SAS Publishing

SAS/STAT® 9.1

User's Guide

The Power to Know®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS/STAT® 9.1 User's Guide*. Cary, NC: SAS Institute Inc.

SAS/STAT® 9.1 User's Guide

ISBN 1-59047-243-8

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, January 2004

Contents

Acknowledgments

Credits

Documentation

Editing	Virginia Clark
Documentation Support	Tim Arnold
Technical Review	Rob Agnelli, Craig DeVault, Elizabeth S. Edwards, Phil Gibbs, Duane Hayes, Kathleen Kiernan, Eddie Routten, Annette Sanders, Paul T. Savarese, David Schlotzhauer, Mike Stockstill

Software

The procedures in SAS/STAT software were implemented by members of the Analytical Solutions Division. Program development includes design, programming, debugging, support, documentation, and technical review. In the following list, the names of the developers currently supporting the procedure are listed first. Other developers previously worked on the procedure.

ACECLUS	Ann Kuo, Warren S. Sarle, Donna Lucas Watts
ANOVA	Randall D. Tobias, Yang C. Yuan
BOXPLOT	Bucky Ransdell, Robert N. Rodriguez
CALIS	Yiu-Fai Yung, Wolfgang M. Hartmann
CANCORR	Ann Kuo, Warren S. Sarle, Donna Lucas Watts
CANDISC	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
CATMOD	Robert E. Derr, John P. Sall, Donna Lucas Watts
CLUSTER	Bart Killam, Warren S. Sarle
CORRESP	Warren F. Kuhfeld
DISCRIM	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
DISTANCE	Ann Kuo
FACTOR	Yiu-Fai Yung, Wolfgang M. Hartmann, John P. Sall, Warren S. Sarle
FASTCLUS	Bart Killam, Warren S. Sarle, Donna Lucas Watts
FREQ	Donna Lucas Watts, John P. Sall
GENMOD	Gordon Johnston
GAM	Randall D. Tobias, Robert Cohen, Dong Xiang

GLM	Randall D. Tobias, James H. Goodnight, John P. Sall, Warren S. Sarle, Yang C. Yuan
GLMMOD	Randall D. Tobias
GLMPOWER	John Castelloe
INBREED	Wendy Czika, Anthony Baiching An, Meltem Narter
KDE	Bucky Ransdell, Russell D. Wolfinger
KRIGE2D	Bart Killam
LATTICE	Oliver Schabenberger, Russell D. Wolfinger
LIFEREG	Colin Chen, Gordon Johnston
LIFETEST	Ying So
LOESS	Robert Cohen
LOGISTIC	Robert E. Derr, Ying So
MDS	Warren S. Sarle
MI	Yang C. Yuan
MIANALYZE	Yang C. Yuan
MIXED	Oliver Schabenberger, Russell D. Wolfinger
MODECLUS	Ann Kuo, Warren S. Sarle
MULTTEST	Robert E. Derr, Russell D. Wolfinger
NESTED	Randall D. Tobias, Leigh A. Ihnen
NLIN	Don Erdman, James H. Goodnight, Leigh A. Ihnen
NLMIXED	Oliver Schabenberger, Russell D. Wolfinger
NPAR1WAY	Donna Lucas Watts, Jane Pierce, John P. Sall
ORTHOREG	Randall D. Tobias, Wolfgang M. Hartmann, John P. Sall
PHREG	Ying So
PLAN	Randall D. Tobias, Leigh A. Ihnen
PLS	Randall D. Tobias
POWER	John Castelloe
PRINCOMP	Ann Kuo, Warren S. Sarle
PRINQUAL	Warren F. Kuhfeld
PROBIT	Colin Chen, Gordon Johnston
REG	Robert Cohen, Leigh A. Ihnen, John P. Sall
ROBUSTREG	Colin Chen
RSREG	Randall D. Tobias, John P. Sall
SCORE	Ann Kuo, Donna Lucas Watts
SIM2D	Bart Killam
STDIZE	Ann Kuo
STEPDISC	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
SURVEYFREQ	Donna Lucas Watts
SURVEYLOGISTIC	Anthony Baiching An
SURVEYMEANS	Anthony Baiching An
SURVEYSELECT	Donna Lucas Watts
SURVEYREG	Anthony Baiching An
TPSPLINE	Randall D. Tobias, Dong Xiang
TRANSREG	Warren F. Kuhfeld
TREE	Bucky Ransdell, Warren S. Sarle
TTEST	John Castelloe, James H. Goodnight, Padraic Neville, Warren S. Sarle,
VARCLUS	Warren S. Sarle, Wolfgang M. Hartmann

VARCOMP	Oliver Schabenberger, James H. Goodnight, Randall D. Tobias, Russell D. Wolfinger
VARIOGRAM	Bart Killam
Probability Routines	Georges Guirguis
Numerical and Graphical Routines	Robert Cohen, Mike Cybrynski, Georges Guirguis, Warren F. Kuhfeld, Warren S. Sarle, Randall D. Tobias, Donna Lucas Watts, Yang C. Yuan, Marc Cohen, Wolfgang M. Hartmann, Leigh A. Ihnen, Richard D. Langston, Katherine Ng, Jane Pierce, John P. Sall, Brian T. Schellenberger

The following people contribute to SAS/STAT software with their leadership and support: Warren F. Kuhfeld, Robert N. Rodriguez, Warren S. Sarle, Maura E. Stokes, and Randall D. Tobias.

Testing

Shu An, Sumita Biswas, Jack J. Berry, Ming-Chun Chang, Bruce Elsheimer, Kelly M. Graham, Gregory D. Goodwin, Gerardo I. Hurtado, Ming Liao, Mark Traccarella, Greg Weier, Li Yang, Fouad G. Younan

Technical Support

Members	Rob Agnelli, Craig DeVault, Phil Gibbs, Duane Hayes, Elizabeth S. Edwards, Kathleen Kiernan, Eddie Routten, Annette Sanders, Paul T. Savarese, David Schlotzhauer, Mike Stockstill

Acknowledgments

Many people make significant and continuing contributions to the development of SAS Institute's software products. The following persons are some of the people who have contributed significant amounts of their time to help us make improvements to SAS/STAT software. This includes research and consulting, testing, or reviewing documentation. We are grateful for the involvement of these members of the statistical community and the many others who are not mentioned here for their feedback, suggestions, and consulting.

Alan Agresti, University of Florida; Douglas Bates, University of Wisconsin; David Binder, Statistics Canada; Suzette Blanchard, Frontier Science Technology Research Foundation; Mary Butler Moore, formerly of University of Florida at Gainesville; Wilbert P. Byrd, Clemson University; Vincent Carey, Harvard University; Sally Carson, RAND; Love Casanova, CSC-FSG; Helene Cavior, Abacus Concepts; George Chao, DuPont Merek Pharmaceutical Company; Daniel M. Chilko, West Virginia University; Jan de Leeuw, University of California, Los Angeles; Dave DeLong, Duke University; Sandra Donaghy, North Carolina State University; David B. Duncan, Johns Hopkins University; Michael Farrell, Oak Ridge National Laboratory; Stewart Fossceco, SLF Consulting; Michael Friendly, York University: Rudolf J. Freund, Texas A & M University; Wayne Fuller, Iowa State University; Andrzej Galecki, University of Michigan; A. Ronald Gallant, University of North Carolina at Chapel Hill; Joseph Gardiner, Michigan State University; Charles Gates, Texas A & M University; Thomas M. Gerig, North Carolina State University; Francis Giesbrecht, North Carolina State University; Harvey J. Gold, North Carolina State University; Kenneth Goldberg, Wyeth-Ayerst Research; Donald Guthrie, University of California, Los Angeles; Gerald Hajian, Schering Plough Research Institute; Bob Hamer, UMDNJ-RWJ Medical School; Frank E. Harrell, Jr., University of Virginia; Walter Harvey, Ohio State University; Douglas Hawkins, University of Minnesota; Ronald W. Helms, University of North Carolina at Chapel Hill; Joseph Hilbe, Arizona State University; Gerry Hobbs, West Virginia University; Ronald R. Hocking, Texas A & M University; Julian Horwich; Camp Conference Company; Jason C. Hsu, Ohio State University; David Hurst, University of Alabama at Birmingham; Emilio A. Icaza, Louisiana State University; Joerg Kaufman, Schering AG; William Kennedy, Iowa State University; Gary Koch, University of North Carolina at Chapel Hill; Kenneth L. Koonce, Louisiana State University; Rich La Valley, Strategic Technology Solutions; Russell V. Lenth, University of Iowa; Charles Lin, U.S. Census Burea; Danyu Lin, University of North Carolina; Ardell C. Linnerud, North Carolina State University; Ramon C. Littel, University of Florida; George MacKenzie, University of Oregon; J. Jack McArdle, University of Virginia; Roderick P. McDonald, Macquarie University; Alfio Marazzi, University of Lausanne; J. Philip Miller, Washington University Medical School; George Milliken, Kansas State University; Robert J. Monroe, North Carolina State University; Robert D. Morrison, Oklahoma State University; Keith Muller, University of North Carolina at Chapel Hill; Anupama Narayanen, Sabre Technologies; Ralph G. O'Brien, Cleveland Clinic Foundation; Kenneth Offord, Mayo Clinic; Christopher R. Olinger, ThotWave Technologies; Robert Parks, Washington University; Richard M. Patterson, Auburn University; Virginia Patterson, University of Tennessee; Cliff Pereira, Oregon State University;

Hans-Peter Piepho, Universitaet Kassel; Edward Pollak, Iowa State University; C. H. Proctor, North Carolina State University; Dana Quade, University of North Carolina at Chapel Hill; Bill Raynor, Kimberly Clark; Georgia Roberts, Statistics Canada; James Roger, Live Data Process Limited; Peter Rousseeuw, University of Antwerp; Donald Rubin, Harvard University; Joseph L. Schafer, Pennsylvania State University; Robert Schechter, AstraZeneca Pharmaceuticals; Shayle Searle, Cornell University; Pat Hermes Smith, formerly of Ciba-Geigy; Roger Smith, formerly of USDA; Phil Spector, University of California, Berkeley; Michael Speed, Texas A & M University at College Station; William Stanish, Statistical Insight; Rodney Strand, Orion Enterprises, LLC; Walter Stroup, University of Nebraska; Robert Teichman, ICI Americas Inc.; Terry M. Therneau, Mayo Clinic; Edward Vonesh, Baxter Healthcare Corporation; Grace Wahba, University of Wisconsin at Madison; Glenn Ware, University of Georgia; Peter H. Westfall, Texas Tech University; Edward W. Whitehorne, CI Partners, LLC; William Wigton, USDA; William Wilson, University of North Florida; Philip Wittall, Unilever; Victor Yohai, University of Buenos Aires; Forrest W. Young, University of North Carolina at Chapel Hill; Ruben Zamar, University of British Columbia; and Scott Zeger, Johns Hopkins University.

The final responsibility for the SAS System lies with SAS Institute alone. We hope that you will always let us know your opinions about the SAS System and its documentation. It is through your participation that SAS software is continuously improved.

Please see Feedback at www.sas.com/statistics/ for your comments.

What’s New in SAS/STAT 9 and 9.1

Contents

What’s New in SAS/STAT 9 and 9.1

Overview

This release brings several new procedures to SAS/STAT software. The MI and MIANALYZE procedures implement the multiple imputation strategy for missing data. Experimental in Releases 8.1 and 8.2, these procedures are now production. The ROBUSTREG procedure analyzes data that may include outliers; it provides stable results in their presence. The TPHREG procedure is a test release of the PHREG procedure that incorporates the CLASS statement.

Power and sample size computations also become available in SAS 9.1. New procedures POWER and GLMPOWER provide these computations for a number of analyses, and the Power and Sample Size Application surfaces them through a point-and-click interface.

SAS 9.1 introduces two new procedures for the analysis of survey data. The SURVEYFREQ procedure produces one-way to *n*-way frequency and crosstabulation tables for data collected from surveys. These tables include estimates of totals and proportions (overall, row percentages, column percentages) and the corresponding standard errors. The SURVEYLOGISTIC procedure performs logistic regression for survey data, and it can also fit links such as the cumulative logit, generalized logit, probit, and complementary log-log functions. Both of these procedures incorporate complex survey sample designs, including designs with stratification, clustering, and unequal weighting, in their computations.

In addition, this release includes numerous enhancements to existing procedures. For example, conditional logistic regression is available in the LOGISTIC procedure through the new STRATA statement, and scoring of data sets is available through the new SCORE statement. The GLM procedure now provides the ability to form classification groups using the full formatted length of the CLASS variable levels. In addition, the SURVIVAL statement in the LIFETEST procedure enables you to create confidence bands (also known as simultaneous confidence intervals) for the survivor function $S(t)$ and to specify a transformation for computing the confidence bands and the pointwise confidence intervals.

More information about the changes and enhancements to SAS/STAT software follows. Features new in SAS 9.1 are indicated with a 9.1 icon; other features were available with SAS 9. Details can be found in the documentation for the individual procedures.

Parallelization

Selected functionalities in the GLM, LOESS, REG, and ROBUSTREG procedures have been multithreaded to exploit hardware with multiple CPUs. Refer to Cohen (2002) for more details.

Statistical Graphics Using ODS

A number of SAS/STAT procedures are using an experimental extension to the Output Delivery System (ODS) that enables them to create statistical graphics automatically. The facility is invoked when you include an ODS GRAPHICS statement before your procedure statements. Graphics are then created automatically, or when you specify procedure options for graphics. Procedures taking advantage of ODS graphics are the ANOVA, CORRESP, GAM, GENMOD, GLM, KDE, LIFETEST, LOESS, LOGISTIC, MI, MIXED, PHREG, PLS, PRINCOMP, PRINQUAL, REG, ROBUSTREG, and TPSLINE procedures. The plots produced and the corresponding options are described in the documentation for the individual procedures.

CATMOD Procedure

9.1 Memory handling has been improved in the CATMOD procedure. The PARAM=REFERENCE option has been added to the MODEL statement and produces reference cell parameterization. Other new options include the ITPRINT, DESIGN, and PROFILE|POPPROFILE options in the PROC statement.

DISTANCE Procedure

9.1 The new DISTANCE procedure computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. These proximity measures are stored as a lower triangular matrix or a square matrix in an output data set (depending on the SHAPE= option) that can then be used as input to the CLUSTER, MDS, and MODECLUS procedures. The input data set may contain numeric or character variables, or both, depending on which proximity measure is used. PROC DISTANCE also provides various nonparametric and parametric methods for standardizing variables. Distance matrices are used frequently in data mining, genomics, marketing, financial analysis, management science, education, chemistry, psychology, biology, and various other fields.

FACTOR Procedure

The NOPROMAXNORM option turns off the default row normalization of the prerotated factor pattern, which is used in computing the promax target matrix.

You can now produce standard errors and confidence limits with the METHOD=ML option for the PROMAX factor solutions. You can obtain the standard errors with the SE option, control the coverage displays with the COVER= option, and set the coverage level with the ALPHA= option.

FREQ Procedure

The BDT option includes Tarone's adjustment in the Breslow-Day test for homogeneity of odds ratios. Refer to Agresti (1996) and Tarone (1985).

The ZEROS option in the WEIGHT statement includes zero-weight observations in the analysis. (By default, PROC FREQ does not process zero-weight observations.) With the ZEROS option, PROC FREQ displays zero-weight levels in crosstabulation and frequency tables. For one-way tables, the ZEROS option includes zero-weight levels in chi-square tests and binomial statistics. For multiway tables, the ZEROS option includes zero-weight levels in kappa statistics.

The CROSSLIST option displays crosstabulation tables in ODS column format. Unlike the default crosstabulation table, the CROSSLIST table has a table definition that you can customize with PROC TEMPLATE. The NLEVELS option provides a table with the number of levels for all TABLES statement variables.

The FREQ procedure now produces exact confidence limits for the common odds ratio and related tests. **9.1**

GENMOD Procedure

The GENMOD procedure now forms classification groups using the full formatted length of the CLASS variable levels. Several new full-rank CLASS variable parameterizations are now available: polynomial, orthogonal polynomial, effect, orthogonal effect, reference, orthogonal reference, ordinal, and orthogonal ordinal. The default parameterization remains the same less-than-full-rank parameterization used in previous releases.

Zero is now a valid value for the negative binomial dispersion parameter corresponding to the Poisson distribution. If a fixed value of zero is specified, a score test for overdispersion (Cameron and Trivedi 1998) is computed.

As an experimental feature, PROC GENMOD now provides model assessment based on aggregates of residuals. **9.1**

GLM Procedure

The GLM procedure now forms classification groups using the full formatted length of the CLASS variable levels.

In addition, you can compute exact *p*-values for three of the four multivariate tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root) and an improved F-approximation for the fourth (Pillai's Trace). The default MSTAT=FAPPROX in the MANOVA and REPEATED statements produces multivariate tests using approximations based on the F distribution. Specifying MSTAT=EXACT computes exact *p*-values for three of the four tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root) and an improved F-approximation for the fourth (Pillai's Trace).

GLMPOWER Procedure

9.1 The GLMPOWER procedure performs prospective analyses for linear models, with a variety of goals:

- determining the sample size required to obtain a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

You specify the design and the cell means using an exemplary data set, a data set of artificial values constructed to represent the intended sampling design and the surmised response means in the underlying population. You specify the model and contrasts using MODEL and CONTRAST statements similar to those in the GLM procedure. You specify the remaining parameters with the POWER statement, which is similar to analysis statements in the new POWER procedure.

KDE Procedure

9.1 The new UNIVAR and BIVAR statements provide improved syntax. The BIVAR statement lists variables in the input data set for which bivariate kernel density estimates are to be computed. The UNIVAR statement lists variables in the input data set for which univariate kernel density estimates are to be computed.

LIFETEST Procedure

The new SURVIVAL statement enables you to create confidence bands (also known as simultaneous confidence intervals) for the survivor function $S(t)$ and to specify a transformation for computing the confidence bands and the pointwise confidence intervals. It contains the following options.

- The OUT= option names the output SAS data set that contains survival estimates as in the OUTSURV= option in the PROC LIFETEST statement.
- The CONFTYPE= option specifies the transformation applied to $S(t)$ to obtain the pointwise confidence intervals and the confidence bands. Four transforms are available: the arcsine-square root transform, the complementary log-log transform, the logarithmic transform, and the logit transform.
- The CONFBAND= option specifies the confidence bands to add to the OUT= data set. You can choose the equal precision confidence bands (Nair 1984), or the Hall-Wellner bands (Hall and Wellner 1980), or both.
- The BANDMAX= option specifies the maximum time for the confidence bands.
- The BANDMIN= option specifies the minimum time for the confidence bands.

- The STDERR option adds the column of standard error of the estimated survivor function to the OUT= data set.
- The ALPHA= option sets the confidence level for pointwise confidence intervals as well as the confidence bands.

The LIFETEST procedure now provides additional tests for comparing two or more samples of survival data, including the Tarone-Ware test, Peto-Peto test, modified Peto-Peto test, and the Fleming-Harrington G_ρ family of tests. Trend tests for ordered alternatives can be requested. Also available are stratified tests for comparing survival function while adjusting for prognostic factors that affect the event rates. **9.1**

LOESS Procedure

The LOESS procedure now performs DF computations using a sparse method when appropriate. In addition, the DFMETHOD=APPROX option is available. **9.1**

LOGISTIC Procedure

The new SCORE statement enables you to score new data sets and compute fit statistics and ROC curves without refitting the model. Information for a fitted model can be saved to a SAS data set with the OUTMODEL= option, while the INMODEL= option inputs the model information required for the scoring.

The new STRATA statement enables you to perform conditional logistic regression on highly stratified data using the method of Gail, Lubin, and Rubenstein (1981). The OFFSET option is now enabled for logistic regression.

The LOGISTIC procedure now forms classification groups using the full formatted length of the CLASS variable levels.

Several new CLASS parameterizations are available: ordinal, orthogonal effect, orthogonal reference, and orthogonal ordinal.

You can now output the design matrix using the new OUTDESIGN= option.

The definition of concordance has been changed to make it more meaningful for ordinal models. The new definition is consistent with that used in previous releases for the binary response model.

Enhancements for the exact computations include **9.1**

- improved performance
- Monte Carlo method
- mid-*p* confidence intervals

For an exact conditional analysis, specifying the STRATA statement performs an efficient stratified analysis. The method of Mehta, Patel, and Senchaudhuri (1992), which is more efficient than the Hirji, Tsiatis, and Mehta (1989) algorithm for many problems, is now available with the METHOD=NETWORK option.

MI Procedure

The INITIAL= option in the EM statement sets the initial estimates for the EM algorithm. Either the means and covariances from complete cases or the means and standard deviations from available cases can be used as the initial estimates for the EM algorithm. You can also specify the correlations for the initial estimates from available cases.

9.1 For data sets with monotone missingness, the REGPMM option in the MONOTONE statement uses the predictive mean matching method to impute a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model.

You can specify more than one method in the MONOTONE statement, and for each imputed variable, the covariates can be specified separately.

The DETAILS option in the MONOTONE statement requests the display of the model parameters used for each imputation.

The experimental CLASS statement is now available to specify categorical variables. These classification variables are used either as covariates for imputed variables or as imputed variables for data sets with monotone missing patterns.

The experimental options LOGISTIC and DISCRIM in the MONOTONE statement impute missing categorical variables by logistic and discriminant methods, respectively.

MIANALYZE Procedure

You can now specify the PARMS= data set without specifying either the COVB= or XPXI= option when the data set contains the standard errors for the parameter estimates.

The DATA= option includes data sets that contain both parameter estimates and their associated standard errors in each observation of the data set.

The BCOV, WCOV, and TCOV options control the display of the between-imputation, within-imputation, and total covariance matrices.

A TEST statement tests linear hypotheses about the parameters, $H_0\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$. For each TEST statement, the procedure combines the estimate and associated standard error for each linear component (a row of $\mathbf{L}\boldsymbol{\beta}$). It can also combine the estimates and associated covariance matrix for all linear components.

The MODELEFFECTS statement lists the effects in the data set to be analyzed. Each effect is a variable or a combination of variables, and is specified with a special notation using variable names and operators. The STDERR statement lists the standard errors associated with the effects in the MODELEFFECTS statement when both parameter estimates and standard errors are saved as variables in the same DATA= data set.

The experimental CLASS statement specifies categorical variables. PROC MIANALYZE reads and combines parameter estimates and covariance matrices for parameters with CLASS variables.

MIXED Procedure

The MIXED procedure now supports geometrically anisotropic covariance structures and covariance models in the Matern class. The LCOMPONENTS option in the MODEL statement produces one degree of freedom tests for fixed effects that correspond to individual estimable functions for Type I, II, and III effects. ***9.1***

The experimental RESIDUAL option of the MODEL statement computes Pearson-type and (internally) studentized residuals. The experimental INFLUENCE option in the MODEL statement computes influence diagnostics by noniterative or iterative methods. Experimental ODS graphics display the results for both of these options. ***9.1***

NPAR1WAY Procedure

The new D option provides the one-sided $D+$ and $D-$ statistics for the asymptotic two-sample Kolmogorov-Smirnov test, in addition to the two-sided D statistic given by the EDF option. The KS option in the EXACT statement gives exact tests for the Kolmogorov-Smirnov $D+$, $D-$, and D for two-sample problems.

PHREG Procedure

The new WEIGHT statement enables you to specify case weights when you are using the BRESLOW or EFRON method for handling ties. Robust sandwich variance estimators of Binder (1992) are computed for the estimated regression parameters. You can specify the option NORMALIZE to normalize the weights so that they add up the actual sample size.

Two options have been added to the TEST statement: AVERAGE and E. The AVERAGE option enables you to compute a combined estimate of all the effects in the given TEST statement. This option gives you an easy way to carry out inferences of the common value of (say) the treatment effects had they been assumed equal. The E option specifies that the linear coefficients and constants be printed. When the AVERAGE option is specified along with the E option, the optimal weights of the average effect are also printed in the same tables as the coefficients.

The recurrence algorithm of Gail, Lubin, and Rubinstein (1981) for computing the exact discrete partial likelihood and its partial derivatives has been modified to use the logarithmic scale. This enables a much larger number of ties to be handled without the numeric problems of overflowing and underflowing.

You can use the PHREG procedure to fit the rate/mean model for the recurrent events data and obtain prediction of the cumulative mean function for a given pattern of fixed covariates. ***9.1***

As an experimental feature, the PHREG procedure now can produce model assessments based on cumulative residuals.

POWER Procedure

9.1 The POWER procedure performs prospective analyses for a variety of goals such as the following:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

This procedure covers a variety of statistical analyses such as *t* tests, equivalence tests, and confidence intervals for means; exact binomial, chi-square, Fisher's exact, and McNemar tests for proportions; multiple regression and correlation; one-way analysis of variance; and rank tests for comparing survival curves.

The POWER procedure is one of several tools available in SAS/STAT software for power and sample size analysis. PROC GLMPOWER covers more complex linear models, and the Power and Sample Size Application provides a user interface and implements many of the analyses supported in the procedures.

Power and Sample Size Application

The Power and Sample Size Application (PSS) is an interface that provides power and sample size computations. The application includes tasks for determining sample size and power for a variety of statistical analyses, including *t*-tests, ANOVA, proportions, equivalence testing, linear models, survival analysis, and table statistics. The application provides multiple input parameter options, stores results in a project format, displays power curves, and produces appropriate narratives for the results. Note that this application is included with SAS/STAT software but needs to be installed from the Mid Tier CD.

ROBUSTREG Procedure

9.1 The ROBUSTREG procedure provides resistant (stable) results in the presence of outliers by limiting the influence of outliers. In statistical applications of outlier detection and robust regression, the methods most commonly used today are Huber (1973) M estimation, high breakdown value estimation, and combinations of these two methods. The ROBUSTREG procedure provides four such methods: M estimation, LTS estimation, S estimation, and MM estimation. With these four methods, the ROBUSTREG procedure acts as an integrated tool for outlier detection and robust regression with various contaminated data. The ROBUSTREG procedure is scalable such that it can be used for applications in data cleansing and data mining.

SURVEYFREQ Procedure

9.1 The SURVEYFREQ procedure produces one-way to n-way frequency and crosstabulation tables for survey data. These tables include estimates of totals and proportions (overall, row percentages, column percentages) and the corresponding standard errors. Like the other survey procedures, PROC SURVEYFREQ computes these variance estimates based on the sample design used to obtain the survey data. The design can be a complex sample survey design with stratification, clustering, and unequal weighting. PROC SURVEYFREQ also provides design-based tests of association between variables.

SURVEYLOGISTIC Procedure

9.1 The SURVEYLOGISTIC procedure performs logistic regression on data that arise from a survey sampling scheme. PROC SURVEYLOGISTIC incorporates complex survey sample designs, including designs with stratification, clustering, and unequal weighting, in its estimation process. Variances of the regression parameters and odds ratios are computed using a Taylor expansion approximation. The SURVEYLOGISTIC procedure is similar in syntax to the LOGISTIC procedure, and it can fit link functions such as the logit, cumulative logit, generalized logit, probit, and complementary log-log functions. Maximum likelihood estimation of the regression coefficients is carried out with either the Fisher-scoring algorithm or the Newton-Raphson algorithm.

SURVEYMEANS Procedure

The STACKING option requests the procedure to produce the output data sets using a stacking table structure, which was the default in earlier releases. The new default is to produce a rectangular table structure in the output data sets. The STACKING option affects the Domain, Ratio, Statistics, and StrataInfo tables.

One-sided confidence limits are now available for descriptive statistics. **9.1**

SURVEYREG Procedure

The SURVEYREG procedure now provides the ability to form classification groups using the full formatted length of the CLASS variable levels, instead of just the first 16 characters of the levels. The ANOVA option in the MODEL statement requests that the ANOVA table be included in the output.

SURVEYSELECT Procedure

The OUTALL option produces an output data set that includes all observations from the DATA= input data set, both those observations selected for the sample and those observations not selected. With the OUTALL option, the OUT= data set contains a variable Selected that indicates whether or not the observation was selected. The OUTALL option is available for equal probability selection methods (METHOD=SRS, URS, SYS, and SEQ).

The SELECTALL option includes all stratum observations in the sample when the stratum sample size exceeds the number of observations in the stratum. The SELECTALL option is available for without-replacement selection methods (METHOD=SRS, SYS, SEQ, PPS, and PPS_SAMPFORD). It is not available for with-replacement or with-minimum-replacement methods, or for those PPS methods that select two units per stratum.

9.1 The OUTSEED option includes the initial seed for each stratum in the output data set. Additionally, you can input initial seeds by strata with the SEED=SAS-data-set option.

TPHREG Procedure

The experimental TPHREG procedure adds the CLASS statement to the PHREG procedure. The CLASS statement enables you to specify categorical variables (also known as CLASS variables) as explanatory variables. Explanatory effects for the model, including covariates, main effects, interactions, and nested effects, can be specified in the same way as in the GLM procedure. The CLASS statement supports less-than-full-rank parameterization as well as various full-rank parameterizations such as reference coding and effect coding. Other CLASS statement features that are found in PROC LOGISTIC, such as specifying specific categories as reference levels, are also available.

The TPHREG procedure also enables you to specify CONTRAST statements for testing customized hypotheses concerning the regression parameters. Each CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful in comparing the hazards between the categories of a CLASS explanatory variable.

TPSPLINE Procedure

9.1 The COEF option in the OUTPUT statement enables you to output coefficients of the fitted function.

TRANSREG Procedure

The TRANSREG procedure has new transformation options for centering and standardizing variables, CENTER and Z, before the transformations. The new EXKNOTS= option specifies exterior knots for SPLINE and MSPLINE transformations and BSPLINE expansions.

The new algorithm option INDIVIDUAL with METHOD=MORALS fits each model for each dependent variable individually and independently of the other dependent variables.

With hypothesis tests, the TRANSREG procedure now produces a table with the number of observations, and, when there are CLASS variables, a class level information table.

References

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Binder, D.A. (1983), “On the Variances of Asymptotically Normal Estimators from Complex Surveys,” *International Statistical Review*, 51, 279–292.

Binder, D.A. (1992), “Fitting Cox’s Proportional Hazards Models from Survey Data,” *Biometrika*, 79, 139–47.

Cameron, A.C. and Trivedi, P.K. (1998), “Regression Analysis of Count Data,” Cambridge: Cambridge University Press.

Cohen, R. (2002), “SAS Meets Big Iron: High Performance Computing in SAS Analytical Procedures,” *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*.

Gail, M.H., Lubin, J.H., and Rubinstein, L.V. (1981), “Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Survival Times,” *Biometrika*, 78, 703–7.

Hall, W.J. and Wellner, J.A. (1980), “Confidence Bands for a Survival Curve for Censored Data,” *Biometrika 69*, 133–143.

Hirji, K.F., Mehta, C.R., and Patel, N.R. (1987), “Computing Distributions for Exact Logistic Regression,” *Journal of the American Statistical Association*, 82, 1110–1117.

Hirji, K.F., Tsiatis, A.A., and Mehta, C.R. (1989), “Median Unbiased Estimation for Binary Data,” *American Statistician*, 43, 7–11.

Huber, P.J. (1973), “Robust Regression: Asymptotics, Conjectures and Monte Carlo,” *Annals of Statistics*, 1, 799-821.

Mehta, C.R., Patel, N., and Senchaudhuri, P. (1992), “Exact Stratified Linear Rank Tests for Ordered Categorical and Binary Data,” *Journal of Computational and Graphical Statistics*, 1, 21–40.

Nair, V.N. (1984), "Confidence Bands for Survival Functions with Censored Data: A Comparative Study," *Technometrics*, 14, 265–275.

Tarone, R. (1985), "On Heterogeneity Tests Based on Efficient Scores," *Biometrika*, 72, 91–95.

Chapter 1
Introduction

Chapter Contents

Chapter 1
Introduction

Overview of SAS/STAT Software

SAS/STAT software provides comprehensive statistical tools for a wide range of statistical analyses, including analysis of variance, categorical data analysis, cluster analysis, multiple imputation, multivariate analysis, nonparametric analysis, power and sample size computations, psychometric analysis, regression, survey data analysis, and survival analysis. A few examples include nonlinear mixed models, generalized linear models, correspondence analysis, and robust regression. The software is constantly being updated to reflect new methodology.

In addition to over sixty procedures for statistical analysis, SAS/STAT software also includes the Market Research Application (MRA), a point-and-click interface to commonly used techniques in market research. The Analyst Application provides convenient access to some of the more commonly used statistical analyses in SAS/STAT software including analysis of variance, regression, logistic regression, mixed models, survival analysis, and some multivariate techniques. Also, the new Power and Sample Size Application (PSS) is an interface to power and sample size computations. These applications are documented separately.

About This Book

Since SAS/STAT software is a part of the SAS System, this book assumes that you are familiar with Base SAS software and with the books *SAS Language Reference: Dictionary*, *SAS Language Reference: Concepts*, and the *SAS Procedures Guide*. It also assumes that you are familiar with basic SAS System concepts such as creating SAS data sets with the DATA step and manipulating SAS data sets with the procedures in Base SAS software (for example, the PRINT and SORT procedures).

Chapter Organization

This book is organized as follows.

“What’s New in SAS/STAT 9 and 9.1” provides information about the changes and enhancements to SAS/STAT software in SAS 9 and SAS 9.1. It describes several new procedures as well as numerous new features. SAS 9.1 features are indicated by a 9.1 icon in the margins of these pages.

Chapter 1, this chapter, provides an overview of SAS/STAT software and summarizes related information, products, and services. The next twelve chapters provide some introduction to the broad areas covered by SAS/STAT software.

Chapter 14, “Using the Output Delivery System,” explains the fundamentals of using the Output Delivery System (ODS) to manage your SAS output. Chapter 15, “Statistical Graphics Using ODS,” describes the experimental extension to ODS that enables a number of statistical procedures to create statistical graphics as easily as tables.

Subsequent chapters describe the SAS procedures that make up SAS/STAT software. These chapters appear in alphabetical order by procedure name and are organized as follows:

- The “Overview” section provides a brief description of the analysis provided by the procedure.
- The “Getting Started” section provides a quick introduction to the procedure through a simple example.
- The “Syntax” section describes the SAS statements and options that control the procedure.
- The “Details” section discusses methodology and miscellaneous details, such as ODS tables and ODS graphics.
- The “Examples” section contains examples using the procedure.
- The “References” section contains references for the methodology and for examples of the procedure.

Following the chapters on the SAS/STAT procedures, Appendix A, “Special SAS Data Sets,” documents the special SAS data sets associated with SAS/STAT procedures.

Typographical Conventions

This book uses several type styles for presenting information. The following list explains the meaning of the typographical conventions used in this book:

roman is the standard type style used for most text.

UPPERCASE ROMAN is used for SAS statements, options, and other SAS language elements when they appear in the text. However, you can enter these elements in your own SAS programs in lowercase, uppercase, or a mixture of the two.

UPPERCASE BOLD is used in the “Syntax” sections’ initial lists of SAS statements and options.

oblique is used for user-supplied values for options in the syntax definitions. In the text, these values are written in *italic*.

helvetica is used for the names of variables and data sets when they appear in the text.

bold is used to refer to matrices and vectors.

italic is used for terms that are defined in the text, for emphasis, and for references to publications.

`monospace` is used for example code. In most cases, this book uses lowercase type for SAS code.

Options Used in Examples

Output of Examples

Most of the output shown in this book is produced with the following SAS System options:

```
options linesize=80 pagesize=200 nonumber nodate;
```

The template STATDOC.TPL is used to create the HTML output that appears in the online documentation. A style template controls stylistic HTML elements such as colors, fonts, and presentation attributes. The style template is specified in the ODS HTML statement as follows:

```
ODS HTML style=statdoc;
```

If you run the examples, you may get slightly different output. This is a function of the SAS System options used and the precision used by your computer for floating-point calculations.

Graphics Options

Some of the graphical output displayed in the examples is generated with the experimental ODS graphics system. Other examples use SAS/GRAPH software, including the GOPTIONS statement and PROC GPLOT. The rest of this section provides information on the specific set of options and symbol statements used to generate graphical output using SAS/GRAPH software.

The code you see in the examples creates the color graphics that appear in the online version of this book. A slightly different set of options and statements is used to create the black-and-white graphics that appear in the printed version of the book.

If you run the examples, you may get slightly different results. This may occur because not all graphic options for color devices translate directly to black-and-white output formats. For complete information on SAS/GRAPH software and graphics options, refer to *SAS/GRAPH Software: Reference*.

The following GOPTIONS statement is used to create the online (color) version of the graphic output.

```
filename GSASFILE  '<file-specification>';

goptions gsfname=GSASFILE   gsfmode =replace
         fileonly
         transparency       dev     = gif
         ftext   = swiss    lfactor = 1
         htext   = 4.0pct   htitle  = 4.5pct
         hsize   = 5.625in  vsize   = 3.5in
         noborder           cback   = white
         horigin = 0in      vorigin = 0in ;
```

The following GOPTIONS statement is used to create the black-and-white version of the graphic output, which appears in the printed version of the manual.

```
filename GSASFILE  '<file-specification>';

goptions gsfname=GSASFILE   gsfmode =replace
         gaccess = sasgaedt fileonly
         dev     = pslepsf
         ftext   = swiss    lfactor = 1
         htext   = 3.0pct   htitle  = 3.5pct
         hsize   = 5.625in  vsize   = 3.5in
         border             cback   = white
         horigin = 0in      vorigin = 0in ;
```

In most of the online examples, the plot symbols are specified as follows:

```
symbol1 value=dot color=white height=3.5pct;
```

The SYMBOL*n* statements used in online examples order the symbol colors as follows: white, yellow, cyan, green, orange, blue, and black.

In the examples appearing in the printed manual, symbol statements specify COLOR=BLACK and order the plot symbols as follows: dot, square, triangle, circle, plus, x, diamond, and star.

The %PLOTIT Macro

Examples that use the %PLOTIT macro are generated by defining a special macro variable to specify graphics options. See Appendix B, “Using the %PLOTIT Macro,” for details on the options specified in these examples.

Where to Turn for More Information

This section describes other sources of information about SAS/STAT software.

Accessing the SAS/STAT Sample Library

The SAS/STAT sample library includes many examples that illustrate the use of SAS/STAT software, including the examples used in this documentation. To access these sample programs, select the **Help** pull-down menu and then select **Getting Started with SAS Software**. From the **Contents** list, choose **Learning to Use SAS** and then **Sample SAS Programs**. Select the **SAS/STAT** product.

Online Documentation

This documentation is available online with the SAS System. If you are using the SAS windowing environment with pull-down menus, you can select **SAS Help and Documentation** from the **Help** menu. Under the **Contents** tab select **SAS/STAT** and then select **SAS/STAT User's Guide** from the list of available topics.

Alternatively, you can type **help STAT** in the command line. Note that you can use the online facility to search the documentation. You can also access the documentation from the SAS Web site. Go to **support.sas.com** and select **Documentation** for more detail.

SAS Institute Technical Support Services

As with all SAS Institute products, the SAS Institute Technical Support staff is available to respond to problems and answer technical questions regarding the use of SAS/STAT software.

Related SAS Software

Many features not found in SAS/STAT software are available in other parts of the SAS System. If you don't find something you need in SAS/STAT software, try looking for the feature in the following SAS software products.

Base SAS Software

The features provided by SAS/STAT software are in addition to the features provided by Base SAS software. Many data management and reporting capabilities you will need are part of Base SAS software. Refer to *SAS Language Reference: Concepts*, *SAS Language Reference: Dictionary*, and the *SAS Procedures Guide* for documentation of Base SAS software.

SAS DATA Step

The DATA step is your primary tool for reading and processing data in the SAS System. The DATA step provides a powerful general purpose programming language that enables you to perform all kinds of data processing tasks. The DATA step is documented in *SAS Language Reference: Concepts*.

Base SAS Procedures

Base SAS software includes many useful SAS procedures. Base SAS procedures are documented in the *SAS Procedures Guide*. The following is a list of Base SAS procedures you may find useful:

CORR	compute correlations
RANK	compute rankings or order statistics
STANDARD	standardize variables to a fixed mean and variance
MEANS	compute descriptive statistics and summarizing or collapsing data over cross sections

TABULATE	print descriptive statistics in tabular format
UNIVARIATE	compute descriptive statistics

SAS/ETS Software

SAS/ETS software provides SAS procedures for econometrics and time series analysis. It includes capabilities for forecasting, systems modeling and simulation, seasonal adjustment, and financial analysis and reporting. In addition, SAS/ETS software includes an interactive time series forecasting system.

SAS/GRAPH Software

SAS/GRAPH software includes procedures that create two- and three-dimensional high-resolution color graphics plots and charts. You can generate output that graphs the relationship of data values to one another, enhance existing graphs, or simply create graphics output that is not tied to data.

SAS/IML Software

SAS/IML software gives you access to a powerful and flexible programming language (Interactive Matrix Language) in a dynamic, interactive environment. The fundamental object of the language is a data matrix. You can use SAS/IML software interactively (at the statement level) to see results immediately, or you can store statements in a module and execute them later. The programming is dynamic because necessary activities such as memory allocation and dimensioning of matrices are done automatically. SAS/IML software is of interest to users of SAS/STAT software because it enables you to program your methods in the SAS System.

SAS/INSIGHT Software

SAS/INSIGHT software is a highly interactive tool for data analysis. You can explore data through a variety of interactive graphs including bar charts, scatter plots, box plots, and three-dimensional rotating plots. You can examine distributions and perform parametric and nonparametric regression, analyze general linear models and generalized linear models, examine correlation matrixes, and perform principal component analyses. Any changes you make to your data show immediately in all graphs and analyses. You can also configure SAS/INSIGHT software to produce graphs and analyses tailored to the way you work.

SAS/INSIGHT software may be of interest to users of SAS/STAT software for interactive graphical viewing of data, editing data, exploratory data analysis, and checking distributional assumptions.

SAS/OR Software

SAS/OR software provides SAS procedures for operations research and project planning and includes a point-and-click interface to project management. Its capabilities include the following:

- solving transportation problems
- linear, integer, and mixed-integer programming
- nonlinear programming
- scheduling projects
- plotting Gantt charts
- drawing network diagrams
- solving optimal assignment problems
- network flow programming

SAS/OR software may be of interest to users of SAS/STAT software for its mathematical programming features. In particular, the NLP procedure in SAS/OR software solves nonlinear programming problems, and it can be used for constrained and unconstrained maximization of user-defined likelihood functions.

SAS/QC Software

SAS/QC software provides a variety of procedures for statistical quality control and quality improvement. SAS/QC software includes procedures for

- Shewhart control charts
- cumulative sum control charts
- moving average control charts
- process capability analysis
- Ishikawa diagrams
- Pareto charts
- experimental design

SAS/QC software also includes the ADX interface for experimental design.

Chapter 2
Introduction to Regression Procedures

Chapter Contents

Chapter 2
Introduction to Regression Procedures

Overview

This chapter reviews SAS/STAT software procedures that are used for regression analysis: CATMOD, GLM, LIFEREG, LOESS, LOGISTIC, NLIN, ORTHOREG, PLS, PROBIT, ROBUSTREG, REG, RSREG, and TRANSREG. The REG procedure provides the most general analysis capabilities; the other procedures give more specialized analyses. This chapter also briefly mentions several procedures in SAS/ETS software.

Introduction

Many SAS/STAT procedures, each with special features, perform regression analysis. The following procedures perform at least one type of regression analysis:

CATMOD — analyzes data that can be represented by a contingency table. PROC CATMOD fits linear models to functions of response frequencies, and it can be used for linear and logistic regression. See Chapter 4, "Introduction to Categorical Data Analysis Procedures," and Chapter 22, "The CATMOD Procedure," for more information.

GENMOD — fits generalized linear models. PROC GENMOD is especially suited for responses with discrete outcomes, and it performs logistic regression and Poisson regression as well as fitting Generalized Estimating Equations for repeated measures data. See Chapter 4, "Introduction to Categorical Data Analysis Procedures," and Chapter 31, "The GENMOD Procedure," for more information.

GLM — uses the method of least squares to fit general linear models. In addition to many other analyses, PROC GLM can perform simple, multiple, polynomial, and weighted regression. PROC GLM has many of the same input/output capabilities as PROC REG, but it does not provide as many diagnostic tools or allow interactive changes in the model or data. See Chapter 3, "Introduction to Analysis-of-Variance Procedures," and Chapter 32, "The GLM Procedure," for more information.

LIFEREG — fits parametric models to failure-time data that may be right censored. These types of models are commonly used in survival analysis. See Chapter 9, "Introduction to Survival Analysis Procedures," and Chapter 39, "The LIFEREG Procedure," for more information.

LOESS fits nonparametric models using a local regression method. PROC LOESS is suitable for modeling regression surfaces where the underlying parametric form is unknown and where robustness in the presence of ouliers is required. See Chapter 41, "The LOESS Procedure," for more information.

LOGISTIC fits logistic models for binomial and ordinal outcomes. PROC LOGISTIC provides a wide variety of model-building methods and computes numerous regression diagnostics. See Chapter 4, "Introduction to Categorical Data Analysis Procedures," and Chapter 42, "The LOGISTIC Procedure," for more information.

NLIN builds nonlinear regression models. Several different iterative methods are available. See Chapter 50, "The NLIN Procedure," for more information.

ORTHOREG performs regression using the Gentleman-Givens computational method. For ill-conditioned data, PROC ORTHOREG can produce more accurate parameter estimates than other procedures such as PROC GLM and PROC REG. See Chapter 53, "The ORTHOREG Procedure," for more information.

PLS performs partial least squares regression, principal components regression, and reduced rank regression, with cross validation for the number of components. See Chapter 56, "The PLS Procedure," for more information.

PROBIT performs probit regression as well as logistic regression and ordinal logistic regression. The PROBIT procedure is useful when the dependent variable is either dichotomous or polychotomous and the independent variables are continuous. See Chapter 60, "The PROBIT Procedure," for more information.

REG performs linear regression with many diagnostic capabilities, selects models using one of nine methods, produces scatter plots of raw data and statistics, highlights scatter plots to identify particular observations, and allows interactive changes in both the regression model and the data used to fit the model. See Chapter 61, "The REG Procedure," for more information.

ROBUSTREG performs robust regression using Huber M estimation and high breakdown value estimation. PROC ROBUSTREG is suitable for detecting outliers and providing resistant (stable) results in the presence of outliers. See Chapter 62, "The ROBUSTREG Procedure," for more information.

RSREG builds quadratic response-surface regression models. PROC RSREG analyzes the fitted response surface to determine the factor levels of optimum response and performs a ridge analysis to search for the region of optimum response. See Chapter 63, "The RSREG Procedure," for more information.

TRANSREG fits univariate and multivariate linear models, optionally with spline and other nonlinear transformations. Models include ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. See Chapter 75, "The TRANSREG Procedure," for more information.

Several SAS/ETS procedures also perform regression. The following procedures are documented in the *SAS/ETS User's Guide*.

AUTOREG implements regression models using time-series data where the errors are autocorrelated. Refer to **??** for more details.

PDLREG performs regression analysis with polynomial distributed lags. Refer to **??** for more details.

SYSLIN handles linear simultaneous systems of equations, such as econometric models. Refer to **??** for more details.

MODEL handles nonlinear simultaneous systems of equations, such as econometric models. Refer to **??** for more details.

Introductory Example

Regression analysis is the analysis of the relationship between one variable and another set of variables. The relationship is expressed as an equation that predicts a *response variable* (also called a *dependent variable* or *criterion*) from a function of *regressor variables* (also called *independent variables, predictors, explanatory variables, factors,* or *carriers*) and *parameters*. The parameters are adjusted so that a measure of fit is optimized. For example, the equation for the ith observation might be

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where y_i is the response variable, x_i is a regressor variable, β_0 and β_1 are unknown parameters to be estimated, and ϵ_i is an error term.

You might use regression analysis to find out how well you can predict a child's weight if you know that child's height. Suppose you collect your data by measuring heights and weights of 19 school children. You want to estimate the intercept β_0 and the slope β_1 of a line described by the equation

$$\text{Weight} = \beta_0 + \beta_1 \text{Height} + \epsilon$$

where

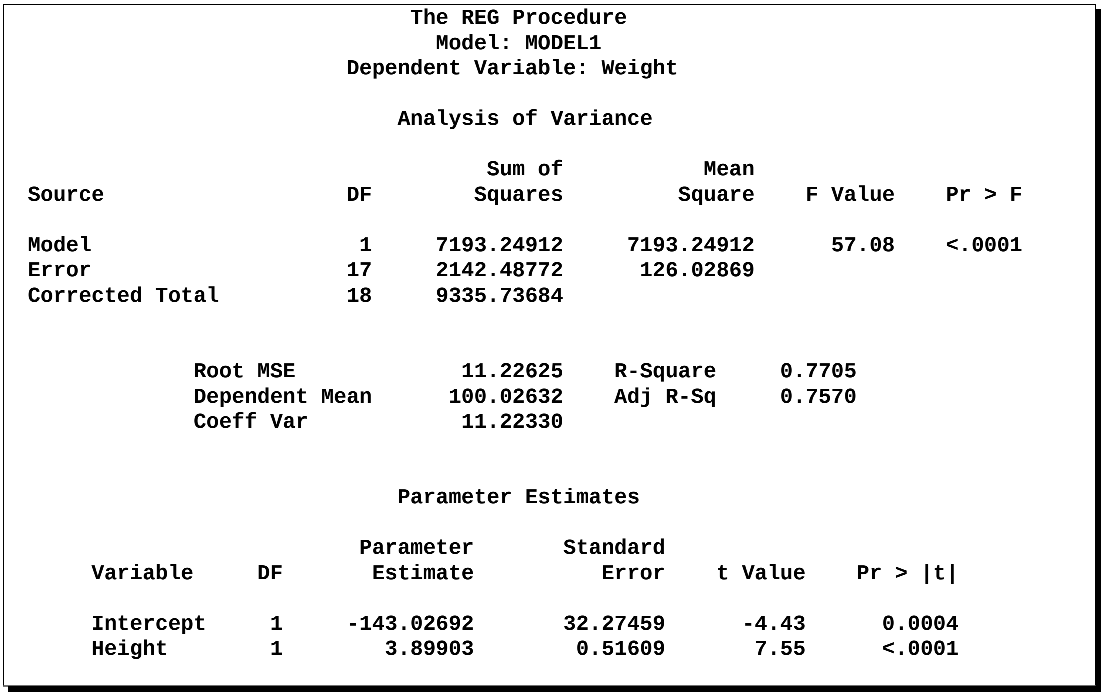
Weight is the response variable.

β_0, β_1 are the unknown parameters.

Height	is the regressor variable.
ϵ	is the unknown error.

The data are included in the following program. The results are displayed in Figure 2.1 and Figure 2.2.

```
data class;
   input Name $ Height Weight Age;
   datalines;
Alfred  69.0 112.5 14
Alice   56.5  84.0 13
Barbara 65.3  98.0 13
Carol   62.8 102.5 14
Henry   63.5 102.5 14
James   57.3  83.0 12
Jane    59.8  84.5 12
Janet   62.5 112.5 15
Jeffrey 62.5  84.0 13
John    59.0  99.5 12
Joyce   51.3  50.5 11
Judy    64.3  90.0 14
Louise  56.3  77.0 12
Mary    66.5 112.0 15
Philip  72.0 150.0 16
Robert  64.8 128.0 12
Ronald  67.0 133.0 15
Thomas  57.5  85.0 11
William 66.5 112.0 15
;
symbol1 v=dot c=blue height=3.5pct;
proc reg;
   model Weight=Height;
   plot Weight*Height/cframe=ligr;
run;
```



```
                              The REG Procedure
                                Model: MODEL1
                          Dependent Variable: Weight

                             Analysis of Variance

                                        Sum of           Mean
Source                    DF           Squares         Square    F Value    Pr > F

Model                      1        7193.24912     7193.24912      57.08    <.0001
Error                     17        2142.48772      126.02869
Corrected Total           18        9335.73684


             Root MSE             11.22625    R-Square     0.7705
             Dependent Mean      100.02632    Adj R-Sq     0.7570
             Coeff Var            11.22330


                             Parameter Estimates

                             Parameter       Standard
     Variable      DF         Estimate          Error    t Value    Pr > |t|

     Intercept      1       -143.02692       32.27459      -4.43      0.0004
     Height         1          3.89903        0.51609       7.55      <.0001
```

Figure 2.1. Regression for Weight and Height Data

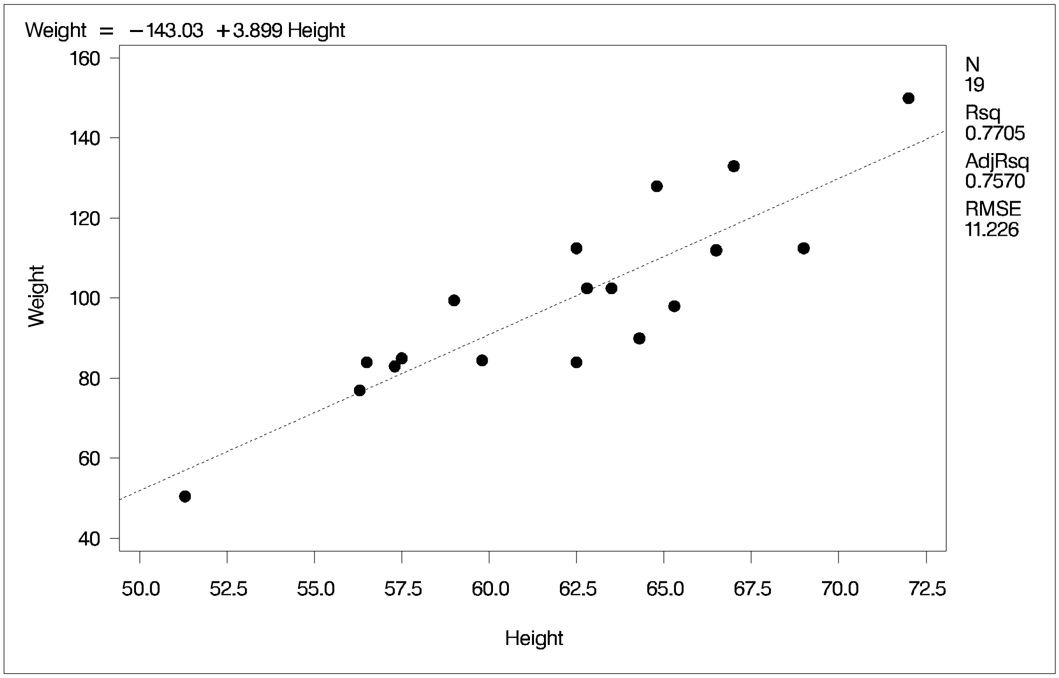


Figure 2.2. Regression for Weight and Height Data

Estimates of β_0 and β_1 for these data are $b_0 = -143.0$ and $b_1 = 3.9$, so the line is described by the equation

$$\text{Weight} = -143.0 + 3.9 * \text{Height}$$

Regression is often used in an exploratory fashion to look for empirical relationships, such as the relationship between Height and Weight. In this example, Height is not the cause of Weight. You would need a controlled experiment to confirm scientifically the relationship. See the "Comments on Interpreting Regression Statistics" section on page 42 for more information.

The method most commonly used to estimate the parameters is to minimize the sum of squares of the differences between the actual response value and the value predicted by the equation. The estimates are called *least-squares estimates*, and the criterion value is called the *error sum of squares*

$$\text{SSE} = \sum_{i=1}^{n} (y_i - b_0 - b_1 x_i)^2$$

where b_0 and b_1 are the estimates of β_0 and β_1 that minimize SSE.

For a general discussion of the theory of least-squares estimation of linear models and its application to regression and analysis of variance, refer to one of the applied regression texts, including Draper and Smith (1981), Daniel and Wood (1980), Johnston (1972), and Weisberg (1985).

SAS/STAT regression procedures produce the following information for a typical regression analysis.

- parameter estimates using the least-squares criterion
- estimates of the variance of the error term
- estimates of the variance or standard deviation of the sampling distribution of the parameter estimates
- tests of hypotheses about the parameters

SAS/STAT regression procedures can produce many other specialized diagnostic statistics, including

- collinearity diagnostics to measure how strongly regressors are related to other regressors and how this affects the stability and variance of the estimates (REG)
- influence diagnostics to measure how each individual observation contributes to determining the parameter estimates, the SSE, and the fitted values (LOGISTIC, REG, RSREG)
- lack-of-fit diagnostics that measure the lack of fit of the regression model by comparing the error variance estimate to another pure error variance that is not dependent on the form of the model (CATMOD, PROBIT, RSREG)

- diagnostic scatter plots that check the fit of the model and highlighted scatter plots that identify particular observations or groups of observations (REG)
- predicted and residual values, and confidence intervals for the mean and for an individual value (GLM, LOGISTIC, REG)
- time-series diagnostics for equally spaced time-series data that measure how much errors may be related across neighboring observations. These diagnostics can also measure functional goodness of fit for data sorted by regressor or response variables (REG, SAS/ETS procedures).

General Regression: The REG Procedure

The REG procedure is a general-purpose procedure for regression that

- handles multiple regression models
- provides nine model-selection methods
- allows interactive changes both in the model and in the data used to fit the model
- allows linear equality restrictions on parameters
- tests linear hypotheses and multivariate hypotheses
- produces collinearity diagnostics, influence diagnostics, and partial regression leverage plots
- saves estimates, predicted values, residuals, confidence limits, and other diagnostic statistics in output SAS data sets
- generates plots of data and of various statistics
- "paints" or highlights scatter plots to identify particular observations or groups of observations
- uses, optionally, correlations or crossproducts for input

Model-selection Methods in PROC REG

The nine methods of model selection implemented in PROC REG are

NONE no selection. This method is the default and uses the full model given in the MODEL statement to fit the linear regression.

FORWARD forward selection. This method starts with no variables in the model and adds variables one by one to the model. At each step, the variable added is the one that maximizes the fit of the model. You can also specify groups of variables to treat as a unit during the selection process. An option enables you to specify the criterion for inclusion.

BACKWARD backward elimination. This method starts with a full model and eliminates variables one by one from the model. At each step, the variable with the smallest contribution to the model is deleted. You can also specify groups of variables to treat as a unit during the selection process. An option enables you to specify the criterion for exclusion.

STEPWISE stepwise regression, forward and backward. This method is a modification of the forward-selection method in that variables already in the model do not necessarily stay there. You can also specify groups of variables to treat as a unit during the selection process. Again, options enable you to specify criteria for entry into the model and for remaining in the model.

MAXR maximum R^2 improvement. This method tries to find the best one-variable model, the best two-variable model, and so on. The MAXR method differs from the STEPWISE method in that many more models are evaluated with MAXR, which considers all switches before making any switch. The STEPWISE method may remove the "worst" variable without considering what the "best" remaining variable might accomplish, whereas MAXR would consider what the "best" remaining variable might accomplish. Consequently, MAXR typically takes much longer to run than STEPWISE.

MINR minimum R^2 improvement. This method closely resembles MAXR, but the switch chosen is the one that produces the smallest increase in R^2.

RSQUARE finds a specified number of models having the highest R^2 in each of a range of model sizes.

CP finds a specified number of models with the lowest C_p within a range of model sizes.

ADJRSQ finds a specified number of models having the highest adjusted R^2 within a range of model sizes.

Nonlinear Regression: The NLIN Procedure

The NLIN procedure implements iterative methods that attempt to find least-squares estimates for nonlinear models. The default method is Gauss-Newton, although several other methods, such as Newton and Marquardt, are available. You must specify parameter names, starting values, and expressions for the model. All necessary analytical derivatives are calculated automatically for you. Grid search is also available to select starting values for the parameters. Since nonlinear models are often difficult to estimate, PROC NLIN may not always find the globally optimal least-squares estimates.

Response Surface Regression: The RSREG Procedure

The RSREG procedure fits a quadratic response-surface model, which is useful in searching for factor values that optimize a response. The following features in PROC RSREG make it preferable to other regression procedures for analyzing response surfaces:

- automatic generation of quadratic effects
- a lack-of-fit test
- solutions for critical values of the surface
- eigenvalues of the associated quadratic form
- a ridge analysis to search for the direction of optimum response

Partial Least Squares Regression: The PLS Procedure

The PLS procedure fits models using any one of a number of linear predictive methods, including *partial least squares* (PLS). Ordinary least-squares regression, as implemented in SAS/STAT procedures such as PROC GLM and PROC REG, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for *new* observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called *factors* (also called *components* or *latent vectors*), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation.

Regression for Ill-conditioned Data: The ORTHOREG Procedure

The ORTHOREG procedure performs linear least-squares regression using the Gentleman-Givens computational method, and it can produce more accurate parameter estimates for ill-conditioned data. PROC GLM and PROC REG produce very accurate estimates for most problems. However, if you have very ill-conditioned data, consider using the ORTHOREG procedure. The collinearity diagnostics in PROC REG can help you to determine whether PROC ORTHOREG would be useful.

Local Regression: The LOESS Procedure

The LOESS procedure implements a nonparametric method for estimating regression surfaces pioneered by Cleveland, Devlin, and Grosse (1988). The LOESS procedure allows great flexibility because no assumptions about the parametric form of the regression surface are needed. Furthermore, the LOESS procedure is suitable when there are outliers in the data and a robust fitting method is necessary.

Robust Regression: The ROBUSTREG Procedure

The ROBUSTREG procedure implements algorithms to detect outliers and provide resistant (stable) results in the presence of outliers. The ROBUSTREG procedure provides four such methods: M estimation, LTS estimation, S estimation, and MM estimation.

- M estimation was introduced by Huber (1973), and it is the simplest approach both computationally and theoretically. Although it is not robust with respect to leverage points, it is still used extensively in analyzing data for which it can be assumed that the contamination is mainly in the response direction.
- Least Trimmed Squares (LTS) estimation is a high breakdown value method introduced by Rousseeuw (1984). The breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness.
- S estimation is a high breakdown value method introduced by Rousseeuw and Yohai (1984). With the same breakdown value, it has a higher statistical efficiency than LTS estimation.
- MM estimation, introduced by Yohai (1987), combines high breakdown value estimation and M estimation. It has both the high breakdown property and a higher statistical efficiency than S estimation.

Logistic Regression: The LOGISTIC Procedure

The LOGISTIC procedure fits logistic models, in which the response can be either dichotomous or polychotomous. Stepwise model selection is available. You can request regression diagnostics, and predicted and residual values.

Regression with Transformations: The TRANSREG Procedure

The TRANSREG procedure can fit many standard linear models. In addition, PROC TRANSREG can find nonlinear transformations of data and fit a linear model to the transformed variables. This is in contrast to PROC REG and PROC GLM, which fit linear models to data, or PROC NLIN, which fits nonlinear models to data. The TRANSREG procedure fits many types of linear models, including

- ordinary regression and ANOVA

- metric and nonmetric conjoint analysis
- metric and nonmetric vector and ideal point preference mapping
- simple, multiple, and multivariate regression with variable transformations
- redundancy analysis with variable transformations
- canonical correlation analysis with variable transformations
- response surface regression with variable transformations

Regression Using the GLM, CATMOD, LOGISTIC, PROBIT, and LIFEREG Procedures

The GLM procedure fits general linear models to data, and it can perform regression, analysis of variance, analysis of covariance, and many other analyses. The following features for regression distinguish PROC GLM from other regression procedures:

- direct specification of polynomial effects
- ease of specifying categorical effects (PROC GLM automatically generates dummy variables for class variables)

Most of the statistics based on predicted and residual values that are available in PROC REG are also available in PROC GLM. However, PROC GLM does not produce collinearity diagnostics, influence diagnostics, or scatter plots. In addition, PROC GLM allows only one model and fits the full model.

See Chapter 3, “Introduction to Analysis-of-Variance Procedures,” and Chapter 32, “The GLM Procedure,” for more details.

The CATMOD procedure can perform linear regression and logistic regression of response functions for data that can be represented in a contingency table. See Chapter 4, “Introduction to Categorical Data Analysis Procedures,” and Chapter 22, “The CATMOD Procedure,” for more details.

The LOGISTIC and PROBIT procedures can perform logistic and ordinal logistic regression. See Chapter 4, “Introduction to Categorical Data Analysis Procedures,” Chapter 42, “The LOGISTIC Procedure,” and Chapter 60, “The PROBIT Procedure,” for additional details.

The LIFEREG procedure is useful in fitting equations to data that may be right-censored. See Chapter 9, “Introduction to Survival Analysis Procedures,” and Chapter 39, “The LIFEREG Procedure,” for more details.

Interactive Features in the CATMOD, GLM, and REG Procedures

The CATMOD, GLM, and REG procedures do not stop after processing a RUN statement. More statements can be submitted as a continuation of the previous statements. Many new features in these procedures are useful to request after you have reviewed the results from previous statements. The procedures stop if a DATA step or another procedure is requested or if a QUIT statement is submitted.

Statistical Background

The rest of this chapter outlines the way many SAS/STAT regression procedures calculate various regression quantities. Exceptions and further details are documented with individual procedures.

Linear Models

In matrix algebra notation, a linear model is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{X}$ is the $n \times k$ design matrix (rows are observations and columns are the regressors), $\boldsymbol{\beta}$ is the $k \times 1$ vector of unknown parameters, and $\boldsymbol{\epsilon}$ is the $n \times 1$ vector of unknown errors. The first column of $\mathbf{X}$ is usually a vector of 1s used in estimating the intercept term.

The statistical theory of linear models is based on strict classical assumptions. Ideally, the response is measured with all the factors controlled in an experimentally determined environment. If you cannot control the factors experimentally, some tests must be interpreted as being conditional on the observed values of the regressors.

Other assumptions are that

- the form of the model is correct (all important explanatory variables have been included)
- regressor variables are measured without error
- the expected value of the errors is zero
- the variance of the error (and thus the dependent variable) for the i^{th} observation is σ^2/w_i, where w_i is a known weight factor. Usually, $w_i = 1$ for all i and thus σ^2 is the common, constant variance.
- the errors are uncorrelated across observations

When hypotheses are tested, the additional assumption is made that the errors are normally distributed.

Statistical Model

If the model satisfies all the necessary assumptions, the least-squares estimates are the best linear unbiased estimates (BLUE). In other words, the estimates have minimum variance among the class of estimators that are unbiased and are linear functions of the responses. If the additional assumption that the error term is normally distributed is also satisfied, then

- the statistics that are computed have the proper sampling distributions for hypothesis testing

- parameter estimates are normally distributed
- various sums of squares are distributed proportional to chi-square, at least under proper hypotheses
- ratios of estimates to standard errors are distributed as Student's t under certain hypotheses
- appropriate ratios of sums of squares are distributed as F under certain hypotheses

When regression analysis is used to model data that do not meet the assumptions, the results should be interpreted in a cautious, exploratory fashion. The significance probabilities under these circumstances are unreliable.

Box (1966) and Mosteller and Tukey (1977, chaps. 12 and 13) discuss the problems that are encountered with regression data, especially when the data are not under experimental control.

Parameter Estimates and Associated Statistics

Parameter estimates are formed using least-squares criteria by solving the normal equations

$$(\mathbf{X}'W\mathbf{X})\mathbf{b} = \mathbf{X}'W\mathbf{y}$$

for the parameter estimates $\mathbf{b}$, where W is a diagonal matrix with the observed weights on the diagonal, yielding

$$\mathbf{b} = (\mathbf{X}'W\mathbf{X})^{-1}\mathbf{X}'W\mathbf{y}$$

Assume for the present that $\mathbf{X}'W\mathbf{X}$ has full column rank k (this assumption is relaxed later). The variance of the error σ^2 is estimated by the mean square error

$$s^2 = \text{MSE} = \frac{\text{SSE}}{n-k} = \frac{1}{n-k}\sum_{i=1}^{n} w_i\left(y_i - \mathbf{x}_i\mathbf{b}\right)^2$$

where $\mathbf{x}_i$ is the ith row of regressors. The parameter estimates are unbiased:

$$\begin{aligned} E(\mathbf{b}) &= \boldsymbol{\beta} \\ E(s^2) &= \sigma^2 \end{aligned}$$

The covariance matrix of the estimates is

$$\text{VAR}(\mathbf{b}) = (\mathbf{X}'W\mathbf{X})^{-1}\sigma^2$$

The estimate of the covariance matrix is obtained by replacing σ^2 with its estimate, s^2, in the formula preceding:

$$\text{COVB} = (\mathbf{X}'W\mathbf{X})^{-1}s^2$$

The correlations of the estimates are derived by scaling to 1s on the diagonal.

Let

$$\begin{aligned}\mathbf{S} &= \text{diag}\left((\mathbf{X}'W\mathbf{X})^{-1}\right)^{-\frac{1}{2}} \\ \text{CORRB} &= \mathbf{S}\left(\mathbf{X}'W\mathbf{X}\right)^{-1}\mathbf{S}\end{aligned}$$

Standard errors of the estimates are computed using the equation

$$\text{STDERR}(b_i) = \sqrt{(\mathbf{X}'W\mathbf{X})^{-1}_{ii}s^2}$$

where $(\mathbf{X}'W\mathbf{X})^{-1}_{ii}$ is the ith diagonal element of $(\mathbf{X}'W\mathbf{X})^{-1}$. The ratio

$$t = \frac{b_i}{\text{STDERR}(b_i)}$$

is distributed as Student's t under the hypothesis that β_i is zero. Regression procedures display the t ratio and the significance probability, which is the probability under the hypothesis $\beta_i = 0$ of a larger absolute t value than was actually obtained. When the probability is less than some small level, the event is considered so unlikely that the hypothesis is rejected.

Type I SS and Type II SS measure the contribution of a variable to the reduction in SSE. Type I SS measure the reduction in SSE as that variable is entered into the model in sequence. Type II SS are the increment in SSE that results from removing the variable from the full model. Type II SS are equivalent to the Type III and Type IV SS reported in the GLM procedure. If Type II SS are used in the numerator of an F test, the test is equivalent to the t test for the hypothesis that the parameter is zero. In polynomial models, Type I SS measure the contribution of each polynomial term after it is orthogonalized to the previous terms in the model. The four types of SS are described in Chapter 11, "The Four Types of Estimable Functions."

Standardized estimates are defined as the estimates that result when all variables are standardized to a mean of 0 and a variance of 1. Standardized estimates are computed by multiplying the original estimates by the sample standard deviation of the regressor variable and dividing by the sample standard deviation of the dependent variable.

R^2 is an indicator of how much of the variation in the data is explained by the model. It is defined as

$$R^2 = 1 - \frac{\text{SSE}}{\text{TSS}}$$

where SSE is the sum of squares for error and TSS is the corrected total sum of squares. The Adjusted R^2 statistic is an alternative to R^2 that is adjusted for the number of parameters in the model. This is calculated as

$$\text{ADJRSQ} = 1 - \frac{n-i}{n-p}\left(1 - R^2\right)$$

where n is the number of observations used to fit the model, p is the number of parameters in the model (including the intercept), and i is 1 if the model includes an intercept term, and 0 otherwise.

Tolerances and variance inflation factors measure the strength of interrelationships among the regressor variables in the model. If all variables are orthogonal to each other, both tolerance and variance inflation are 1. If a variable is very closely related to other variables, the tolerance goes to 0 and the variance inflation gets very large. Tolerance (TOL) is 1 minus the R^2 that results from the regression of the other variables in the model on that regressor. Variance inflation (VIF) is the diagonal of $(\mathbf{X}'W\mathbf{X})^{-1}$ if $(\mathbf{X}'W\mathbf{X})$ is scaled to correlation form. The statistics are related as

$$\text{VIF} = \frac{1}{\text{TOL}}$$

Models Not of Full Rank

If the model is not full rank, then a generalized inverse can be used to solve the normal equations to minimize the SSE:

$$\mathbf{b} = (\mathbf{X}'W\mathbf{X})^{-}\mathbf{X}'W\mathbf{y}$$

However, these estimates are not unique since there are an infinite number of solutions using different generalized inverses. PROC REG and other regression procedures choose a nonzero solution for all variables that are linearly independent of previous variables and a zero solution for other variables. This corresponds to using a generalized inverse in the normal equations, and the expected values of the estimates are the Hermite normal form of $\mathbf{X}'W\mathbf{X}$ multiplied by the true parameters:

$$E(\mathbf{b}) = (\mathbf{X}'W\mathbf{X})^{-}(\mathbf{X}'W\mathbf{X})\boldsymbol{\beta}$$

Degrees of freedom for the zeroed estimates are reported as zero. The hypotheses that are not testable have t tests displayed as missing. The message that the model is not full rank includes a display of the relations that exist in the matrix.

Comments on Interpreting Regression Statistics

In most applications, regression models are merely useful approximations. Reality is often so complicated that you cannot know what the true model is. You may have to choose a model more on the basis of what variables can be measured and what kinds of models can be estimated than on a rigorous theory that explains how the universe really works. However, even in cases where theory is lacking, a regression model may be an excellent predictor of the response if the model is carefully formulated from a large sample. The interpretation of statistics such as parameter estimates may nevertheless be highly problematical.

Statisticians usually use the word “prediction” in a technical sense. *Prediction* in this sense does not refer to “predicting the future” (statisticians call that *forecasting*) but rather to guessing the response from the values of the regressors in an observation taken under the same circumstances as the sample from which the regression equation was estimated. If you developed a regression model for predicting consumer preferences in 1958, it may not give very good predictions in 1988 no matter how well it did in 1958. If it is the future you want to predict, your model must include whatever relevant factors may change over time. If the process you are studying does in fact change over time, you must take observations at several, perhaps many, different times. Analysis of such data is the province of SAS/ETS procedures such as AUTOREG and STATESPACE. Refer to the *SAS/ETS User’s Guide* for more information on these procedures.

The comments in the rest of this section are directed toward linear least-squares regression. Nonlinear regression and non-least-squares regression often introduce further complications. For more detailed discussions of the interpretation of regression statistics, see Darlington (1968), Mosteller and Tukey (1977), Weisberg (1985), and Younger (1979).

Interpreting Parameter Estimates from a Controlled Experiment

Parameter estimates are easiest to interpret in a controlled experiment in which the regressors are manipulated independently of each other. In a well-designed experiment, such as a randomized factorial design with replications in each cell, you can use lack-of-fit tests and estimates of the standard error of prediction to determine whether the model describes the experimental process with adequate precision. If so, a regression coefficient estimates the amount by which the mean response changes when the regressor is changed by one unit while all the other regressors are unchanged. However, if the model involves interactions or polynomial terms, it may not be possible to interpret individual regression coefficients. For example, if the equation includes both linear and quadratic terms for a given variable, you cannot physically change the value of the linear term without also changing the value of the quadratic term. Sometimes it may be possible to recode the regressors, for example by using orthogonal polynomials, to make the interpretation easier.

If the nonstatistical aspects of the experiment are also treated with sufficient care (including such things as use of placebos and double blinds), then you can state conclusions in causal terms; that is, this change in a regressor causes that change in the response. Causality can never be inferred from statistical results alone or from an observational study.

If the model that you fit is not the true model, then the parameter estimates may depend strongly on the particular values of the regressors used in the experiment. For example, if the response is actually a quadratic function of a regressor but you fit a linear function, the estimated slope may be a large negative value if you use only small values of the regressor, a large positive value if you use only large values of the regressor, or near zero if you use both large and small regressor values. When you report the results of an experiment, it is important to include the values of the regressors. It is also important to avoid extrapolating the regression equation outside the range of regressors in the sample.

Interpreting Parameter Estimates from an Observational Study

In an observational study, parameter estimates can be interpreted as the expected difference in response of two observations that differ by one unit on the regressor in question and that have the same values for all other regressors. You cannot make inferences about "changes" in an observational study since you have not actually changed anything. It may not be possible even in principle to change one regressor independently of all the others. Neither can you draw conclusions about causality without experimental manipulation.

If you conduct an observational study and if you do not know the true form of the model, interpretation of parameter estimates becomes even more convoluted. A coefficient must then be interpreted as an average over the sampled population of expected differences in response of observations that differ by one unit on only one regressor. The considerations that are discussed under controlled experiments for which the true model is not known also apply.

Comparing Parameter Estimates

Two coefficients in the same model can be directly compared only if the regressors are measured in the same units. You can make any coefficient large or small just by changing the units. If you convert a regressor from feet to miles, the parameter estimate is multiplied by 5280.

Sometimes standardized regression coefficients are used to compare the effects of regressors measured in different units. Standardizing the variables effectively makes the standard deviation the unit of measurement. This makes sense only if the standard deviation is a meaningful quantity, which usually is the case only if the observations are sampled from a well-defined population. In a controlled experiment, the standard deviation of a regressor depends on the values of the regressor selected by the experimenter. Thus, you can make a standardized regression coefficient large by using a large range of values for the regressor.

In some applications you may be able to compare regression coefficients in terms of the practical range of variation of a regressor. Suppose that each independent variable in an industrial process can be set to values only within a certain range. You can rescale the variables so that the smallest possible value is zero and the largest possible value is one. Then the unit of measurement for each regressor is the maximum possible range of the regressor, and the parameter estimates are comparable in that sense. Another possibility is to scale the regressors in terms of the cost of setting a regressor to a particular value, so comparisons can be made in monetary terms.

Correlated Regressors

In an experiment, you can often select values for the regressors such that the regressors are orthogonal (not correlated with each other). Orthogonal designs have enormous advantages in interpretation. With orthogonal regressors, the parameter estimate for a given regressor does not depend on which other regressors are included in the model, although other statistics such as standard errors and p-values may change.

If the regressors are correlated, it becomes difficult to disentangle the effects of one regressor from another, and the parameter estimates may be highly dependent on which regressors are used in the model. Two correlated regressors may be nonsignificant when tested separately but highly significant when considered together. If two regressors have a correlation of 1.0, it is impossible to separate their effects.

It may be possible to recode correlated regressors to make interpretation easier. For example, if X and Y are highly correlated, they could be replaced in a linear regression by $X+Y$ and $X-Y$ without changing the fit of the model or statistics for other regressors.

Errors in the Regressors

If there is error in the measurements of the regressors, the parameter estimates must be interpreted with respect to the measured values of the regressors, not the true values. A regressor may be statistically nonsignificant when measured with error even though it would have been highly significant if measured accurately.

Probability Values (p-values)

Probability values (p-values) do not necessarily measure the importance of a regressor. An important regressor can have a large (nonsignificant) p-value if the sample is small, if the regressor is measured over a narrow range, if there are large measurement errors, or if another closely related regressor is included in the equation. An unimportant regressor can have a very small p-value in a large sample. Computing a confidence interval for a parameter estimate gives you more useful information than just looking at the p-value, but confidence intervals do not solve problems of measurement errors in the regressors or highly correlated regressors.

The p-values are always approximations. The assumptions required to compute exact p-values are never satisfied in practice.

Interpreting R^2

R^2 is usually defined as the proportion of variance of the response that is predictable from (that can be explained by) the regressor variables. It may be easier to interpret $\sqrt{1-R^2}$, which is approximately the factor by which the standard error of prediction is reduced by the introduction of the regressor variables.

R^2 is easiest to interpret when the observations, including the values of both the regressors and response, are randomly sampled from a well-defined population.

Nonrandom sampling can greatly distort R^2. For example, excessively large values of R^2 can be obtained by omitting from the sample observations with regressor values near the mean.

In a controlled experiment, R^2 depends on the values chosen for the regressors. A wide range of regressor values generally yields a larger R^2 than a narrow range. In comparing the results of two experiments on the same variables but with different ranges for the regressors, you should look at the standard error of prediction (root mean square error) rather than R^2.

Whether a given R^2 value is considered to be large or small depends on the context of the particular study. A social scientist might consider an R^2 of 0.30 to be large, while a physicist might consider 0.98 to be small.

You can always get an R^2 arbitrarily close to 1.0 by including a large number of completely unrelated regressors in the equation. If the number of regressors is close to the sample size, R^2 is very biased. In such cases, the adjusted R^2 and related statistics discussed by Darlington (1968) are less misleading.

If you fit many different models and choose the model with the largest R^2, all the statistics are biased and the p-values for the parameter estimates are not valid. Caution must be taken with the interpretation of R^2 for models with no intercept term. As a general rule, no-intercept models should be fit only when theoretical justification exists and the data appear to fit a no-intercept framework. The R^2 in those cases is measuring something different (refer to Kvalseth 1985).

Incorrect Data Values

All regression statistics can be seriously distorted by a single incorrect data value. A decimal point in the wrong place can completely change the parameter estimates, R^2, and other statistics. It is important to check your data for outliers and influential observations. The diagnostics in PROC REG are particularly useful in this regard.

Predicted and Residual Values

After the model has been fit, predicted and residual values are usually calculated and output. The predicted values are calculated from the estimated regression equation; the residuals are calculated as actual minus predicted. Some procedures can calculate standard errors of residuals, predicted mean values, and individual predicted values.

Consider the ith observation where $\mathbf{x}_i$ is the row of regressors, $\mathbf{b}$ is the vector of parameter estimates, and s^2 is the mean squared error.

Let

$$h_i = w_i \mathbf{x}_i (\mathbf{X}' W \mathbf{X})^{-1} \mathbf{x}_i' \ \ \text{(the leverage)}$$

where $\mathbf{X}$ is the design matrix for the observed data, $\mathbf{x}_i$ is an arbitrary regressor vector (possibly but not necessarily a row of $\mathbf{X}$), W is a diagonal matrix with the observed weights on the diagonal, and w_i is the weight corresponding to $\mathbf{x}_i$.

Then

$$\begin{aligned}\hat{y}_i &= \mathbf{x}_i\mathbf{b} \ \text{ (the predicted mean value)} \\ \text{STDERR}(\hat{y}_i) &= \sqrt{h_i s^2/w_i} \ \text{ (the standard error of the predicted mean)}\end{aligned}$$

The standard error of the individual (future) predicted value y_i is

$$\text{STDERR}(y_i) = \sqrt{(1+h_i)s^2/w_i}$$

If the predictor vector $\mathbf{x}_i$ corresponds to an observation in the analysis data, then the residual for that observation is defined as

$$\begin{aligned}\text{RESID}_i &= y_i - \mathbf{x}_i\mathbf{b} \ \text{ (the residual)} \\ \text{STDERR}(\text{RESID}_i) &= \sqrt{(1-h_i)s^2/w_i} \ \text{ (the standard error of the residual)}\end{aligned}$$

The ratio of the residual to its standard error, called the *studentized residual*, is sometimes shown as

$$\text{STUDENT}_i = \frac{\text{RESID}_i}{\text{STDERR}(\text{RESID}_i)}$$

There are two kinds of confidence intervals for predicted values. One type of confidence interval is an interval for the mean value of the response. The other type, sometimes called a *prediction* or *forecasting interval*, is an interval for the actual value of a response, which is the mean value plus error.

For example, you can construct for the ith observation a confidence interval that contains the true mean value of the response with probability $1-\alpha$. The upper and lower limits of the confidence interval for the mean value are

$$\begin{aligned}\text{LowerM} &= \mathbf{x}_i\mathbf{b} - t_{\alpha/2}\sqrt{h_i s^2/w_i} \\ \text{UpperM} &= \mathbf{x}_i\mathbf{b} + t_{\alpha/2}\sqrt{h_i s^2/w_i}\end{aligned}$$

where $t_{\alpha/2}$ is the tabulated t statistic with degrees of freedom equal to the degrees of freedom for the mean squared error.

The limits for the confidence interval for an actual individual response are

$$\begin{aligned}\text{LowerI} &= \mathbf{x}_i\mathbf{b} - t_{\alpha/2}\sqrt{(1+h_i)s^2/w_i} \\ \text{UpperI} &= \mathbf{x}_i\mathbf{b} + t_{\alpha/2}\sqrt{(1+h_i)s^2/w_i}\end{aligned}$$

Influential observations are those that, according to various criteria, appear to have a large influence on the parameter estimates. One measure of influence, Cook's D, measures the change to the estimates that results from deleting each observation:

$$\text{COOKD} = \frac{1}{k}\text{STUDENT}^2 \left(\frac{\text{STDERR}(\hat{y})}{\text{STDERR}(\text{RESID})} \right)^2$$

where k is the number of parameters in the model (including the intercept). For more information, refer to Cook (1977, 1979).

The *predicted residual* for observation i is defined as the residual for the ith observation that results from dropping the ith observation from the parameter estimates. The sum of squares of predicted residual errors is called the *PRESS statistic*:

$$\begin{aligned} \text{PRESID}_i &= \frac{\text{RESID}_i}{1 - h_i} \\ \text{PRESS} &= \sum_{i=1}^{n} w_i \text{PRESID}_i^2 \end{aligned}$$

Testing Linear Hypotheses

The general form of a linear hypothesis for the parameters is

$$\mathbf{H}_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is $q \times k$, $\boldsymbol{\beta}$ is $k \times 1$, and $\mathbf{c}$ is $q \times 1$. To test this hypothesis, the linear function is taken with respect to the parameter estimates:

$$\mathbf{Lb} - \mathbf{c}$$

This has variance

$$\text{Var}(\mathbf{Lb} - \mathbf{c}) = \mathbf{L}\text{Var}(\mathbf{b})\mathbf{L}' = \mathbf{L}(\mathbf{X}'W\mathbf{X})^{-}\mathbf{L}'\sigma^2$$

where $\mathbf{b}$ is the estimate of $\boldsymbol{\beta}$.

A quadratic form called the *sum of squares due to the hypothesis* is calculated:

$$\text{SS}(\mathbf{Lb} - \mathbf{c}) = (\mathbf{Lb} - \mathbf{c})'(\mathbf{L}(\mathbf{X}'W\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb} - \mathbf{c})$$

If you assume that this is testable, the SS can be used as a numerator of the F test:

$$F = \frac{\text{SS}(\mathbf{Lb} - \mathbf{c})/q}{s^2}$$

This is compared with an F distribution with q and dfe degrees of freedom, where dfe is the degrees of freedom for residual error.

Multivariate Tests

Multivariate hypotheses involve several dependent variables in the form

$$\mathbf{H}_0 : \mathbf{L}\boldsymbol{\beta}\mathbf{M} = \mathbf{d}$$

where $\mathbf{L}$ is a linear function on the regressor side, $\boldsymbol{\beta}$ is a matrix of parameters, $\mathbf{M}$ is a linear function on the dependent side, and $\mathbf{d}$ is a matrix of constants. The special case (handled by PROC REG) in which the constants are the same for each dependent variable is written

$$(\mathbf{L}\boldsymbol{\beta} - \mathbf{cj})\mathbf{M} = \mathbf{0}$$

where $\mathbf{c}$ is a column vector of constants and $\mathbf{j}$ is a row vector of 1s. The special case in which the constants are 0 is

$$\mathbf{L}\boldsymbol{\beta}\mathbf{M} = \mathbf{0}$$

These multivariate tests are covered in detail in Morrison (1976); Timm (1975); Mardia, Kent, and Bibby (1979); Bock (1975); and other works cited in Chapter 5, “Introduction to Multivariate Procedures.”

To test this hypothesis, construct two matrices, $\mathbf{H}$ and $\mathbf{E}$, that correspond to the numerator and denominator of a univariate F test:

$$\begin{aligned}
\mathbf{H} &= \mathbf{M}'(\mathbf{LB} - \mathbf{cj})'(\mathbf{L}(\mathbf{X}'W\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{LB} - \mathbf{cj})\mathbf{M} \\
\mathbf{E} &= \mathbf{M}'\left(\mathbf{Y}'W\mathbf{Y} - \mathbf{B}'(\mathbf{X}'W\mathbf{X})\mathbf{B}\right)\mathbf{M}
\end{aligned}$$

Four test statistics, based on the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ or $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$, are formed. Let λ_i be the ordered eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ (if the inverse exists), and let ξ_i be the ordered eigenvalues of $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$. It happens that $\xi_i = \lambda_i/(1+\lambda_i)$ and $\lambda_i = \xi_i/(1-\xi_i)$, and it turns out that $\rho_i = \sqrt{\xi_i}$ is the ith canonical correlation.

Let p be the rank of $(\mathbf{H}+\mathbf{E})$, which is less than or equal to the number of columns of $\mathbf{M}$. Let q be the rank of $\mathbf{L}(\mathbf{X}'W\mathbf{X})^{-}\mathbf{L}'$. Let v be the error degrees of freedom and $s = \min(p,q)$. Let $m = (|p-q|-1)/2$, and let $n = (v-p-1)/2$. Then the following statistics test the multivariate hypothesis in various ways, and their p-values can be approximated by F distributions. Note that in the special case that the rank of $\mathbf{H}$ is 1, all of these F statistics will be the same and the corresponding p-values will in fact be exact, since in this case the hypothesis is really univariate.

Wilks' Lambda

If

$$\Lambda = \frac{\text{det}(\mathbf{E})}{\text{det}(\mathbf{H}+\mathbf{E})} = \prod_{i=1}^{n} \frac{1}{1+\lambda_i} = \prod_{i=1}^{n}(1-\xi_i)$$

then

$$F = \frac{1-\Lambda^{1/t}}{\Lambda^{1/t}} \cdot \frac{rt-2u}{pq}$$

is approximately F, where

$$\begin{aligned} r &= v - \frac{p-q+1}{2} \\ u &= \frac{pq-2}{4} \\ t &= \begin{cases} \sqrt{\frac{p^2q^2-4}{p^2+q^2-5}} & \text{if } p^2+q^2-5>0 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

The degrees of freedom are pq and $rt-2u$. The distribution is exact if $\min(p,q) \leq 2$. (Refer to Rao 1973, p. 556.)

Pillai's Trace

If

$$\mathbf{V} = \text{trace}\left(\mathbf{H}(\mathbf{H}+\mathbf{E})^{-1}\right) = \sum_{i=1}^{n} \frac{\lambda_i}{1+\lambda_i} = \sum_{i=1}^{n} \xi_i$$

then

$$F = \frac{2n+s+1}{2m+s+1} \cdot \frac{\mathbf{V}}{s-\mathbf{V}}$$

is approximately F with $s(2m+s+1)$ and $s(2n+s+1)$ degrees of freedom.

Hotelling-Lawley Trace

If

$$\mathbf{U} = \text{trace}\left(\mathbf{E}^{-1}\mathbf{H}\right) = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \frac{\xi_i}{1-\xi_i}$$

then for $n > 0$

$$F = (\mathbf{U}/c)((4 + (pq + 2)/(b - 1))/(pq))$$

is approximately F with pq and $4 + (pq + 2)/(b - 1)$ degrees of freedom, where $b = (p + 2n)(q + 2n)/(2(2n + 1)(n - 1))$ and $c = (2 + (pq + 2)/(b - 1))/(2n)$; while for $n \leq 0$

$$F = \frac{2(sn + 1)\mathbf{U}}{s^2(2m + s + 1)}$$

is approximately F with $s(2m + s + 1)$ and $2(sn + 1)$ degrees of freedom.

Roy's Maximum Root

If

$$\Theta = \lambda_1$$

then

$$F = \Theta\frac{v - r + q}{r}$$

where $r = \max(p, q)$ is an upper bound on F that yields a lower bound on the significance level. Degrees of freedom are r for the numerator and $v - r + q$ for the denominator.

Tables of critical values for these statistics are found in Pillai (1960).

Exact Multivariate Tests

Beginning with release 9.0 of SAS/STAT software, if you specify the MSTAT=EXACT option on the appropriate statement, *p*-values for three of the four tests are computed exactly (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root), and the *p*-values for the fourth (Pillai's trace) are based on an F-approximation that is more accurate (but occasionally slightly more liberal) than the default. The exact *p*-values for Roy's Greatest Root give an especially dramatic improvement, since in this case the F-approximation only provides a lower bound for the p-value. If you use the F-based p-value for this test in the usual way, declaring a test significant if $p < 0.05$, then your decisions may be very liberal. For example, instead of the nominal 5% Type I error rate, such a procedure can easily have an actual Type I error rate in excess of 30%. By contrast, basing such a procedure on the exact *p*-values will result in the appropriate 5% Type I error rate, under the usual regression assumptions.

The exact p-values are based on the following sources:

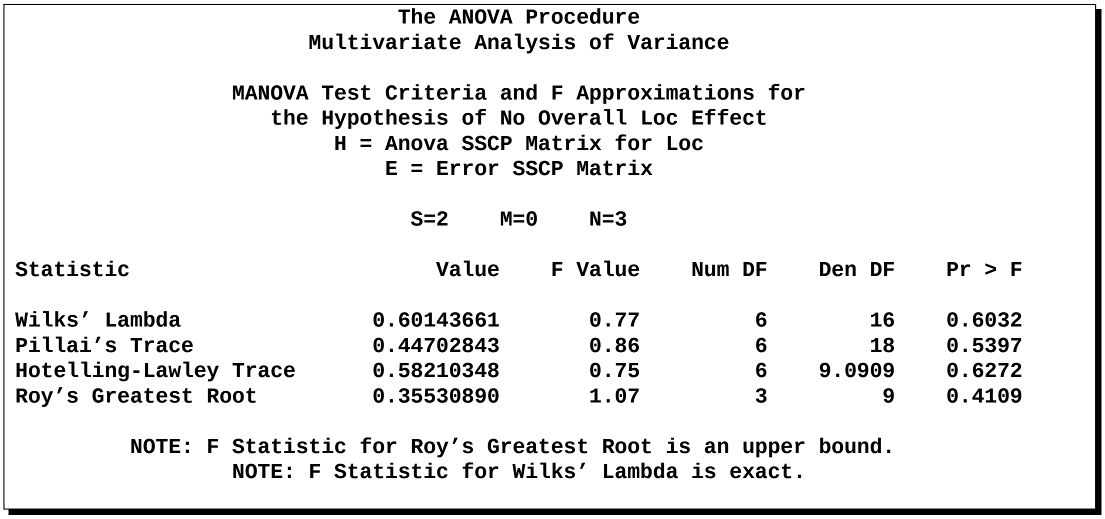
- **Wilks' Lambda:** Lee (1972), Davis (1979)
- **Pillai's Trace:** Muller (1998)
- **Hotelling-Lawley Trace:** Davis (1970), Davis (1980)
- **Roy's Greatest Root:** Davis (1972), Pillai and Flury (1984)

Note that although the MSTAT=EXACT p-value for Pillai's Trace is still approximate, it has "substantially greater accuracy" than the default approximation (Muller 1998).

Since most of the MSTAT=EXACT p-values are not based on the F-distribution, the columns in the multivariate tests table corresponding to this approximation—in particular, the F value and the numerator and denominator degrees of freedom—are no longer displayed, and the column containing the p-values is labeled "P Value" instead of "Pr > F". Thus, for example, suppose you use the following PROC ANOVA code to perform a multivariate analysis of an archaeological data set:

```
data Skulls;
   input Loc $20. Basal Occ Max;
datalines;
Minas Graes, Brazil  2.068 2.070 1.580
Minas Graes, Brazil  2.068 2.074 1.602
Minas Graes, Brazil  2.090 2.090 1.613
Minas Graes, Brazil  2.097 2.093 1.613
Minas Graes, Brazil  2.117 2.125 1.663
Minas Graes, Brazil  2.140 2.146 1.681
Matto Grosso, Brazil 2.045 2.054 1.580
Matto Grosso, Brazil 2.076 2.088 1.602
Matto Grosso, Brazil 2.090 2.093 1.643
Matto Grosso, Brazil 2.111 2.114 1.643
Santa Cruz, Bolivia  2.093 2.098 1.653
Santa Cruz, Bolivia  2.100 2.106 1.623
Santa Cruz, Bolivia  2.104 2.101 1.653
;

proc anova data=Skulls;
   class Loc;
   model Basal Occ Max = Loc / nouni;
   manova h=Loc;
   ods select MultStat;
run;
```

The default multivariate tests, based on the F-approximations, are shown in Figure 2.3.



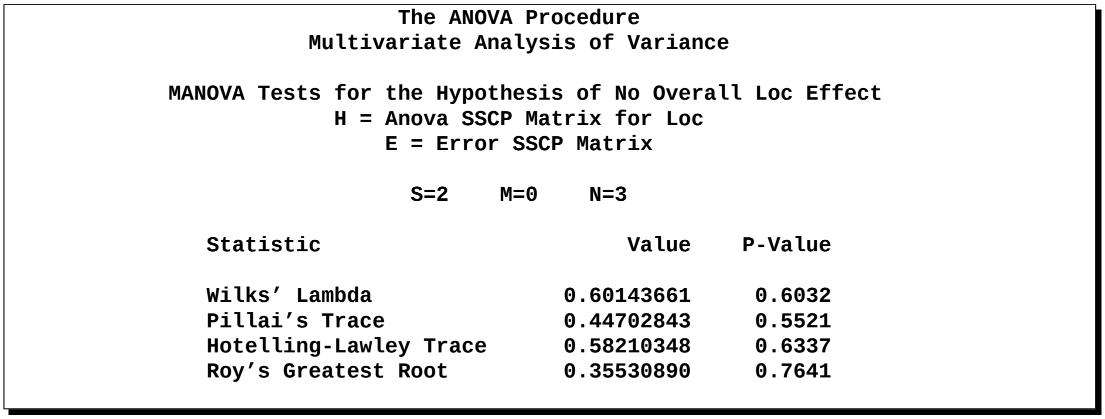
```
                          The ANOVA Procedure
                   Multivariate Analysis of Variance

             MANOVA Test Criteria and F Approximations for
                the Hypothesis of No Overall Loc Effect
                     H = Anova SSCP Matrix for Loc
                         E = Error SSCP Matrix

                           S=2    M=0    N=3

Statistic                        Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda               0.60143661       0.77         6        16    0.6032
Pillai's Trace              0.44702843       0.86         6        18    0.5397
Hotelling-Lawley Trace      0.58210348       0.75         6    9.0909    0.6272
Roy's Greatest Root         0.35530890       1.07         3         9    0.4109

         NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                 NOTE: F Statistic for Wilks' Lambda is exact.
```

Figure 2.3. Default Multivariate Tests

If you specify MSTAT=EXACT on the MANOVA statement

```
proc anova data=Skulls;
   class Loc;
   model Basal Occ Max = Loc / nouni;
   manova h=Loc / mstat=exact;
   ods select MultStat;
run;
```

then the displayed output is the much simpler table shown in Figure 2.4.



```
                          The ANOVA Procedure
                   Multivariate Analysis of Variance

          MANOVA Tests for the Hypothesis of No Overall Loc Effect
                     H = Anova SSCP Matrix for Loc
                         E = Error SSCP Matrix

                           S=2    M=0    N=3

               Statistic                        Value    P-Value

               Wilks' Lambda               0.60143661     0.6032
               Pillai's Trace              0.44702843     0.5521
               Hotelling-Lawley Trace      0.58210348     0.6337
               Roy's Greatest Root         0.35530890     0.7641
```

Figure 2.4. Multivariate Tests with MSTAT=EXACT

Notice that the p-value for Roy's Greatest Root is substantially larger in the new table, and correspondingly more in line with the p-values for the other tests.

If you reference the underlying ODS output object for the table of multivariate statistics, it is important to note that its structure does not depend on the value of the

MSTAT= specification. In particular, it always contains columns corresponding to both the default MSTAT=FAPPROX and the MSTAT=EXACT tests. Moreover, since the MSTAT=FAPPROX tests are relatively cheap to compute, the columns corresponding to them are always filled in, even though they are not displayed when you specify MSTAT=EXACT. On the other hand, for MSTAT=FAPPROX (which is the default), the column of exact p-values contains missing values, and is not displayed.

References

Allen, D.M. (1971), "Mean Square Error of Prediction as a Criterion for Selecting Variables," *Technometrics*, 13, 469–475.

Allen, D.M. and Cady, F.B. (1982), *Analyzing Experimental Data by Regression*, Belmont, CA: Lifetime Learning Publications.

Belsley, D.A., Kuh, E., and Welsch, R.E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons, Inc.

Bock, R.D. (1975), *Multivariate Statistical Methods in Behavioral Research*, New York: McGraw-Hill Book Co.

Box, G.E.P. (1966), "The Use and Abuse of Regression," *Technometrics*, 8, 625–629.

Cleveland, W.S., Devlin, S.J., and Grosse, E. (1988), "Regression by Local Fitting," *Journal of Econometrics*, 37, 87–114.

Cook, R.D. (1977), "Detection of Influential Observations in Linear Regression," *Technometrics*, 19, 15–18.

Cook, R.D. (1979), "Influential Observations in Linear Regression," *Journal of the American Statistical Association*, 74, 169–174.

Daniel, C. and Wood, F. (1980), *Fitting Equations to Data*, Revised Edition, New York: John Wiley & Sons, Inc.

Darlington, R.B. (1968), "Multiple Regression in Psychological Research and Practice," *Psychological Bulletin*, 69, 161–182.

Davis, A. W. (1970), "Differential Equation of Hotelling's Generalized T^2," *Annals of Statistics*, 39, 815–832.

Davis, A. W. (1972), "On the Marginal Distributions of the Latent Roots of the Multivariate Beta Matrix," *Biometrika*, 43, 1664–1670.

Davis, A. W. (1979), "On the Differential Equation for Meijer $G^{0,0}_{p,p}$ Function, and Further Wilks's Likelihood Ratio Criterion," *Biometrika*, 66, 519–531.

Davis, A. W. (1980), "Further Tabulation of Hotelling's Generalized T^2," *Communications in Statistics, Part B*, 9, 321–336.

Draper, N. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Durbin, J. and Watson, G.S. (1951), "Testing for Serial Correlation in Least Squares Regression," *Biometrika*, 37, 409–428.

Freund, R.J., Littell, R.C., and Spector P.C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Freund, R.J. and Littell, R.C. (1986), *SAS System for Regression, 1986 Edition*, Cary, NC: SAS Institute Inc.

Goodnight, J.H. (1979), "A Tutorial on the SWEEP Operator," *The American Statistician*, 33, 149−158. (Also available as SAS Technical Report R-106, *The Sweep Operator: Its Importance in Statistical Computing*, Cary, NC: SAS Institute Inc.)

Hawkins, D.M. (1980), "A Note on Fitting a Regression With No Intercept Term," *The American Statistician*, 34, 233.

Hosmer, D.W, Jr and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons, Inc.

Huber, P.J. (1973), "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Annals of Statistics*, 1, 799-821.

Johnston, J. (1972), *Econometric Methods*, New York: McGraw-Hill Book Co.

Kennedy, W.J. and Gentle, J.E. (1980), *Statistical Computing*, New York: Marcel Dekker, Inc.

Kvalseth, T.O. (1985), "Cautionary Note About R^2," *The American Statistician*, 39, 279.

Lee, Y. (1972), "Some Results on the Distribution of Wilk's Likelihood Ratio Criterion," *Biometrika*, 95, 649.

Mallows, C.L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–75.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis*, London: Academic Press.

Morrison, D.F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill Book Co.

Mosteller, F. and Tukey, J.W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley Publishing Co., Inc.

Muller, K. (1998), "A New F Approximation for the Pillai-Bartlett Trace Under H0," *Journal of Computational and Graphical Statistics*, 7, 131–137.

Neter, J. and Wasserman, W. (1974), *Applied Linear Statistical Models*, Homewood, IL: Irwin.

Pillai, K.C.S. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of Philippines.

Pillai, K.C.S. and Flury, B.N. (1984), "Percentage Points of the Largest Characteristic Root of the Multivariate Beta Matrix," *Communications in Statistics, Part A*, 13, 2199–2237.

Pindyck, R.S. and Rubinfeld, D.L. (1981), *Econometric Models and Econometric Forecasts*, Second Edition, New York: McGraw-Hill Book Co.

Rao, C.R. (1973), *Linear Statistical Inference and Its Applications*, Second Edition, New York: John Wiley & Sons, Inc.

Rawlings, J.O. (1988), *Applied Regression Analysis: A Research Tool*, Pacific Grove, California: Wadsworth & Brooks/Cole Advanced Books & Software.

Rousseeuw, P.J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871-880.

Rousseeuw, P.J. and Yohai, V. (1984), "Robust Regression by Means of S Estimators," in *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R.D. Martin, Lecture Notes in Statistics, 26, New York: Springer-Verlag, 256-274.

Timm, N.H. (1975), *Multivariate Analysis with Applications in Education and Psychology*, Monterey, CA: Brooks-Cole Publishing Co.

Weisberg, S. (1985), *Applied Linear Regression*, Second Edition. New York: John Wiley & Sons, Inc.

Yohai V.J. (1987), "High Breakdown Point and High Efficiency Robust Estimates for Regression," *Annals of Statistics*, 15, 642-656.

Younger, M.S. (1979), *Handbook for Linear Regression*, North Scituate, MA: Duxbury Press.

Chapter 3
Introduction to Analysis-of-Variance Procedures

Chapter Contents

Chapter 3
Introduction to Analysis-of-Variance Procedures

Overview

This chapter reviews the SAS/STAT software procedures that are used for analysis of variance: GLM, ANOVA, CATMOD, MIXED, NESTED, NPAR1WAY, TRANSREG, TTEST, and VARCOMP. Also discussed are SAS/STAT and SAS/QC software procedures for constructing analysis of variance designs: PLAN, FACTEX, and OPTEX.

The flagship analysis-of-variance procedure is the GLM procedure, which handles most standard problems. The following are descriptions of PROC GLM and other procedures that are used for more specialized situations:

ANOVA performs analysis of variance, multivariate analysis of variance, and repeated measures analysis of variance for *balanced* designs. PROC ANOVA also performs several multiple comparison tests.

CATMOD fits linear models and performs analysis of variance and repeated measures analysis of variance for categorical responses.

GENMOD fits generalized linear models and performs analysis of variance in the generalized linear models framework. The methods are particularly suited for discrete response outcomes.

GLM performs analysis of variance, regression, analysis of covariance, repeated measures analysis, and multivariate analysis of variance. PROC GLM produces several diagnostic measures, performs tests for random effects, provides contrasts and estimates for customized hypothesis tests, performs several multiple comparison tests, and provides tests for means adjusted for covariates.

MIXED performs mixed-model analysis of variance and repeated measures analysis of variance via covariance structure modeling. Using likelihood-based or method-of-moment estimates, PROC MIXED constructs statistical tests and intervals, allows customized contrasts and estimates, and computes empirical Bayes predictions.

NESTED performs analysis of variance and analysis of covariance for purely nested random models.

NPAR1WAY performs nonparametric one-way analysis of rank scores.

TTEST compares the means of two groups of observations.

TRANSREG	fits univariate and multivariate linear models, optionally with spline and other nonlinear transformations.
VARCOMP	estimates variance components for random or mixed models.

The following section presents an overview of some of the fundamental features of analysis of variance. Subsequent sections describe how this analysis is performed with procedures in SAS/STAT software. For more detail, see the chapters for the individual procedures. Additional sources are described in the "References" section on page 67.

Statistical Details for Analysis of Variance

Definitions

Analysis of variance (ANOVA) is a technique for analyzing experimental data in which one or more *response* (or *dependent* or simply Y) variables are measured under various conditions identified by one or more classification variables. The combinations of levels for the classification variables form the cells of the experimental design for the data. For example, an experiment may measure weight change (the dependent variable) for men and women who participated in three different weight-loss programs. The six cells of the design are formed by the six combinations of sex (men, women) and program (A, B, C).

In an analysis of variance, the variation in the response is separated into variation attributable to differences between the classification variables and variation attributable to random error. An analysis of variance constructs tests to determine the significance of the classification effects. A typical goal in an analysis of variance is to compare means of the response variable for various combinations of the classification variables.

An analysis of variance may be written as a linear model. Analysis of variance procedures in SAS/STAT software use the model to predict the response for each observation. The difference between the actual and predicted response is the *residual error*. Most of the procedures fit model parameters that minimize the sum of squares of residual errors. Thus, the method is called *least squares regression*. The variance due to the random error, σ^2, is estimated by the mean squared error (MSE or s^2).

Fixed and Random Effects

The explanatory classification variables in an ANOVA design may represent fixed or random effects. The levels of a classification variable for a fixed effect give all the levels of interest, while the levels of a classification variable for a random effect are typically a subset of levels selected from a population of levels. The following are examples.

- In a large drug trial, the levels that correspond to types of drugs are usually considered to comprise a fixed effect, but the levels corresponding to the various clinics where the drugs are administered comprise a random effect.

- In agricultural experiments, it is common to declare locations (or plots) as random because the levels are chosen randomly from a large population of locations and you assume fertility to vary normally across locations.
- In repeated-measures experiments with people or animals as subjects, subjects are declared random because they are selected from the larger population to which you want to generalize.

A typical assumption is that random effects have values drawn from a normally distributed random process with mean zero and common variance. Effects are declared random when the levels are randomly selected from a large population of possible levels. Inferences are made using only a few levels but can be generalized across the whole population of random effects levels.

The consequence of having random effects in your model is that some observations are no longer uncorrelated but instead have a covariance that depends on the variance of the random effect. In fact, a more general approach to random effect models is to model the covariance between observations.

Tests of Effects

Analysis of variance tests are constructed by comparing independent mean squares. To test a particular null hypothesis, you compute the ratio of two mean squares that have the same expected value under that hypothesis; if the ratio is much larger than 1, then that constitutes significant evidence against the null. In particular, in an analysis-of-variance model with fixed effects only, the expected value of each mean square has two components: quadratic functions of fixed parameters and random variation. For example, for a fixed effect called A, the expected value of its mean square is

$$E(\text{MS(A)}) = \text{Q}(\boldsymbol{\beta}) + \sigma_e^2$$

Under the null hypothesis of no A effect, the fixed portion Q($\boldsymbol{\beta}$) of the expected mean square is zero. This mean square is then compared to another mean square, say MS(E), that is independent of the first and has expected value σ_e^2. The ratio of the two mean squares

$$F = \frac{\text{MS(A)}}{\text{MS(E)}}$$

has the F distribution under the null hypothesis. When the null hypothesis is false, the numerator term has a larger expected value, but the expected value of the denominator remains the same. Thus, large F values lead to rejection of the null hypothesis. The probability of getting an F value at least as large as the one observed given that the null hypothesis is true is called the *significance probability value* (or the p-value). A p-value of less than 0.05, for example, indicates that data with *no* real A effect will yield F values as large as the one observed less than 5% of the time. This is usually considered moderate evidence that there *is* a real A effect. Smaller p-values constitute even stronger evidence. Larger p-values indicate that the effect of interest is less than random noise. In this case, you can conclude either that there is no effect at all or that you do not have enough data to detect the differences being tested.

General Linear Models

An analysis-of-variance model can be written as a linear model, which is an equation that predicts the response as a linear function of parameters and design variables. In general,

$$y_i = \beta_0 x_{0i} + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + \epsilon_i \quad i = 1, 2, \ldots, n$$

where y_i is the response for the ith observation, β_k are unknown parameters to be estimated, and x_{ij} are design variables. Design variables for analysis of variance are indicator variables; that is, they are always either 0 or 1.

The simplest model is to fit a single mean to all observations. In this case there is only one parameter, β_0, and one design variable, x_{0i}, which always has the value of 1:

$$\begin{aligned} y_i &= \beta_0 x_{0i} + \epsilon_i \\ &= \beta_0 + \epsilon_i \end{aligned}$$

The least-squares estimator of β_0 is the mean of the y_i. This simple model underlies all more complex models, and all larger models are compared to this simple mean model. In writing the parameterization of a linear model, β_0 is usually referred to as the *intercept*.

A one-way model is written by introducing an indicator variable for each level of the classification variable. Suppose that a variable A has four levels, with two observations per level. The indicator variables are created as follows:

Intercept	A1	A2	A3	A4
1	1	0	0	0
1	1	0	0	0
1	0	1	0	0
1	0	1	0	0
1	0	0	1	0
1	0	0	1	0
1	0	0	0	1
1	0	0	0	1

The linear model for this example is

$$y_i = \beta_0 + \beta_1 A1_i + \beta_2 A2_i + \beta_3 A3_i + \beta_4 A4_i$$

To construct crossed and nested effects, you can simply multiply out all combinations of the main-effect columns. This is described in detail in "Specification of Effects" in Chapter 32, "The GLM Procedure."

Linear Hypotheses

When models are expressed in the framework of linear models, hypothesis tests are expressed in terms of a linear function of the parameters. For example, you may want to test that $\beta_2 - \beta_3 = 0$. In general, the coefficients for linear hypotheses are some set of Ls:

$$H_0\colon L_0\beta_0 + L_1\beta_1 + \cdots + L_k\beta_k = 0$$

Several of these linear functions can be combined to make one joint test. These tests can be expressed in one matrix equation:

$$H_0\colon \mathbf{L}\boldsymbol{\beta} = 0$$

For each linear hypothesis, a sum of squares (SS) due to that hypothesis can be constructed. These sums of squares can be calculated either as a quadratic form of the estimates

$$\mathrm{SS}(\mathbf{L}\boldsymbol{\beta} = 0) = (\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$$

or, equivalently, as the increase in sums of squares for error (SSE) for the model constrained by the null hypothesis

$$\mathrm{SS}(\mathbf{L}\beta = 0) = \mathrm{SSE(constrained)} - \mathrm{SSE(full)}$$

This SS is then divided by appropriate degrees of freedom and used as a numerator of an F statistic.

Analysis of Variance for Fixed Effect Models

PROC GLM for General Linear Models

The GLM procedure is the flagship tool for analysis of variance in SAS/STAT software. It performs analysis of variance by using least squares regression to fit general linear models, as described in the section “General Linear Models” on page 62. Among the statistical methods available in PROC GLM are regression, analysis of variance, analysis of covariance, multivariate analysis of variance, and partial correlation.

While PROC GLM can handle most common analysis of variance problems, other procedures are more efficient or have more features than PROC GLM for certain specialized analyses, or they can handle specialized models that PROC GLM cannot. Much of the rest of this chapter is concerned with comparing PROC GLM to other procedures.

PROC ANOVA for Balanced Designs

When you design an experiment, you choose how many experimental units to assign to each combination of levels (or cells) in the classification. In order to achieve good statistical properties and simplify the computations, you typically attempt to assign the same number of units to every cell in the design. Such designs are called *balanced designs*.

In SAS/STAT software, you can use the ANOVA procedure to perform analysis of variance for balanced data. The ANOVA procedure performs computations for analysis of variance that assume the balanced nature of the data. These computations are simpler and more efficient than the corresponding general computations performed by PROC GLM. Note that PROC ANOVA can be applied to certain designs that are not balanced in the strict sense of equal numbers of observations for all cells. These additional designs include all one-way models, regardless of how unbalanced the cell counts are, as well as Latin squares, which do not have data in all cells. In general, however, the ANOVA procedure is recommended only for balanced data. **If you use ANOVA to analyze a design that is not balanced, you must assume responsibility for the validity of the output.** You are responsible for recognizing incorrect results, which may include negative values reported for the sums of squares. If you are not certain that your data fit into a balanced design, then you probably need the framework of general linear models in the GLM procedure.

Comparing Group Means with PROC ANOVA and PROC GLM

When you have more than two means to compare, an F test in PROC ANOVA or PROC GLM tells you whether the means are significantly different from each other, but it does not tell you which means differ from which other means.

If you have specific comparisons in mind, you can use the CONTRAST statement in PROC GLM to make these comparisons. However, if you make many comparisons using some given significance level (0.05, for example), you are more likely to make a type 1 error (incorrectly rejecting a hypothesis that the means are equal) simply because you have more chances to make the error.

Multiple comparison methods give you more detailed information about the differences among the means and enable you to control error rates for a multitude of comparisons. A variety of multiple comparison methods are available with the MEANS statement in both the ANOVA and GLM procedures, as well as the LSMEANS statement in the GLM and MIXED procedures. These are described in detail in “Multiple Comparisons” in Chapter 32, “The GLM Procedure.”

PROC TTEST for Comparing Two Groups

If you want to perform an analysis of variance and have only one classification variable with two levels, you can use PROC TTEST. In this special case, the results generated by PROC TTEST are equivalent to the results generated by PROC ANOVA or PROC GLM.

In addition to testing for differences between two groups, PROC TTEST performs a test for unequal variances. You can use PROC TTEST with balanced or unbalanced groups. The PROC NPAR1WAY procedure performs nonparametric analogues to t tests. See Chapter 12, "Introduction to Nonparametric Analysis," for an overview and Chapter 52 for details on PROC NPAR1WAY.

Analysis of Variance for Mixed and Random Effect Models

Just as PROC GLM is the flagship procedure for fixed-effect linear models, the MIXED procedure is the flagship procedure for random- and mixed-effect linear models. PROC MIXED fits a variety of mixed linear models to data and enables you to use these fitted models to make statistical inferences about the data. The default fitting method maximizes the restricted likelihood of the data under the assumption that the data are normally distributed and any missing data are missing at random. This general framework accommodates many common correlated-data methods, including variance component models and repeated measures analyses.

A few other procedures in SAS/STAT software offer limited mixed-linear-model capabilities. PROC GLM fits some random-effects and repeated-measures models, although its methods are based on method-of-moments estimation and a portion of the output applies only to the fixed-effects model. PROC NESTED fits special nested designs and may be useful for large data sets because of its customized algorithms. PROC VARCOMP estimates variance components models, but all of its methods are now available in PROC MIXED. PROC LATTICE fits special balanced lattice designs, but, again, the same models are available in PROC MIXED. In general, PROC MIXED is recommended for nearly all of your linear mixed-model applications.

PROC NLMIXED handles models in which the fixed or random effects enter nonlinearly. It requires that you specify a conditional distribution of the data given the random effects, with available distributions including the normal, binomial, and Poisson. You can alternatively code your own distribution with SAS programming statements. Under a normality assumption for the random effects, PROC NLMIXED performs maximum likelihood estimation via adaptive Gaussian quadrature and a dual quasi-Newton optimization algorithm. Besides standard maximum likelihood results, you can obtain empirical Bayes predictions of the random effects and estimates of arbitrary functions of the parameters with delta-method standard errors. PROC NLMIXED has a wide variety of applications, two of the most common being nonlinear growth curves and overdispersed binomial data.

Analysis of Variance for Categorical Data and Generalized Linear Models

A *categorical variable* is defined as one that can assume only a limited number of values. For example, a person's sex is a categorical variable that can assume one of two values. Variables with levels that simply name a group are said to be measured on a *nominal scale*. Categorical variables can also be measured using an *ordinal scale*, which means that the levels of the variable are ordered in some way. For example, responses to an opinion poll are usually measured on an ordinal scale, with levels ranging from “strongly disagree” to “no opinion” to “strongly agree.”

For two categorical variables, one measured on an ordinal scale and one measured on a nominal scale, you may assign scores to the levels of the ordinal variable and test whether the mean scores for the different levels of the nominal variable are significantly different. This process is analogous to performing an analysis of variance on continuous data, which can be performed by PROC CATMOD. If there are n nominal variables, rather than 1, then PROC CATMOD can do an n-way analysis of variance of the mean scores.

For two categorical variables measured on a nominal scale, you can test whether the distribution of the first variable is significantly different for the levels of the second variable. This process is an analysis of variance of proportions, rather than means, and can be performed by PROC CATMOD. The corresponding n-way analysis of variance can also be performed by PROC CATMOD.

See Chapter 4, “Introduction to Categorical Data Analysis Procedures,” and Chapter 22, “The CATMOD Procedure,” for more information.

The GENMOD procedure uses maximum likelihood estimation to fit generalized linear models. This family includes models for categorical data such as logistic, probit, and complementary log-log regression for binomial data and Poisson regression for count data, as well as continuous models such as ordinary linear regression, gamma and inverse Gaussian regression models. PROC GENMOD performs analysis of variance through likelihood ratio and Wald tests of fixed effects in generalized linear models, and provides contrasts and estimates for customized hypothesis tests. It performs analysis of repeated measures data with generalized estimating equation (GEE) methods.

See Chapter 4, “Introduction to Categorical Data Analysis Procedures,” and Chapter 31, “The GENMOD Procedure,” for more information.

Nonparametric Analysis of Variance

Analysis of variance is sensitive to the distribution of the error term. If the error term is not normally distributed, the statistics based on normality can be misleading. The traditional test statistics are called *parametric tests* because they depend on the specification of a certain probability distribution except for a set of free parameters. Parametric tests are said to depend on distributional assumptions. Nonparametric methods perform the tests without making any strict distributional assumptions. Even if the data are distributed normally, nonparametric methods are often almost as powerful as parametric methods.

Most nonparametric methods are based on taking the ranks of a variable and analyzing these ranks (or transformations of them) instead of the original values. The NPAR1WAY procedure performs a nonparametric one-way analysis of variance. Other nonparametric tests can be performed by taking ranks of the data (using the RANK procedure) and using a regular parametric procedure (such as GLM or ANOVA) to perform the analysis. Some of these techniques are outlined in the description of PROC RANK in the *SAS Procedures Guide* and in Conover and Iman (1981).

Constructing Analysis of Variance Designs

Analysis of variance is most often used for data from designed experiments. You can use the PLAN procedure to construct designs for many experiments. For example, PROC PLAN constructs designs for completely randomized experiments, randomized blocks, Latin squares, factorial experiments, and balanced incomplete block designs.

Randomization, or randomly assigning experimental units to cells in a design and to treatments within a cell, is another important aspect of experimental design. For either a new or an existing design, you can use PROC PLAN to randomize the experimental plan.

Additional features for design of experiments are available in SAS/QC software. The FACTEX and OPTEX procedures can construct a wide variety of designs, including factorials, fractional factorials, and D-optimal or A-optimal designs. These procedures, as well as the ADX Interface, provide features for randomizing and replicating designs; saving the design in an output data set; and interactively changing the design by changing its size, use of blocking, or the search strategies used. For more information, see *SAS/QC Software: Reference*.

References

Analysis of variance was pioneered by R.A. Fisher (1925). For a general introduction to analysis of variance, see an intermediate statistical methods textbook such as Steel and Torrie (1980), Snedecor and Cochran (1980), Milliken and Johnson (1984), Mendenhall (1968), John (1971), Ott (1977), or Kirk (1968). A classic source is Scheffé (1959). Freund, Littell, and Spector (1991) bring together a treatment of these statistical methods and SAS/STAT software procedures. Schlotzhauer and Littell (1997) cover how to perform *t* tests and one-way analysis of variance with SAS/STAT procedures. Texts on linear models include Searle (1971), Graybill (1976), and Hocking (1984). Kennedy and Gentle (1980) survey the computing aspects.

Conover, W.J. and Iman, R.L. (1981), “Rank Transformations as a Bridge Between Parametric and Nonparametric Statistics,” *The American Statistician*, 35, 124–129.

Fisher, R.A. (1925), *Statistical Methods for Research Workers*, Edinburgh: Oliver & Boyd.

Freund, R.J., Littell, R.C., and Spector, P.C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Graybill, F.A. (1976), *Theory and Applications of the Linear Model*, North Scituate, MA: Duxbury Press.

Hocking, R.R. (1984), *Analysis of Linear Models*, Monterey, CA: Brooks-Cole Publishing Co.

John, P. (1971), *Statistical Design and Analysis of Experiments*, New York: Macmillan Publishing Co.

Kennedy, W.J., Jr. and Gentle, J.E. (1980), *Statistical Computing*, New York: Marcel Dekker, Inc.

Kirk, R.E. (1968), *Experimental Design: Procedures for the Behavioral Sciences*, Monterey, CA: Brooks-Cole Publishing Co.

Mendenhall, W. (1968), *Introduction to Linear Models and the Design and Analysis of Experiments*, Belmont, CA: Duxbury Press.

Milliken, G.A. and Johnson, D.E. (1984), *Analysis of Messy Data Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Ott, L. (1977), *Introduction to Statistical Methods and Data Analysis*, Second Edition, Belmont, CA: Duxbury Press.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons, Inc.

Schlotzhauer, S.D. and Littell, R.C. (1997), *SAS System for Elementary Statistical Analysis*, Cary, NC: SAS Institute Inc.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Snedecor, G.W. and Cochran, W.G. (1980), *Statistical Methods*, Seventh Edition, Ames, IA: Iowa State University Press.

Steel R.G.D. and Torrie, J.H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill Book Co.

Chapter 4
Introduction to Categorical Data Analysis Procedures

Chapter Contents

Chapter 4
Introduction to Categorical Data Analysis Procedures

Overview

Several procedures in SAS/STAT software can be used for the analysis of categorical data:

CATMOD
fits linear models to functions of categorical data, facilitating such analyses as regression, analysis of variance, linear modeling, log-linear modeling, logistic regression, and repeated measures analysis. Maximum likelihood estimation is used for the analysis of logits and generalized logits, and weighted least squares analysis is used for fitting models to other response functions. Iterative proportional fitting (IPF), which avoids the need for parameter estimation, is available for fitting hierarchical log-linear models when there is a single population.

CORRESP
performs simple and multiple correspondence analyses, using a contingency table, Burt table, binary table, or raw categorical data as input. For more on PROC CORRESP, see Chapter 5, "Introduction to Multivariate Procedures," and Chapter 24, "The CORRESP Procedure."

FREQ
builds frequency tables or contingency tables and can produce numerous statistics. For one-way frequency tables, it can perform tests for equal proportions, specified proportions, or the binomial proportion. For contingency tables, it can compute various tests and measures of association and agreement including chi-square statistics, odds ratios, correlation statistics, Fisher's exact test for any size two-way table, kappa, and trend tests. In addition, it performs stratified analysis, computing Cochran-Mantel-Haenszel statistics and estimates of the common relative risk. Exact p-values and confidence intervals are available for various test statistics and measures.

GENMOD
fits generalized linear models with maximum-likelihood methods. This family includes logistic, probit, and complementary log-log regression models for binomial data, Poisson and negative binomial regression models for count data, and multinomial models for ordinal response data. It performs likelihood ratio and Wald tests for type I, type III, and user-defined contrasts. It analyzes repeated measures data with generalized estimating equation (GEE) methods.

LOGISTIC fits linear logistic regression models for discrete response data with maximum-likelihood methods. It provides four variable selection methods and computes regression diagnostics. It can also perform stratified conditional logistic regression analysis for binary response data and exact conditional regression analysis for binary and nominal response data. The logit link function in the logistic regression models can be replaced by the probit function or the complementary log-log function.

PROBIT fits models with probit, logit, or complementary log-log links for quantal assay or other discrete event data. It is mainly designed for dose-response analysis with a natural response rate. It computes the fiducial limits for the dose variable and provides various graphical displays for the analysis.

Other procedures that perform analyses for categorical data are the TRANSREG and PRINQUAL procedures. PROC PRINQUAL is summarized in Chapter 5, "Introduction to Multivariate Procedures," and PROC TRANSREG is summarized in Chapter 2, "Introduction to Regression Procedures."

A *categorical variable* is defined as one that can assume only a limited number of discrete values. The measurement scale for such a variable is unrestricted. It can be *nominal*, which means that the observed levels are not ordered. It can be *ordinal*, which means that the observed levels are ordered in some way. Or it can be *interval*, which means that the observed levels are ordered and numeric and that any interval of one unit on the scale of measurement represents the same amount, regardless of its location on the scale. One example of a categorical variable is litter size; another is the number of times a subject has been married. A variable that lies on a nominal scale is sometimes called a *qualitative* or *classification variable*.

Categorical data result from observations on multiple subjects where one or more categorical variables are observed for each subject. If there is only one categorical variable, then the data are generally represented by a *frequency table*, which lists each observed value of the variable and its frequency of occurrence.

If there are two or more categorical variables, then a subject's *profile* is defined as the subject's observed values for each of the variables. Such categorical data can be represented by a frequency table that lists each observed profile and its frequency of occurrence.

If there are exactly two categorical variables, then the data are often represented by a two-dimensional *contingency table*, which has one row for each level of variable 1 and one column for each level of variable 2. The intersections of rows and columns, called *cells*, correspond to variable profiles, and each cell contains the frequency of occurrence of the corresponding profile.

If there are more than two categorical variables, then the data can be represented by a *multidimensional contingency table*. There are two commonly used methods for displaying such tables, and both require that the variables be divided into two sets.

In the first method, one set contains a row variable and a column variable for a two-dimensional contingency table, and the second set contains all of the other variables. The variables in the second set are used to form a set of profiles. Thus, the data are represented as a series of two-dimensional contingency tables, one for each profile. This is the data representation used by PROC FREQ. For example, if you request tables for RACE*SEX*AGE*INCOME, the FREQ procedure represents the data as a series of contingency tables: the row variable is AGE, the column variable is INCOME, and the combinations of levels of RACE and SEX form a set of profiles.

In the second method, one set contains the independent variables, and the other set contains the dependent variables. Profiles based on the independent variables are called *population profiles*, whereas those based on the dependent variables are called *response profiles*. A two-dimensional contingency table is then formed, with one row for each population profile and one column for each response profile. Since any subject can have only one population profile and one response profile, the contingency table is uniquely defined. This is the data representation used by PROC CATMOD.

Sampling Frameworks and Distribution Assumptions

This section discusses the sampling frameworks and distribution assumptions for the CATMOD and FREQ procedures.

Simple Random Sampling: One Population

Suppose you take a simple random sample of 100 people and ask each person the following question: Of the three colors red, blue, and green, which is your favorite? You then tabulate the results in a frequency table as shown in Table 4.1.

Table 4.1. One-Way Frequency Table

	Favorite Color			
	Red	Blue	Green	Total
Frequency	52	31	17	100
Proportion	0.52	0.31	0.17	1.00

In the population you are sampling, you assume there is an unknown probability that a population member, selected at random, would choose any given color. In order to estimate that probability, you use the sample proportion

$$p_j = \frac{n_j}{n}$$

where n_j is the frequency of the jth response and n is the total frequency.

Because of the random variation inherent in any random sample, the frequencies have a probability distribution representing their relative frequency of occurrence in a hypothetical series of samples. For a simple random sample, the distribution of frequencies for a frequency table with three levels is as follows. The probability that the first frequency is n_1, the second frequency is n_2, and the third is $n_3 = n - n_1 - n_2$, is given by

$$\Pr(n_1, n_2, n_3) = \frac{n!}{n_1! n_2! n_3!} \pi_1^{n_1} \pi_2^{n_2} \pi_3^{n_3}$$

where π_j is the true probability of observing the jth response level in the population.

This distribution, called the *multinomial distribution*, can be generalized to any number of response levels. The special case of two response levels is called the *binomial distribution*.

Simple random sampling is the type of sampling required by PROC CATMOD when there is one population. PROC CATMOD uses the multinomial distribution to estimate a probability vector and its covariance matrix. If the sample size is sufficiently large, then the probability vector is approximately normally distributed as a result of central limit theory. PROC CATMOD uses this result to compute appropriate test statistics for the specified statistical model.

Stratified Simple Random Sampling: Multiple Populations

Suppose you take two simple random samples, 50 men and 50 women, and ask the same question as before. You are now sampling two different populations that may have different response probabilities. The data can be tabulated as shown in Table 4.2.

Table 4.2. Two-Way Contingency Table: Sex by Color

	Favorite Color			
Sex	Red	Blue	Green	Total
Male	30	10	10	50
Female	20	10	20	50
Total	50	20	30	100

Note that the row marginal totals (50, 50) of the contingency table are fixed by the sampling design, but the column marginal totals (50, 20, 30) are random. There are six probabilities of interest for this table, and they are estimated by the sample proportions

$$p_{ij} = \frac{n_{ij}}{n_i}$$

where n_{ij} denotes the frequency for the ith population and the jth response, and n_i is the total frequency for the ith population. For this contingency table, the sample proportions are shown in Table 4.3.

Table 4.3. Table of Sample Proportions by Sex

	Favorite Color			
Sex	Red	Blue	Green	Total
Male	0.60	0.20	0. 20	1.00
Female	0.40	0. 20	0.40	1.00

The probability distribution of the six frequencies is the *product multinomial distribution*

$$\Pr(n_{11}, n_{12}, n_{13}, n_{21}, n_{22}, n_{23}) = \frac{n_1! n_2! \pi_{11}^{n_{11}} \pi_{12}^{n_{12}} \pi_{13}^{n_{13}} \pi_{21}^{n_{21}} \pi_{22}^{n_{22}} \pi_{23}^{n_{23}}}{n_{11}! n_{12}! n_{13}! n_{21}! n_{22}! n_{23}!}$$

where π_{ij} is the true probability of observing the jth response level in the ith population. The product multinomial distribution is simply the product of two or more individual multinomial distributions since the populations are independent. This distribution can be generalized to any number of populations and response levels.

Stratified simple random sampling is the type of sampling required by PROC CATMOD when there is more than one population. PROC CATMOD uses the product multinomial distribution to estimate a probability vector and its covariance matrix. If the sample sizes are sufficiently large, then the probability vector is approximately normally distributed as a result of central limit theory, and PROC CATMOD uses this result to compute appropriate test statistics for the specified statistical model. The statistics are known as Wald statistics, and they are approximately distributed as chi-square when the null hypothesis is true.

Observational Data: Analyzing the Entire Population

Sometimes the observed data do not come from a random sample but instead represent a complete set of observations on some population. For example, suppose a class of 100 students is classified according to sex and favorite color. The results are shown in Table 4.4.

In this case, you could argue that all of the frequencies are fixed since the entire population is observed; therefore, there is no sampling error. On the other hand, you could hypothesize that the observed table has only fixed marginals and that the cell frequencies represent one realization of a conceptual process of assigning color preferences to individuals. The assignment process is open to hypothesis, which means that you can hypothesize restrictions on the joint probabilities.

Table 4.4. Two-Way Contingency Table: Sex by Color

	Favorite Color			
Sex	Red	Blue	Green	Total
Male	16	21	20	57
Female	12	20	11	43
Total	28	41	31	100

The usual hypothesis (sometimes called *randomness*) is that the distribution of the column variable (Favorite Color) does not depend on the row variable (Sex). This implies that, for each row of the table, the assignment process corresponds to a simple random sample (without replacement) from the finite population represented by the column marginal totals (or by the column marginal subtotals that remain after sampling other rows). The hypothesis of randomness induces a probability distribution on the frequencies in the table; it is called the *hypergeometric distribution*.

If the same row and column variables are observed for each of several populations, then the probability distribution of all the frequencies can be called the *multiple hypergeometric distribution*. Each population is called a *stratum*, and an analysis that draws information from each stratum and then summarizes across them is called a *stratified analysis* (or a *blocked analysis* or a *matched analysis*). PROC FREQ does such a stratified analysis, computing test statistics and measures of association.

In general, the populations are formed on the basis of cross-classifications of independent variables. Stratified analysis is a method of adjusting for the effect of these variables without being forced to estimate parameters for them.

The multiple hypergeometric distribution is the one used by PROC FREQ for the computation of Cochran-Mantel-Haenszel statistics. These statistics are in the class of *randomization model test statistics*, which require minimal assumptions for their validity. PROC FREQ uses the multiple hypergeometric distribution to compute the mean and the covariance matrix of a function vector in order to measure the deviation between the observed and expected frequencies with respect to a particular type of alternative hypothesis. If the cell frequencies are sufficiently large, then the function vector is approximately normally distributed as a result of central limit theory, and FREQ uses this result to compute a quadratic form that has a chi-square distribution when the null hypothesis is true.

Randomized Experiments

Consider a *randomized experiment* in which patients are assigned to one of two treatment groups according to a randomization process that allocates 50 patients to each group. After a specified period of time, each patient's status (cured or uncured) is recorded. Suppose the data shown in Table 4.5 give the results of the experiment. The null hypothesis is that the two treatments are equally effective. Under this hypothesis, treatment is a randomly assigned label that has no effect on the cure rate of the patients. But this implies that each row of the table represents a simple random sample from the finite population whose cure rate is described by the column marginal totals. Therefore, the column marginals (58, 42) are fixed under the hypothesis. Since the row marginals (50, 50) are fixed by the allocation process, the hypergeometric distribution is induced on the cell frequencies. Randomized experiments can also be specified in a stratified framework, and Cochran-Mantel-Haenszel statistics can be computed relative to the corresponding multiple hypergeometric distribution.

Table 4.5. Two-Way Contingency Table: Treatment by Status

	Status		
Treatment	Cured	Uncured	Total
1	36	14	50
2	22	28	50
Total	58	42	100

Relaxation of Sampling Assumptions

As indicated previously, the CATMOD procedure assumes that the data are from a stratified simple random sample, so it uses the product multinomial distribution. If the data are not from such a sample, then in many cases it is still possible to use PROC CATMOD by arguing that each row of the contingency table *does* represent a simple random sample from some hypothetical population. The extent to which the inferences are generalizable depends on the extent to which the hypothetical population is perceived to resemble the target population.

Similarly, the Cochran-Mantel-Haenszel statistics use the multiple hypergeometric distribution, which requires fixed row and column marginal totals in each contingency table. If the sampling process does not yield a table with fixed margins, then it is usually possible to fix the margins through conditioning arguments similar to the ones used by Fisher when he developed the Exact Test for 2×2 tables. In other words, if you want fixed marginal totals, you can generally make your analysis conditional on those observed totals.

For more information on sampling models for categorical data, see Bishop, Fienberg, and Holland (1975, Chapter 13).

Comparison of FREQ and CATMOD Procedures

PROC FREQ is used primarily to investigate the relationship between two variables; any confounding variables are taken into account by stratification rather than by parameter estimation. PROC CATMOD is used to investigate the relationship among many variables, all of which are integrated into a parametric model.

When PROC CATMOD estimates the covariance matrix of the frequencies, it assumes that the frequencies were obtained by a stratified simple random sampling procedure. However, PROC CATMOD can also analyze input data that consist of a function vector and a covariance matrix. Therefore, if the sampling procedure is different, you can estimate the covariance matrix of the frequencies in the appropriate manner before submitting the data to PROC CATMOD.

For the FREQ procedure, Fisher's Exact Test and Cochran-Mantel-Haenszel statistics are based on the hypergeometric distribution, which corresponds to fixed marginal totals. However, by conditioning arguments, these tests are generally applicable to a wide range of sampling procedures. Similarly, the Pearson and likelihood-ratio chi-square statistics can be derived under a variety of sampling situations.

PROC FREQ can do some traditional nonparametric analysis (such as the Kruskal-Wallis test and Spearman's correlation) since it can generate rank scores internally. Fisher's Exact Test and the Cochran-Mantel-Haenszel statistics are also inherently nonparametric. However, the main vehicle for nonparametric analyses in the SAS System is the NPAR1WAY procedure.

A large sample size is required for the validity of the chi-square distributions, the standard errors, and the covariance matrices for both PROC FREQ and PROC CATMOD. If sample size is a problem, then PROC FREQ has the advantage with its CMH statistics because it does not use any degrees of freedom to estimate parameters for confounding variables. In addition, PROC FREQ can compute exact p-values for any two-way table, provided that the sample size is sufficiently small in relation to the size of the table. It can also produce exact p-values for many tests, including the test of binomial proportions, the Cochran-Armitage test for trend, and the Jonckheere-Terpstra test for ordered differences among classes.

See the chapters on the FREQ and CATMOD procedures for more information. In addition, some well-known texts that deal with analyzing categorical data are listed in the "References" section of this chapter.

Comparison of CATMOD, GENMOD, LOGISTIC, and PROBIT Procedures

The CATMOD, GENMOD, LOGISTIC, and PROBIT procedures can all be used for statistical modeling of categorical data. The CATMOD procedure provides maximum likelihood estimation for logistic regression, including the analysis of logits for dichotomous outcomes and the analysis of generalized logits for polychotomous outcomes. It provides weighted least squares estimation of many other response functions, such as means, cumulative logits, and proportions, and you can also compute and analyze other response functions that can be formed from the proportions corresponding to the rows of a contingency table. In addition, a user can input and analyze a set of response functions and user-supplied covariance matrix with weighted least squares. With the CATMOD procedure, by default, all explanatory (independent) variables are treated as classification variables.

The GENMOD procedure is also a general statistical modeling tool which fits generalized linear models to data: it fits several useful models to categorical data including logistic regression, the proportional odds model, and Poisson regression. The GENMOD procedure also provides a facility for fitting generalized estimating equations to correlated response data that are categorical, such as repeated dichotomous outcomes. The GENMOD procedure fits models using maximum likelihood estimation, and you include classification variables in your models with a CLASS statement. PROC GENMOD can perform type I and type III tests, and it provides predicted values and residuals.

The LOGISTIC procedure is specifically designed for logistic regression. It performs the usual logistic regression analysis for dichotomous outcomes and it fits the proportional odds model and the generalized logit model for ordinal and nominal outcomes, respectively, by the method of maximum likelihood. With the CLASS statement, you can include independent CLASS variables in the model. This procedure has capabilities for a variety of model-building techniques, including stepwise, forward, and backward selection. It computes predicted values, the receiver operating characteristics (ROC) curve and the area beneath the curve, and a number of regression diagnostics. It can create output data sets containing these values and other statistics. PROC LOGISTIC can perform a conditional logistic regression analysis (matched-set and case-controlled) for binary response data. For small data sets, PROC LOGISTIC can perform the exact conditional logistic analysis of Hirji, Mehta, and Patel (1987) and Mehta, Patel, and Senchaudhuri (1992).

The PROBIT procedure is designed for quantal assay or other discrete event data. In additional to performing the logistic regression analysis, it can estimate the threshold response rate. PROC PROBIT can also estimate the values of independent variables that yield a desired response. With the CLASS statement, you can include CLASS variables in the model. PROC PROBIT allows only the less-than-full-rank parameterization for the CLASS variables.

Stokes, Davis, and Koch (2000) provide substantial discussion of these procedures, particularly the use of the FREQ, LOGISTIC, GENMOD, and CATMOD procedures for statistical modeling.

Logistic Regression

Dichotomous Response

You have many choices of performing logistic regression in the SAS System. The CATMOD, GENMOD, LOGISTIC, and PROBIT procedures fit the usual logistic regression model.

PROC LOGISTIC provides the capability of model-building, and performs conditional logistic regression analysis for case-control studies and exact conditional logistic regression analysis. You may choose to use it for these reasons.

PROC CATMOD may not be efficient when there are continous independent variables with large numbers of different values. For a continuous variable with a very limited number of values, PROC CATMOD may be useful. You list the continuous variables in the DIRECT statement.

The LOGISTIC, GENMOD, and PROBIT procedures can analyze summarized data by enabling you to input the numbers of events and trials; the ratio of events to trials must be between 0 and 1. PROC PROBIT enables you to estimate the natural response rate and compute fiducial limits for the dose variable.

Ordinal Response

PROC LOGISTIC fits the proportional odds model to the ordinal response data by default. PROC PROBIT fits this model if you specify the logistic distribution, and PROC GENMOD fits the same model if you specify the CLOGIT link and the multinomial distribution.

Nominal Response

When the response variable is nominal, there is no concept of ordering of the response values. PROC CATMOD fits a logistic model to response functions called *generalized logits*. PROC LOGISTIC fits the generalized logit model if you specify the GLOGIT link.

Parameterization

There are some differences in the way that models are parameterized, which means that you might get different parameter estimates if you were to perform logistic regression in each of these procedures.

- Parameter estimates from the procedures may differ in sign, depending on the ordering of response levels, which you can change if you want.
- The parameter estimates associated with a categorical independent variable may differ among the procedures, since the estimates depend on the coding of the indicator variables in the design matrix. By default, the design matrix column produced by PROC CATMOD for a binary independent variable is coded using the values 1 and -1 . The same column produced by the CLASS statement of PROC PROBIT is coded using 1 and 0. PROC CATMOD uses the deviation from the mean coding, which is a full-rank parameterization, and PROC PROBIT uses the less-than-full-rank coding. As a result, the parameter estimate printed by PROC CATMOD is one-half of the estimate produced by PROC PROBIT. Both PROC GENMOD and PROC LOGISTIC allow either a full-rank parameterization or the less-than-full-rank parameterization. See the "Details" sections in the chapters on the CATMOD, GENMOD, LOGISTIC, and PROBIT procedures for more information on the generation of the design matrices used by these procedures.
- The maximum-likelihood algorithm used differs among the procedures. PROC LOGISTIC uses the Fisher's scoring method by default, while PROC PROBIT, PROC GENMOD, and PROC CATMOD use the Newton-Raphson method. The parameter estimates should be the same for all three procedures, and the standard errors should be the same for the logistic model. For the normal and extreme-value (Gompertz) distributions in PROC PROBIT, which correspond to the probit and cloglog links, respectively, in PROC GENMOD and PROC LOGISTIC, the standard errors may differ. In general, tests computed using the standard errors from the Newton-Raphson method will be more conservative.
- The LOGISTIC, GENMOD, and PROBIT procedures can be used to fit a cumulative regression model for ordinal response data using maximum-likelihood estimation. PROC LOGISTIC and PROC GENMOD use a different parameterization from that of PROC PROBIT, which results in different intercept parameters. Estimates of the slope parameters, however, should be the same for both procedures. The estimated standard errors of the slope estimates are slightly different between the two procedures because of the different computational algorithms used as default.

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons, Inc.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Bishop, Y., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman and Hall.

Cox, D.R. and Snell, E.J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman and Hall.

Dobson, A. (1990), *An Introduction to Generalized Linear Models*, London: Chapman and Hall.

Fleiss, J.L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons, Inc.

Freeman, D.H., (1987), *Applied Categorical Data Analysis*, New York: Marcel-Dekker.

Grizzle, J.E., Starmer, C.F., and Koch, G.G. (1969), “Analysis of Categorical Data by Linear Models,” *Biometrics*, 25, 489–504.

Hirji, K.F., Mehta, C.R., and Patel, N.R. (1987), “Computing Distributions for Exact Logistic Regression,” *Journal of the American Statistical Association*, 82, 1110–1117.

Hosmer, D.W, Jr. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons, Inc.

Mehta, C.R., Patel, N. and Senchaudhuri, P. (1992), “Exact Stratified Linear Rank Tests for Ordered Categorical and Binary Data,” *Journal of Computational and Graphical Statistics*, 1, 21–40.

McCullagh, P. and Nelder, J.A. (1989), *Generalized Linear Models*, London: Chapman and Hall.

Stokes, M.E., Davis, C.S., and Koch, G.G (2000), *Categorical Data Analysis Using the SAS System*, Cary NC: SAS Institute Inc.

Chapter 5
Introduction to Multivariate Procedures

Chapter Contents

Chapter 5
Introduction to Multivariate Procedures

Overview

The procedures discussed in this chapter investigate relationships among variables without designating some as independent and others as dependent. Principal component analysis and common factor analysis examine relationships within a single set of variables, whereas canonical correlation looks at the relationship between two sets of variables. The following is a brief description of SAS/STAT multivariate procedures:

CORRESP performs simple and multiple correspondence analyses, using a contingency table, Burt table, binary table, or raw categorical data as input. Correspondence analysis is a weighted form of principal component analysis that is appropriate for frequency data.

PRINCOMP performs a principal component analysis and outputs standardized or unstandardized principal component scores.

PRINQUAL performs a principal component analysis of qualitative data and multidimensional preference analysis.

FACTOR performs principal component and common factor analyses with rotations and outputs component scores or estimates of common factor scores.

CANCORR performs a canonical correlation analysis and outputs canonical variable scores.

Many other SAS/STAT procedures can also analyze multivariate data, for example, the CATMOD, GLM, REG, CALIS, and TRANSREG procedures as well as the procedures for clustering and discriminant analysis.

The purpose of *principal component analysis* (Rao 1964) is to derive a small number of linear combinations (principal components) of a set of variables that retain as much of the information in the original variables as possible. Often a small number of principal components can be used in place of the original variables for plotting, regression, clustering, and so on. Principal component analysis can also be viewed as an attempt to uncover approximate linear dependencies among variables.

The purpose of *common factor analysis* (Mulaik 1972) is to explain the correlations or covariances among a set of variables in terms of a limited number of unobservable, latent variables. The latent variables are not generally computable as linear combinations of the original variables. In common factor analysis, it is assumed that the variables are linearly related if not for uncorrelated random error or *unique variation* in each variable; both the linear relations and the amount of unique variation can be estimated.

Principal component and common factor analysis are often followed by rotation of the components or factors. *Rotation* is the application of a nonsingular linear transformation to components or common factors to aid interpretation.

The purpose of *canonical correlation analysis* (Mardia, Kent, and Bibby 1979) is to explain or summarize the relationship between two sets of variables by finding a small number of linear combinations from each set of variables that have the highest possible between-set correlations. Plots of the canonical variables can be useful in examining multivariate dependencies. If one of the two sets of variables consists of dummy variables generated from a classification variable, the canonical correlation is equivalent to canonical discriminant analysis (see Chapter 21, “The CANDISC Procedure,”). If both sets of variables are dummy variables, canonical correlation is equivalent to simple correspondence analysis.

The purpose of *correspondence analysis* (Lebart, Morineau, and Warwick 1984; Greenacre 1984; Nishisato 1980) is to summarize the associations between a set of categorical variables in a small number of dimensions. Correspondence analysis computes scores on each dimension for each row and column category in a contingency table. Plots of these scores show the relationships among the categories.

The PRINQUAL procedure obtains linear and nonlinear transformations of variables using the method of alternating least squares (Young 1981) to optimize properties of the transformed variables’ covariance or correlation matrix. PROC PRINQUAL nonlinearly transforms variables, improving their fit to a principal component model. The name, PRINQUAL, for principal components of qualitative data, comes from the special case analysis of fitting a principal component model to nominal and ordinal scale of measurement variables (Young, Takane, and de Leeuw 1978). However, PROC PRINQUAL also has facilities for smoothly transforming continuous variables. All of PROC PRINQUAL’s transformations are also available in the TRANSREG procedure, which fits regression models with nonlinear transformations. PROC PRINQUAL can also perform metric and nonmetric multidimensional preference (MDPREF) analyses (Carroll 1972). The PRINQUAL procedure produces very little displayed output; the results are available in an output data set.

Comparison of the PRINCOMP and FACTOR Procedures

Although PROC FACTOR can be used for common factor analysis, the default method is principal components. PROC FACTOR produces the same results as PROC PRINCOMP except that scoring coefficients from PROC FACTOR are normalized to give principal component scores with unit variance, whereas PROC PRINCOMP by default produces principal component scores with variance equal to the corresponding eigenvalue. PROC PRINCOMP can also compute scores standardized to unit variance.

PROC PRINCOMP has the following advantages over PROC FACTOR:

- PROC PRINCOMP is slightly faster if a small number of components is requested.
- PROC PRINCOMP can analyze somewhat larger problems in a fixed amount of memory.
- PROC PRINCOMP can output scores from an analysis of a partial correlation or covariance matrix.
- PROC PRINCOMP is simpler to use.

PROC FACTOR has the following advantages over PROC PRINCOMP for principal component analysis:

- PROC FACTOR produces more output, including the scree (eigenvalue) plot, pattern matrix, and residual correlations.
- PROC FACTOR does rotations.

If you want to perform a common factor analysis, you must use PROC FACTOR instead of PROC PRINCOMP. Principal component analysis should never be used if a common factor solution is desired (Dziuban and Harris 1973; Lee and Comrey 1979).

Comparison of the PRINCOMP and PRINQUAL Procedures

The PRINCOMP procedure performs principal component analysis. The PRINQUAL procedure finds linear and nonlinear transformations of variables to optimize properties of the transformed variables' covariance or correlation matrix. One property is the sum of the first n eigenvalues, which is a measure of the fit of a principal component model with n components. Use PROC PRINQUAL to find nonlinear transformations of your variables or to perform a multidimensional preference analysis. Use PROC PRINCOMP to fit a principal component model to your data or to PROC PRINQUAL's output data set. PROC PRINCOMP produces a report of the principal component analysis and output data sets. PROC PRINQUAL produces only an output data set and an iteration history table.

Comparison of the PRINCOMP and CORRESP Procedures

As summarized previously, PROC PRINCOMP performs a principal component analysis of interval-scaled data. PROC CORRESP performs correspondence analysis, which is a weighted form of principal component analysis that is appropriate for frequency data. If your data are categorical, use PROC CORRESP instead of PROC PRINCOMP. Both procedures produce an output data set that can be used with the %PLOTIT macro. The plots produced from the PROC CORRESP output data set graphically show relationships among the categories of the categorical variables.

Comparison of the PRINQUAL and CORRESP Procedures

Both PROC PRINQUAL and PROC CORRESP can be used to summarize associations among variables measured on a nominal scale. PROC PRINQUAL searches for a single nonlinear transformation of the original scoring of each nominal variable that optimizes some aspect of the covariance matrix of the transformed variables. For example, PROC PRINQUAL could be used to find scorings that maximize the fit of a principal component model with one component. PROC CORRESP uses the crosstabulations of nominal variables, not covariances, and produces multiple scores for each category of each nominal variable. The main conceptual difference between PROC PRINQUAL and PROC CORRESP is that PROC PRINQUAL assumes that the categories of a nominal variable correspond to values of a single underlying interval variable, whereas PROC CORRESP assumes that there are multiple underlying interval variables and therefore uses different category scores for each dimension of the correspondence analysis. PROC CORRESP scores on the first dimension match the single set of PROC PRINQUAL scores (with appropriate standardizations for both analyses).

Comparison of the TRANSREG and PRINQUAL Procedures

Both the TRANSREG and PRINQUAL procedures are data transformation procedures that have many of the same transformations. These procedures can either directly perform the specified transformation (such as taking the logarithm of the variable) or search for an optimal transformation (such as a spline with a specified number of knots). Both procedures can use an iterative, alternating-least-squares analysis. Both procedures create an output data set that can be used as input to other procedures. PROC PRINQUAL displays very little output, whereas PROC TRANSREG displays many results. PROC TRANSREG has two sets of variables, usually dependent and independent, and it fits linear models such as ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. In contrast, PROC

PRINQUAL has one set of variables, fits a principal component model or multidimensional preference analysis, and can also optimize other properties of a correlation or covariance matrix. PROC TRANSREG performs hypothesis testing and can be used to code experimental designs prior to their use in other analyses.

See Chapter 2, “Introduction to Regression Procedures,” for more comparisons of the TRANSREG and REG procedures.

References

Carroll J.D. (1972), “Individual Differences and Multidimensional Scaling,” in R.N. Shepard, A.K. Romney, and S.B. Nerlove (eds.), *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, New York: Seminar Press.

Dziuban, C.D. and Harris, C.W. (1973), “On the Extraction of Components and the Applicability of the Factor Model,” *American Educational Research Journal*, 10, 93–99.

Greenacre, M.J. (1984), *Theory and Applications of Correspondence Analysis*, London: Academic Press.

Hanson, R.J. and Norris, M.J. (1981), “Analysis of Measurements Based on the Singular Value Decomposition,” *SIAM Journal on Scientific and Statistical Computing*, 2, 363–373.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker, Inc.

Lebart, L., Morineau, A., and Warwick, K.M. (1984), *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*, New York: John Wiley & Sons, Inc.

Lee, H.B. and Comrey, A.L. (1979), “Distortions in a Commonly Used Factor Analytic Procedure,” *Multivariate Behavioral Research*, 14, 301–321.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis*, London: Academic Press.

Mulaik, S.A. (1972), *The Foundations of Factor Analysis*, New York: McGraw-Hill Book Co.

Nishisato, S. (1980), *Analysis of Categorical Data: Dual Scaling and Its Applications*, Toronto: University of Toronto Press.

Rao, C.R. (1964), “The Use and Interpretation of Principal Component Analysis in Applied Research,” *Sankhya A*, 26, 329–358.

Van den Wollenberg, A.L. (1977), “Redundancy Analysis—An Alternative to Canonical Correlation Analysis,” *Psychometrika*, 42, 207–219.

Young, F.W. (1981), “Quantitative Analysis of Qualitative Data,” *Psychometrika*, 46, 357–388.

Young, F.W., Takane, Y., and de Leeuw, J. (1978), “The Principal Components of Mixed Measurement Level Multivariate Data: An Alternating Least Squares Method with Optimal Scaling Features,” *Psychometrika*, 43, 279–281.

Chapter 6
Introduction to Discriminant Procedures

Chapter Contents

Chapter 6
Introduction to Discriminant Procedures

Overview

The SAS procedures for discriminant analysis treat data with one classification variable and several quantitative variables. The purpose of discriminant analysis can be to find one or more of the following:

- a mathematical rule, or *discriminant function*, for guessing to which class an observation belongs, based on knowledge of the quantitative variables only
- a set of linear combinations of the quantitative variables that best reveals the differences among the classes
- a subset of the quantitative variables that best reveals the differences among the classes

The SAS discriminant procedures are as follows:

DISCRIM computes various discriminant functions for classifying observations. Linear or quadratic discriminant functions can be used for data with approximately multivariate normal within-class distributions. Nonparametric methods can be used without making any assumptions about these distributions.

CANDISC performs a canonical analysis to find linear combinations of the quantitative variables that best summarize the differences among the classes.

STEPDISC uses forward selection, backward elimination, or stepwise selection to try to find a subset of quantitative variables that best reveals differences among the classes.

Background

The term *discriminant analysis* (Fisher 1936; Cooley and Lohnes 1971; Tatsuoka 1971; Kshirsagar 1972; Lachenbruch 1975, 1979; Gnanadesikan 1977; Klecka 1980; Hand 1981,1982; Silverman, 1986) refers to several different types of analysis. Classificatory discriminant analysis is used to classify observations into two or more known groups on the basis of one or more quantitative variables. Classification can be done by either a parametric method or a nonparametric method in the DISCRIM procedure. A parametric method is appropriate only for approximately normal within-class distributions. The method generates either a linear discriminant function (the within-class covariance matrices are assumed to be equal) or a quadratic discriminant function (the within-class covariance matrices are assumed to be unequal).

When the distribution within each group is not assumed to have any specific distribution or is assumed to have a distribution different from the multivariate normal distribution, nonparametric methods can be used to derive classification criteria. These methods include the kernel method and nearest-neighbor methods. The kernel method uses uniform, normal, Epanechnikov, biweight, or triweight kernels in estimating the group-specific density at each observation. The within-group covariance matrices or the pooled covariance matrix can be used to scale the data.

The performance of a discriminant function can be evaluated by estimating error rates (probabilities of misclassification). Error count estimates and posterior probability error rate estimates can be evaluated with PROC DISCRIM. When the input data set is an ordinary SAS data set, the error rates can also be estimated by cross validation.

In multivariate statistical applications, the data collected are largely from distributions different from the normal distribution. Various forms of nonnormality can arise, such as qualitative variables or variables with underlying continuous but nonnormal distributions. If the multivariate normality assumption is violated, the use of parametric discriminant analysis may not be appropriate. When a parametric classification criterion (linear or quadratic discriminant function) is derived from a nonnormal population, the resulting error rate estimates may be biased.

If your quantitative variables are not normally distributed, or if you want to classify observations on the basis of categorical variables, you should consider using the CATMOD or LOGISTIC procedure to fit a categorical linear model with the classification variable as the dependent variable. Press and Wilson (1978) compare logistic regression and parametric discriminant analysis and conclude that logistic regression is preferable to parametric discriminant analysis in cases for which the variables do not have multivariate normal distributions within classes. However, if you do have normal within-class distributions, logistic regression is less efficient than parametric discriminant analysis. Efron (1975) shows that with two normal populations having a common covariance matrix, logistic regression is between one half and two thirds as effective as the linear discriminant function in achieving asymptotically the same error rate.

Do not confuse discriminant analysis with cluster analysis. All varieties of discriminant analysis require prior knowledge of the classes, usually in the form of a sample from each class. In cluster analysis, the data do not include information on class membership; the purpose is to construct a classification. See Chapter 7, “Introduction to Clustering Procedures.”

Canonical discriminant analysis is a dimension-reduction technique related to principal components and canonical correlation, and it can be performed by both the CANDISC and DISCRIM procedures. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of a discriminant criterion, you should use PROC CANDISC. Stepwise discriminant analysis is a variable-selection technique implemented by the STEPDISC procedure. After selecting a subset of variables with PROC STEPDISC, use any of the other discriminant procedures to obtain more detailed analyses. PROC CANDISC and PROC STEPDISC perform hypothesis tests that require the within-class distributions to be approximately normal, but these procedures can be used descriptively with nonnormal data.

Another alternative to discriminant analysis is to perform a series of univariate one-way ANOVAs. All three discriminant procedures provide summaries of the univariate ANOVAs. The advantage of the multivariate approach is that two or more classes that overlap considerably when each variable is viewed separately may be more distinct when examined from a multivariate point of view.

Example: Contrasting Univariate and Multivariate Analyses

Consider the two classes indicated by 'H' and 'O' in Figure 6.1. The results are shown in Figure 6.2.

```
data random;
   drop n;

   Group = 'H';
   do n = 1 to 20;
      X = 4.5 + 2 * normal(57391);
      Y = X + .5 + normal(57391);
      output;
   end;

   Group = 'O';
   do n = 1 to 20;
      X = 6.25 + 2 * normal(57391);
      Y = X - 1 + normal(57391);
      output;
   end;

run;

symbol1 v='H' c=blue;
symbol2 v='O' c=yellow;
proc gplot;
   plot Y*X=Group / cframe=ligr nolegend;
run;

proc candisc anova;
   class Group;
   var X Y;
run;
```

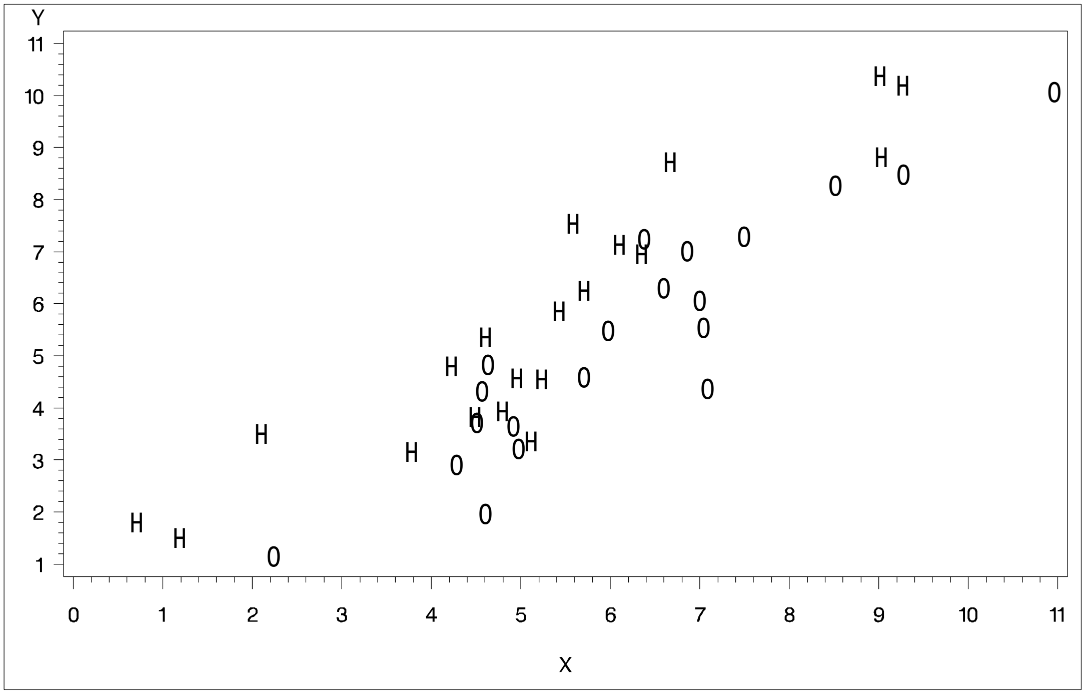
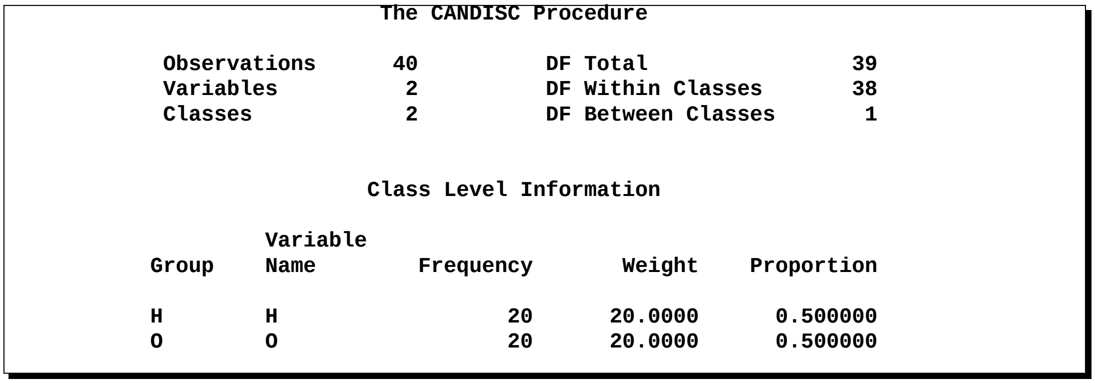


Figure 6.1. Groups for Contrasting Univariate and Multivariate Analyses



```
                         The CANDISC Procedure

       Observations      40          DF Total                39
       Variables          2          DF Within Classes       38
       Classes            2          DF Between Classes       1


                        Class Level Information

                 Variable
      Group      Name         Frequency         Weight    Proportion

      H          H                   20        20.0000      0.500000
      O          O                   20        20.0000      0.500000
```

Figure 6.2. Contrasting Univariate and Multivariate Analyses

```
                             The CANDISC Procedure

                          Univariate Test Statistics

                    F Statistics,    Num DF=1,   Den DF=38

              Total     Pooled    Between
           Standard   Standard   Standard             R-Square
Variable  Deviation  Deviation  Deviation  R-Square  / (1-RSq)  F Value  Pr > F

X            2.1776     2.1498     0.6820    0.0503     0.0530     2.01  0.1641
Y            2.4215     2.4486     0.2047    0.0037     0.0037     0.14  0.7105


                               Average R-Square

                       Unweighted              0.0269868
                       Weighted by Variance    0.0245201


                Multivariate Statistics and Exact F Statistics

                             S=1    M=0    N=17.5

Statistic                        Value    F Value    Num DF    Den DF    Pr > F

Wilks’ Lambda               0.64203704      10.31         2        37    0.0003
Pillai’s Trace              0.35796296      10.31         2        37    0.0003
Hotelling-Lawley Trace      0.55754252      10.31         2        37    0.0003
Roy’s Greatest Root         0.55754252      10.31         2        37    0.0003
```

```
                             The CANDISC Procedure

                                    Adjusted    Approximate        Squared
                    Canonical      Canonical       Standard      Canonical
                  Correlation    Correlation          Error    Correlation

             1       0.598300       0.589467       0.102808       0.357963

                                   Eigenvalues of Inv(E)*H
                                     = CanRsq/(1-CanRsq)

                    Eigenvalue    Difference    Proportion    Cumulative

               1        0.5575                      1.0000        1.0000

                  Test of H0: The canonical correlations in the
                    current row and all that follow are zero

                  Likelihood    Approximate
                       Ratio        F Value    Num DF    Den DF    Pr > F

             1    0.64203704          10.31         2        37    0.0003

                        NOTE: The F statistic is exact.
```

```
                          The CANDISC Procedure

                        Total Canonical Structure

                        Variable              Can1

                        X                -0.374883
                        Y                 0.101206


                        Between Canonical Structure

                        Variable              Can1

                        X                -1.000000
                        Y                 1.000000


                     Pooled Within Canonical Structure

                        Variable              Can1

                        X                -0.308237
                        Y                 0.081243
```

```
                          The CANDISC Procedure

             Total-Sample Standardized Canonical Coefficients

                        Variable              Can1

                        X             -2.625596855
                        Y              2.446680169


          Pooled Within-Class Standardized Canonical Coefficients

                        Variable              Can1

                        X             -2.592150014
                        Y              2.474116072


                        Raw Canonical Coefficients

                        Variable              Can1

                        X             -1.205756217
                        Y              1.010412967


                    Class Means on Canonical Variables

                         Group              Can1

                         H          0.7277811475
                         O          -.7277811475
```

The univariate R^2s are very small, 0.0503 for X and 0.0037 for Y, and neither variable shows a significant difference between the classes at the 0.10 level.

The multivariate test for differences between the classes is significant at the 0.0003 level. Thus, the multivariate analysis has found a highly significant difference, whereas the univariate analyses failed to achieve even the 0.10 level. The Raw Canonical Coefficients for the first canonical variable, Can1, show that the classes differ most widely on the linear combination -1.205756217 X + 1.010412967 Y or approximately Y - 1.2 X. The R^2 between Can1 and the class variable is 0.357963 as given by the Squared Canonical Correlation, which is much higher than either univariate R^2.

In this example, the variables are highly correlated within classes. If the within-class correlation were smaller, there would be greater agreement between the univariate and multivariate analyses.

References

Cooley, W.W. and Lohnes, P.R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons, Inc.

Dillion, W. and Goldstein, M. (1984), *Multivariate Analysis: Methods and Applications*, New York: John Wiley & Sons, Inc.

Efron, B. (1975), “The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis,” *Journal of the American Statistical Association*, 70, 892–898.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, New York: John Wiley & Sons, Inc.

Hand, D.J. (1981), *Discrimination and Classification*, New York: John Wiley & Sons, Inc.

Hand, D.J. (1982), *Kernel Discriminant Analysis*, New York: Research Studies Press.

Hora, S.C. and Wilcox, J.B. (1982), “Estimation of Error Rates in Several-Population Discriminant Analysis,” *Journal of Marketing Research*, XIX, 57–61.

Klecka, W.R. (1980), *Discriminant Analysis*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-019. Beverly Hills, CA: Sage Publications.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker, Inc.

Lachenbruch, P.A. (1975), *Discriminant Analysis*, New York: Hafner.

Lachenbruch, P.A. (1979), “Discriminant Analysis,” *Biometrics*, 35, 69–85.

Press, S.J. and Wilson, S. (1978), “Choosing Between Logistic Regression and Discriminant Analysis,” *Journal of the American Statistical Association*, 73, 699–705.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall.

Tatsuoka, M.M. (1971), *Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Chapter 7
Introduction to Clustering Procedures

Chapter Contents

Chapter 7
Introduction to Clustering Procedures

Overview

You can use SAS clustering procedures to cluster the observations or the variables in a SAS data set. Both hierarchical and disjoint clusters can be obtained. Only numeric variables can be analyzed directly by the procedures, although the DISTANCE procedure can compute a distance matrix using character or numeric variables.

The purpose of cluster analysis is to place objects into groups or clusters suggested by the data, not defined a priori, such that objects in a given cluster tend to be similar to each other in some sense, and objects in different clusters tend to be dissimilar. You can also use cluster analysis for summarizing data rather than for finding "natural" or "real" clusters; this use of clustering is sometimes called *dissection* (Everitt 1980).

Any generalization about cluster analysis must be vague because a vast number of clustering methods have been developed in several different fields, with different definitions of clusters and similarity among objects. The variety of clustering techniques is reflected by the variety of terms used for cluster analysis: botryology, classification, clumping, competitive learning, morphometrics, nosography, nosology, numerical taxonomy, partitioning, Q-analysis, systematics, taximetrics, taxonorics, typology, unsupervised pattern recognition, vector quantization, and winner-take-all learning. Good (1977) has also suggested aciniformics and agminatics.

Several types of clusters are possible:

- Disjoint clusters place each object in one and only one cluster.
- Hierarchical clusters are organized so that one cluster may be entirely contained within another cluster, but no other kind of overlap between clusters is allowed.
- Overlapping clusters can be constrained to limit the number of objects that belong simultaneously to two clusters, or they can be unconstrained, allowing any degree of overlap in cluster membership.
- Fuzzy clusters are defined by a probability or grade of membership of each object in each cluster. Fuzzy clusters can be disjoint, hierarchical, or overlapping.

The data representations of objects to be clustered also take many forms. The most common are

- a square distance or similarity matrix, in which both rows and columns correspond to the objects to be clustered. A correlation matrix is an example of a similarity matrix.
- a coordinate matrix, in which the rows are observations and the columns are variables, as in the usual SAS multivariate data set. The observations, the variables, or both may be clustered.

The SAS procedures for clustering are oriented toward disjoint or hierarchical clusters from coordinate data, distance data, or a correlation or covariance matrix. The following procedures are used for clustering:

CLUSTER performs hierarchical clustering of observations using eleven agglomerative methods applied to coordinate data or distance data.

FASTCLUS finds disjoint clusters of observations using a k-means method applied to coordinate data. PROC FASTCLUS is especially suitable for large data sets.

MODECLUS finds disjoint clusters of observations with coordinate or distance data using nonparametric density estimation. It can also perform approximate nonparametric significance tests for the number of clusters.

VARCLUS performs both hierarchical and disjoint clustering of variables by oblique multiple-group component analysis.

TREE draws tree diagrams, also called *dendrograms* or *phenograms*, using output from the CLUSTER or VARCLUS procedures. PROC TREE can also create a data set indicating cluster membership at any specified level of the cluster tree.

The following procedures are useful for processing data prior to the actual cluster analysis:

ACECLUS attempts to estimate the pooled within-cluster covariance matrix from coordinate data without knowledge of the number or the membership of the clusters (Art, Gnanadesikan, and Kettenring 1982). PROC ACECLUS outputs a data set containing canonical variable scores to be used in the cluster analysis proper.

DISTANCE computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. PROC DISTANCE also provides various nonparametric and parametric methods for standardizing variables. Different variables can be standardized with different methods.

PRINCOMP performs a principal component analysis and outputs principal component scores.

STDIZE standardizes variables using any of a variety of location and scale measures, including mean and standard deviation, minimum and range, median and absolute deviation from the median, various *m* estimators and *a* estimators, and some scale estimators designed specifically for cluster analysis.

Massart and Kaufman (1983) is the best elementary introduction to cluster analysis. Other important texts are Anderberg (1973), Sneath and Sokal (1973), Duran and Odell (1974), Hartigan (1975), Titterington, Smith, and Makov (1985), McLachlan and Basford (1988), and Kaufmann and Rousseeuw (1990). Hartigan (1975) and Spath (1980) give numerous FORTRAN programs for clustering. Any prospective user of cluster analysis should study the Monte Carlo results of Milligan (1980), Milligan and Cooper (1985), and Cooper and Milligan (1984). Important references on the statistical aspects of clustering include MacQueen (1967), Wolfe (1970), Scott and Symons (1971), Hartigan (1977; 1978; 1981; 1985), Symons (1981), Everitt (1981), Sarle (1983), Bock (1985), and Thode et al. (1988). Bayesian methods have important advantages over maximum likelihood; refer to Binder (1978; 1981), Banfield and Raftery (1993), and Bensmail et al, (1997). For fuzzy clustering, refer to Bezdek (1981) and Bezdek and Pal (1992). The signal-processing perspective is provided by Gersho and Gray (1992). Refer to Blashfield and Aldenderfer (1978) for a discussion of the fragmented state of the literature on cluster analysis.

Clustering Variables

Factor rotation is often used to cluster variables, but the resulting clusters are fuzzy. It is preferable to use PROC VARCLUS if you want hard (nonfuzzy), disjoint clusters. Factor rotation is better if you want to be able to find overlapping clusters. It is often a good idea to try both PROC VARCLUS and PROC FACTOR with an oblique rotation, compare the amount of variance explained by each, and see how fuzzy the factor loadings are and whether there seem to be overlapping clusters.

You can use PROC VARCLUS to harden a fuzzy factor rotation; use PROC FACTOR to create an output data set containing scoring coefficients and initialize PROC VARCLUS with this data set:

```
proc factor rotate=promax score outstat=fact;
run;

proc varclus initial=input proportion=0;
run;
```

You can use any rotation method instead of the PROMAX method. The SCORE and OUTSTAT= options are necessary in the PROC FACTOR statement. PROC VARCLUS reads the correlation matrix from the data set created by PROC FACTOR. The INITIAL=INPUT option tells PROC VARCLUS to read initial scoring coefficients from the data set. The option PROPORTION=0 keeps PROC VARCLUS from splitting any of the clusters.

Clustering Observations

PROC CLUSTER is easier to use than PROC FASTCLUS because one run produces results from one cluster up to as many as you like. You must run PROC FASTCLUS once for each number of clusters.

The time required by PROC FASTCLUS is roughly proportional to the number of observations, whereas the time required by PROC CLUSTER with most methods varies with the square or cube of the number of observations. Therefore, you can use PROC FASTCLUS with much larger data sets than PROC CLUSTER.

If you want to hierarchically cluster a data set that is too large to use with PROC CLUSTER directly, you can have PROC FASTCLUS produce, for example, 50 clusters, and let PROC CLUSTER analyze these 50 clusters instead of the entire data set. The MEAN= data set produced by PROC FASTCLUS contains two special variables:

- The variable _FREQ_ gives the number of observations in the cluster.
- The variable _RMSSTD_ gives the root-mean-square across variables of the cluster standard deviations.

These variables are automatically used by PROC CLUSTER to give the correct results when clustering clusters. For example, you could specify Ward's minimum variance method (Ward 1963),

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

proc cluster method=ward outtree=tree;
   var x y z;
run;
```

or Wong's hybrid method (Wong 1982):

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

proc cluster method=density hybrid outtree=tree;
   var x y z;
run;
```

More detailed examples are given in Chapter 23, "The CLUSTER Procedure."

Characteristics of Methods for Clustering Observations

Many simulation studies comparing various methods of cluster analysis have been performed. In these studies, artificial data sets containing known clusters are produced using pseudo-random-number generators. The data sets are analyzed by a variety of clustering methods, and the degree to which each clustering method recovers the known cluster structure is evaluated. Refer to Milligan (1981) for a review of such studies. In most of these studies, the clustering method with the best overall performance has been either average linkage or Ward's minimum variance method. The method with the poorest overall performance has almost invariably been single linkage. However, in many respects, the results of simulation studies are inconsistent and confusing.

When you attempt to evaluate clustering methods, it is essential to realize that most methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least-squares criterion (Sarle 1982), such as k-means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation such as single linkage and density linkage.

Most simulation studies have generated compact (often multivariate normal) clusters of roughly equal size or dispersion. Such studies naturally favor average linkage and Ward's method over most other hierarchical methods, especially single linkage. It would be easy, however, to design a study using elongated or irregular clusters in which single linkage would perform much better than average linkage or Ward's method (see some of the following examples). Even studies that compare clustering methods using "realistic" data may unfairly favor particular methods. For example, in all the data sets used by Mezzich and Solomon (1980), the clusters established by field experts are of equal size. When interpreting simulation or other comparative studies, you must, therefore, decide whether the artificially generated clusters in the study resemble the clusters you suspect may exist in your data in terms of size, shape, and dispersion. If, like many people doing exploratory cluster analysis, you have no idea what kinds of clusters to expect, you should include at least one of the relatively unbiased methods, such as density linkage, in your analysis.

The rest of this section consists of a series of examples that illustrate the performance of various clustering methods under various conditions. The first, and simplest example, shows a case of well-separated clusters. The other examples show cases of poorly separated clusters, clusters of unequal size, parallel elongated clusters, and nonconvex clusters.

Well-Separated Clusters

If the population clusters are sufficiently well separated, almost any clustering method performs well, as demonstrated in the following example using single linkage. In this and subsequent examples, the output from the clustering procedures is not shown, but cluster membership is displayed in scatter plots. The following SAS statements produce Figure 7.1:

```
data compact;
   keep x y;
   n=50; scale=1;
   mx=0; my=0; link generate;
   mx=8; my=0; link generate;
   mx=4; my=8; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
run;

proc cluster data=compact outtree=tree
             method=single noprint;
run;

proc tree noprint out=out n=3;
   copy x y;
run;

legend1 frame cframe=ligr  cborder=black
        position=center value=(justify=center);
axis1 minor=none label=(angle=90 rotate=0);
axis2 minor=none;
proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Well-Separated,
           Compact Clusters';
run;
```

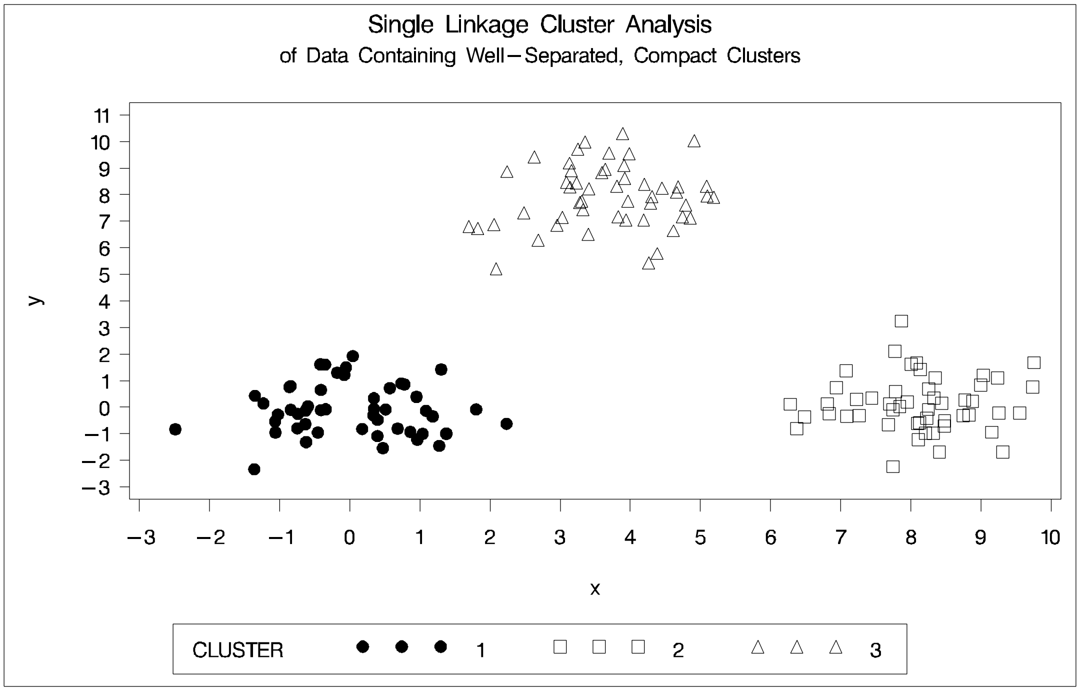


Figure 7.1. Data Containing Well-Separated, Compact Clusters: PROC CLUSTER with METHOD=SINGLE and PROC GPLOT

Poorly Separated Clusters

To see how various clustering methods differ, you must examine a more difficult problem than that of the previous example.

The following data set is similar to the first except that the three clusters are much closer together. This example demonstrates the use of PROC FASTCLUS and five hierarchical methods available in PROC CLUSTER. To help you compare methods, this example plots true, generated clusters. Also included is a bubble plot of the density estimates obtained in conjunction with two-stage density linkage in PROC CLUSTER. The following SAS statements produce Figure 7.2:

```
data closer;
   keep x y c;
   n=50; scale=1;
   mx=0; my=0; c=3; link generate;
   mx=3; my=0; c=1; link generate;
   mx=1; my=2; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(9)*scale+mx;
      y=rannor(9)*scale+my;
      output;
   end;
   return;
run;

title 'True Clusters for Data Containing Poorly Separated,
       Compact Clusters';
proc gplot;
    plot y*x=c/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
run;
```

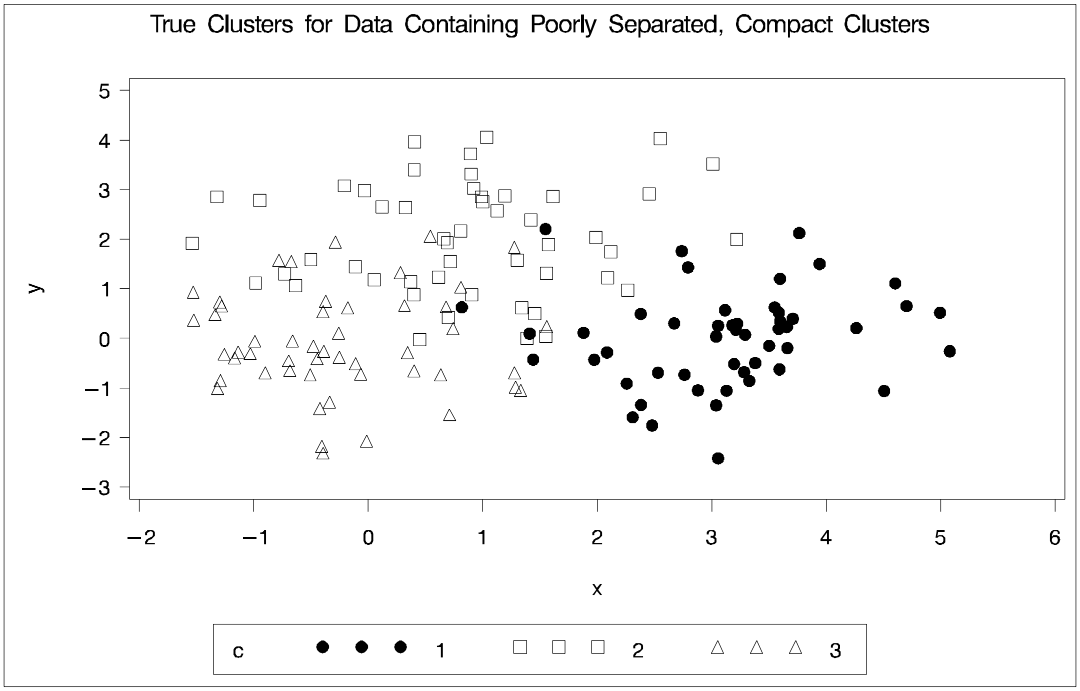


Figure 7.2. Data Containing Poorly Separated, Compact Clusters: Plot of True Clusters

The following statements use the FASTCLUS procedure to find three clusters and the GPLOT procedure to plot the clusters. Since the GPLOT step is repeated several times in this example, it is contained in the PLOTCLUS macro. The following statements produce Figure 7.3.

```
%macro plotclus;
   legend1 frame cframe=ligr  cborder=black
           position=center value=(justify=center);
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   proc gplot;
      plot y*x=cluster/frame cframe=ligr
           vaxis=axis1 haxis=axis2 legend=legend1;
   run;
%mend plotclus;

proc fastclus data=closer out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;
%plotclus;
```

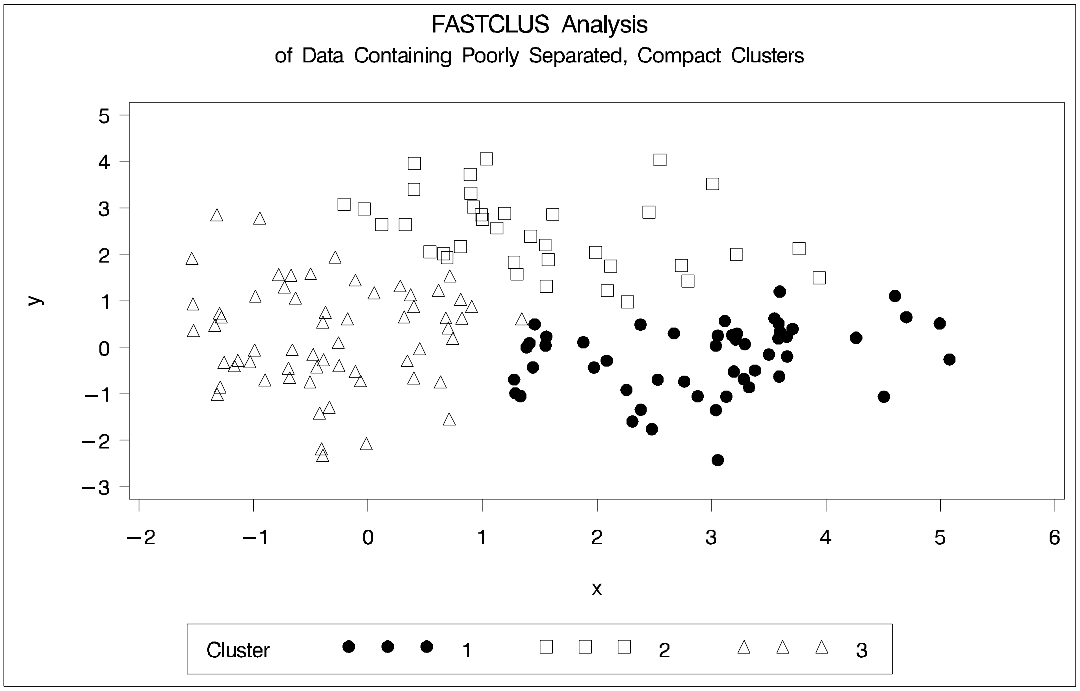


Figure 7.3. Data Containing Poorly Separated, Compact Clusters: PROC FASTCLUS

The following SAS statements produce Figure 7.4:

```
proc cluster data=closer outtree=tree method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;

%plotclus;
```

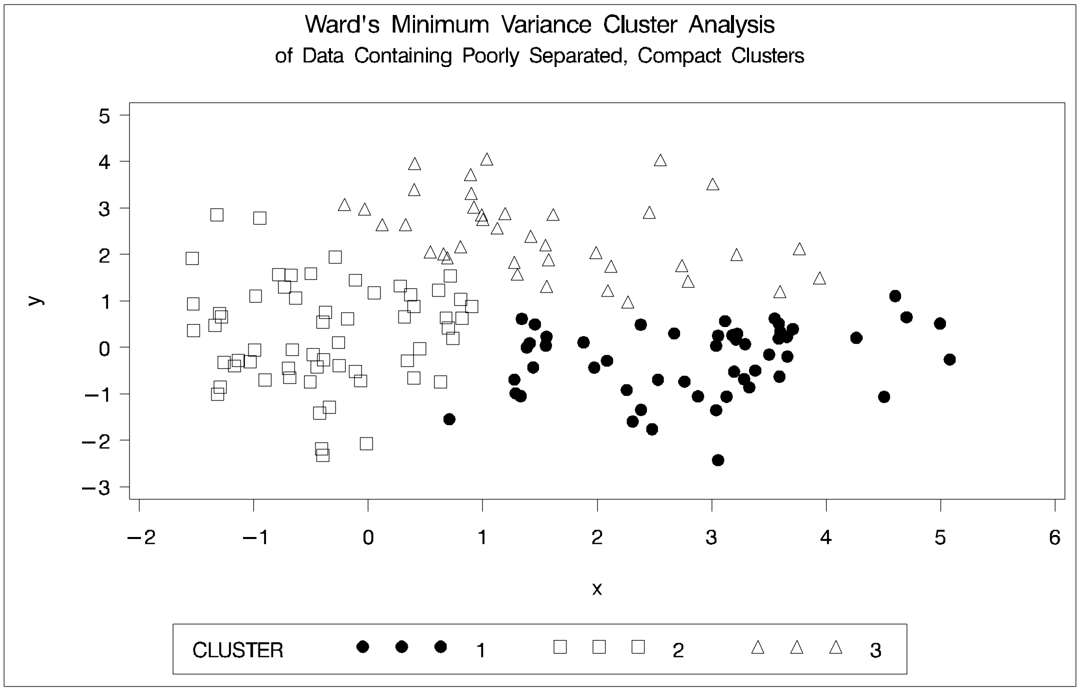


Figure 7.4. Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=WARD

The following SAS statements produce Figure 7.5:

```
proc cluster data=closer outtree=tree method=average noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;

%plotclus;
```

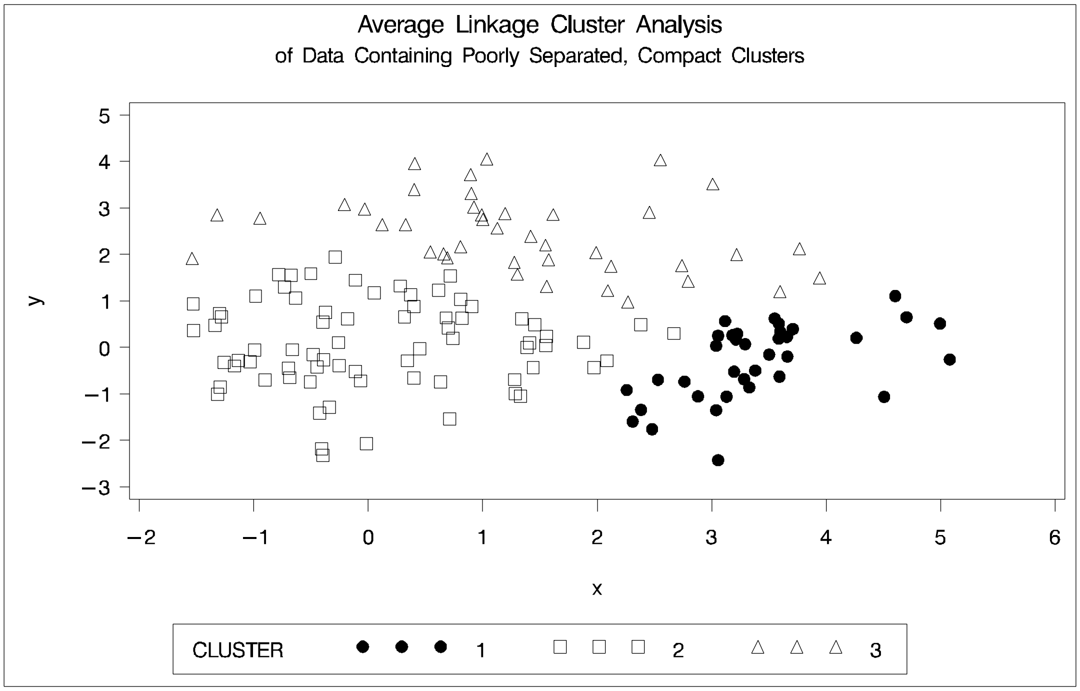


Figure 7.5. Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=AVERAGE

The following SAS statements produce Figure 7.6:

```
proc cluster data=closer outtree=tree
             method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;

%plotclus;
```

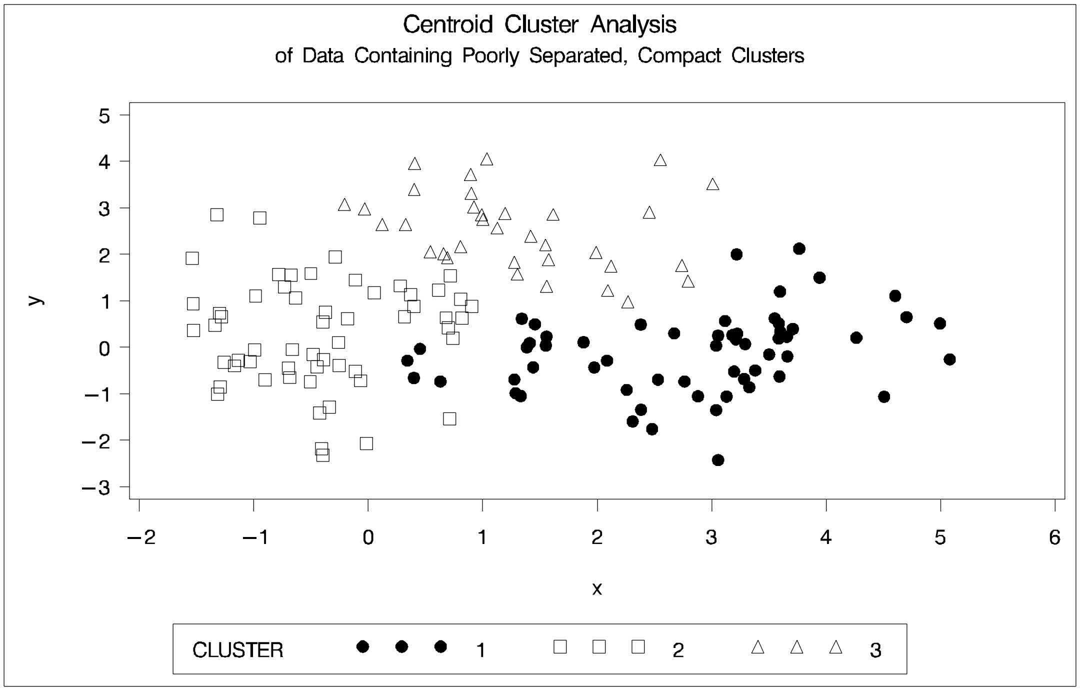


Figure 7.6. Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=CENTROID

The following SAS statements produce Figure 7.7:

```
proc cluster data=closer outtree=tree
             method=twostage k=10 noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;

%plotclus;

proc gplot;
   bubble y*x=_dens_/frame cframe=ligr
          vaxis=axis1 haxis=axis2;
   title 'Estimated Densities';
   title2 'for Data Containing Poorly Separated,
           Compact Clusters';
run;
```

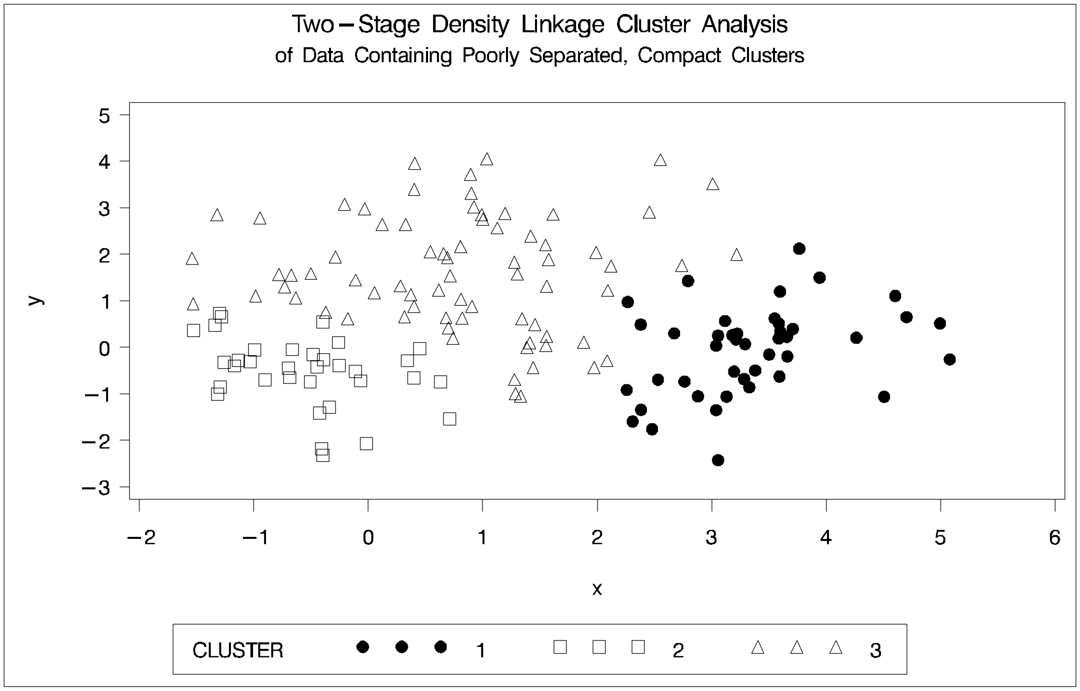
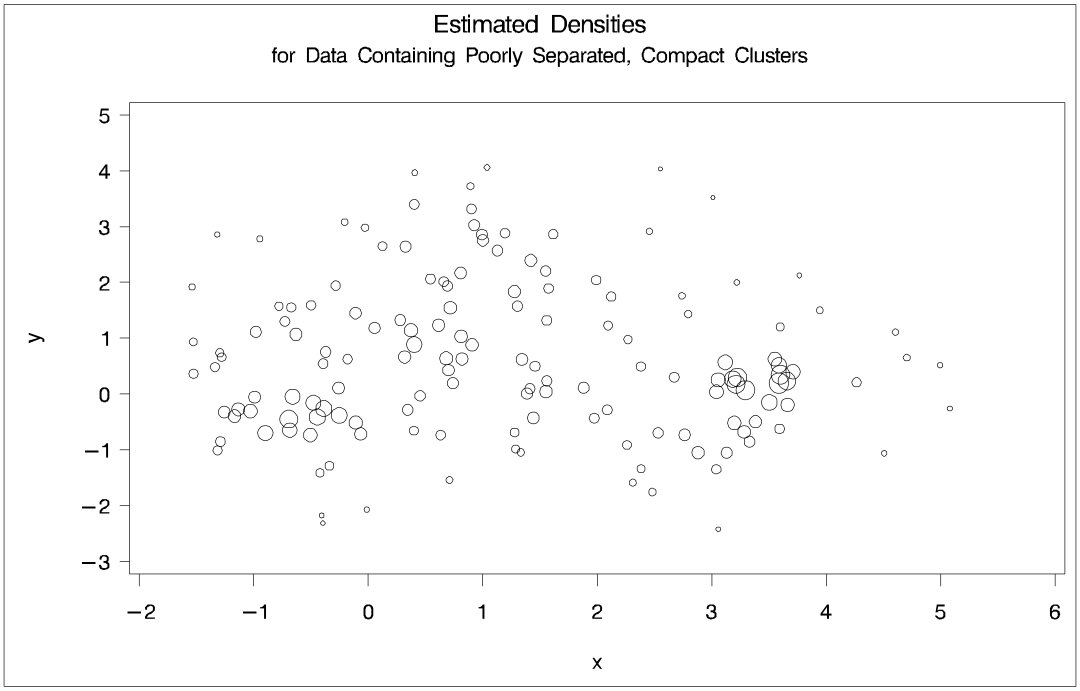


Figure 7.7. Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=TWOSTAGE



In two-stage density linkage, each cluster is a region surrounding a local maximum of the estimated probability density function. If you think of the estimated density function as a landscape with mountains and valleys, each mountain is a cluster, and the boundaries between clusters are placed near the bottoms of the valleys.

The following SAS statements produce Figure 7.8:

```
proc cluster data=closer outtree=tree
             method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated,
           Compact Clusters';
run;

%plotclus;
```

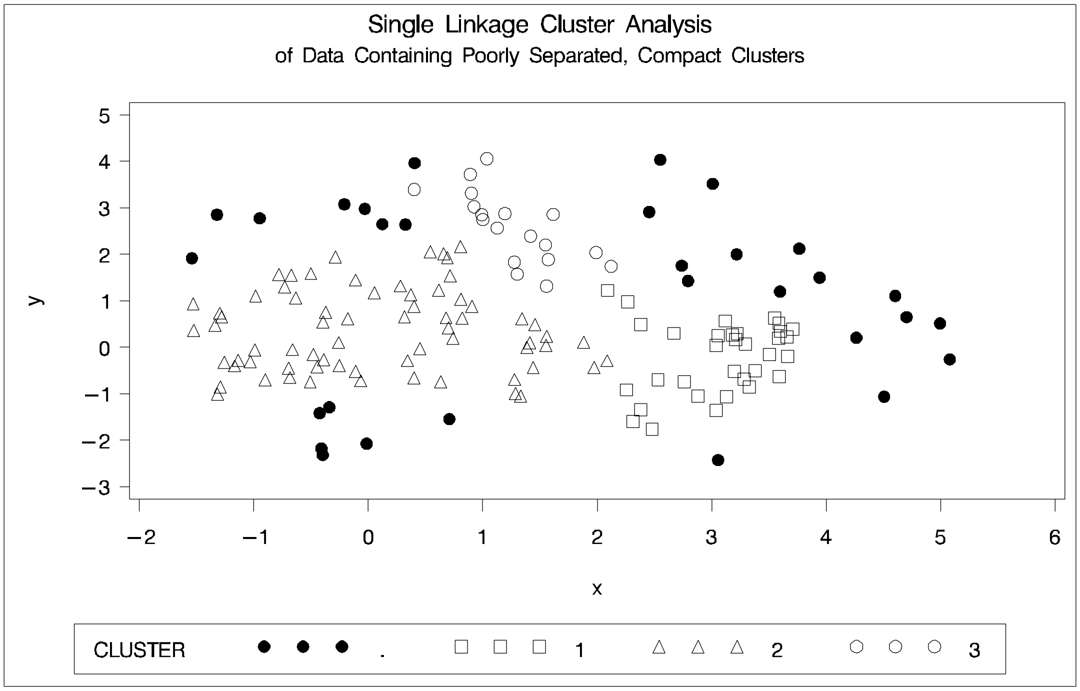


Figure 7.8. Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=SINGLE

The two least-squares methods, PROC FASTCLUS and Ward's, yield the most uniform cluster sizes and the best recovery of the true clusters. This result is expected since these two methods are biased toward recovering compact clusters of equal size. With average linkage, the lower-left cluster is too large; with the centroid method, the lower-right cluster is too large; and with two-stage density linkage, the top cluster is too large. The single linkage analysis resembles average linkage except for the large number of outliers resulting from the DOCK= option in the PROC TREE statement; the outliers are plotted as dots (missing values).

Multinormal Clusters of Unequal Size and Dispersion

In this example, there are three multinormal clusters that differ in size and dispersion. PROC FASTCLUS and five of the hierarchical methods available in PROC CLUSTER are used. To help you compare methods, the true, generated clusters are plotted. The following SAS statements produce Figure 7.9:

```
data unequal;
   keep x y c;
   mx=1; my=0; n=20; scale=.5; c=1; link generate;
   mx=6; my=0; n=80; scale=2.; c=3; link generate;
   mx=3; my=4; n=40; scale=1.; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
run;

title 'True Clusters for Data Containing Multinormal
       Clusters';
title2 'of Unequal Size';
proc gplot;
   plot y*x=c/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
run;
```

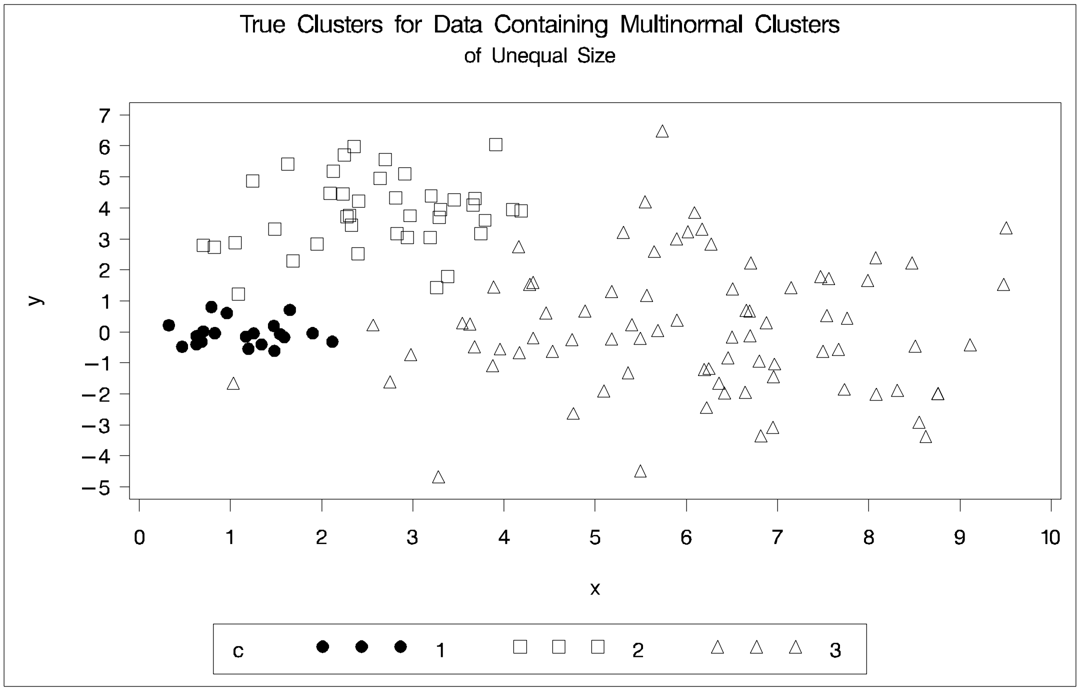


Figure 7.9. Data Containing Generated Clusters of Unequal Size

The following statements use the FASTCLUS procedure to find three clusters and the PLOTCLUS macro to plot the clusters. The statements produce Figure 7.10.

```
proc fastclus data=unequal out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;
```

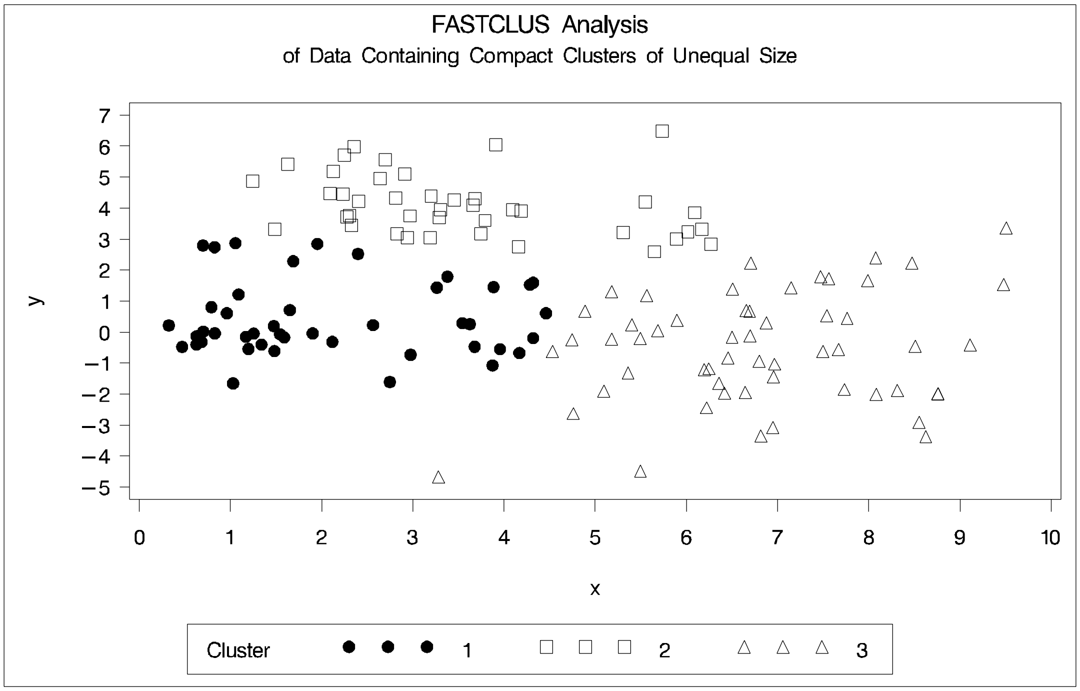


Figure 7.10. Data Containing Compact Clusters of Unequal Size: PROC FASTCLUS

The following SAS statements produce Figure 7.11:

```
proc cluster data=unequal outtree=tree
             method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;
```

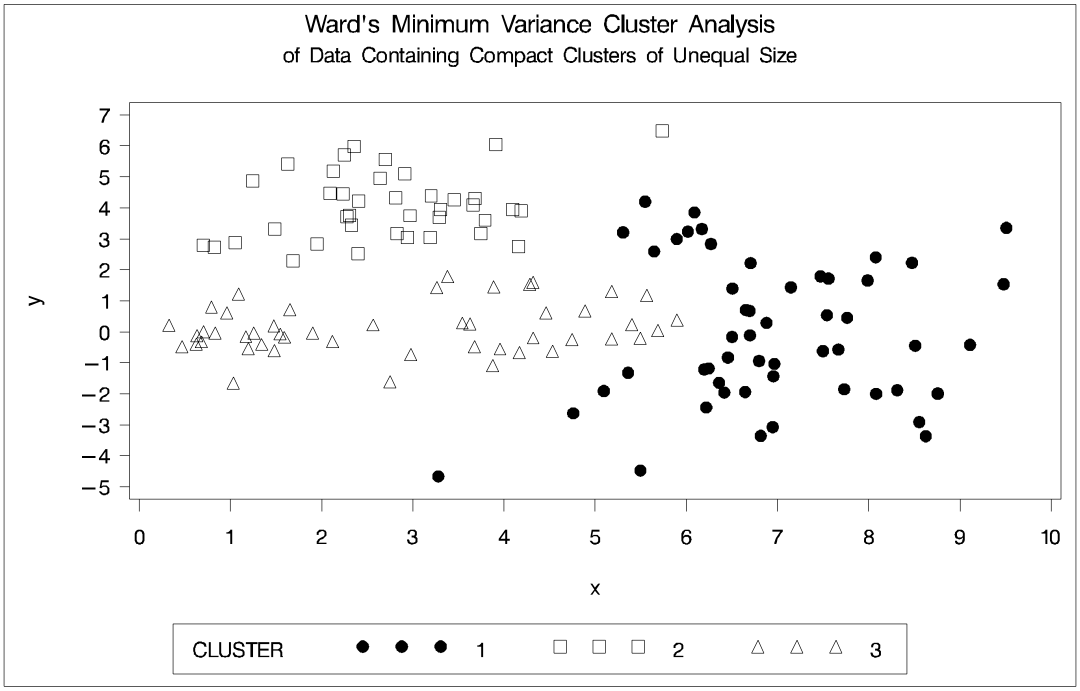


Figure 7.11. Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=WARD

The following SAS statements produce Figure 7.12:

```
proc cluster data=unequal outtree=tree method=average
             noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;
```

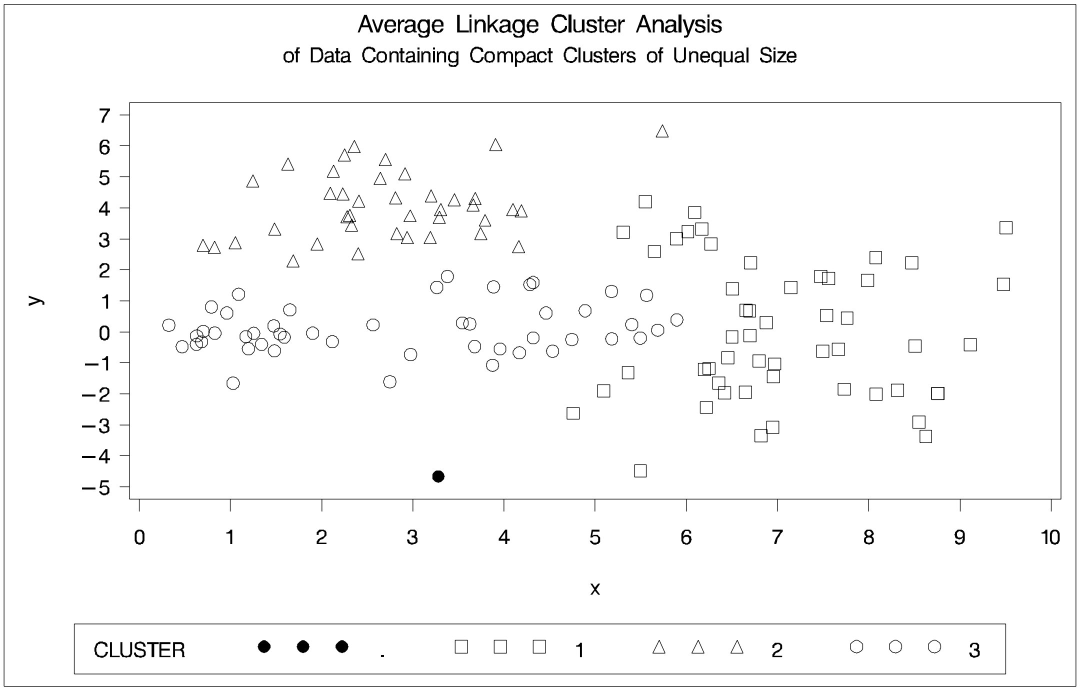


Figure 7.12. Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=AVERAGE

The following SAS statements produce Figure 7.13:

```
proc cluster data=unequal outtree=tree
             method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;
```

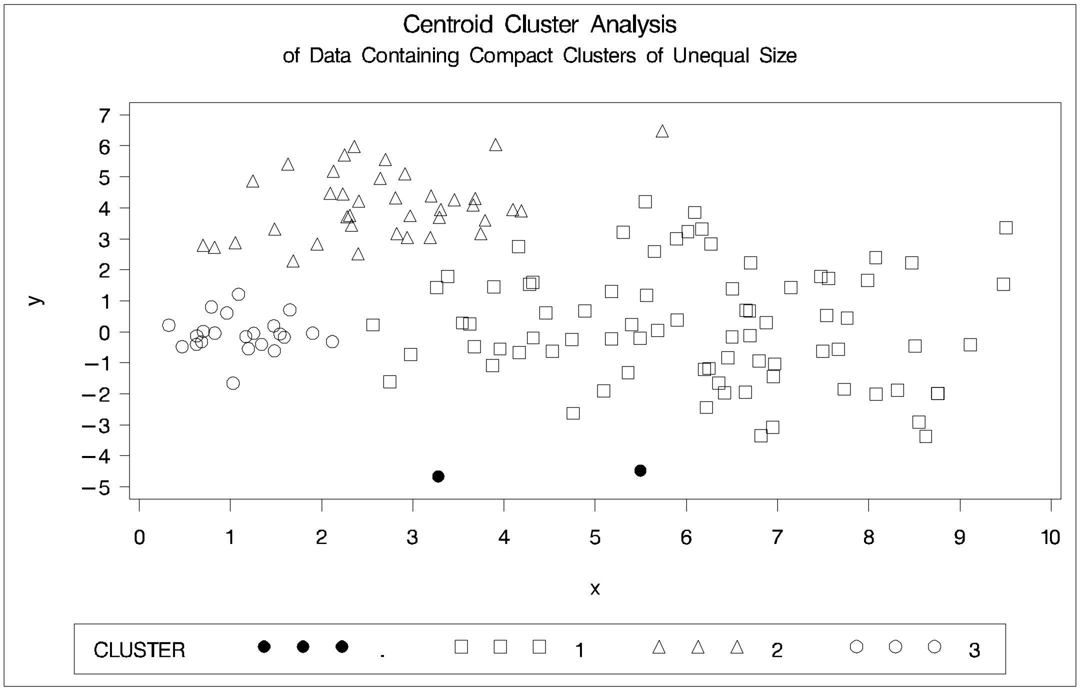


Figure 7.13. Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=CENTROID

The following SAS statements produce Figure 7.14:

```
proc cluster data=unequal outtree=tree method=twostage
             k=10 noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;

proc gplot;
   bubble y*x=_dens_/frame cframe=ligr
          vaxis=axis1 haxis=axis2 ;
   title 'Estimated Densities';
   title2 'for Data Containing Compact Clusters of
           Unequal Size';
run;
```

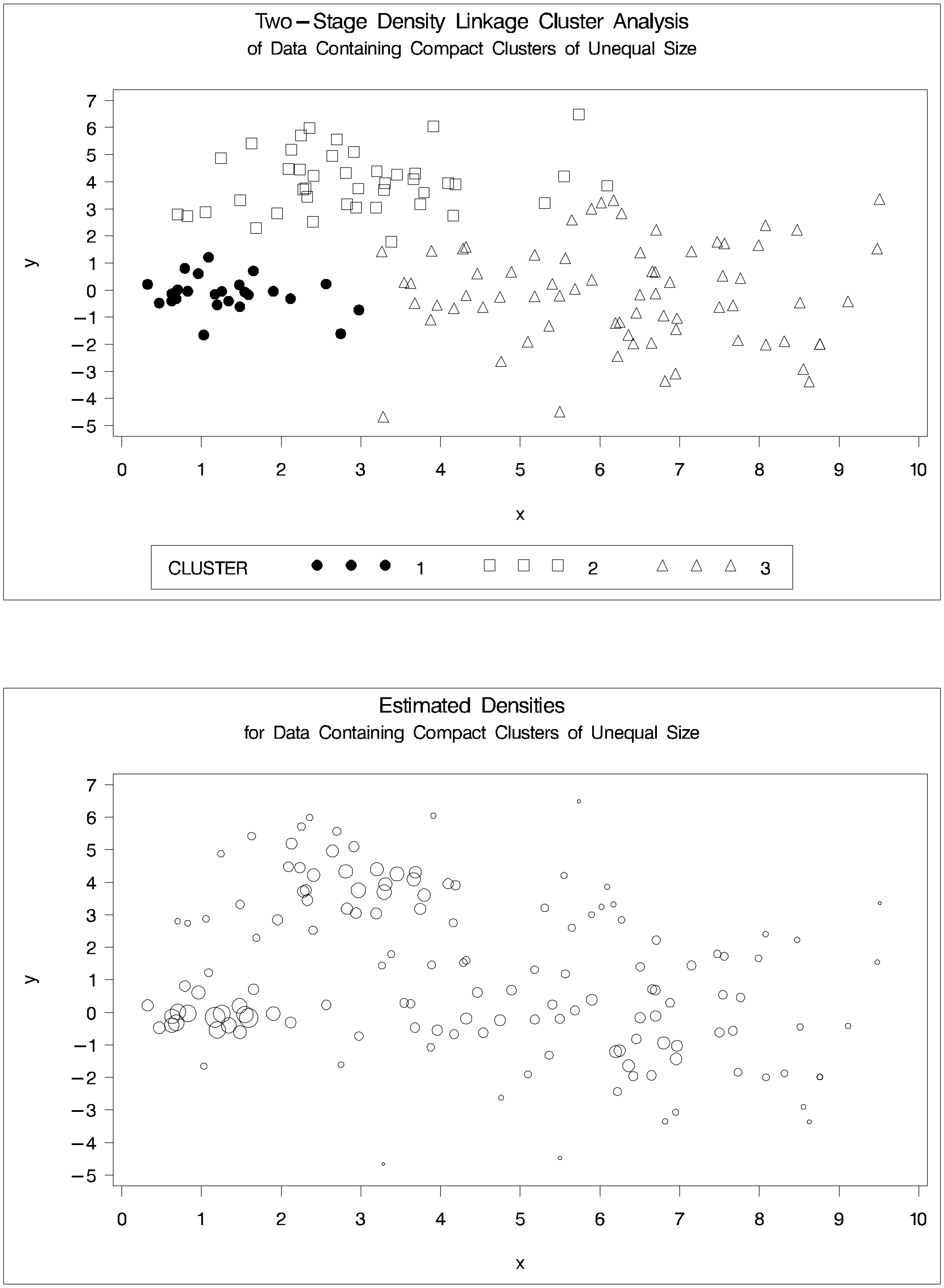


Figure 7.14. Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=TWOSTAGE

The following SAS statements produce Figure 7.15:

```
proc cluster data=unequal outtree=tree
             method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of
           Unequal Size';
run;

%plotclus;
```

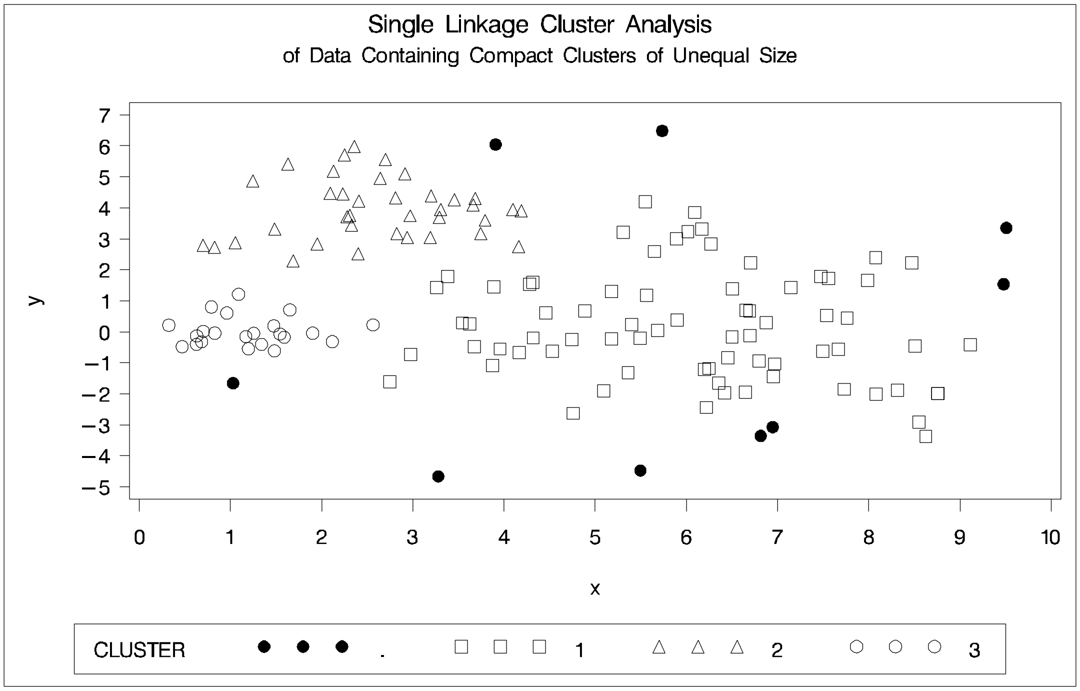


Figure 7.15. Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=SINGLE

In the PROC FASTCLUS analysis, the smallest cluster, in the bottom left of the plot, has stolen members from the other two clusters, and the upper-left cluster has also acquired some observations that rightfully belong to the larger, lower-right cluster. With Ward's method, the upper-left cluster is separated correctly, but the lower-left cluster has taken a large bite out of the lower-right cluster. For both of these methods, the clustering errors are in accord with the biases of the methods to produce clusters of equal size. In the average linkage analysis, both the upper- and lower-left clusters have encroached on the lower-right cluster, thereby making the variances more nearly equal than in the true clusters. The centroid method, which lacks the size and dispersion biases of the previous methods, obtains an essentially correct partition.

Two-stage density linkage does almost as well even though the compact shapes of these clusters favor the traditional methods. Single linkage also produces excellent results.

Elongated Multinormal Clusters

In this example, the data are sampled from two highly elongated multinormal distributions with equal covariance matrices. The following SAS statements produce Figure 7.16:

```
data elongate;
   keep x y;
   ma=8; mb=0; link generate;
   ma=6; mb=8; link generate;
   stop;
generate:
   do i=1 to 50;
      a=rannor(7)*6+ma;
      b=rannor(7)+mb;
      x=a-b;
      y=a+b;
      output;
   end;
   return;
run;

proc fastclus data=elongate out=out maxc=2 noprint;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

Notice that PROC FASTCLUS found two clusters, as requested by the MAXC= option. However, it attempted to form spherical clusters, which are obviously inappropriate for this data.

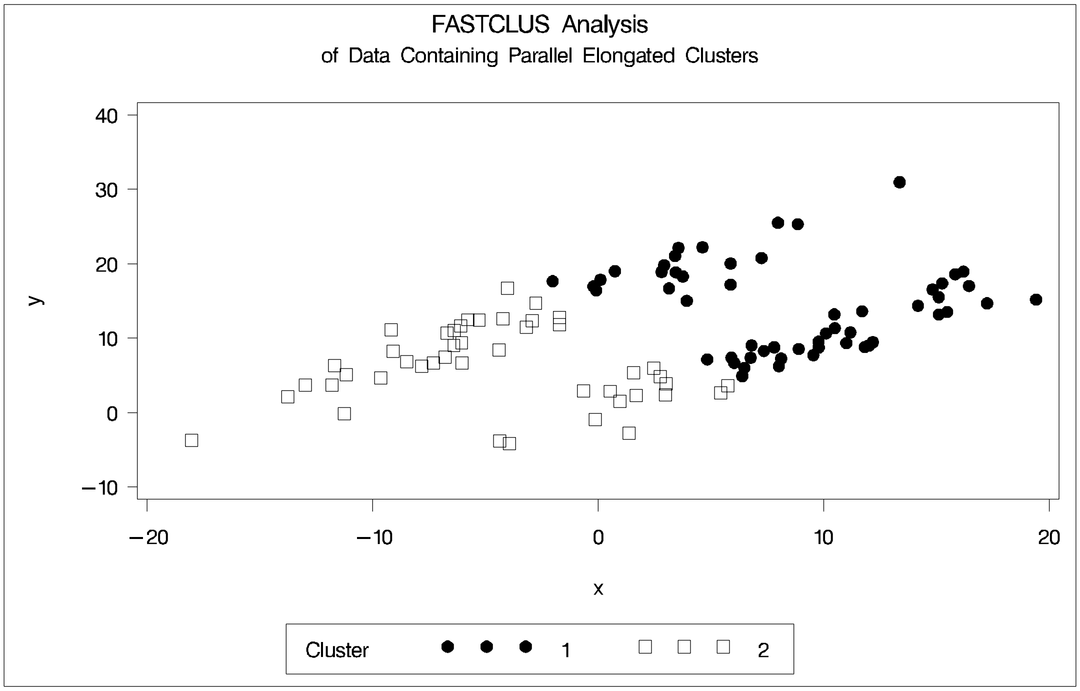


Figure 7.16. Data Containing Parallel Elongated Clusters: PROC FASTCLUS

The following SAS statements produce Figure 7.17:

```
proc cluster data=elongate outtree=tree
             method=average noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

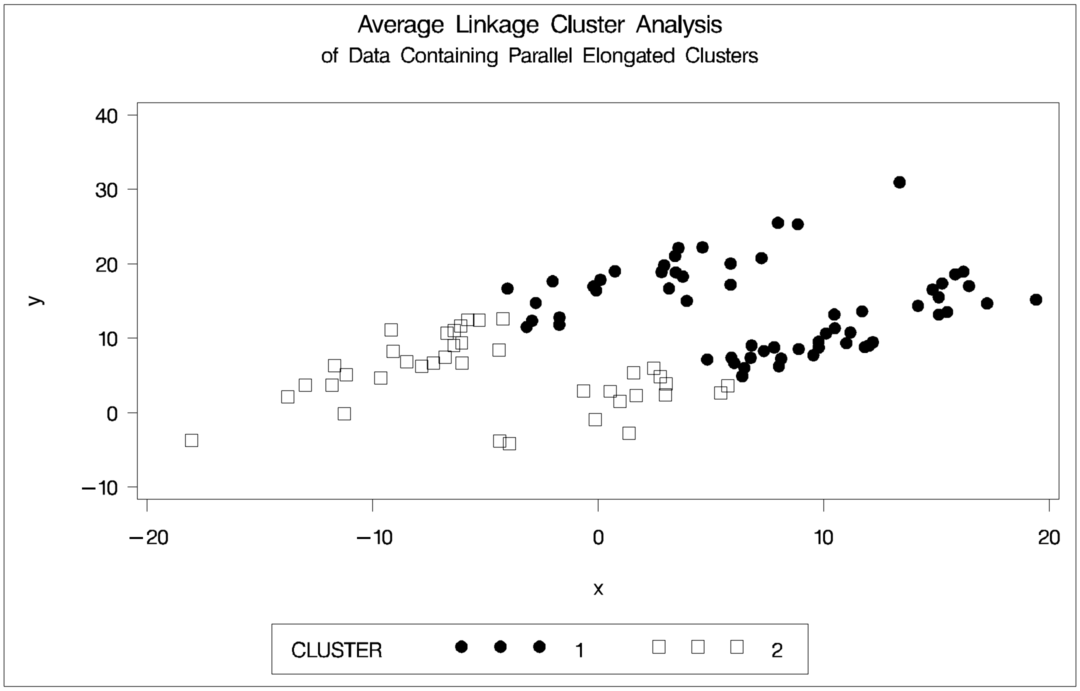


Figure 7.17. Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=AVERAGE

The following SAS statements produce Figure 7.18:

```
proc cluster data=elongate outtree=tree
             method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

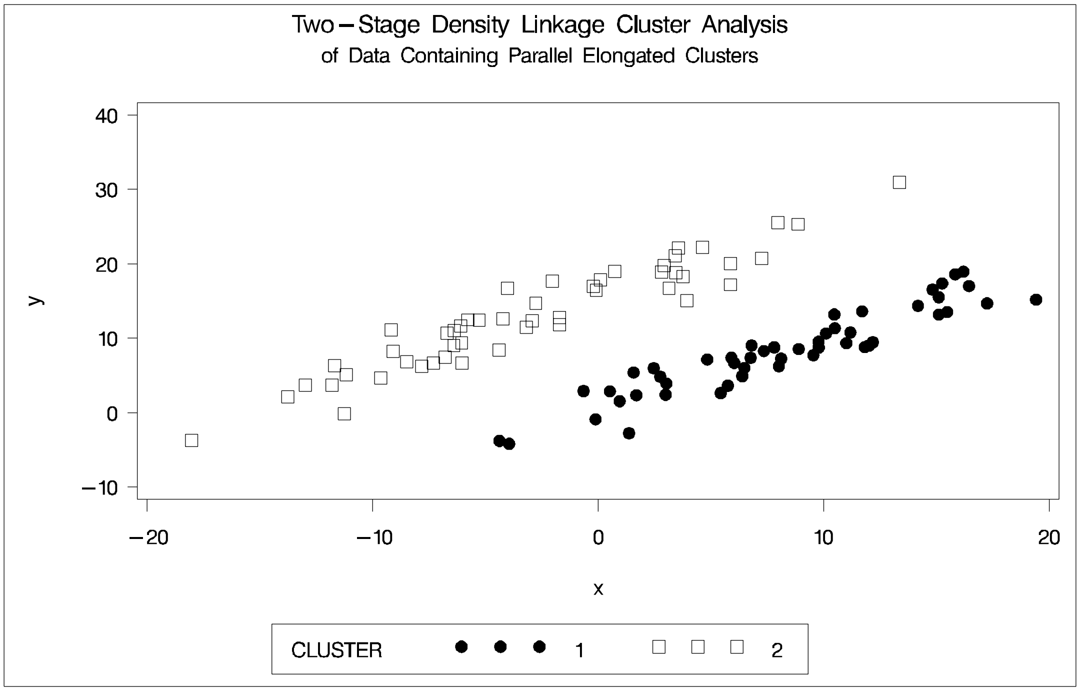


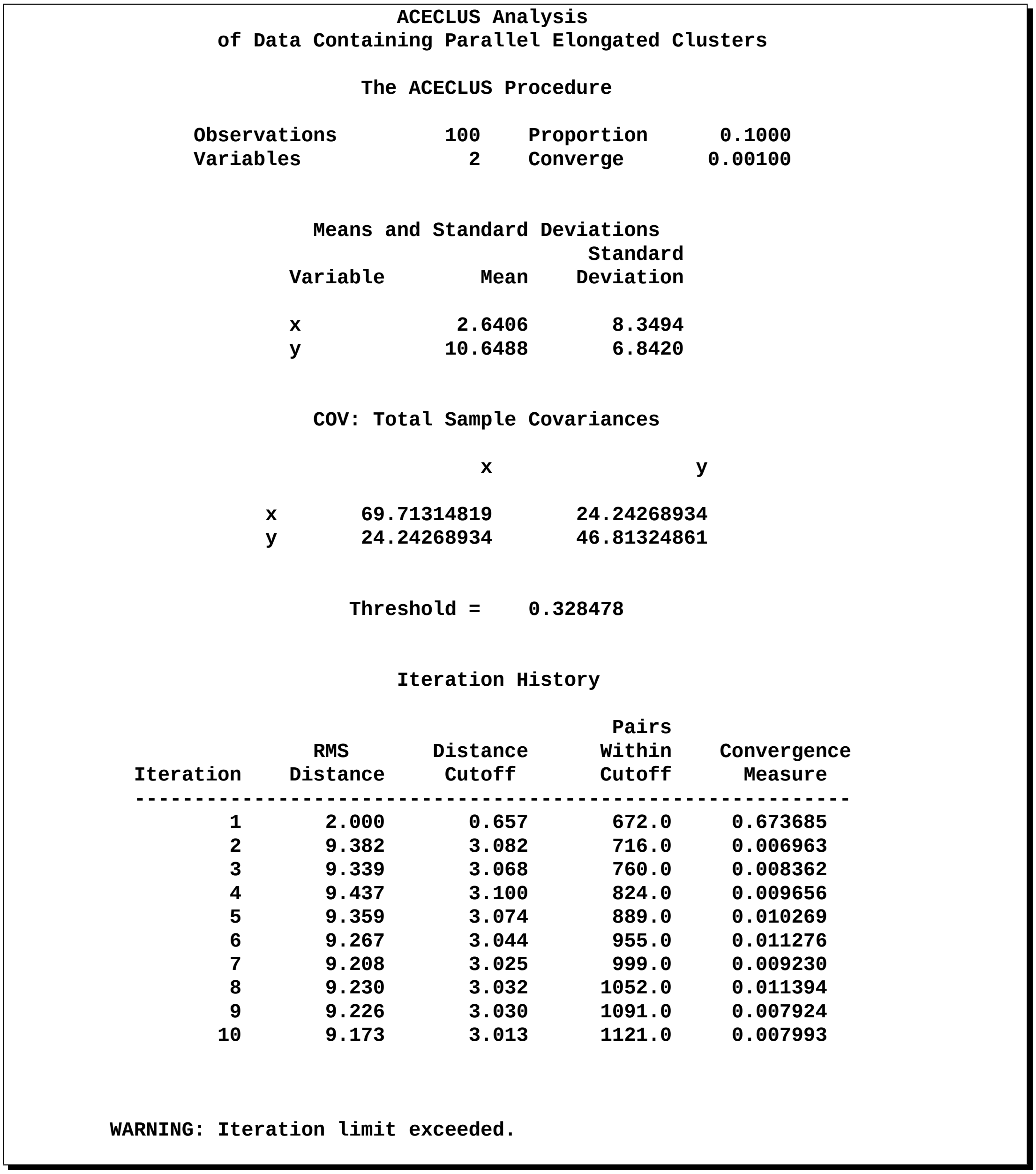
Figure 7.18. Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=TWOSTAGE

PROC FASTCLUS and average linkage fail miserably. Ward's method and the centroid method, not shown, produce almost the same results. Two-stage density linkage, however, recovers the correct clusters. Single linkage, not shown, finds the same clusters as two-stage density linkage except for some outliers.

In this example, the population clusters have equal covariance matrices. If the within-cluster covariances are known, the data can be transformed to make the clusters spherical so that any of the clustering methods can find the correct clusters. But when you are doing a cluster analysis, you do not know what the true clusters are, so you cannot calculate the within-cluster covariance matrix. Nevertheless, it is sometimes possible to estimate the within-cluster covariance matrix without knowing the cluster membership or even the number of clusters, using an approach invented by Art, Gnanadesikan, and Kettenring (1982). A method for obtaining such an estimate is available in the ACECLUS procedure.

In the following analysis, PROC ACECLUS transforms the variables X and Y into canonical variables CAN1 and CAN2. The latter are plotted and then used in a cluster analysis by Ward's method. The clusters are then plotted with the original variables X and Y. The following SAS statements produce Figure 7.19:

```
proc aceclus data=elongate out=ace p=.1;
   var x y;
   title 'ACECLUS Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;


proc gplot;
   plot can2*can1/frame cframe=ligr;
   title 'Data Containing Parallel Elongated Clusters';
   title2 'After Transformation by PROC ACECLUS';
run;
```



```
                        ACECLUS Analysis
         of Data Containing Parallel Elongated Clusters

                     The ACECLUS Procedure

       Observations         100    Proportion      0.1000
       Variables              2    Converge       0.00100


                 Means and Standard Deviations
                                         Standard
               Variable         Mean    Deviation

               x              2.6406       8.3494
               y             10.6488       6.8420


                 COV: Total Sample Covariances

                               x                 y

             x       69.71314819       24.24268934
             y       24.24268934       46.81324861


                    Threshold =    0.328478


                        Iteration History

                                          Pairs
                 RMS       Distance      Within    Convergence
  Iteration    Distance     Cutoff       Cutoff      Measure
  ------------------------------------------------------------
          1       2.000       0.657       672.0     0.673685
          2       9.382       3.082       716.0     0.006963
          3       9.339       3.068       760.0     0.008362
          4       9.437       3.100       824.0     0.009656
          5       9.359       3.074       889.0     0.010269
          6       9.267       3.044       955.0     0.011276
          7       9.208       3.025       999.0     0.009230
          8       9.230       3.032      1052.0     0.011394
          9       9.226       3.030      1091.0     0.007924
         10       9.173       3.013      1121.0     0.007993



WARNING: Iteration limit exceeded.
```

Figure 7.19. Data Containing Parallel Elongated Clusters: PROC ACECLUS

```
                          ACECLUS Analysis
               of Data Containing Parallel Elongated Clusters

                        The ACECLUS Procedure

             ACE: Approximate Covariance Estimate Within Clusters

                                 x                  y

                   x       9.299329632       8.215362614
                   y       8.215362614       8.937753936


                     Eigenvalues of Inv(ACE)*(COV-ACE)

                  Eigenvalue    Difference    Proportion    Cumulative

             1       36.7091       33.1672        0.9120        0.9120
             2        3.5420                      0.0880        1.0000


                 Eigenvectors (Raw Canonical Coefficients)

                              Can1          Can2

                       x      -.748392      0.109547
                       y      0.736349      0.230272


                    Standardized Canonical Coefficients

                              Can1          Can2

                       x      -6.24866       0.91466
                       y       5.03812       1.57553
```

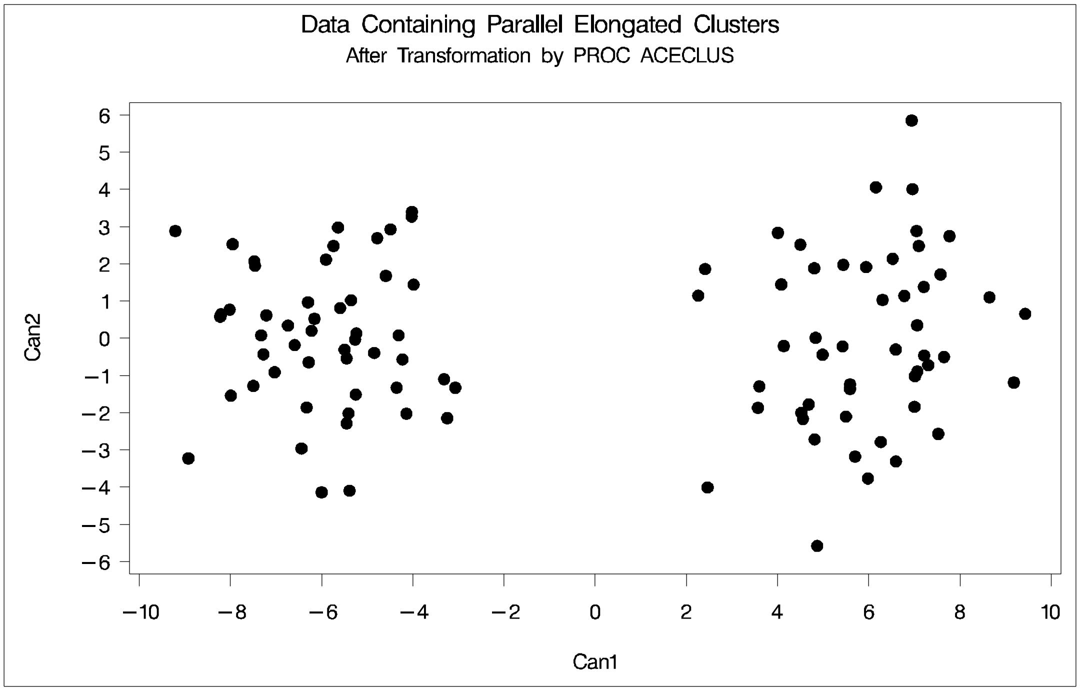


Figure 7.20. Data Containing Parallel Elongated Clusters After Transformation by PROC ACECLUS

The following SAS statements produce Figure 7.21:

```
proc cluster data=ace outtree=tree method=ward noprint;
   var can1 can2;
   copy x y;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
   title3 'After Transformation by PROC ACECLUS';
run;
```

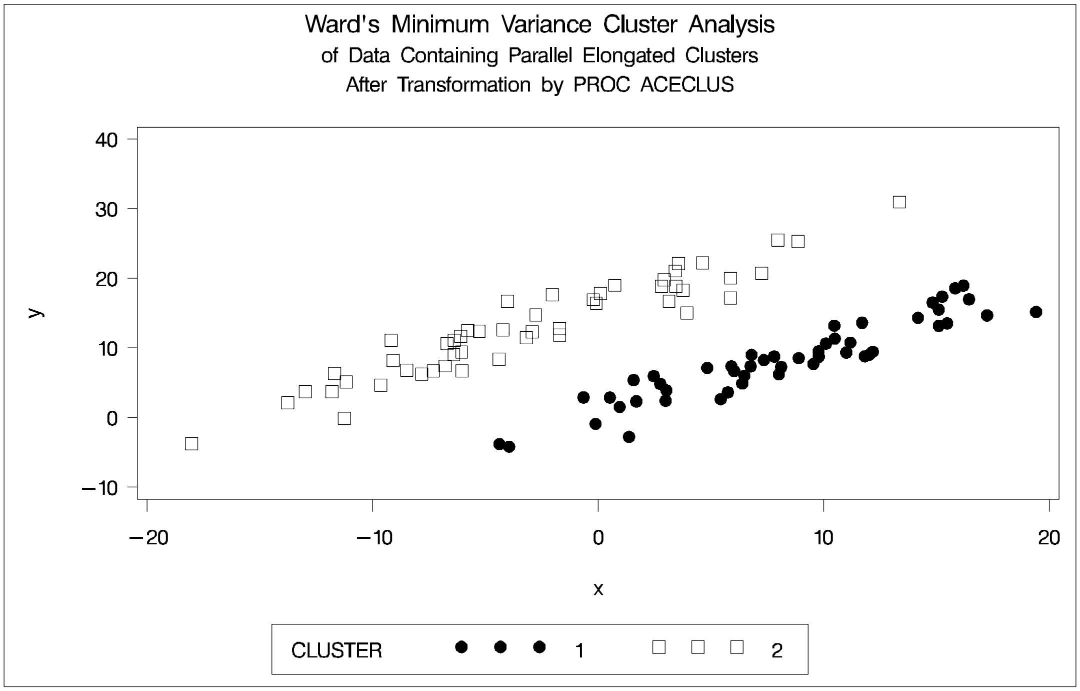


Figure 7.21. Transformed Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=WARD

Nonconvex Clusters

If the population clusters have very different covariance matrices, using PROC ACECLUS is of no avail. Although methods exist for estimating multinormal clusters with unequal covariance matrices (Wolfe 1970; Symons 1981; Everitt and Hand 1981; Titterington, Smith, and Makov 1985; McLachlan and Basford 1988, these methods tend to have serious problems with initialization and may converge to degenerate solutions. For unequal covariance matrices or radically nonnormal distributions, the best approach to cluster analysis is through nonparametric density estimation, as in density linkage. The next example illustrates population clusters with nonconvex density contours. The following SAS statements produce Figure 7.22.

```
data noncon;
   keep x y;
   do i=1 to 100;
      a=i*.0628319;
      x=cos(a)+(i>50)+rannor(7)*.1;
      y=sin(a)+(i>50)*.3+rannor(7)*.1;
      output;
   end;
run;

proc fastclus data=noncon out=out maxc=2 noprint;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

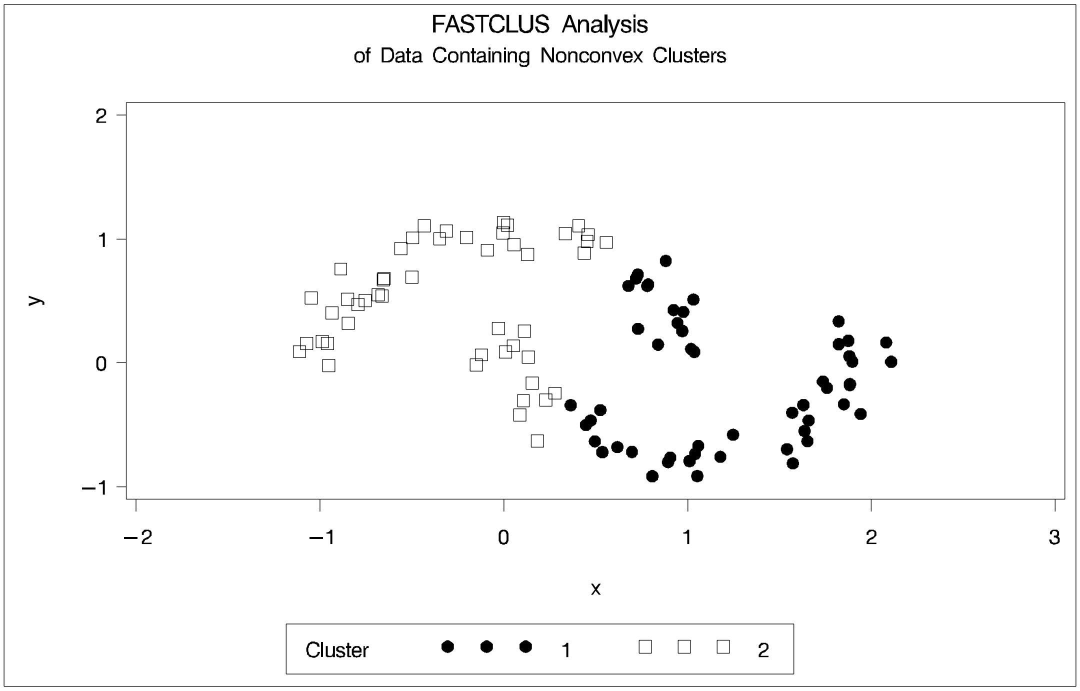


Figure 7.22. Data Containing Nonconvex Clusters: PROC FASTCLUS

The following SAS statements produce Figure 7.23.

```
proc cluster data=noncon outtree=tree
             method=centroid noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

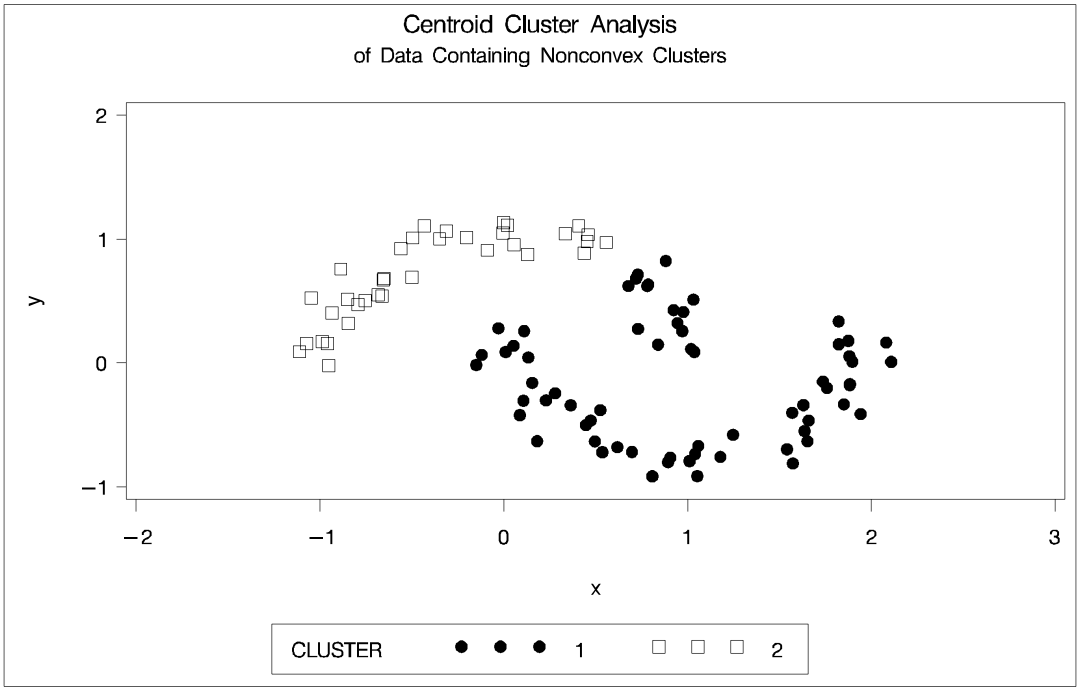


Figure 7.23. Data Containing Nonconvex Clusters: PROC CLUSTER with METHOD=CENTROID

The following SAS statements produce Figure 7.24.

```
proc cluster data=noncon outtree=tree
             method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc gplot;
   plot y*x=cluster/frame cframe=ligr
        vaxis=axis1 haxis=axis2 legend=legend1;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

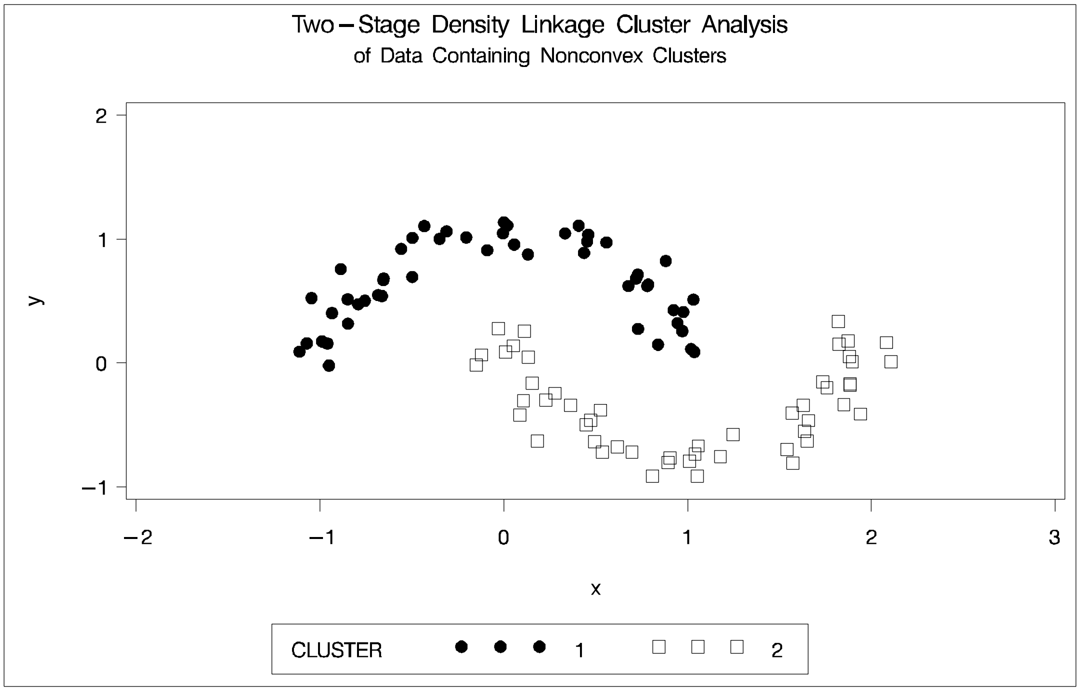


Figure 7.24. Data Containing Nonconvex Clusters: PROC CLUSTER with METHOD=TWOSTAGE

Ward's method and average linkage, not shown, do better than PROC FASTCLUS but not as well as the centroid method. Two-stage density linkage recovers the correct clusters, as does single linkage, which is not shown.

The preceding examples are intended merely to illustrate some of the properties of clustering methods in common use. If you intend to perform a cluster analysis, you should consult more systematic and rigorous studies of the properties of clustering methods, such as Milligan (1980).

The Number of Clusters

There are no completely satisfactory methods for determining the number of population clusters for any type of cluster analysis (Everitt 1979; Hartigan 1985; Bock 1985).

If your purpose in clustering is dissection, that is, to summarize the data without trying to uncover real clusters, it may suffice to look at R^2 for each variable and pooled over all variables. Plots of R^2 against the number of clusters are useful.

It is always a good idea to look at your data graphically. If you have only two or three variables, use PROC GPLOT to make scatter plots identifying the clusters. With more variables, use PROC CANDISC to compute canonical variables for plotting.

Ordinary significance tests, such as analysis of variance F tests, are not valid for testing differences between clusters. Since clustering methods attempt to maximize the separation between clusters, the assumptions of the usual significance tests, parametric or nonparametric, are drastically violated. For example, if you take a sample of 100 observations from a single univariate normal distribution, have PROC FASTCLUS divide it into two clusters, and run a t test between the clusters, you usually obtain a *p*-value of less than 0.0001. For the same reason, methods that purport to test for clusters against the null hypothesis that objects are assigned randomly to clusters (such as McClain and Rao 1975; Klastorin 1983) are useless.

Most valid tests for clusters either have intractable sampling distributions or involve null hypotheses for which rejection is uninformative. For clustering methods based on distance matrices, a popular null hypothesis is that all permutations of the values in the distance matrix are equally likely (Ling 1973; Hubert 1974). Using this null hypothesis, you can do a permutation test or a rank test. The trouble with the permutation hypothesis is that, with any real data, the null hypothesis is implausible even if the data do not contain clusters. Rejecting the null hypothesis does not provide any useful information (Hubert and Baker 1977).

Another common null hypothesis is that the data are a random sample from a multivariate normal distribution (Wolfe 1970, 1978; Duda and Hart 1973; Lee 1979). The multivariate normal null hypothesis arises naturally in normal mixture models (Titterington, Smith, and Makov 1985; McLachlan and Basford 1988). Unfortunately, the likelihood ratio test statistic does not have the usual asymptotic chi-squared distribution because the regularity conditions do not hold. Approximations to the asymptotic distribution of the likelihood ratio have been suggested (Wolfe 1978), but the adequacy of these approximations is debatable (Everitt 1981; Thode, Mendell, and Finch 1988). For small samples, bootstrapping seems preferable (McLachlan and Basford 1988). Bayesian inference provides a promising alternative to likelihood ratio tests for the number of mixture components for both normal mixtures and other types of distributions (Binder 1978, 1981; Banfield and Raftery 1993; Bensmail et al. 1997).

The multivariate normal null hypothesis is better than the permutation null hypothesis, but it is not satisfactory because there is typically a high probability of rejection if the data are sampled from a distribution with lower kurtosis than a normal distribution, such as a uniform distribution. The tables in Englemann and Hartigan (1969), for example, generally lead to rejection of the null hypothesis when the data are sampled from a uniform distribution. Hawkins, Muller, and ten Krooden (1982, pp. 337–340) discuss a highly conservative Bonferroni method for hypothesis testing. The conservativeness of this approach may compensate to some extent for the liberalness exhibited by tests based on normal distributions when the population is uniform.

Perhaps a better null hypothesis is that the data are sampled from a uniform distribution (Hartigan 1978; Arnold 1979; Sarle 1983). The uniform null hypothesis leads to conservative error rates when the data are sampled from a strongly unimodal distribution such as the normal. However, in two or more dimensions and depending on the test statistic, the results can be very sensitive to the shape of the region of support of the uniform distribution. Sarle (1983) suggests using a hyperbox with sides proportional in length to the singular values of the centered coordinate matrix.

Given that the uniform distribution provides an appropriate null hypothesis, there are still serious difficulties in obtaining sampling distributions. Some asymptotic results are available (Hartigan 1978, 1985; Pollard 1981; Bock 1985) for the within-cluster sum of squares, the criterion that PROC FASTCLUS and Ward's minimum variance method attempt to optimize. No distributional theory for finite sample sizes has yet appeared. Currently, the only practical way to obtain sampling distributions for realistic sample sizes is by computer simulation.

Arnold (1979) used simulation to derive tables of the distribution of a criterion based on the determinant of the within-cluster sum of squares matrix $|\mathbf{W}|$. Both normal and uniform null distributions were used. Having obtained clusters with either PROC FASTCLUS or PROC CLUSTER, you can compute Arnold's criterion with the ANOVA or CANDISC procedure. Arnold's tables provide a conservative test because PROC FASTCLUS and PROC CLUSTER attempt to minimize the trace of $\mathbf{W}$ rather than the determinant. Marriott (1971, 1975) also provides useful information on $|\mathbf{W}|$ as a criterion for the number of clusters.

Sarle (1983) used extensive simulations to develop the cubic clustering criterion (CCC), which can be used for crude hypothesis testing and estimating the number of population clusters. The CCC is based on the assumption that a uniform distribution on a hyperrectangle will be divided into clusters shaped roughly like hypercubes. In large samples that can be divided into the appropriate number of hypercubes, this assumption gives very accurate results. In other cases the approximation is generally conservative. For details about the interpretation of the CCC, consult Sarle (1983).

Milligan and Cooper (1985) and Cooper and Milligan (1988) compared thirty methods for estimating the number of population clusters using four hierarchical clustering methods. The three criteria that performed best in these simulation studies with a high degree of error in the data were a pseudo F statistic developed by Calinski and Harabasz (1974), a statistic referred to as $J_e(2)/J_e(1)$ by Duda and Hart (1973) that can be transformed into a pseudo t^2 statistic, and the cubic clustering criterion. The pseudo F statistic and the CCC are displayed by PROC FASTCLUS; these two statistics and the pseudo t^2 statistic, which can be applied only to hierarchical methods, are displayed by PROC CLUSTER. It may be advisable to look for consensus among the three statistics, that is, local peaks of the CCC and pseudo F statistic combined with a small value of the pseudo t^2 statistic and a larger pseudo t^2 for the next cluster fusion. It must be emphasized that these criteria are appropriate only for compact or slightly elongated clusters, preferably clusters that are roughly multivariate normal.

Recent research has tended to de-emphasize mixture models in favor of nonparametric models in which clusters correspond to modes in the probability density function. Hartigan and Hartigan (1985) and Hartigan (1985) developed a test of unimodality versus bimodality in the univariate case.

Nonparametric tests for the number of clusters can also be based on nonparametric density estimates. This approach requires much weaker assumptions than mixture models, namely, that the observations are sampled independently and that the distribution can be estimated nonparametrically. Silverman (1986) describes a bootstrap test for the number of modes using a Gaussian kernel density estimate, but problems have been reported with this method under the uniform null distribution. Further developments in nonparametric methods are given by Mueller and Sawitzki (1991), Minnotte (1992), and Polonik (1993). All of these methods suffer from heavy computational requirements.

One useful descriptive approach to the number-of-clusters problem is provided by Wong and Schaack (1982), based on a kth-nearest-neighbor density estimate. The kth-nearest-neighbor clustering method developed by Wong and Lane (1983) is applied with varying values of k. Each value of k yields an estimate of the number of modal clusters. If the estimated number of modal clusters is constant for a wide range of k values, there is strong evidence of at least that many modes in the population. A plot of the estimated number of modes against k can be highly informative. Attempts to derive a formal hypothesis test from this diagnostic plot have met with difficulties, but a simulation approach similar to Silverman's (1986) does seem to work (Girman 1994). The simulation, of course, requires considerable computer time.

Sarle and Kuo (1993) document a less expensive approximate nonparametric test for the number of clusters that has been implemented in the MODECLUS procedure. This test sacrifices statistical efficiency for computational efficiency. The method for conducting significance tests is described in the chapter on the MODECLUS procedure. This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the *p*-values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from

20 to 2000 indicate that the *p*-values are almost always conservative. The only case discovered so far in which the *p*-values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

References

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, New York: Academic Press, Inc.

Arnold, S.J. (1979), "A Test for Clusters," *Journal of Marketing Research*, 16, 545–551.

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), "Data-based Metrics for Cluster Analysis," *Utilitas Mathematica*, 21A, 75–99.

Banfield, J.D. and Raftery, A.E. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821.

Bensmail, H., Celeux, G., Raftery, A.E., and Robert, C.P. (1997), "Inference in Model-Based Cluster Analysis," *Statistics and Computing*, 7, 1–10.

Binder, D.A. (1978), "Bayesian Cluster Analysis," *Biometrika*, 65, 31–38.

Binder, D.A. (1981), "Approximations to Bayesian Clustering Rules," *Biometrika*, 68, 275–285.

Blashfield, R.K. and Aldenderfer, M.S. (1978), "The Literature on Cluster Analysis," *Multivariate Behavioral Research*, 13, 271–295.

Bock, H.H. (1985), "On Some Significance Tests in Cluster Analysis," *Journal of Classification*, 2, 77–108.

Calinski, T. and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, 3, 1–27.

Cooper, M.C. and Milligan, G.W. (1988), "The Effect of Error on Determining the Number of Clusters," *Proceedings of the International Workshop on Data Analysis, Decision Support and Expert Knowledge Representation in Marketing and Related Areas of Research*, 319–328.

Duda, R.O. and Hart, P.E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons, Inc.

Duran, B.S. and Odell, P.L. (1974), *Cluster Analysis*, New York: Springer-Verlag.

Englemann, L. and Hartigan, J.A. (1969), "Percentage Points of a Test for Clusters," *Journal of the American Statistical Association*, 64, 1647–1648.

Everitt, B.S. (1979), "Unresolved Problems in Cluster Analysis," *Biometrics*, 35, 169–181.

Everitt, B.S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books Ltd.

Everitt, B.S. (1981), “A Monte Carlo Investigation of the Likelihood Ratio Test for the Number of Components in a Mixture of Normal Distributions,” *Multivariate Behavioral Research*, 16, 171–80.

Everitt, B.S. and Hand, D.J. (1981), *Finite Mixture Distributions*, New York: Chapman and Hall.

Girman, C.J. (1994), “Cluster Analysis and Classification Tree Methodology as an Aid to Improve Understanding of Benign Prostatic Hyperplasia,” Ph.D. thesis, Chapel Hill, NC: Department of Biostatistics, University of North Carolina.

Good, I.J. (1977), “The Botryology of Botryology,” in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press, Inc.

Harman, H.H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Hartigan, J.A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons, Inc.

Hartigan, J.A. (1977), “Distribution Problems in Clustering,” in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press, Inc.

Hartigan, J.A. (1978), “Asymptotic Distributions for Clustering Criteria,” *Annals of Statistics*, 6, 117–131.

Hartigan, J.A. (1981), “Consistency of Single Linkage for High-Density Clusters,” *Journal of the American Statistical Association*, 76, 388–394.

Hartigan, J.A. (1985), “Statistical Theory in Clustering,” *Journal of Classification*, 2, 63–76.

Hartigan, J.A. and Hartigan, P.M. (1985), “The Dip Test of Unimodality,” *Annals of Statistics*, 13, 70–84.

Hartigan, P.M. (1985), “Computation of the Dip Statistic to Test for Unimodality,” *Applied Statistics*, 34, 320–325.

Hawkins, D.M., Muller, M.W., and ten Krooden, J.A. (1982), “Cluster Analysis,” in *Topics in Applied Multivariate Analysis*, ed. D.M. Hawkins, Cambridge: Cambridge University Press.

Hubert, L. (1974), “Approximate Evaluation Techniques for the Single-Link and Complete-Link Hierarchical Clustering Procedures,” *Journal of the American Statistical Association*, 69, 698–704.

Hubert, L.J. and Baker, F.B. (1977), “An Empirical Comparison of Baseline Models for Goodness-of-Fit in r-Diameter Hierarchical Clustering,” in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press, Inc.

Klastorin, T.D. (1983), “Assessing Cluster Analysis Results,” *Journal of Marketing Research*, 20, 92–98.

Lee, K.L. (1979), “Multivariate Tests for Clusters,” *Journal of the American Statistical Association*, 74, 708–714.

Ling, R.F (1973), “A Probability Theory of Cluster Analysis,” *Journal of the American Statistical Association*, 68, 159–169.

MacQueen, J.B. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

Marriott, F.H.C. (1971), "Practical Problems in a Method of Cluster Analysis," *Biometrics*, 27, 501–514.

Marriott, F.H.C. (1975), "Separating Mixtures of Normal Distributions," *Biometrics*, 31, 767–769.

Massart, D.L. and Kaufman, L. (1983), *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*, New York: John Wiley & Sons, Inc.

McClain, J.O. and Rao, V.R. (1975), "CLUSTISZ: A Program to Test for the Quality of Clustering of a Set of Objects," *Journal of Marketing Research*, 12, 456–460.

McLachlan, G.J. and Basford, K.E. (1988), *Mixture Models*, New York: Marcel Dekker, Inc.

Mezzich, J.E and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press, Inc.

Milligan, G.W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, 45, 325–342.

Milligan, G.W. (1981), "A Review of Monte Carlo Tests of Cluster Analysis," *Multivariate Behavioral Research*, 16, 379–407.

Milligan, G.W. and Cooper, M.C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, 50, 159–179.

Minnotte, M.C. (1992), "A Test of Mode Existence with Applications to Multimodality," Ph.D. thesis, Rice University, Department of Statistics.

Mueller, D.W. and Sawitzki, G. (1991), "Excess Mass Estimates and Tests for Multimodality," JASA 86, 738–746.

Pollard, D. (1981), "Strong Consistency of k-Means Clustering," *Annals of Statistics*, 9, 135–140.

Polonik, W. (1993), "Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach," Technical Report, Beitraege zur Statistik Nr. 7, Universitaet Heidelberg.

Sarle, W.S. (1982), "Cluster Analysis by Least Squares," *Proceedings of the Seventh Annual SAS Users Group International Conference*, 651–653.

Sarle, W.S. (1983), *Cubic Clustering Criterion*, SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Sarle, W.S and Kuo, An-Hsiang (1993), *The MODECLUS Procedure*, SAS Technical Report P-256, Cary, NC: SAS Institute Inc.

Scott, A.J. and Symons, M.J. (1971), "Clustering Methods Based on Likelihood Ratio Criteria," *Biometrics*, 27, 387–397.

Silverman, B.W. (1986), *Density Estimation*, New York: Chapman and Hall.

Sneath, P.H.A. and Sokal, R.R. (1973), *Numerical Taxonomy*, San Francisco: W.H. Freeman.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, England: Ellis Horwood.

Symons, M.J. (1981), "Clustering Criteria and Multivariate Normal Mixtures," *Biometrics*, 37, 35–43.

Thode, H.C. Jr., Mendell, N.R., and Finch, S.J. (1988), "Simulated Percentage Points for the Null Distribution of the Likelihood Ratio Test for a Mixture of Two Normals," *Biometrics*, 44, 1195–1201.

Titterington, D.M., Smith, A.F.M., and Makov, U.E. (1985), *Statistical Analysis of Finite Mixture Distributions*, New York: John Wiley & Sons, Inc.

Ward, J.H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244.

Wolfe, J.H. (1970), "Pattern Clustering by Multivariate Mixture Analysis," *Multivariate Behavioral Research*, 5, 329–350.

Wolfe, J.H. (1978), "Comparative Cluster Analysis of Patterns of Vocational Interest," *Multivariate Behavioral Research*, 13, 33–44.

Wong, M.A. (1982), "A Hybrid Clustering Method for Identifying High-Density Clusters," *Journal of the American Statistical Association*, 77, 841–847.

Wong, M.A. and Lane, T. (1983), "A kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M.A. and Schaack, C. (1982), "Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 8
Introduction to Scoring, Standardization, and Ranking Procedures

Chapter Contents

Chapter 8
Introduction to Scoring, Standardization, and Ranking Procedures

Overview

Several SAS/STAT procedures are utilities that produce an output data set with new variables that are transformations of data in the input data set. SAS/STAT software includes four of these procedures. The RANK procedure produces rank scores across observations, the SCORE procedure constructs functions across the variables, and the STANDARD and STDIZE procedures transform each variable individually.

RANK ranks the observations of each numeric variable from low to high and outputs ranks or rank scores. For a complete discussion of the RANK procedure, refer to the *SAS Procedures Guide*.

SCORE constructs new variables that are linear combinations of old variables according to a scoring data set. This procedure is used with the FACTOR procedure and other procedures that output scoring coefficients.

STANDARD standardizes variables to a given mean and standard deviation. For a complete discussion of PROC STANDARD, refer to the *SAS Procedures Guide*.

STDIZE standardizes variables by subtracting a location measure and dividing by a scale measure. A variety of location and scale measures are provided. Such measures include the mean, median, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate.

Chapter 9
Introduction to Survival Analysis Procedures

Chapter Contents

Chapter 9
Introduction to Survival Analysis Procedures

Overview

Data that measure lifetime or the length of time until the occurrence of an event are called *lifetime*, *failure time*, or *survival* data. For example, variables of interest might be the lifetime of diesel engines, the length of time a person stayed on a job, or the survival time for heart transplant patients. Such data have special considerations that must be incorporated into any analysis.

Background

Survival data consist of a response (event time, failure time, or survival time) variable that measures the duration of time until a specified event occurs and possibly a set of independent variables thought to be associated with the failure time variable. These independent variables (concomitant variables, covariates, or prognostic factors) can be either discrete, such as sex or race, or continuous, such as age or temperature. The system that gives rise to the event of interest can be biological, as for most medical data, or physical, as for engineering data. The purpose of survival analysis is to model the underlying distribution of the failure time variable and to assess the dependence of the failure time variable on the independent variables.

An intrinsic characteristic of survival data is the possibility for censoring of observations, that is, the actual time until the event is not observed. Such censoring can arise from withdrawal from the experiment or termination of the experiment. Because the response is usually a duration, some of the possible events may not yet have occurred when the period for data collection has terminated. For example, clinical trials are conducted over a finite period of time with staggered entry of patients. That is, patients enter a clinical trial over time and thus the length of follow-up varies by individuals; consequently, the time to the event may not be ascertained on all patients in the study. Additionally, some of the responses may be lost to follow-up (for example, a participant may move or refuse to continue to participate) before termination of data collection. In either case, only a lower bound on the failure time of the censored observations is known. These observations are said to be *right censored*. Thus, an additional variable is incorporated into the analysis indicating which responses are observed event times and which are censored times. More generally, the failure time may only be known to be smaller than a given value (*left censored*) or known to be within a given interval (*interval censored*). There are numerous possible censoring schemes that arise in survival analysis. The monograph by Maddala (1983) discusses several related types of censoring situations, and the text by Kalbfleisch and Prentice (1980) also discusses several censoring schemes. Data with censored observations cannot be analyzed by ignoring the censored observations because, among other considerations, the longer-lived individuals are generally more likely to be censored. The method of analysis must take the censoring into account and correctly use the censored observations as well as the uncensored observations.

Another characteristic of survival data is that the response cannot be negative. This suggests that a transformation of the survival time such as a log transformation may be necessary or that specialized methods may be more appropriate than those that assume a normal distribution for the error term. It is especially important to check any underlying assumptions as a part of the analysis because some of the models used are very sensitive to these assumptions.

Survival Analysis Procedures

There are three SAS procedures for analyzing survival data: LIFEREG, LIFETEST and PHREG. PROC LIFEREG is a parametric regression procedure for modeling the distribution of survival time with a set of concomitant variables. PROC LIFETEST is a nonparametric procedure for estimating the survivor function, comparing the underlying survival curves of two or more samples, and testing the association of survival time with other variables. PROC PHREG is a semiparametric procedure that fits the Cox proportional hazards model.

The LIFEREG Procedure

The LIFEREG procedure fits parametric accelerated failure time models to survival data that may be left, right, or interval censored. The parametric model is of the form

$$y = \mathbf{x}'\boldsymbol{\beta} + \sigma\epsilon$$

where y is usually the log of the failure time variable, $\mathbf{x}$ is a vector of covariate values, $\boldsymbol{\beta}$ is a vector of unknown regression parameters, σ is an unknown scale parameter, and ϵ is an error term. The baseline distribution of the error term can be specified as one of several possible distributions, including, but not limited to, the log normal, log logistic, and Weibull distributions. Several texts that discuss these parametric models are Kalbfleisch and Prentice (1980), Lawless (1982), and Nelson (1990).

The LIFETEST Procedure

The LIFETEST procedure computes nonparametric estimates of the survival distribution function. You can request either the product-limit (Kaplan and Meier 1958) or the life-table (actuarial) estimate of the distribution. The texts by Cox and Oakes (1984) and Kalbfleisch and Prentice (1980) provide good discussions of the product-limit estimator, and the texts by Lee (1992) and Elandt-Johnson and Johnson (1980) include detailed discussions of the life-table estimator. PROC LIFETEST computes nonparametric tests to compare the survival curves of two or more groups. The procedure also computes rank tests of association of the survival time variable with other concomitant variables as given in Kalbfleisch and Prentice (1980, Chapter 6).

The PHREG Procedure

The PHREG procedure fits the proportional hazards model of Cox (1972; 1975) to survival data that may be right censored. The Cox model is a semiparametric model in which the hazard function of the survival time is given by

$$\lambda(t;\mathbf{x}) = \lambda_0(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{x}(t)}$$

where $\lambda_0(t)$ is an unspecified baseline hazard function, $\mathbf{x}(t)$ is a vector of covariate values, possibly time-dependent, and $\boldsymbol{\beta}$ is a vector of unknown regression parameters. The model is referred to as a semiparametric model since part of the model involves the unspecified baseline function over time (which is infinite dimensional) and the other part involves a finite number of regression parameters. Several texts that discuss the Cox regression models are Collett (1994), Cox and Oakes (1984), Kalbfleisch and Prentice (1980), and Lawless (1982).

Survival Analysis with SAS/STAT Procedures

The typical goal in survival analysis is to characterize the distribution of the survival time for a given population, to compare the survival distributions among different groups, or to study the relationship between the survival time and some concomitant variables.

A first step in the analysis of a set of survival data is to use PROC LIFETEST to compute and plot the estimate of the distribution of the survival time. In many applications, there will often be several survival curves to compare. For example, you want to compare the survival experiences of patients who receive different treatments for their disease. The association between covariates and the survival time variable can be investigated by computing estimates of the survival distribution function within strata defined by the covariates. In particular, if the proportional hazards model is appropriate, the estimates of the log(-log(SURVIVAL)) plotted against the log(TIME) variable should give approximately parallel lines, where SURVIVAL is the survival distribution estimate and TIME is the failure time variable. Additionally, these lines should be approximately straight if the Weibull model is appropriate.

Statistics that test for association between failure time and covariates can be used to select covariates for further investigation. The LIFETEST procedure computes linear rank statistics using either Wilcoxon or log-rank scores. These statistics and their estimated covariance matrix can be used with the REG procedure with the option METHOD=RSQUARE to find the subset of variables that produce the largest joint test statistic for association. An example of this method of variable selection is given in the “Examples” section of Chapter 40, “The LIFETEST Procedure.”

Another approach to examine the relationship between the concomitant variables and survival time is through a regression model in which the survival time has a distribution that depends on the concomitant variables. The regression coefficients may be interpreted as describing the direction and strength of the relationship of each explanatory variable on the effect of the survival time.

In many biological systems, the Cox model may be a reasonable description of the relationship between the distribution of the survival time and the prognostic factors. You use PROC PHREG to fit the Cox regression model. The regression coefficient is interpreted as the increase of the log hazard ratio resulting in the increase of one unit in the covariate. However, the underlying hazard function is left unspecified and, as in any other model, the results can be misleading if the proportional hazards assumptions do not hold.

Accelerated failure time models are popular for survival data of physical systems. In many cases, the underlying survival distribution is known empirically. You use PROC LIFEREG to fit these parametric models. Also, PROC LIFEREG can accommodate data with interval-censored observations, which are not allowed in PROC PHREG.

A common technique for checking the validity of a regression model is to embed it in a larger model and use the likelihood ratio test to check whether the reduction to the actual model is valid. Other techniques include examining the residuals. Both PROC LIFEREG and PROC PHREG produce predicted values, residuals, and other computed values that can be used to assess the model adequacy.

References

Collett, D. (1994), *Modeling Survival Data in Medical Research*, London: Chapman and Hall.

Cox, D. R. (1972), “Regression Models and Life Tables,” *Journal of the Royal Statistical Society, Series B*, 20, 187–220 (with discussion).

Cox, D. R. (1975), “Partial Likelihood,” *Biometrika*, 62, 269–276.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman and Hall.

Elandt-Johnson, R. C. and Johnson, N. L. (1980), *Survival Models and Data Analysis*, New York: John Wiley & Sons, Inc.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons, Inc.

Kaplan, E. and Meier, P. (1958), “Nonparametric Estimation from Incomplete Observations,” *Journal of the American Statistical Association*, 53, 457–481.

Lawless, J. F. (1982), *Statistical Methods and Methods for Lifetime Data*, New York: John Wiley & Sons, Inc.

Lee, E. T. (1992), *Statistical Methods for Survival Data Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Maddala, G. S. (1983), *Limited-Dependent and Qualitative Variables in Econometrics*, New York: Cambridge University Press.

Nelson, W. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, New York: John Wiley & Sons, Inc.

Chapter 10
Introduction to Survey Procedures

Chapter Contents

Chapter 10
Introduction to Survey Sampling and Analysis Procedures

Overview

This chapter introduces the SAS/STAT procedures for survey sampling and describes how you can use these procedures to analyze survey data.

Researchers often use sample survey methodology to obtain information about a large population by selecting and measuring a sample from that population. Due to variability among items, researchers apply scientific probability-based designs to select the sample. This reduces the risk of a distorted view of the population and allows statistically valid inferences to be made from the sample. Refer to Lohr (1999), Kalton (1983), Cochran (1977), and Kish (1965) for more information on statistical sampling and analysis of complex survey data. To select probability-based random samples from a study population, you can use the SURVEYSELECT procedure, which provides a variety of methods for probability sampling. To analyze sample survey data, you can use the SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures, which incorporate the sample design into the analyses.

Many SAS/STAT procedures, such as the MEANS, FREQ, GLM and LOGISTIC procedures, can compute sample means, produce crosstabulation tables, and estimate regression relationships. However, in most of these procedures, statistical inference is based on the assumption that the sample is drawn from an infinite population by simple random sampling. If the sample is in fact selected from a finite population using a complex survey design, these procedures generally do not calculate the estimates and their variances according to the design actually used. Using analyses that are not appropriate for your sample design can lead to incorrect statistical inferences.

The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures do properly analyze complex survey data, taking into account the sample design. These procedures can be used for multistage designs or for single-stage designs, with or without stratification, and with or without unequal weighting. The procedures use the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method is appropriate for all designs where the first-stage sample is selected with replacement, or where the first-stage sampling fraction is small, as it often is in practice.

The following table briefly describes the sampling and analysis procedures in SAS/STAT software.

Table 10.1. Sampling and Analysis Procedures in SAS/STAT Software

SURVEYSELECT	
Sampling Methods	simple random sampling unrestricted random sampling (with replacement) systematic sequential selection probability proportional to size (PPS) with and without replacement PPS systematic PPS for two units per stratum sequential PPS with minimum replacement
SURVEYMEANS	
Statistics	estimates of population means and totals estimates of population proportions standard errors confidence limits hypothesis tests domain analyses ratio estimates
SURVEYFREQ	
Analyses	one-way frequency tables two-way and multiway crosstabulation tables estimates of population totals and proportions standard errors confidence limits tests of goodness-of-fit tests of independence
SURVEYREG	
Analyses	linear regression model fitting regression coefficients covariance matrices hypothesis tests confidence limits estimable functions contrasts
SURVEYLOGISTIC	
Analyses	cumulative logit regression model fitting logit, complementary log-log, and probit link functions generalized logit regression model fitting regression coefficients covariance matrices

Table 10.1. (continued)

	hypothesis tests model diagnostics odds ratios confidence limits estimable functions contrasts

The Survey Procedures

The SURVEYSELECT procedure provides methods for probability sample selection. The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures provide statistical analyses for sample survey data. The following sections contain brief descriptions of these procedures. See the chapters on these procedures for more detailed information.

PROC SURVEYSELECT

The SURVEYSELECT procedure provides a variety of methods for selecting probability-based random samples. The procedure can select a simple random sample or a sample according to a complex multistage sample design that includes stratification, clustering, and unequal probabilities of selection. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population.

PROC SURVEYSELECT provides methods for both equal probability sampling and sampling with probability proportional to size (PPS). In PPS sampling, a unit's selection probability is proportional to its size measure. PPS sampling is often used in cluster sampling, where you select clusters (groups of sampling units) of varying size in the first stage of selection. Available PPS methods include without replacement, with replacement, systematic, and sequential with minimum replacement. The procedure can apply these methods for stratified and replicated sample designs.

PROC SURVEYMEANS

The SURVEYMEANS procedure produces estimates of population means and totals from sample survey data. You can use PROC SURVEYMEANS to compute the following statistics:

- estimates of population means, with corresponding standard errors and *t* tests
- estimates of population totals, with corresponding standard deviations and *t* tests
- estimates of proportions for categorical variables, with standard errors and *t* tests
- ratio estimates of population means and proportions, and their standard errors

- confidence limits for population means, totals, and proportions
- data summary information

It is common practice to compute statistics for subpopulations or domains, in addition to computing statistics for the entire study population. Formation of these subpopulations may be unrelated to the sample design, so the domain sample sizes may actually be random variables. *Domain analysis* takes into account this variability, using the entire sample when estimating the variance of domain estimates. This is also known as subgroup analysis, subpopulation analysis, or subdomain analysis. For more information on domain analysis, refer to Lohr (1999) and Cochran (1977).

You can use the SURVEYMEANS procedure to perform domain analysis to compute the following statistics:

- domain (subpopulation) estimates of means, with corresponding standard errors and t tests
- domain (subpopulation) estimates of totals, with corresponding standard deviations and t tests
- proportion estimates within domains for categorical variables, with standard errors and t tests
- confidence limits for domain statistics

PROC SURVEYFREQ

The SURVEYFREQ procedure produces one-way to n-way frequency and crosstabulation tables from sample survey data. These tables include estimates of population totals, population proportions (overall proportions, and also row and column proportions), and corresponding standard errors. Confidence limits, coefficients of variation, and design effects are also available. The procedure also provides a variety of options to customize your table display.

For one-way frequency tables, PROC SURVEYFREQ provides Rao-Scott chi-square goodness-of-fit tests, which are adjusted for the sample design. You can test a null hypothesis of equal proportions for a one-way frequency table, or you can input other null hypothesis proportions for the test. For two-way frequency tables, PROC SURVEYFREQ provides design-adjusted tests of independence, or no association, between the row and column variables. These tests include the Rao-Scott chi-square test, the Rao-Scott likelihood-ratio test, the Wald chi-square test, and the Wald log-linear chi-square test.

PROC SURVEYREG

The SURVEYREG procedure fits linear models for survey data and computes regression coefficients and their variance-covariance matrix. The procedure allows you to specify classification effects using the same syntax as in the GLM procedure. The procedure also provides hypothesis tests for the model effects, for any specified estimable linear functions of the model parameters, and for custom hypothesis tests for linear combinations of the regression parameters. The procedure also computes the confidence limits of the parameter estimates and their linear estimable functions.

PROC SURVEYLOGISTIC

The SURVEYLOGISTIC procedure investigates the relationship between discrete responses and a set of explanatory variables for survey data. The procedure fits linear logistic regression models for discrete response survey data by the method of maximum likelihood, incorporating the sample design into the analysis. The SURVEYLOGISTIC procedure enables you to use categorical classification variables (also known as CLASS variables) as explanatory variables in an explanatory model, using the familiar syntax for main effects and interactions employed in the GLM and LOGISTIC procedures.

The following link functions are available for regression in PROC SURVEYLOGISTIC: the cumulative logit function (CLOGIT), the generalized logit function (GLOGIT), the probit function (PROBIT), and the complementary log-log function (CLOGLOG). The procedure performs maximum likelihood estimation of the regression coefficients with either the Fisher-scoring algorithm or the Newton-Raphson algorithm. Variances of the regression parameters and the odds ratios are computed with a Taylor expansion approximation; refer to Binder (1983) and Morel (1989).

Survey Design Specification

Survey sampling is the process of selecting a probability-based sample from a finite population according to a sample design. You then collect data from these selected units and use them to estimate characteristics of the entire population.

A *sample design* encompasses the rules and operations by which you select sampling units from the population and the computation of sample statistics, which are estimates of the population values of interest. The objective of your survey often determines appropriate sample designs and valid data collection methodology. A complex sample design can include stratification, clustering, multiple stages of selection, and unequal weighting. The survey procedures can be used for single-stage designs or for multistage designs, with or without stratification, and with or without unequal weighting.

To analyze your survey data with the SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures, you need to specify sample design information to the procedures. This information includes design strata, clusters, and sampling weights. All the survey analysis procedures use the same syntax for specifying sample design information. You provide sample design information with the STRATA, CLUSTER, and WEIGHT statements, and with the RATE= or TOTAL= option in the PROC statement.

When there are clusters, or PSUs, in the sample design, the procedures estimate variance from the variance among PSUs, as described in the section "Variance Estimation" on page 166. For a multistage sample design, the variance estimation method depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

The following sections provide brief descriptions of basic sample design concepts and terminology used in the survey procedures. Refer to Lohr (1999), Kalton (1983), Cochran (1977), and Kish (1965).

Population

Population refers to the target population or group of individuals of interest for study. Often, the primary objective is to estimate certain characteristics of this population, called *population values*. A *sampling unit* is an element or an individual in the target population. A sample is a subset of the population that is selected for the study.

Before you use the survey procedures, you should have a well-defined target population, sampling units, and an appropriate sample design.

In order to select a sample according to your sample design, you need to have a list of sampling units in the population. This is called a *sampling frame*. PROC SURVEYSELECT selects a sample using this sampling frame.

Stratification

Stratified sampling involves selecting samples independently within strata, which are nonoverlapping subgroups of the survey population. Stratification controls the distribution of the sample size in the strata. It is widely used to meet a variety of survey objectives. For example, with stratification you can ensure adequate sample sizes for subgroups of interest, including small subgroups, or you can use stratification to improve the precision of overall estimates. To improve precision, units within strata should be as homogeneous as possible for the characteristics of interest.

Clustering

Cluster sampling involves selecting clusters, which are groups of sampling units. For example, clusters may be schools, hospitals, or geographical areas, and sampling units may be students, patients, or citizens. Cluster sampling can provide efficiency in frame construction and other survey operations. However, it can also result in a loss in precision of your estimates, compared to a nonclustered sample of the same size. To minimize this effect, units within clusters should be as heterogeneous as possible for the characteristics of interest.

Multistage Sampling

In *multistage sampling*, you select an initial or first-stage sample based on groups of elements in the population, called *primary sampling units* or *PSUs*.

Then you create a second-stage sample by drawing a subsample from each selected PSU in the first-stage sample. By repeating this operation, you can select a higher-stage sample. If you include all the elements from a selected primary sampling unit, then the two-stage sample is a cluster sample.

Sampling Weights

Sampling weights, or *survey weights*, are positive values associated with each unit in your sample. Ideally, the weight of a sampling unit should be the “frequency” that the sampling unit represents in the target population.

Often, sampling weights are the reciprocals of the selection probabilities for the sampling units. When you use PROC SURVEYSELECT, the procedure generates the sampling weight component for each stage of the design, and you can multiply these sampling weight components to obtain the final sampling weights. Sometimes, sampling weights also include nonresponse adjustments, post-sampling stratification, or regression adjustments using supplemental information.

When the sampling units have unequal weights, you must provide the weights to the survey analysis procedures. If you do not specify sampling weights, the procedures use equal weights in the analyses.

Population Totals and Sampling Rates

The ratio of the sample size (the number of sampling units in the sample) n and the population size (the total number of sampling units in the target population) N is written as

$$f = \frac{n}{N}$$

This ratio is called the *sampling rate* or the *sampling fraction*. If you select a sample without replacement, the extra efficiency compared to selecting a sample with replacement can be measured by the *finite population correction* (*fpc*) factor, $(1 - f)$.

If your analysis includes a finite population correction factor, you can input either the sampling rate or the population total. Otherwise, the procedures do not use the *fpc* when computing variance estimates. For fairly small sampling fractions, it is appropriate to ignore this correction. Refer to Cochran (1977) and Kish (1965).

As discussed in the following section “Variance Estimation,” for a multistage sample design, the variance estimation method depends only on the first stage of the sample design. Therefore, if you are specifying the sampling rate, you should input the *first-stage sampling rate*, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the target population.

Variance Estimation

The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures use the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method is appropriate for all designs where the first-stage sample is selected with replacement, or where the first-stage sampling fraction is small, as it often is in practice. The Taylor expansion method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Fuller 1975, Woodruff 1971). When there are clusters, or primary sampling units (PSUs), in the sample design, the procedures estimate the variance from the variation among the PSUs. When the design is stratified, the procedures pool stratum variance estimates to compute the overall variance estimate.

For a multistage sample design, the variance estimation method depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

For more information on variance estimation for sample survey data, refer to Lohr (1999), Särndal, Swenson, and Wretman (1992), Lee, Forthoffer, and Lorimor (1989), Wolter (1985), Cochran (1977), Kish (1965), and Hansen, Hurwitz, and Madow (1953).

In addition to the traditional Taylor expansion method, other methods for variance estimation for survey data include balanced repeated replication and jackknife repeated replication. These methods usually give similar, satisfactory results (Särndal, Swenson, and Wretman 1992; Wolter 1985); the SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures currently provide only the Taylor expansion method.

See Chapter 70, “The SURVEYMEANS Procedure,” Chapter 68, “The SURVEYFREQ Procedure,” Chapter 71, “The SURVEYREG Procedure,” and Chapter 69, “The SURVEYLOGISTIC Procedure,” for complete details.

Example

This section demonstrates how you can use the survey procedures to select a probability-based sample and then analyze the survey data to make inferences about the population. The analyses includes descriptive statistics and regression analysis. This example is a survey of income and expenditures for a group of households in North Carolina and South Carolina. The goals of the survey are to

- estimate total income and total basic living expenses
- investigate the linear relationship between income and living expenses

Sample Selection

To select a sample with PROC SURVEYSELECT, you input a SAS data set that contains the sampling frame or list of units from which the sample is to be selected. You also specify the selection method, the desired sample size or sampling rate, and other selection parameters.

In this example, the sample design is a stratified simple random sample design, with households as the sampling units. The sampling frame (the list of the group of the households) is stratified by State and Region. Within strata, households are selected by simple random sampling. Using this design, the following PROC SURVEYSELECT statements select a probability sample of households from the HHSample data set:

```
proc surveyselect data=HHSample out=Sample
                  method=srs n=(3, 5, 3, 6, 2);
   strata State Region;
run;
```

The STRATA statement names the stratification variables State and Region. In the PROC SURVEYSELECT statement, the DATA= option names the SAS data set HHSample as the input data set (the sampling frame) from which to select the sample. The OUT= option stores the sample in the SAS data set named Sample. The METHOD=SRS option specifies simple random sampling as the sample selection method. The N= option specifies the stratum sample sizes.

The SURVEYSELECT procedure then selects a stratified random sample of households and produces the output data set Sample, which contains the selected households together with their selection probabilities and sampling weights. The data set Sample also contains the sampling unit identification variable Id and the stratification variables State and Region from the data set HHSample.

Survey Data Analysis

You can use the SURVEYMEANS and SURVEYREG procedures to estimate population values and to perform regression analyses for survey data. The following example briefly shows the capabilities of each procedure. See Chapter 70, “The SURVEYMEANS Procedure,” and Chapter 71, “The SURVEYREG Procedure,” for more detailed information.

To estimate the total income and expenditure in the population from the sample, you specify the input data set containing the sample, the statistics to be computed, the variables to be analyzed, and any stratification variables. The statements to compute the descriptive statistics are as follows:

```
proc surveymeans data=Sample sum clm;
   var Income Expense;
   strata State Region;
   weight Weight;
run;
```

The PROC SURVEYMEANS statement invokes the procedure, specifies the input data set, and requests estimates of population totals and their standard deviations for the analysis variables (SUM), as well as confidence limits for the estimates (CLM).

The VAR statement specifies the two analysis variables, Income and Expense. The STRATA statement identifies State and Region as the stratification variables in the sample design. The WEIGHT statement specifies the sampling weight variable Weight.

You can also use the SURVEYREG procedure to perform regression analysis for sample survey data. Suppose that, in order to explore the relationship between the total income and the total basic living expenses of a household in the survey population, you choose the following linear model to describe the relationship:

$$\text{Expense} = \alpha + \beta * \text{Income} + \text{error}$$

The following statements fit this linear model:

```
proc surveyreg data=Sample;
   strata State Region ;
   model  Expense = Income;
   weight Weight;
run;
```

In the PROC SURVEYREG statement, the DATA= option specifies the input sample survey data as Sample. The STRATA statement identifies the stratification variables as State and Region. The MODEL statement specifies the model, with Expense as the dependent variable and Income as the independent variable. The WEIGHT statement specifies the sampling weight variable Weight.

References

Binder, D. A. (1983), “On the Variances of Asymptotically Normal Estimators from Complex Surveys,” *International Statistical Review*, 51, 279–292.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Fuller, W. A. (1975), “Regression Analysis for Sample Survey,” *Sankhyā*, 37 (3), Series C, 117–132.

Hansen, M. H., Hurwitz, W. N., and Madow, W.G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons, Inc.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications, Inc.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Lee, E. S., Forthoffer, R. N., and Lorimor, R. J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications, Inc.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Morel, G. (1989) "Logistic Regression under Complex Survey Designs," *Survey Methodology*, 15, 203–223.

Särndal, C. E., Swenson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag Inc.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag Inc.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 11
The Four Types of Estimable Functions

Chapter Contents

Chapter 11
The Four Types of Estimable Functions

Overview

The GLM procedure and many other SAS/STAT procedures label the tests and Sums of Squares (SS) associated with the various effects in the model as Type I, Type II, Type III, and Type IV. These four types of hypotheses may not always be sufficient for a statistician to perform all desired hypothesis tests, but they should suffice for the vast majority of analyses. This chapter explains the hypotheses tested by each of the four types of SS. For additional discussion, see Freund, Littell, and Spector (1991) or Milliken and Johnson (1984).

Estimability

For linear models such as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \epsilon$$

with $E(\mathbf{Y}) = \mathbf{X}\beta$, a primary analytical goal is to estimate or test for the significance of certain linear combinations of the elements of $\boldsymbol{\beta}$. This is accomplished by computing linear combinations of the observed $\mathbf{Y}$s. An unbiased linear estimate of a specific linear function of the individual βs, say $\mathbf{L}\boldsymbol{\beta}$, is a linear combination of the $\mathbf{Y}$s that has an expected value of $\mathbf{L}\boldsymbol{\beta}$. Hence, the following definition:

> A linear combination of the parameters $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if a linear combination of the $\mathbf{Y}$s exists that has expected value $\mathbf{L}\boldsymbol{\beta}$.

Any linear combination of the $\mathbf{Y}$s, for instance $\mathbf{KY}$, will have expectation $E(\mathbf{KY}) = \mathbf{KX}\boldsymbol{\beta}$. Thus, the expected value of any linear combination of the $\mathbf{Y}$s is equal to that same linear combination of the rows of $\mathbf{X}$ multiplied by $\boldsymbol{\beta}$. Therefore,

> $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if there is a linear combination of the rows of $\mathbf{X}$ that is equal to $\mathbf{L}$—that is, if and only if there is a $\mathbf{K}$ such that $\mathbf{L} = \mathbf{KX}$.

Thus, the rows of $\mathbf{X}$ form a generating set from which any estimable $\mathbf{L}$ can be constructed. Since the row space of $\mathbf{X}$ is the same as the row space of $\mathbf{X'X}$, the rows of $\mathbf{X'X}$ also form a generating set from which all estimable $\mathbf{L}$s can be constructed. Similarly, the rows of $(\mathbf{X'X})^{-}\mathbf{X'X}$ also form a generating set for $\mathbf{L}$.

Therefore, if $\mathbf{L}$ can be written as a linear combination of the rows of $\mathbf{X}$, $\mathbf{X}'\mathbf{X}$, or $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, then $\mathbf{L}\boldsymbol{\beta}$ is estimable.

Once an estimable $\mathbf{L}$ has been formed, $\mathbf{L}\boldsymbol{\beta}$ can be estimated by computing $\mathbf{Lb}$, where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$. From the general theory of linear models, the unbiased estimator $\mathbf{Lb}$ is, in fact, the *best* linear unbiased estimator of $\mathbf{L}\boldsymbol{\beta}$ in the sense of having minimum variance as well as maximum likelihood when the residuals are normal. To test the hypothesis that $\mathbf{L}\boldsymbol{\beta} = 0$, compute SS $(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = 0) = (\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}\mathbf{Lb}$ and form an F test using the appropriate error term.

General Form of an Estimable Function

This section demonstrates a shorthand technique for displaying the generating set for any estimable $\mathbf{L}$. Suppose

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}$$

$\mathbf{X}$ is a generating set for $\mathbf{L}$, but so is the smaller set

$$\mathbf{X}^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{X}^*$ is formed from $\mathbf{X}$ by deleting duplicate rows.

Since all estimable $\mathbf{L}$s must be linear functions of the rows of $\mathbf{X}^*$ for $\mathbf{L}\boldsymbol{\beta}$ to be estimable, an $\mathbf{L}$ for a single-degree-of-freedom estimate can be represented symbolically as

$$L1 \times (1\,1\,0\,0) + L2 \times (1\,0\,1\,0) + L3 \times (1\,0\,0\,1)$$

or

$$\mathbf{L} = (L1 + L2 + L3,\ L1,\ L2,\ L3)$$

For this example, $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if the first element of $\mathbf{L}$ is equal to the sum of the other elements of $\mathbf{L}$ or if

$$\mathbf{L}\boldsymbol{\beta} = (L1 + L2 + L3) \times \mu + L1 \times A_1 + L2 \times A_2 + L3 \times A_3$$

is estimable for any values of $L1$, $L2$, and $L3$.

If other generating sets for $\mathbf{L}$ are represented symbolically, the symbolic notation looks different. However, the inherent nature of the rules is the same. For example, if row operations are performed on $\mathbf{X}^*$ to produce an identity matrix in the first 3×3 submatrix of the resulting matrix

$$\mathbf{X}^{**} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

then $\mathbf{X}^{**}$ is also a generating set for $\mathbf{L}$. An estimable $\mathbf{L}$ generated from $\mathbf{X}^{**}$ can be represented symbolically as

$$\mathbf{L} = (L1,\ L2,\ L3,\ L1 - L2 - L3)$$

Note that, again, the first element of $\mathbf{L}$ is equal to the sum of the other elements.

With multiple generating sets available, the question arises as to which one is the best to represent $\mathbf{L}$ symbolically. Clearly, a generating set containing a minimum of rows (of full row rank) and a maximum of zero elements is desirable. The generalized inverse of $\mathbf{X}'\mathbf{X}$ computed by the GLM procedure has the property that $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ usually contains numerous zeros. For this reason, PROC GLM uses the nonzero rows of $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ to represent $\mathbf{L}$ symbolically.

If the generating set represented symbolically is of full row rank, the number of symbols $(L1, L2, \ldots)$ represents the maximum rank of any testable hypothesis (in other words, the maximum number of linearly independent rows for any $\mathbf{L}$ matrix that can be constructed). By letting each symbol in turn take on the value of 1 while the others are set to 0, the original generating set can be reconstructed.

Introduction to Reduction Notation

Reduction notation can be used to represent differences in Sums of Squares for two models. The notation $R(\mu, A, B, C)$ denotes the complete main effects model for effects A, B, and C. The notation

$$R(A \mid \mu, B, C)$$

denotes the difference between the model SS for the complete main effects model containing A, B, and C and the model SS for the reduced model containing only B and C.

In other words, this notation represents the differences in Model SS produced by

```
proc glm;
   class a b c;
   model y=a b c;
run;
```

and

```
proc glm;
   class b c;
   model y=b c;
run;
```

As another example, consider a regression equation with four independent variables. The notation $R(\beta_3, \beta_4 | \beta_1, \beta_2)$ denotes the differences in Model SS between

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$$

and

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

With PROC REG, this is the difference in Model SS for the models produced by

```
model y=x1 x2 x3 x4;
```

and

```
model y=x1 x2;
```

Examples

A One-Way Classification Model

For the model

$$Y = \mu + A_i + \epsilon \qquad i = 1, 2, 3$$

the general form of estimable functions $\mathbf{Lb}$ is (from the previous example)

$$\mathbf{L}\boldsymbol{\beta} = L1 \times \mu + L2 \times A_1 + L3 \times A_2 + (L1 - L2 - L3) \times A_3$$

Thus,

$$\mathbf{L} = (L1, L2, L3, L1 - L2 - L3)$$

Tests involving only the parameters A_1, A_2, and A_3 must have an $\mathbf{L}$ of the form

$$\mathbf{L} = (0, L2, L3, -L2 - L3)$$

Since the preceding $\mathbf{L}$ involves only two symbols, hypotheses with at most two degrees-of-freedom can be constructed. For example, let $L2 = 1$ and $L3 = 0$; then let $L2 = 0$ and $L3 = 1$:

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

The preceding $\mathbf{L}$ can be used to test the hypothesis that $A_1 = A_2 = A_3$. For this example, any $\mathbf{L}$ with two linearly independent rows with column 1 equal to zero produces the same Sum of Squares. For example, a pooled linear quadratic

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 1 & -2 & 1 \end{bmatrix}$$

gives the same SS. In fact, for any $\mathbf{L}$ of full row rank and any nonsingular matrix $\mathbf{K}$ of conformable dimensions,

$$\text{SS}(H_0\colon \mathbf{L}\boldsymbol{\beta} = 0) = \text{SS}(H_0\colon \mathbf{KL}\boldsymbol{\beta} = 0)$$

A Three-Factor Main Effects Model

Consider a three-factor main effects model involving the CLASS variables A, B, and C, as shown in Table 11.1.

Table 11.1. Three-Factor Main Effects Model

Obs	A	B	C
1	1	2	1
2	1	1	2
3	2	1	3
4	2	2	2
5	2	2	2

The general form of an estimable function is shown in Table 11.2.

Table 11.2. General Form of an Estimable Function for Three-Factor Main Effects Model

Parameter	Coefficient
μ (Intercept)	$L1$
$A1$	$L2$
$A2$	$L1 - L2$
$B1$	$L4$
$B2$	$L1 - L4$
$C1$	$L6$
$C2$	$L1 + L2 - L4 - 2 \times L6$
$C3$	$-L2 + L4 + L6$

Since only four symbols ($L1$, $L2$, $L4$, and $L6$) are involved, any testable hypothesis will have at most four degrees of freedom. If you form an $\mathbf{L}$ matrix with four linearly independent rows according to the preceding rules, then

$$\text{SS}(H_0\colon \mathbf{L}\boldsymbol{\beta} = 0) = R(\mu, A, B, C)$$

In a main effects model, the usual hypothesis of interest for a main effect is the equality of all the parameters. In this example, it is not possible to test such a hypothesis because of confounding. One way to proceed is to construct a maximum rank hypothesis (MRH) involving only the parameters of the main effect in question. This can be done using the general form of estimable functions. Note the following:

- To get an MRH involving only the parameters of A, the coefficients of $\mathbf{L}$ associated with μ, $B1$, $B2$, $C1$, $C2$, and $C3$ must be equated to zero. Starting at the top of the general form, let $L1 = 0$, then $L4 = 0$, then $L6 = 0$. If $C2$ and $C3$ are not to be involved, then $L2$ must also be zero. Thus, $A1 - A2$ is not estimable; that is, the MRH involving only the A parameters has zero rank and $R(A \mid \mu, B, C) = 0$.
- To obtain the MRH involving only the B parameters, let $L1 = L2 = L6 = 0$. But then to remove $C2$ and $C3$ from the comparison, $L4$ must also be set to 0. Thus, $B1 - B2$ is not estimable and $R(B \mid \mu, A, C) = 0$.
- To obtain the MRH involving only the C parameters, let $L1 = L2 = L4 = 0$. Thus, the MRH involving only C parameters is

  $$C1 - 2 \times C2 + C3 = K \qquad \text{(for any } K\text{)}$$

  or any multiple of the left-hand side equal to K. Furthermore,

  $$\text{SS}(H_0\colon\ C1 - 2 \times C2 + C3 = 0) = R(C \mid \mu, A, B)$$

A Multiple Regression Model

Suppose

$$E(Y) = \beta_0 + \beta_1 \times X1 + \beta_2 \times X2 + \beta_3 \times X3$$

If the $\mathbf{X}'\mathbf{X}$ matrix is of full rank, the general form of estimable functions is as shown in Table 11.3.

Table 11.3. General Form of Estimable Functions for a Multiple Regression Model When $\mathbf{X}'\mathbf{X}$ Matrix Is of Full Rank

Parameter	Coefficient
β_0	$L1$
β_1	$L2$
β_2	$L3$
β_3	$L4$

To test, for example, the hypothesis that $\beta_2 = 0$, let $L1 = L2 = L4 = 0$ and let $L3 = 1$. Then $\text{SS}(\mathbf{L}\boldsymbol{\beta} = 0) = R(\beta_2 \mid \beta_0, \beta_1, \beta_3)$. In the full-rank case, all parameters, as well as any linear combination of parameters, are estimable.

Suppose, however, that $X3 = 2 \times X1 + 3 \times X2$. The general form of estimable functions is shown in Table 11.4.

Table 11.4. General Form of Estimable Functions for a Multiple Regression Model When $\mathbf{X}'\mathbf{X}$ Matrix Is Not of Full Rank

Parameter	Coefficient
β_0	$L1$
β_1	$L2$
β_2	$L3$
β_3	$2 \times L2 + 3 \times L3$

For this example, it is possible to test H_0: $\beta_0 = 0$. However, β_1, β_2, and β_3 are not jointly estimable; that is,

$$\begin{aligned} R(\beta_1 \mid \beta_0, \beta_2, \beta_3) &= 0 \\ R(\beta_2 \mid \beta_0, \beta_1, \beta_3) &= 0 \\ R(\beta_3 \mid \beta_0, \beta_1, \beta_2) &= 0 \end{aligned}$$

Using Symbolic Notation

The preceding examples demonstrate the ability to manipulate the symbolic representation of a generating set. Note that any operations performed on the symbolic notation have corresponding row operations that are performed on the generating set itself.

Estimable Functions

Type I SS and Estimable Functions

The Type I SS and the associated hypotheses they test are by-products of the modified sweep operator used to compute a generalized inverse of $\mathbf{X}'\mathbf{X}$ and a solution to the normal equations. For the model $E(Y) = X1 \times B1 + X2 \times B2 + X3 \times B3$, the Type I SS for each effect correspond to

Effect	Type I SS
$B1$	$R(B1)$
$B2$	$R(B2 \vert B1)$
$B3$	$R(B3 \vert B1, B2)$

The Type I SS are model-order dependent; each effect is adjusted only for the preceding effects in the model.

There are numerous ways to obtain a Type I hypothesis matrix $\mathbf{L}$ for each effect. One way is to form the $\mathbf{X}'\mathbf{X}$ matrix and then reduce $\mathbf{X}'\mathbf{X}$ to an upper triangular matrix by row operations, skipping over any rows with a zero diagonal. The nonzero rows of the resulting matrix associated with $X1$ provide an $\mathbf{L}$ such that

$$\text{SS}(H_0\text{: } \mathbf{L}\boldsymbol{\beta} = 0) = R(B1)$$

The nonzero rows of the resulting matrix associated with $X2$ provide an $\mathbf{L}$ such that

$$\text{SS}(H_0\text{: } \mathbf{L}\boldsymbol{\beta} = 0) = R(B2|B1)$$

The last set of nonzero rows (associated with $X3$) provide an $\mathbf{L}$ such that

$$\text{SS}(H_0\text{: } \mathbf{L}\boldsymbol{\beta} = 0) = R(B3|B1, B2)$$

Another more formalized representation of Type I generating sets for $B1$, $B2$, and $B3$, respectively, is

$$\begin{array}{rclccccccc}
\mathbf{G}_1 & = & (& \mathbf{X}_1'\mathbf{X}_1 & | & \mathbf{X}_1'\mathbf{X}_2 & | & \mathbf{X}_1'\mathbf{X}_3 &) \\
\mathbf{G}_2 & = & (& 0 & | & \mathbf{X}_2'\mathbf{M}_2\mathbf{X}_2 & | & \mathbf{X}_2'\mathbf{M}_2\mathbf{X}_3 &) \\
\mathbf{G}_3 & = & (& 0 & | & 0 & | & \mathbf{X}_3'\mathbf{M}_3\mathbf{X}_3 &)
\end{array}$$

where

$$\mathbf{M}_1 \quad = \quad \mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{X}_1'$$

and

$$\mathbf{M}_2 \quad = \quad \mathbf{M}_1 - \mathbf{M}_1\mathbf{X}_2(\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2)^{-}\mathbf{X}_2'\mathbf{M}_1$$

Using the Type I generating set $\mathbf{G}_2$ (for example), if an $\mathbf{L}$ is formed from linear combinations of the rows of $\mathbf{G}_2$ such that $\mathbf{L}$ is of full row rank and of the same row rank as $\mathbf{G}_2$, then $\text{SS}(H_0 : \mathbf{L}\boldsymbol{\beta} = 0) = R(B2|B1)$.

In the GLM procedure, the Type I estimable functions displayed symbolically when the E1 option is requested are

$$\begin{array}{rcl}
\mathbf{G}_1^* & = & (\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{G}_1 \\
\mathbf{G}_2^* & = & (\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2)^{-}\mathbf{G}_2 \\
\mathbf{G}_3^* & = & (\mathbf{X}_3'\mathbf{M}_2\mathbf{X}_3)^{-}\mathbf{G}_3
\end{array}$$

As can be seen from the nature of the generating sets $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, only the Type I estimable functions for $B3$ are guaranteed not to involve the $B1$ and $B2$ parameters. The Type I hypothesis for $B2$ can (and usually does) involve $B3$ parameters. The Type I hypothesis for $B1$ usually involves $B2$ and $B3$ parameters.

There are, however, a number of models for which the Type I hypotheses are considered appropriate. These are

- balanced ANOVA models specified in proper sequence (that is, interactions do not precede main effects in the MODEL statement and so forth)
- purely nested models (specified in the proper sequence)
- polynomial regression models (in the proper sequence).

Type II SS and Estimable Functions

For main effects models and regression models, the general form of estimable functions can be manipulated to provide tests of hypotheses involving only the parameters of the effect in question. The same result can also be obtained by entering each effect in turn as the last effect in the model and obtaining the Type I SS for that effect. These are the *Type II SS*. Using a modified reversible sweep operator, it is possible to obtain the Type II SS without actually rerunning the model.

Thus, the **Type II SS correspond to the R notation in which each effect is adjusted for all other effects possible**. For a regression model such as

$$E(Y) = X1 \times B1 + X2 \times B2 + X3 \times B3$$

the Type II SS correspond to

Effect	**Type II SS**
$B1$	$R(B1 \mid B2, B3)$
$B2$	$R(B2 \mid B1, B3)$
$B3$	$R(B3 \mid B1, B2)$

For a main effects model (A, B, and C as classification variables), the Type II SS correspond to

Effect	**Type II SS**
A	$R(A \mid B, C)$
B	$R(B \mid A, C)$
C	$R(C \mid A, B)$

As the discussion in the section "A Three-Factor Main Effects Model" on page 177 indicates, for regression and main effects models the Type II SS provide an MRH for each effect that does not involve the parameters of the other effects.

For models involving interactions and nested effects, in the absence of a priori parametric restrictions, it is not possible to obtain a test of a hypothesis for a main effect free of parameters of higher-level effects with which the main effect is involved.

It is reasonable to assume, then, that any test of a hypothesis concerning an effect should involve the parameters of that effect and only those other parameters with which that effect is involved.

Contained Effect

Given two effects $F1$ and $F2$, $F1$ is said to be *contained in* $F2$ provided that

- both effects involve the same continuous variables (if any)
- $F2$ has more CLASS variables than does $F1$, and if $F1$ has CLASS variables, they all appear in $F2$

Note that the interaction effect μ is contained in all pure CLASS effects, but it is not contained in any effect involving a continuous variable. No effect is contained by μ.

Type II, Type III, and Type IV estimable functions rely on this definition, and they all have one thing in common: the estimable functions involving an effect $F1$ also involve the parameters of all effects that contain $F1$, and they do not involve the parameters of effects that do not contain $F1$ (other than $F1$).

Hypothesis Matrix for Type II Estimable Functions

The Type II estimable functions for an effect $F1$ have an $\mathbf{L}$ (before reduction to full row rank) of the following form:

- All columns of $\mathbf{L}$ associated with effects not containing $F1$ (except $F1$) are zero.
- The submatrix of $\mathbf{L}$ associated with effect $F1$ is $(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)^{-}(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)$.
- Each of the remaining submatrices of $\mathbf{L}$ associated with an effect $F2$ that contains $F1$ is $(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)^{-}(\mathbf{X}_1'\mathbf{M}\mathbf{X}_2)$.

In these submatrices,

$$\begin{aligned}
\mathbf{X}_0 &= \text{the columns of } \mathbf{X} \text{ whose associated effects do not contain } F1. \\
\mathbf{X}_1 &= \text{the columns of } \mathbf{X} \text{ associated with } F1. \\
\mathbf{X}_2 &= \text{the columns of } \mathbf{X} \text{ associated with an } F2 \text{ effect that contains } F1. \\
\mathbf{M} &= \mathbf{I} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{X}_0)^{-}\mathbf{X}_0'.
\end{aligned}$$

For the model $Y = A\ B\ A*B$, the Type II SS correspond to

$$R(A\,|\,\mu, B), \quad R(B\,|\,\mu, A), \quad R(A*B\,|\,\mu, A, B)$$

for effects A, B, and $A*B$, respectively. For the model $Y = A\ B(A)\ C(A\ B)$, the Type II SS correspond to

$$R(A\,|\,\mu), \;\; R(B(A)\,|\,\mu, A), \;\; R(C(AB)\,|\,\mu, A, B(A))$$

for effects A, $B(A)$ and $C(AB)$, respectively. For the model $Y = X\ X*X$, the Type II SS correspond to

$$R(X\,|\,\mu, X*X) \;\text{ and }\; R(X*X\,|\,\mu, X)$$

for X and $X*X$, respectively.

Example of Type II Estimable Functions

For a 2×2 factorial with w observations per cell, the general form of estimable functions is shown in Table 11.5. Any nonzero values for $L2$, $L4$, and $L6$ can be used to construct $\mathbf{L}$ vectors for computing the Type II SS for A, B, and $A * B$, respectively.

Table 11.5. General Form of Estimable Functions for 2×2 Factorial

Effect	Coefficient
μ	$L1$
$A1$	$L2$
$A2$	$L1 - L2$
$B1$	$L4$
$B2$	$L1 - L4$
$AB11$	$L6$
$AB12$	$L2 - L6$
$AB21$	$L4 - L6$
$AB22$	$L1 - L2 - L4 + L6$

For a balanced 2×2 factorial with the same number of observations in every cell, the Type II estimable functions are shown in Table 11.6.

Table 11.6. Type II Estimable Functions for Balanced 2×2 Factorial

	Coefficients for Effect		
Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$-L2$	0	0
$B1$	0	$L4$	0
$B2$	0	$-L4$	0
$AB11$	$0.5 * L2$	$0.5 * L4$	$L6$
$AB12$	$0.5 * L2$	$-0.5 * L4$	$-L6$
$AB21$	$-0.5 * L2$	$0.5 * L4$	$-L6$
$AB22$	$-0.5 * L2$	$-0.5 * L4$	$L6$

For an unbalanced 2×2 factorial (with two observations in every cell except the $AB22$ cell, which contains only one observation), the general form of estimable functions is the same as if it were balanced since the same effects are still estimable. However, the Type II estimable functions for A and B are not the same as they were for the balanced design. The Type II estimable functions for this unbalanced 2×2 factorial are shown in Table 11.7.

Table 11.7. Type II Estimable Functions for Unbalanced 2×2 Factorial

	Coefficients for Effect		
Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$-L2$	0	0
$B1$	0	$L4$	0
$B2$	0	$-L4$	0
$AB11$	$0.6 * L2$	$0.6 * L4$	$L6$
$AB12$	$0.4 * L2$	$-0.6 * L4$	$-L6$
$AB21$	$-0.6 * L2$	$0.4 * L4$	$-L6$
$AB22$	$-0.4 * L2$	$-0.4 * L4$	$L6$

By comparing the hypothesis being tested in the balanced case to the hypothesis being tested in the unbalanced case for effects A and B, you can note that the Type II hypotheses for A and B are dependent on the cell frequencies in the design. For unbalanced designs in which the cell frequencies are not proportional to the background population, the Type II hypotheses for effects that are contained in other effects are of questionable merit.

However, if an effect is not contained in any other effect, the Type II hypothesis for that effect is an MRH that does not involve any parameters except those associated with the effect in question.

Thus, Type II SS are appropriate for

- any balanced model
- any main effects model
- any pure regression model
- an effect not contained in any other effect (regardless of the model)

In addition to the preceding, the Type II SS is generally accepted by most statisticians for purely nested models.

Type III and IV SS and Estimable Functions

When an effect is contained in another effect, the Type II hypotheses for that effect are dependent on the cell frequencies. The philosophy behind both the Type III and Type IV hypotheses is that the hypotheses tested for any given effect should be the same for all designs with the same general form of estimable functions.

To demonstrate this concept, recall the hypotheses being tested by the Type II SS in the balanced 2×2 factorial shown in Table 11.6. Those hypotheses are precisely the ones that the Type III and Type IV hypotheses employ for all 2×2 factorials that have at least one observation per cell. The Type III and Type IV hypotheses for a design without missing cells usually differ from the hypothesis employed for the same design with missing cells since the general form of estimable functions usually differs.

Type III Estimable Functions

Type III hypotheses are constructed by working directly with the general form of estimable functions. The following steps are used to construct a hypothesis for an effect $F1$:

1. For every effect in the model except $F1$ and those effects that contain $F1$, equate the coefficients in the general form of estimable functions to zero.

   If $F1$ is not contained in any other effect, this step defines the Type III hypothesis (as well as the Type II and Type IV hypotheses). If $F1$ is contained in other effects, go on to step 2. (See the section "Type II SS and Estimable Functions" on page 181 for a definition of when effect $F1$ is contained in another effect.)
2. If necessary, equate new symbols to compound expressions in the $F1$ block in order to obtain the simplest form for the $F1$ coefficients.
3. Equate all symbolic coefficients outside of the $F1$ block to a linear function of the symbols in the $F1$ block in order to make the $F1$ hypothesis orthogonal to hypotheses associated with effects that contain $F1$.

By once again observing the Type II hypotheses being tested in the balanced 2×2 factorial, it is possible to verify that the A and $A * B$ hypotheses are orthogonal and also that the B and $A * B$ hypotheses are orthogonal. This principle of orthogonality between an effect and any effect that contains it holds for all balanced designs. Thus, construction of Type III hypotheses for any design is a logical extension of a process that is used for balanced designs.

The Type III hypotheses are precisely the hypotheses being tested by programs that reparameterize using the usual assumptions (for example, all parameters for an effect summing to zero). When no missing cells exist in a factorial model, Type III SS coincide with Yates' weighted squares-of-means technique. When cells are missing in factorial models, the Type III SS coincide with those discussed in Harvey (1960) and Henderson (1953).

The following steps illustrate the construction of Type III estimable functions for a 2×2 factorial with no missing cells.

To obtain the $A * B$ interaction hypothesis, start with the general form and equate the coefficients for effects μ, A, and B to zero, as shown in Table 11.8.

Table 11.8. Type III Hypothesis for $A * B$ Interaction

Effect	**General Form**	$L1 = L2 = L4 = 0$
μ	$L1$	0
$A1$	$L2$	0
$A2$	$L1 - L2$	0
$B1$	$L4$	0
$B2$	$L1 - L4$	0
$AB11$	$L6$	$L6$
$AB12$	$L2 - L6$	$-L6$
$AB21$	$L4 - L6$	$-L6$
$AB22$	$L1 - L2 - L4 + L6$	$L6$

The last column in Table 11.8 represents the form of the MRH for $A * B$.

To obtain the Type III hypothesis for A, first start with the general form and equate the coefficients for effects μ and B to zero (let $L1 = L4 = 0$). Next let $L6 = K * L2$, and find the value of K that makes the A hypothesis orthogonal to the A*B hypothesis. In this case, K=0.5. Each of these steps is shown in Table 11.9.

In Table 11.9, the fourth column (under $L6 = K * L2$) represents the form of all estimable functions not involving μ, $B1$, or $B2$. The prime difference between the Type II and Type III hypotheses for A is the way K is determined. Type II chooses K as a function of the cell frequencies, whereas Type III chooses K such that the estimable functions for A are orthogonal to the estimable functions for $A * B$.

Table 11.9. Type III Hypothesis for A

Effect	**General Form**	$L1 = L4 = 0$	$L6 = K * L2$	$K = 0.5$
μ	$L1$	0	0	0
$A1$	$L2$	$L2$	$L2$	$L2$
$A2$	$L1 - L2$	$-L2$	$-L2$	$-L2$
$B1$	$L4$	0	0	0
$B2$	$L1 - L4$	0	0	0
$AB11$	$L6$	$L6$	$K * L2$	$0.5 * L2$
$AB12$	$L2 - L6$	$L2 - L6$	$(1 - K) * L2$	$0.5 * L2$
$AB21$	$L4 - L6$	$-L6$	$-K * L2$	$-0.5 * L2$
$AB22$	$L1 - L2 - L4 + L6$	$-L2 + L6$	$(K - 1) * L2$	$-0.5 * L2$

An example of Type III estimable functions in a 3×3 factorial with unequal cell frequencies and missing diagonals is given in Table 11.10 (N_1 through N_6 represent the nonzero cell frequencies).

Table 11.10. A 3×3 Factorial Design with Unequal Cell Frequencies and Missing Diagonals

			B	
		1	2	3
A	1		N_1	N_2
	2	N_3		N_4
	3	N_5	N_6	

For any nonzero values of N_1 through N_6, the Type III estimable functions for each effect are shown in Table 11.11.

Table 11.11. Type III Estimable Functions for 3×3 Factorial Design with Unequal Cell Frequencies and Missing Diagonals

Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$L3$	0	0
$A3$	$-L2 - L3$	0	0
$B1$	0	$L5$	0
$B2$	0	$L6$	0
$B3$	0	$-L5 - L6$	0
$AB12$	$0.667 * L2 + 0.333 * L3$	$0.333 * L5 + 0.667 * L6$	$L8$
$AB13$	$0.333 * L2 - 0.333 * L3$	$-0.333 * L5 - 0.667 * L6$	$-L8$
$AB21$	$0.333 * L2 + 0.667 * L3$	$0.667 * L5 + 0.333 * L6$	$-L8$
$AB23$	$-0.333 * L2 + 0.333 * L3$	$-0.667 * L5 - 0.333 * L6$	$L8$
$AB31$	$-0.333 * L2 - 0.667 * L3$	$0.333 * L5 - 0.333 * L6$	$L8$
$AB32$	$-0.667 * L2 - 0.333 * L3$	$-0.333 * L5 + 0.333 * L6$	$-L8$

Type IV Estimable Functions

By once again looking at the Type II hypotheses being tested in the balanced 2×2 factorial (see Table 11.6), you can see another characteristic of the hypotheses employed for balanced designs: the coefficients of lower-order effects are averaged across each higher-level effect involving the same subscripts. For example, in the A hypothesis, the coefficients of $AB11$ and $AB12$ are equal to one-half the coefficient of $A1$, and the coefficients of $AB21$ and $AB22$ are equal to one-half the coefficient of $A2$. With this in mind then, the basic concept used to construct Type IV hypotheses is that the coefficients of any effect, say $F1$, are distributed equitably across higher-level effects that contain $F1$. When missing cells occur, this same general philosophy is adhered to, but care must be taken in the way the distributive concept is applied.

Construction of Type IV hypotheses begins as does the construction of the Type III hypotheses. That is, for an effect $F1$, equate to zero all coefficients in the general form that do not belong to $F1$ or to any other effect containing $F1$. If $F1$ is not contained in any other effect, then the Type IV hypothesis (and Type II and III) has been found. If $F1$ is contained in other effects, then simplify, if necessary, the coefficients associated with $F1$ so that they are all free coefficients or functions of other free coefficients in the $F1$ block.

To illustrate the method of resolving the free coefficients outside of the $F1$ block, suppose that you are interested in the estimable functions for an effect A and that A is contained in AB, AC, and ABC. (In other words, the main effects in the model are A, B, and C.)

With missing cells, the coefficients of intermediate effects (here they are AB and AC) do not always have an equal distribution of the lower-order coefficients, so the coefficients of the highest-order effects are determined first (here it is ABC). Once the highest-order coefficients are determined, the coefficients of intermediate effects are automatically determined.

The following process is performed for each free coefficient of A in turn. The resulting symbolic vectors are then added together to give the Type IV estimable functions for A.

1. Select a free coefficient of A, and set all other free coefficients of A to zero.
2. If any of the levels of A have zero as a coefficient, equate all of the coefficients of higher-level effects involving that level of A to zero. This step alone usually resolves most of the free coefficients remaining.
3. Check to see if any higher-level coefficients are now zero when the coefficient of the associated level of A is not zero. If this situation occurs, the Type IV estimable functions for A are not unique.
4. For each level of A in turn, if the A coefficient for that level is nonzero, count the number of times that level occurs in the higher-level effect. Then equate each of the higher-level coefficients to the coefficient of that level of A divided by the count.

An example of a 3×3 factorial with four missing cells (N_1 through N_5 represent positive cell frequencies) is shown in Table 11.12.

Table 11.12. 3×3 Factorial Design with Four Missing Cells

			B	
		1	2	3
A	1	N_1	N_2	
	2	N_3	N_4	
	3			N_5

The Type IV estimable functions are shown in Table 11.13.

Table 11.13. Type IV Estimable Functions for 3×3 Factorial Design with Four Missing Cells

Effect	A	B	$A * B$
μ	0	0	0
$A1$	$-L3$	0	0
$A2$	$L3$	0	0
$A3$	0	0	0
$B1$	0	$L5$	0
$B2$	0	$-L5$	0
$B3$	0	0	0
$AB11$	$-0.5 * L3$	$0.5 * L5$	$L8$
$AB12$	$-0.5 * L3$	$-0.5 * L5$	$-L8$
$AB21$	$0.5 * L3$	$0.5 * L5$	$-L8$
$AB22$	$0.5 * L3$	$-0.5 * L5$	$L8$
$AB33$	0	0	0

A Comparison of Type III and Type IV Hypotheses

For the vast majority of designs, Type III and Type IV hypotheses for a given effect are the same. Specifically, they are the same for any effect $F1$ that is not contained in other effects for any design (with or without missing cells). For factorial designs with no missing cells, the Type III and Type IV hypotheses coincide for all effects. When there are missing cells, the hypotheses can differ. By using the GLM procedure, you can study the differences in the hypotheses and then decide on the appropriateness of the hypotheses for a particular model.

The Type III hypotheses for three-factor and higher completely nested designs with unequal Ns in the lowest level differ from the Type II hypotheses; however, the Type IV hypotheses do correspond to the Type II hypotheses in this case.

When missing cells occur in a design, the Type IV hypotheses may not be unique. If this occurs in PROC GLM, you are notified, and you may need to consider defining your own specific comparisons.

References

Freund, R.J., Littell, R.C., and Spector, P.C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Harvey, W. R. (1960),“Least-Squares Analysis of Data with Unequal Subclass Frequencies,” USDA, *Agriculture Research Service*, ARS 20-8, reprinted with corrections as ARS H-4, 1975, also reprinted 1979.

Henderson, C. R. (1953), “Estimation of Variance and Covariance Components,” *Biometrics*, 9, 226–252.

Goodnight, J.H. (1978), *Tests of Hypotheses in Fixed Effects Linear Models*, SAS Technical Report R-101, Cary, NC: SAS Institute Inc.

Milliken, G.A. and Johnson, D.E. (1984), *Analysis of Messy Data, Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Chapter 12

Introduction to Nonparametric Analysis

Chapter Contents

Chapter 12
Introduction to Nonparametric Analysis

Overview

In statistical inference, or hypothesis testing, the traditional tests are called *parametric tests* because they depend on the specification of a probability distribution (such as the normal) except for a set of free parameters. Parametric tests are said to depend on distributional assumptions. *Nonparametric tests*, on the other hand, do not require any strict distributional assumptions. Even if the data are distributed normally, nonparametric methods are often almost as powerful as parametric methods.

Many nonparametric methods analyze the ranks of a variable rather than the original values. Procedures such as PROC NPAR1WAY calculate the ranks for you and then perform appropriate nonparametric tests. However, there are some situations in which you use a procedure such as PROC RANK to calculate ranks and then use another procedure to perform the appropriate test. See the section "Obtaining Ranks" on page 198 for details.

Although the NPAR1WAY procedure is specifically targeted for nonparametric analysis, many other procedures also perform nonparametric analyses. Some general references on nonparametrics include Hollander and Wolfe (1999), Conover (1999), Gibbons and Chakraborti (1992), Hettmansperger (1984), Randles and Wolfe (1979), and Lehmann (1975).

Testing for Normality

Many parametric tests assume an underlying normal distribution for the population. If your data do not meet this assumption, you may prefer to use a nonparametric analysis.

Base SAS software provides several tests for normality in the UNIVARIATE procedure. Depending on your sample size, PROC UNIVARIATE performs the Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling, and Cramér-von Mises tests. For more on PROC UNIVARIATE, refer to the *Base SAS 9.1 Procedures Guide*.

Comparing Distributions

To test the hypothesis that two or more groups of observations have identical distributions, use the NPAR1WAY procedure, which provides empirical distribution function (EDF) statistics. The procedure calculates the Kolmogorov-Smirnov test, the the Cramér-von Mises test, and, when the data are classified into only two samples, the Kuiper test. Exact *p*-values are available for the two-sample Kolmogorov-Smirnov test. To obtain these tests, use the EDF option in the PROC NPAR1WAY statement. For details, see Chapter 52, "The NPAR1WAY Procedure."

One-Sample Tests

Base SAS software provides two one-sample tests in the UNIVARIATE procedure: a sign test and the Wilcoxon signed rank test. Both tests are designed for situations where you want to make an inference about the location (median) of a population. For example, suppose you want to test whether the median resting pulse rate of marathon runners differs from a specified value.

By default, both of these tests examine the hypothesis that the median of the population from which the sample is drawn is equal to a specified value, which is zero by default. The Wilcoxon signed rank test requires that the distribution be symmetric; the sign test does not require this assumption. These tests can also be used for the case of two related samples; see the section "Comparing Two Independent Samples" for more information.

The two tests are automatically provided by the UNIVARIATE procedure. For details, formulas, and examples, refer to the chapter on the UNIVARIATE procedure in the *Base SAS 9.1 Procedures Guide*.

Two-Sample Tests

This section describes tests appropriate for two independent samples (for example, two groups of subjects given different treatments) and for two related samples (for example, before-and-after measurements on a single group of subjects). Related samples are also referred to as paired samples or matched pairs.

Comparing Two Independent Samples

SAS/STAT software provides several nonparametric tests for location and scale differences.

When you perform these tests, your data should consist of a random sample of observations from two different populations. Your goal is either to compare the location parameters (medians) or the scale parameters of the two populations. For example, suppose your data consist of the number of days in the hospital for two groups of patients: those who received a standard surgical procedure and those who received a new, experimental surgical procedure. These patients are a random sample from the population of patients who have received the two types of surgery. Your goal is to decide whether the median hospital stays differ for the two populations.

Tests in the NPAR1WAY Procedure

The NPAR1WAY procedure provides the following location tests: Wilcoxon rank sum test (Mann-Whitney U test), Median test, Savage test, and Van der Waerden test. Also note that the Wilcoxon rank sum test can be obtained from the FREQ procedure. In addition, PROC NPAR1WAY produces the following tests for scale differences: Siegel-Tukey test, Ansari-Bradley test, Klotz test, and Mood test. PROC NPAR1WAY also provides tests using the input data observations as scores, enabling you to produce a wide variety of tests. You can construct any scores with the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank test. You can also directly analyze the raw data this way, producing the permutation test known as Pitman's test.

When data are sparse, skewed, or heavily tied, the usual asymptotic tests may not be appropriate. In these situations, exact tests may be suitable for analyzing your data. The NPAR1WAY procedure can produce exact *p*-values for all of the two-sample tests for location and scale differences.

Chapter 52, "The NPAR1WAY Procedure," provides detailed statistical formulas for these statistics, as well as examples of their use.

Tests in the FREQ Procedure

This procedure provides a test for comparing the location of two groups and for testing for independence between two variables.

The situation in which you want to compare the location of two groups of observations corresponds to a table with two rows. In this case, the asymptotic Wilcoxon rank sum test can be obtained by using SCORES=RANK in the TABLES statement and by looking at either of the following:

- the Mantel-Haenszel statistic in the list of tests for no association. This is labeled as "Mantel Haenszel Chi-Square" and PROC FREQ displays the statistic, the degrees of freedom, and the *p*-value. To obtain this statistic, specify the CHISQ option in the TABLES statement.
- the CMH statistic 2 in the section on Cochran-Mantel-Haenszel statistics. PROC FREQ displays the statistic, the degrees of freedom, and the *p*-value. To obtain this statistic, specify the CMH2 option in the TABLES statement.

When you test for independence, the question being answered is whether the two variables of interest are related in some way. For example, you might want to know if student scores on a standard test are related to whether students attended a public or private school. One way to think of this situation is to consider the data as a two-way table; the hypothesis of interest is whether the rows and columns are independent. In the preceding example, the groups of students would form the two rows, and the scores would form the columns. The special case of a two-category response (Pass/Fail) leads to a 2×2 table; the case of more than two categories for the response (A/B/C/D/F) leads to a $2 \times c$ table, where c is the number of response categories.

For testing whether two variables are independent, PROC FREQ provides Fisher's exact test. For a 2×2 table, PROC FREQ automatically provides Fisher's exact test when you specify the CHISQ option in the TABLES statement. For a $2 \times c$ table, use the FISHER option in the EXACT statement to obtain the test.

See Chapter 29, "The FREQ Procedure," for details, formulas, and examples of these tests.

Comparing Two Related Samples

SAS/STAT software provides the following nonparametric tests for comparing the locations of two related samples:

- Wilcoxon signed rank test
- sign test
- McNemar's test

The first two tests are available in the UNIVARIATE procedure, and the last test is available in the FREQ procedure. When you perform these tests, your data should consist of pairs of measurements for a random sample from a single population. For example, suppose your data consist of SAT scores for students before and after attending a course on how to prepare for the SAT. The pairs of measurements are the scores before and after the course, and the students should be a random sample of students who attended the course. Your goal in analysis is to decide if the median change in scores is significantly different from zero.

Tests in the UNIVARIATE Procedure

By default, PROC UNIVARIATE performs a Wilcoxon signed rank test and a sign test. To use these tests on two related samples, perform the following steps:

1. In the DATA step, create a new variable that contains the differences between the two related variables.
2. Run PROC UNIVARIATE, using the new variable in the VAR statement.

For discussion of the tests, formulas, and examples, refer to the chapter on the UNIVARIATE procedure in the *Base SAS 9.1 Procedures Guide*.

Tests in the FREQ Procedure

The FREQ procedure can be used to obtain McNemar's test, which is simply another special case of a Cochran-Mantel-Haenszel statistic (and also of the sign test). The AGREE option in the TABLES statement produces this test for 2×2 tables, and exact p-values are also available for this test. See Chapter 29, "The FREQ Procedure," for more information.

Tests for k Samples

Comparing k Independent Samples

One goal in comparing k independent samples is to determine whether the location parameters (medians) of the populations are different. Another goal is to determine whether the scale parameters for the populations are different. For example, suppose new employees are randomly assigned to one of three training programs. At the end of the program, the employees receive a standard test that gives a rating score of their job ability. The goal of analysis is to compare the median scores for the three groups and decide whether the differences are real or due to chance alone.

To compare k independent samples, either the NPAR1WAY or the FREQ procedure provides a Kruskal-Wallis test. PROC NPAR1WAY also provides the Savage, median, and Van der Waerden tests. In addition, PROC NPAR1WAY produces the following tests for scale differences: Siegel-Tukey test, Ansari-Bradley test, Klotz test, and Mood test. Note that you can obtain exact *p*-values for all of these tests.

In addition, you can specify the SCORES=DATA option to use the input data observations as scores. This enables you to produce a very wide variety of tests. You can construct any scores using the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank and one-way ANOVA tests. You can also analyze the raw data with the SCORES=DATA option; for two-sample data, this permutation test is known as Pitman's test.

See Chapter 52, "The NPAR1WAY Procedure," for details, formulas, and examples.

To produce a Kruskal-Wallis test in the FREQ procedure, use SCORES=RANK and the CMH2 option in the TABLES statement. Then, look at the second Cochran-Mantel-Haenszel statistic (labeled "Row Mean Scores Differ") to obtain the Kruskal-Wallis test. The FREQ procedure also provides the Jonckheere-Terpstra test, which is more powerful than the Kruskal-Wallis test for comparing k samples against ordered alternatives. The exact test is also available. In addition, you can obtain a ridit analysis, developed by Bross (1958), by specifying SCORES=RIDIT or SCORES=MODRIDIT in the TABLES statement in the FREQ procedure. See Chapter 29, "The FREQ Procedure," for more information.

Comparing k Dependent Samples

Friedman's test enables you to compare the locations of three or more dependent samples. You can obtain Friedman's Chi-square with the FREQ procedure by using the CMH2 option and SCORES=RANK and looking at the second CMH statistic in the output. For an example, see Chapter 29, "The FREQ Procedure." This chapter also contains formulas and other details on the CMH statistics. For a discussion of how to use the RANK and GLM procedures to obtain Friedman's test, refer to Ipe (1987).

Measures of Correlation and Associated Tests

The CORR procedure in Base SAS software provides several nonparametric measures of association and associated tests. It computes Spearman's rank-order correlation, Kendall's tau-b, and Hoeffding's measure of dependence, and it provides tests for each of these statistics. PROC CORR also computes Spearman's partial rank-order correlation and Kendall's partial tau-b. Finally, PROC CORR computes Cronbach's coefficient alpha for raw and standardized variables. This statistic can be used to estimate the reliability coefficient. For a general discussion of correlations, formulas, interpretation, and examples, refer to the chapter on the CORR procedure in the *Base SAS 9.1 Procedures Guide*.

The FREQ procedure also provides some nonparametric measures of association: gamma, Kendall's tau-b, Stuart's tau-c, Somer's D, and the Spearman rank correlation. The output includes the measure, the asymptotic standard error, confidence limits, and the asymptotic test that the measure equals zero. For the Spearman rank correlation, you can optionally request an exact *p*-value that the correlation is equal to zero. For more information, see Chapter 29, "The FREQ Procedure."

Obtaining Ranks

The primary procedure for obtaining ranks is the RANK procedure in Base SAS software. Note that the PRINQUAL and TRANSREG procedures also provide rank transformations. With all three of these procedures, you can create an output data set and use it as input to another SAS/STAT procedure or to the IML procedure. Refer to the *Base SAS 9.1 Procedures Guide* for information on the RANK procedure, and see the chapters in this book for information on the PRINQUAL and TRANSREG procedures.

In addition, you can specify SCORES=RANK in the TABLES statement in the FREQ procedure. PROC FREQ then uses ranks to perform the analyses requested and generates nonparametric analyses.

For more discussion of using the rank transform, refer to Iman and Conover (1979), Conover and Iman (1981), Hora and Conover (1984), Iman, Hora, and Conover (1984), Hora and Iman (1988), and Iman (1988).

Kernel Density Estimation

The KDE procedure performs either univariate or bivariate kernel density estimation. Statistical *density estimation* involves approximating a hypothesized probability density function from observed data. *Kernel density estimation* is a nonparametric technique for density estimation in which a known density function (the kernel) is averaged across the observed data points to create a smooth approximation.

PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. PROC KDE outputs the kernel density estimate to a SAS data set, which you can then use with other procedures for plotting or analysis. PROC KDE also computes a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function.

For more information, see Chapter 36, "The KDE Procedure."

References

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Bross, I.D.J. (1958), “How to Use Ridit Analysis,” *Biometrics*, 14, 18–38.

Conover, W.J. (1999), *Practical Nonparametric Statistics*, Third Edition, New York: John Wiley & Sons, Inc.

Conover, W.J. and Iman, R.L. (1981), “Rank Transformations as a Bridge between Parametric and Nonparametric Statistics,” *The American Statistician*, 35, 124–129.

Gibbons, J.D. and Chakraborti, S. (1992), *Nonparametric Statistical Inference*, Third Edition, New York: Marcel Dekker, Inc.

Hajek, J. (1969), *A Course in Nonparametric Statistics*, San Francisco: Holden-Day.

Hettmansperger, T.P. (1984), *Statistical Inference Based on Ranks*, New York: John Wiley & Sons, Inc.

Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, Second Edition, New York: John Wiley & Sons, Inc.

Hora, S.C. and Conover, W.J. (1984), “The F Statistic in the Two-Way Layout with Rank-Score Transformed Data,” *Journal of the American Statistical Association*, 79, 668–673.

Hora, S.C. and Iman, R.L. (1988), “Asymptotic Relative Efficiencies of the Rank-Transformation Procedure in Randomized Complete Block Designs,” *Journal of the American Statistical Association*, 83, 462–470.

Iman, R.L. and Conover, W.J. (1979), “The Use of the Rank Transform in Regression,” *Technometrics*, 21, 499–509.

Iman, R.L., Hora, S.C., and Conover, W.J. (1984), “Comparison of Asymptotically Distribution-Free Procedures for the Analysis of Complete Blocks,” *Journal of the American Statistical Association*, 79, 674–685.

Iman, R.L. (1988), “The Analysis of Complete Blocks Using Methods Based on Ranks,” *Proceedings of the Thirteenth Annual SAS Users Group International Conference*, 13, 970–978.

Ipe, D. (1987), “Performing the Friedman Test and the Associated Multiple Comparison Test Using PROC GLM,” *Proceedings of the Twelfth Annual SAS Users Group International Conference*, 12, 1146–1148.

Lehmann, E.L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, San Francisco: Holden-Day.

Randles, R.H. and Wolfe, D.A. (1979), *Introduction to the Theory of Nonparametric Statistics*, New York: John Wiley & Sons, Inc.

Chapter 13

Introduction to Structural Equation Modeling

Chapter Contents

Chapter 13

Introduction to Structural Equations with Latent Variables

Overview

You can use the CALIS procedure for analysis of covariance structures, fitting systems of linear structural equations, and path analysis. These terms are more or less interchangeable, but they emphasize different aspects of the analysis. The analysis of covariance structures refers to the formulation of a model for the variances and covariances among a set of variables and the fitting of the model to an observed covariance matrix. In linear structural equations, the model is formulated as a system of equations relating several random variables with assumptions about the variances and covariances of the random variables. In path analysis, the model is formulated as a path diagram, in which arrows connecting variables represent (co)variances and regression coefficients. Path models and linear structural equation models can be converted to models of the covariance matrix and can, therefore, be fitted by the methods of covariance structure analysis. All of these methods allow the use of hypothetical latent variables or measurement errors in the models.

Loehlin (1987) provides an excellent introduction to latent variable models using path diagrams and structural equations. A more advanced treatment of structural equation models with latent variables is given by Bollen (1989). Fuller (1987) provides a highly technical statistical treatment of measurement-error models.

Comparison of the CALIS and SYSLIN Procedures

The SYSLIN procedure in the SAS/ETS product can also fit certain kinds of path models and linear structural equation models. PROC CALIS differs from PROC SYSLIN in that PROC CALIS allows more generality in the use of latent variables in the models. Latent variables are unobserved, hypothetical variables, as distinct from manifest variables, which are the observed data. PROC SYSLIN allows at most one latent variable, the error term, in each equation. PROC CALIS allows several latent variables to appear in an equation—in fact, all the variables in an equation can be latent as long as there are other equations that relate the latent variables to manifest variables.

Both the CALIS and SYSLIN procedures enable you to specify a model as a system of linear equations. When there are several equations, a given variable may be a dependent variable in one equation and an independent variable in other equations. Therefore, additional terminology is needed to describe unambiguously the roles of variables in the system. Variables with values that are determined jointly and simultaneously by the system of equations are called *endogenous variables*. Variables with values that are determined outside the system, that is, in a manner separate from the process described by the system of equations, are called exogenous variables. The purpose of the system of equations is to explain the variation of each endogenous variable in terms of exogenous variables or other endogenous variables or both. Refer to Loehlin (1987, p. 4) for further discussion of endogenous and exogenous variables. In the econometric literature, error and disturbance terms are usually distinguished from exogenous variables, but in systems with more than one latent variable in an equation, the distinction is not always clear.

In PROC SYSLIN, endogenous variables are identified by the ENDOGENOUS statement. When you specify structural equations in PROC CALIS, endogenous variables are assumed to be those that appear on the left-hand sides of the equations; a given variable may appear on the left-hand side of at most one equation.

PROC SYSLIN provides many methods of estimation, some of which are applicable only in special cases. For example, ordinary least-squares estimates are suitable in certain kinds of systems but may be statistically biased and inconsistent in other kinds. PROC CALIS provides three methods of estimation that can be used with most models. Both the CALIS and SYSLIN procedures can do maximum likelihood estimation, which PROC CALIS calls ML and PROC SYSLIN calls FIML. PROC SYSLIN can be much faster than PROC CALIS in those special cases for which it provides computationally efficient estimation methods. However, PROC CALIS has a variety of sophisticated algorithms for maximum likelihood estimation that may be much faster than FIML in PROC SYSLIN.

PROC CALIS can impose a wider variety of constraints on the parameters, including nonlinear constraints, than can PROC SYSLIN. For example, PROC CALIS can constrain error variances or covariances to equal specified constants, or it can constrain two error variances to have a specified ratio.

Model Specification

PROC CALIS provides several ways to specify a model. Structural equations can be transcribed directly in the LINEQS statement. A path diagram can be described in the RAM statement. You can specify a first-order factor model in the FACTOR and MATRIX statements. Higher-order factor models and other complicated models can be expressed in the COSAN and MATRIX statements. For most applications, the LINEQS and RAM statements are easiest to use; the choice between these two statements is a matter of personal preference.

You can save a model specification in an OUTRAM= data set, which can then be used with the INRAM= option to specify the model in a subsequent analysis.

Estimation Methods

The CALIS procedure provides five methods of estimation specified by the METHOD= option:

DWLS	diagonally weighted least squares
ULS	unweighted least squares
GLS	normal theory generalized least squares
ML	maximum likelihood for multivariate normal distributions
WLS	weighted least squares for arbitrary distributions

Each estimation method is based on finding parameter estimates that minimize a badness-of-fit function, which measures the difference between the observed sample covariance matrix and the predicted covariance matrix, given the model and the parameter estimates. See the section "Estimation Methods" on page 574 in Chapter 19, "The CALIS Procedure," for formulas, or refer to Loehlin (1987, pp. 54–62) and Bollen (1989, pp. 104–123) for further discussion.

The default is METHOD=ML, which is the most popular method for applications. The option METHOD=GLS usually produces very similar results to METHOD=ML. Asymptotically, ML and GLS are the same. Both methods assume a multivariate normal distribution in the population. The WLS method with default weight matrix, which is equivalent to the asymptotically distribution free (ADF) method, yields asymptotically normal estimates regardless of the distribution in the population. When the multivariate normal assumption is in doubt, especially if they have high kurtosis, you should seriously consider the WLS method. When a correlation matrix is analyzed, only the WLS may produce correct standard error estimates. However, in order to use the WLS method with the expected statistical properties, sample size must be large. Several thousands may be a minimum requirement.

The ULS and DWLS methods yield reasonable estimates under less restrictive assumptions. You can apply these methods to normal or nonnormal situations, or to covariance or correlation matrices. The drawback is that the statistical qualities of the estimates seem to be unknown. For this reason, PROC CALIS does not provide standard errors or test statistics with these two methods.

You cannot use METHOD=ML if the observed covariance matrix is singular. You could either remove variables involved in the linear dependencies or use less restrictive estimation methods like ULS. Specifying METHOD=GLS assumes that the predicted covariance matrix is nonsingular. If GLS fails because of a singular predicted covariance matrix, you need to examine whether the model specification leads to the singularity. If so, modify the model specification to eliminate the problem. If not, you probably need to use other estimation methods.

You should remove outliers and try to transform variables that are skewed or heavy-tailed. This applies to all estimation methods, since all the estimation methods depend on the sample covariance matrix, and the sample covariance matrix is a poor estimator for distributions with high kurtosis (Bollen 1989, pp. 415–418; Huber 1981; Hampel et al. 1986). PROC CALIS displays estimates of univariate and multivariate kurtosis (Bollen 1989, pp. 418–425) if you specify the KURTOSIS option in the PROC CALIS statement.

Statistical Inference

When you specify the ML, GLS, or WLS estimation with appropriate models, PROC CALIS can compute

- a chi-square goodness-of-fit test of the specified model versus the alternative that the data are from a population with unconstrained covariance matrix (Loehlin 1987, pp. 62–64; Bollen 1989, pp. 110, 115, 263–269)
- approximate standard errors of the parameter estimates (Bollen 1989, pp. 109, 114, 286), displayed with the STDERR option
- various modification indices, requested via the MODIFICATION or MOD option, that give the approximate change in the chi-square statistic that would result from removing constraints on the parameters or constraining additional parameters to zero (Bollen 1989, pp. 293–303)

If you have two models such that one model results from imposing constraints on the parameters of the other, you can test the constrained model against the more general model by fitting both models with PROC CALIS. If the constrained model is correct, the difference between the chi-square goodness-of-fit statistics for the two models has an approximate chi-square distribution with degrees of freedom equal to the difference between the degrees of freedom for the two models (Loehlin 1987, pp. 62–67; Bollen 1989, pp. 291–292).

All of the test statistics and standard errors computed under ML and GLS depend on the assumption of multivariate normality. Normality is a much more important requirement for data with random independent variables than it is for fixed independent variables. If the independent variables are random, distributions with high kurtosis tend to give liberal tests and excessively small standard errors, while low kurtosis tends to produce the opposite effects (Bollen 1989, pp. 266–267, 415–432).

All test statistics and standard errors computed by PROC CALIS are based on asymptotic theory and should not be trusted in small samples. There are no firm guidelines on how large a sample must be for the asymptotic theory to apply with reasonable accuracy. Some simulation studies have indicated that problems are likely to occur with sample sizes less than 100 (Loehlin 1987, pp. 60–61; Bollen 1989, pp. 267–268). Extrapolating from experience with multiple regression would suggest that the sample size should be at least 5 to 20 times the number of parameters to be estimated in order to get reliable and interpretable results. The WLS method may even require that the sample size be over several thousands.

The asymptotic theory requires that the parameter estimates be in the interior of the parameter space. If you do an analysis with inequality constraints and one or more constraints are active at the solution (for example, if you constrain a variance to be nonnegative and the estimate turns out to be zero), the chi-square test and standard errors may not provide good approximations to the actual sampling distributions.

For modeling correlation structures, the only theoretically correct method is the WLS method with the default ASYCOV=CORR option. For other methods, standard error estimates for modeling correlation structures may be inaccurate even for sample sizes as large as 400. The chi-square statistic is generally the same regardless of which matrix is analyzed, provided that the model involves no scale-dependent constraints. However, if the purpose is to obtain reasonable parameter estimates for the correlation structures only, then you may find other estimation methods useful as well.

If you fit a model to a correlation matrix and the model constrains one or more elements of the predicted matrix to equal 1.0, the degrees of freedom of the chi-square statistic must be reduced by the number of such constraints. PROC CALIS attempts to determine which diagonal elements of the predicted correlation matrix are constrained to a constant, but it may fail to detect such constraints in complicated models, particularly when programming statements are used. If this happens, you should add parameters to the model to release the constraints on the diagonal elements.

Goodness-of-fit Statistics

In addition to the chi-square test, there are many other statistics for assessing the goodness of fit of the predicted correlation or covariance matrix to the observed matrix.

Akaike's (1987) information criterion (AIC) and Schwarz's (1978) Bayesian criterion (SBC) are useful for comparing models with different numbers of parameters—the model with the smallest value of AIC or SBC is considered best. Based on both theoretical considerations and various simulation studies, SBC seems to work better, since AIC tends to select models with too many parameters when the sample size is large.

There are many descriptive measures of goodness of fit that are scaled to range approximately from zero to one: the goodness of fit index (GFI) and GFI adjusted for degrees of freedom (AGFI) (Jöreskog and Sörbom 1988), centrality (McDonald 1989), and the parsimonious fit index (James, Mulaik, and Brett 1982). Bentler and Bonett (1980) and Bollen (1986) have proposed measures for comparing the goodness of fit of one model with another in a descriptive rather than inferential sense.

The root mean squared error approximation (RMSEA) proposed by Steiger and Lind (1980) does not assume a true model being fitting to the data. It measures the discrepancy between the fitted model and the covariance matrix in the population. For samples, RMSEA and confidence intervals can be estimated. Statistical tests for determining whether the population RMSEA's fall below certain specified values are available (Browne and Cudeck 1993). In the same vein, Browne and Cudeck (1993) propose the expected cross validation index (ECVI) that measures how good a model is for predicting future sample covariances. Point estimate and confidence intervals for ECVI are also developed.

None of these measures of goodness of fit are related to the goodness of prediction of the structural equations. Goodness of fit is assessed by comparing the observed correlation or covariance matrix with the matrix computed from the model and parameter estimates. Goodness of prediction is assessed by comparing the actual values of the endogenous variables with their predicted values, usually in terms of root mean squared error or proportion of variance accounted for (R^2). For latent endogenous variables, root mean squared error and R^2 can be estimated from the fitted model.

Optimization Methods

PROC CALIS uses a variety of nonlinear optimization algorithms for computing parameter estimates. These algorithms are very complicated and do not always work. PROC CALIS will generally inform you when the computations fail, usually by displaying an error message about the iteration limit being exceeded. When this happens, you may be able to correct the problem simply by increasing the iteration limit (MAXITER= and MAXFUNC=). However, it is often more effective to change the optimization method (OMETHOD=) or initial values. For more details, see the section "Use of Optimization Techniques" on page 664 in Chapter 19, "The CALIS Procedure," and refer to Bollen (1989, pp. 254–256).

PROC CALIS may sometimes converge to a local optimum rather than the global optimum. To gain some protection against local optima, you can run the analysis several times with different initial estimates. The RANDOM= option in the PROC CALIS statement is useful for generating a variety of initial estimates.

Specifying Structural Equation Models

Consider fitting a linear equation to two observed variables, Y and X. Simple linear regression uses the model of a particular form, labeled for purposes of discussion, as Model Form A.

Model Form A

$$Y \quad = \quad \alpha + \beta X + E_Y$$

where α and β are coefficients to be estimated and E_Y is an error term. If the values of X are fixed, the values of E_Y are assumed to be independent and identically distributed realizations of a normally distributed random variable with mean zero and variance Var(E_Y). If X is a random variable, X and E_Y are assumed to have a bivariate normal distribution with zero correlation and variances Var(X) and Var(E_Y), respectively. Under either set of assumptions, the usual formulas hold for the estimates of the coefficients and their standard errors (see Chapter 2, "Introduction to Regression Procedures,").

In the REG or SYSLIN procedure, you would fit a simple linear regression model with a MODEL statement listing only the names of the manifest variables:

```
proc reg;
   model y=x;
run;
```

You can also fit this model with PROC CALIS, but you must explicitly specify the names of the parameters and the error terms (except for the intercept, which is assumed to be present in each equation). The linear equation is given in the LINEQS statement, and the error variance is specified in the STD statement.

```
proc calis cov;
   lineqs y=beta x + ex;
   std ex=vex;
run;
```

The parameters are the regression coefficient beta and the variance vex of the error term ex. You do not need to type an * between beta and x to indicate the multiplication of the variable by the coefficient.

The LINEQS statement uses the convention that the names of error terms begin with the letter E, disturbances (errors terms for latent variables) in equations begin with D, and other latent variables begin with F for “factor.” Names of variables in the input SAS data set can, of course, begin with any letter.

If you leave out the name of a coefficient, the value of the coefficient is assumed to be 1. If you leave out the name of a variance, the variance is assumed to be 0. So if you tried to write the model the same way you would in PROC REG, for example,

```
proc calis cov;
   lineqs y=x;
```

you would be fitting a model that says Y is equal to X plus an intercept, with no error.

The COV option is used because PROC CALIS, like PROC FACTOR, analyzes the correlation matrix by default, yielding standardized regression coefficients. The COV option causes the covariance matrix to be analyzed, producing raw regression coefficients. See Chapter 2, “Introduction to Regression Procedures,” for a discussion of the interpretation of raw and standardized regression coefficients.

Since the analysis of covariance structures is based on modeling the covariance matrix and the covariance matrix contains no information about means, PROC CALIS neglects the intercept parameter by default. To estimate the intercept, change the COV option to UCOV, which analyzes the uncorrected covariance matrix, and use the AUGMENT option, which adds a row and column for the intercept, called INTERCEPT, to the matrix being analyzed. The model can then be specified as

```
proc calis ucov augment;
   lineqs y=alpha intercept + beta x + ex;
   std ex=vex;
run;
```

In the LINEQS statement, intercept represents a variable with a constant value of 1; hence, the coefficient alpha is the intercept parameter.

Other commonly used options in the PROC CALIS statement include

- MODIFICATION to display model modification indices
- RESIDUAL to display residual correlations or covariances
- STDERR to display approximate standard errors
- TOTEFF to display total effects

For ordinary unconstrained regression models, there is no reason to use PROC CALIS instead of PROC REG. But suppose that the observed variables Y and X are contaminated by error, and you want to estimate the linear relationship between their true, error-free scores. The model can be written in several forms. A model of Form B is as follows.

Model Form B

$$\begin{aligned} Y &= \alpha + \beta F_X + E_Y \\ X &= F_X + E_X \\ \text{Cov}(F_X, E_X) &= \text{Cov}(F_X, E_Y) \;=\; \text{Cov}(E_X, E_Y) \;=\; 0 \end{aligned}$$

This model has two error terms, E_Y and E_X, as well as another latent variable F_X representing the true value corresponding to the manifest variable X. The true value corresponding to Y does not appear explicitly in this form of the model.

The assumption in Model Form B is that the error terms and the latent variable F_X are jointly uncorrelated is of critical importance. This assumption must be justified on substantive grounds such as the physical properties of the measurement process. If this assumption is violated, the estimators may be severely biased and inconsistent.

You can express Model Form B in PROC CALIS as follows:

```
proc calis cov;
   lineqs y=beta fx + ey,
          x=fx + ex;
   std fx=vfx,
       ey=vey,
       ex=vex;
run;
```

You must specify a variance for each of the latent variables in this model using the STD statement. You can specify either a name, in which case the variance is considered a parameter to be estimated, or a number, in which case the variance is constrained to equal that numeric value. In general, you must specify a variance for each latent exogenous variable in the model, including error and disturbance terms. The variance of a manifest exogenous variable is set equal to its sample variance by default. The variances of endogenous variables are predicted from the model and are not parameters. Covariances involving latent exogenous variables are assumed to be zero by default. Covariances between manifest exogenous variables are set equal to the sample covariances by default.

Fuller (1987, pp. 18–19) analyzes a data set from Voss (1969) involving corn yields (Y) and available soil nitrogen (X) for which there is a prior estimate of the measurement error for soil nitrogen Var(E_X) of 57. You can fit Model Form B with this constraint using the following SAS statements.

```
data corn(type=cov);
   input _type_ $ _name_ $ y x;
   datalines;
n    . 11       11
mean . 97.4545  70.6364
cov  y 87.6727  .
cov  x 104.8818 304.8545
;


proc calis data=corn cov stderr;
   lineqs y=beta fx + ey,
          x=fx + ex;
   std ex=57,
       fx=vfx,
       ey=vey;
run;
```

In the STD statement, the variance of ex is given as the constant value 57. PROC CALIS produces the following estimates.

```
                            The CALIS Procedure
          Covariance Structure Analysis: Maximum Likelihood Estimation

                 y       =   0.4232*fx       +  1.0000 ey
                 Std Err     0.1658 beta
                 t Value     2.5520
                 x       =   1.0000 fx       +  1.0000 ex


                       Variances of Exogenous Variables

                                                      Standard
           Variable Parameter      Estimate             Error    t Value

           fx       vfx           247.85450         136.33508       1.82
           ey       vey            43.29105          23.92488       1.81
           ex                      57.00000
```

Figure 13.1. Measurement Error Model for Corn Data

PROC CALIS also displays information about the initial estimates that can be useful if there are optimization problems. If there are no optimization problems, the initial estimates are usually not of interest; they are not reproduced in the examples in this chapter.

You can write an equivalent model (labeled here as Model Form C) using a latent variable F_Y to represent the true value corresponding to Y.

Model Form C

$$\begin{aligned}
Y &= F_Y + E_Y \\
X &= F_X + E_X \\
F_Y &= \alpha + \beta F_X \\
\text{Cov}(F_X, E_X) &= \text{Cov}(F_X, E_X) \ = \ \text{Cov}(E_X, E_Y) \ = \ 0
\end{aligned}$$

The first two of the three equations express the observed variables in terms of a true score plus error; these equations are called the measurement model. The third equation, expressing the relationship between the latent true-score variables, is called the structural or causal model. The decomposition of a model into a measurement model and a structural model (Keesling 1972; Wiley 1973; Jöreskog 1973) has been popularized by the program LISREL (Jöreskog and Sörbom 1988). The statements for fitting this model are

```
proc calis cov;
   lineqs y=fy + ey,
          x=fx + ex,
          fy=beta fx;
   std fx=vfx,
       ey=vey,
       ex=vex;
run;
```

You do not need to include the variance of F_Y in the STD statement because the variance of F_Y is determined by the structural model in terms of the variance of F_X, that is, Var(F_Y)=β^2 Var(F_X).

Correlations involving endogenous variables are derived from the model. For example, the structural equation in Model Form C implies that F_Y and F_X are correlated unless β is zero. In all of the models discussed so far, the latent exogenous variables are assumed to be jointly uncorrelated. For example, in Model Form C, E_Y, E_X, and F_X are assumed to be uncorrelated. If you want to specify a model in which E_Y and E_X, say, are correlated, you can use the COV statement to specify the numeric value of the covariance Cov(E_Y, E_X) between E_Y and E_X, or you can specify a name to make the covariance a parameter to be estimated. For example,

```
proc calis cov;
   lineqs y=fy + ey,
          x=fx + ex,
          fy=beta fx;
   std fy=vfy,
       fx=vfx,
       ey=vey,
       ex=vex;
   cov ey ex=ceyex;
run;
```

This COV statement specifies that the covariance between ey and ex is a parameter named ceyex. All covariances that are not listed in the COV statement and that are not determined by the model are assumed to be zero. If the model contained two or more manifest exogenous variables, their covariances would be set to the observed sample values by default.

Identification of Models

Unfortunately, if you try to fit models of Form B or Form C without additional constraints, you cannot obtain unique estimates of the parameters. These models have four parameters (one coefficient and three variances). The covariance matrix of the observed variables Y and X has only three elements that are free to vary, since Cov(Y,X)=Cov(X,Y). The covariance structure can, therefore, be expressed as three equations in four unknown parameters. Since there are fewer equations than unknowns, there are many different sets of values for the parameters that provide a solution for the equations. Such a model is said to be underidentified.

If the number of parameters equals the number of free elements in the covariance matrix, then there may exist a unique set of parameter estimates that exactly reproduce the observed covariance matrix. In this case, the model is said to be just identified or saturated.

If the number of parameters is less than the number of free elements in the covariance matrix, there may exist no set of parameter estimates that reproduces the observed covariance matrix. In this case, the model is said to be overidentified. Various statistical criteria, such as maximum likelihood, can be used to choose parameter estimates that approximately reproduce the observed covariance matrix. If you use ML, GLS, or WLS estimation, PROC CALIS can perform a statistical test of the goodness of fit of the model under the certain statistical assumptions.

If the model is just identified or overidentified, it is said to be identified. If you use ML, GLS, or WLS estimation for an identified model, PROC CALIS can compute approximate standard errors for the parameter estimates. For underidentified models, PROC CALIS obtains approximate standard errors by imposing additional constraints resulting from the use of a generalized inverse of the Hessian matrix.

You cannot guarantee that a model is identified simply by counting the parameters. For example, for any latent variable, you must specify a numeric value for the variance, or for some covariance involving the variable, or for a coefficient of the variable in at least one equation. Otherwise, the scale of the latent variable is indeterminate, and the model will be underidentified regardless of the number of parameters and the size of the covariance matrix. As another example, an exploratory factor analysis with two or more common factors is always underidentified because you can rotate the common factors without affecting the fit of the model.

PROC CALIS can usually detect an underidentified model by computing the approximate covariance matrix of the parameter estimates and checking whether any estimate is linearly related to other estimates (Bollen 1989, pp. 248–250), in which case PROC CALIS displays equations showing the linear relationships among the estimates. Another way to obtain empirical evidence regarding the identification of a model is to run the analysis several times with different initial estimates to see if the same final estimates are obtained.

Bollen (1989) provides detailed discussions of conditions for identification in a variety of models.

The following example is inspired by Fuller (1987, pp. 40–41). The hypothetical data are counts of two types of cells, cells forming rosettes and nucleated cells, in spleen samples. It is reasonable to assume that counts have a Poisson distribution; hence, the square roots of the counts should have a constant error variance of 0.25.

You can use PROC CALIS to fit a model of Form C to the square roots of the counts without constraints on the parameters, as displayed in following statements. The option OMETHOD=QUANEW is used in the PROC CALIS statement because in this case it produces more rapid convergence than the default optimization method.

```
data spleen;
   input rosette nucleate;
   sqrtrose=sqrt(rosette);
   sqrtnucl=sqrt(nucleate);
   datalines;
4 62
5 87
5 117
6 142
8 212
9 120
12 254
13 179
15 125
19 182
28 301
51 357
;
```

```
proc calis data=spleen cov omethod=quanew;
   lineqs sqrtrose=factrose + err_rose,
          sqrtnucl=factnucl + err_nucl,
          factrose=beta factnucl;
   std err_rose=v_rose,
       err_nucl=v_nucl,
       factnucl=v_factnu;
run;
```

This model is underidentified. PROC CALIS displays the following warning:

```
WARNING: Problem not identified: More parameters to estimate ( 4 )
         than given values in data matrix ( 3 ).
```

and diagnoses the indeterminacy as follows:

```
NOTE: Hessian matrix is not full rank. Not all parameters are identified.
      Some parameter estimates are linearly related to other parameter
      estimates as shown in the following equations:

v_nucl  =  -10.554977 - 0.036438 * beta + 1.00000 * v_factnu
           + 0.149564 * v_rose
```

The constraint that the error variances equal 0.25 can be imposed by modifying the STD statement:

```
proc calis data=spleen cov stderr;
   lineqs sqrtrose=factrose + err_rose,
          sqrtnucl=factnucl + err_nucl,
          factrose=beta factnucl;
   std err_rose=.25,
       err_nucl=.25,
       factnucl=v_factnu;
run;
```

The resulting parameter estimates are displayed in Figure 13.2.

```
                        The CALIS Procedure
        Covariance Structure Analysis: Maximum Likelihood Estimation

                      factrose =   0.4034*factnucl
                      Std Err      0.0508 beta
                      t Value      7.9439


                   Variances of Exogenous Variables

                                            Standard
         Variable Parameter      Estimate      Error    t Value

         factnucl v_factnu       10.45846    4.56608       2.29
         err_rose                 0.25000
         err_nucl                 0.25000
```

Figure 13.2. Spleen Data: Parameter Estimates for Overidentified Model

This model is overidentified and the chi-square goodness-of-fit test yields a *p*-value of 0.0219, as displayed in Figure 13.3.

```
        Covariance Structure Analysis: Maximum Likelihood Estimation

        Fit Function                                          0.4775
        Goodness of Fit Index (GFI)                           0.7274
        GFI Adjusted for Degrees of Freedom (AGFI)            0.1821
        Root Mean Square Residual (RMR)                       0.1785
        Parsimonious GFI (Mulaik, 1989)                       0.7274
        Chi-Square                                            5.2522
        Chi-Square DF                                              1
        Pr > Chi-Square                                       0.0219
        Independence Model Chi-Square                         13.273
        Independence Model Chi-Square DF                           1
        RMSEA Estimate                                        0.6217
        RMSEA 90% Lower Confidence Limit                      0.1899
        RMSEA 90% Upper Confidence Limit                      1.1869
        ECVI Estimate                                         0.9775
        ECVI 90% Lower Confidence Limit                            .
        ECVI 90% Upper Confidence Limit                       2.2444
        Probability of Close Fit                              0.0237
        Bentler's Comparative Fit Index                       0.6535
        Normal Theory Reweighted LS Chi-Square                9.5588
        Akaike's Information Criterion                        3.2522
        Bozdogan's (1987) CAIC                                1.7673
        Schwarz's Bayesian Criterion                          2.7673
        McDonald's (1989) Centrality                          0.8376
        Bentler & Bonett's (1980) Non-normed Index            0.6535
        Bentler & Bonett's (1980) NFI                         0.6043
        James, Mulaik, & Brett (1982) Parsimonious NFI        0.6043
        Z-Test of Wilson & Hilferty (1931)                    2.0375
        Bollen (1986) Normed Index Rho1                       0.6043
        Bollen (1988) Non-normed Index Delta2                 0.6535
        Hoelter's (1983) Critical N                               10
```

Figure 13.3. Spleen Data: Fit Statistics for Overidentified Model

The sample size is so small that the *p*-value should not be taken to be accurate, but to get a small *p*-value with such a small sample indicates it is possible that the model is seriously deficient. The deficiency could be due to any of the following:

- The error variances are not both equal to 0.25.
- The error terms are correlated with each other or with the true scores.
- The observations are not independent.
- There is a disturbance in the linear relation between factrose and factnucl.
- The relation between factrose and factnucl is not linear.
- The actual distributions are not adequately approximated by the multivariate normal distribution.

A simple and plausible modification to the model is to add a “disturbance term” or “error in the equation” to the structural model, as follows.

```
proc calis data=spleen cov stderr;
   lineqs sqrtrose=factrose + err_rose,
          sqrtnucl=factnucl + err_nucl,
          factrose=beta factnucl + disturb;
   std err_rose=.25,
       err_nucl=.25,
       factnucl=v_factnu,
       disturb=v_dist;
run;
```

The following parameter estimates are produced.

```
                         The CALIS Procedure
        Covariance Structure Analysis: Maximum Likelihood Estimation

             factrose =   0.3907*factnucl +  1.0000 disturb
             Std Err      0.0771 beta
             t Value      5.0692


                   Variances of Exogenous Variables

                                                  Standard
         Variable Parameter      Estimate            Error    t Value

         factnucl v_factnu       10.50458          4.58577       2.29
         err_rose                 0.25000
         err_nucl                 0.25000
         disturb  v_dist          0.38153          0.28556       1.34
```

Figure 13.4. Spleen Data: Parameter Estimated for Just Identified Model

This model is just identified, so there are no degrees of freedom for the chi-square goodness-of-fit test.

Path Diagrams and the RAM Model

Complicated models are often easier to understand when they are expressed as path diagrams. One advantage of path diagrams over equations is that variances and covariances can be shown directly in the path diagram. Loehlin (1987) provides a detailed discussion of path diagrams.

It is customary to write the names of manifest variables in rectangles and names of latent variables in ovals. The coefficients in each equation are indicated by drawing arrows from the independent variables to the dependent variable. Covariances between exogenous variables are drawn as two-headed arrows. The variance of an exogenous variable can be displayed as a two-headed arrow with both heads pointing to the exogenous variable, since the variance of a variable is the covariance of the variable with itself. Here is a path diagram for the spleen data, explicitly showing all latent variables and variances of exogenous variables.

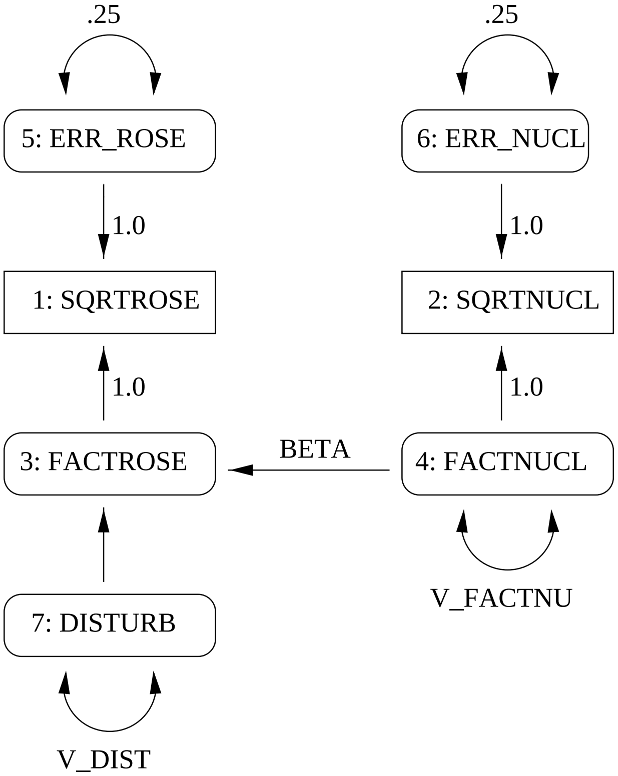


Figure 13.5. Path Diagram: Spleen

There is an easier way to draw the path diagram based on McArdle's reticular action model (RAM) (McArdle and McDonald 1984). McArdle uses the convention that a two-headed arrow that points to an endogenous variable actually refers to the error or disturbance term associated with that variable. A two-headed arrow with both heads pointing to the same endogenous variable represents the error or disturbance variance for the equation that determines the endogenous variable; there is no need to draw a separate oval for the error or disturbance term. Similarly, a two-headed arrow connecting two endogenous variables represents the covariance between the error of disturbance terms associated with the endogenous variables. The RAM conventions allow the previous path diagram to be simplified, as follows.

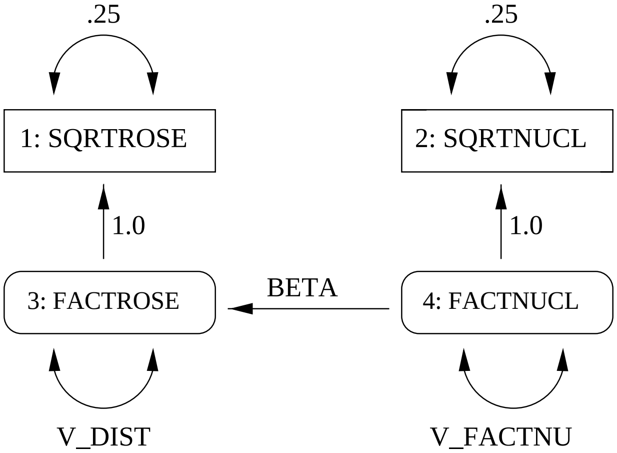


Figure 13.6. Path Diagram: Spleen

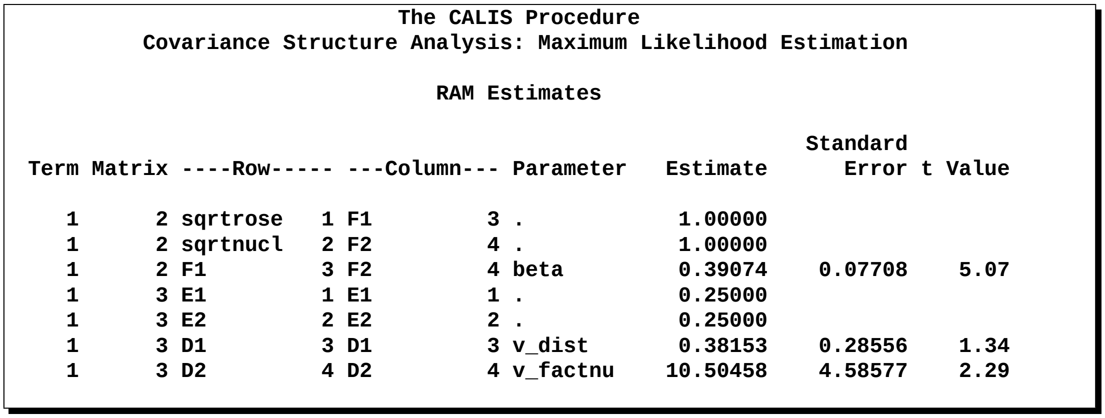
The RAM statement in PROC CALIS provides a simple way to transcribe a path diagram based on the reticular action model. Assign the integers 1, 2, 3,. . . to the variables in the order in which they appear in the SAS data set or in the VAR statement, if you use one. Assign subsequent consecutive integers to the latent variables displayed explicitly in the path diagram (excluding the error and disturbance terms implied by two-headed arrows) in any order. Each arrow in the path diagram can then be identified by two numbers indicating the variables connected by the path. The RAM statement consists of a list of descriptions of all the arrows in the path diagram. The descriptions are separated by commas. Each arrow description consists of three or four numbers and, optionally, a name in the following order:

1. The number of heads the arrow has.
2. The number of the variable the arrow points to, or either variable if the arrow is two-headed.
3. The number of the variable the arrow comes from, or the other variable if the arrow is two-headed.
4. The value of the coefficient or (co)variance that the arrow represents.
5. A name if the arrow represents a parameter to be estimated, in which case the previous number is taken to be the initial estimate of the parameter. Omit the name if the arrow represents a constant. If you specify a name, the fourth number may be omitted.

The model for the spleen data can be specified with the RAM statement, as follows:

```
       /* 1 sqrtrose  */
       /* 2 sqrtnucl  */
       /* 3 factrose  */
       /* 4 factnucl  */
   proc calis data=spleen cov stderr method=ml outram=splram1;
      var sqrtrose sqrtnucl;
      ram 1 1 3 1,         /* sqrtrose <- factrose */
          1 2 4 1,         /* sqrtnucl <- factnucl */
          1 3 4 beta,      /* factrose <- factnucl */
          2 1 1 .25,       /* error variance for sqrtrose */
          2 2 2 .25,       /* error variance for sqrtnucl */
          2 3 3 v_dist,    /* disturbance variance for factrose */
          2 4 4 v_factnu;  /* variance of factnucl */
   run;
```

The resulting output in RAM form is displayed in Figure 13.7.



```
                         The CALIS Procedure
        Covariance Structure Analysis: Maximum Likelihood Estimation

                            RAM Estimates

                                                             Standard
Term Matrix ----Row----- ---Column--- Parameter   Estimate      Error t Value

   1      2 sqrtrose   1 F1         3 .            1.00000
   1      2 sqrtnucl   2 F2         4 .            1.00000
   1      2 F1         3 F2         4 beta         0.39074    0.07708    5.07
   1      3 E1         1 E1         1 .            0.25000
   1      3 E2         2 E2         2 .            0.25000
   1      3 D1         3 D1         3 v_dist       0.38153    0.28556    1.34
   1      3 D2         4 D2         4 v_factnu    10.50458    4.58577    2.29
```

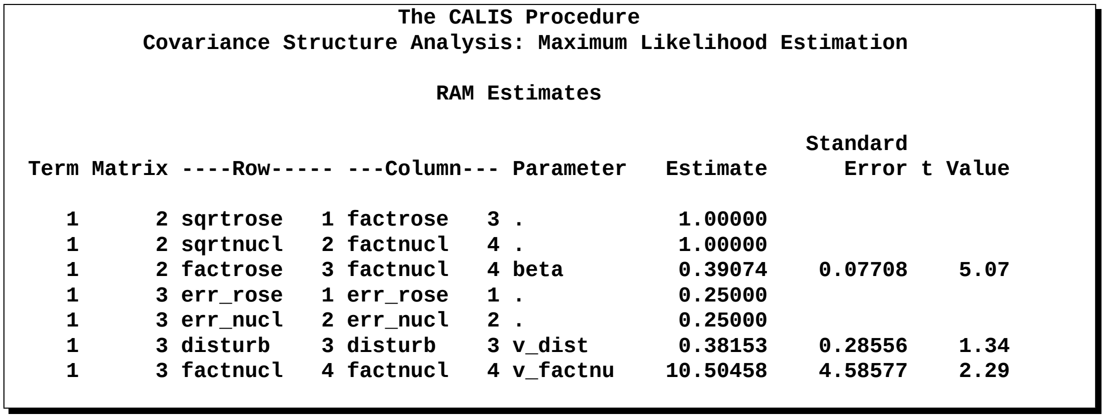
Figure 13.7. Spleen Data: RAM Model

You can request an output data set containing the model specification by using the OUTRAM= option in the PROC CALIS statement. Names for the latent variables can be specified in a VNAMES statement.

```
proc calis data=spleen cov stderr method=ml outram=splram1;
   var sqrtrose sqrtnucl;
   vnames 1 factrose factnucl,
          2 err_rose err_nucl disturb factnucl;
   ram 1 1 3 1,         /* sqrtrose <- factrose */
       1 2 4 1,         /* sqrtnucl <- factnucl */
       1 3 4 beta,      /* factrose <- factnucl */
       2 1 1 .25,       /* error variance for sqrtrose */
       2 2 2 .25,       /* error variance for sqrtnucl */
       2 3 3 v_dist,    /* disturbance variance for factrose */
       2 4 4 v_factnu;  /* variance of factnucl */
run;
```

```
proc print;
run;
```

The RAM output is displayed in Figure 13.8.



```
                          The CALIS Procedure
         Covariance Structure Analysis: Maximum Likelihood Estimation

                              RAM Estimates

                                                                Standard
Term Matrix ----Row----- ---Column--- Parameter   Estimate      Error t Value

   1      2 sqrtrose   1 factrose   3 .            1.00000
   1      2 sqrtnucl   2 factnucl   4 .            1.00000
   1      2 factrose   3 factnucl   4 beta         0.39074    0.07708    5.07
   1      3 err_rose   1 err_rose   1 .            0.25000
   1      3 err_nucl   2 err_nucl   2 .            0.25000
   1      3 disturb    3 disturb    3 v_dist       0.38153    0.28556    1.34
   1      3 factnucl   4 factnucl   4 v_factnu    10.50458    4.58577    2.29
```

Figure 13.8. Spleen Data: RAM Model with Names for Latent Variables

The OUTRAM= data set contains the RAM model as you specified it in the RAM statement, but it contains the final parameter estimates and standard errors instead of the initial values.

```
Obs    _TYPE_     _NAME_      _MATNR_    _ROW_    _COL_    _ESTIM_    _STDERR_

  1    MODEL      _IDE_          1         2        4       1.0000     0.00000
  2    MODEL      _A_            2         4        4       6.0000     2.00000
  3    MODEL      _P_            3         4        4       3.0000     0.00000
  4    VARNAME    sqrtrose       2         .        1        .          .
  5    VARNAME    sqrtnucl       2         .        2        .          .
  6    VARNAME    factrose       2         .        3        .          .
  7    VARNAME    factnucl       2         .        4        .          .
  8    VARNAME    err_rose       3         .        1        .          .
  9    VARNAME    err_nucl       3         .        2        .          .
 10    VARNAME    disturb        3         .        3        .          .
 11    VARNAME    factnucl       3         .        4        .          .
 12    METHOD     ML             .         .        .        .          .
 13    STAT       N              .         .        .      12.0000      .
 14    STAT       FIT            .         .        .       0.0000      .
 15    STAT       GFI            .         .        .       1.0000      .
 16    STAT       AGFI           .         .        .        .          .
 17    STAT       RMR            .         .        .       0.0000      .
 18    STAT       PGFI           .         .        .       0.0000      .
 19    STAT       NPARM          .         .        .       3.0000      .
 20    STAT       DF             .         .        .       0.0000      .
 21    STAT       N_ACT          .         .        .       0.0000      .
 22    STAT       CHISQUAR       .         .        .       0.0000      .
 23    STAT       P_CHISQ        .         .        .       0.0000      .
 24    STAT       CHISQNUL       .         .        .      13.2732      .
 25    STAT       RMSEAEST       .         .        .       0.0000      .
 26    STAT       RMSEALOB       .         .        .        .          .
 27    STAT       RMSEAUPB       .         .        .        .          .
 28    STAT       P_CLOSFT       .         .        .        .          .
 29    STAT       ECVI_EST       .         .        .       0.7500      .
 30    STAT       ECVI_LOB       .         .        .        .          .
 31    STAT       ECVI_UPB       .         .        .        .          .
 32    STAT       COMPFITI       .         .        .       1.0000      .
 33    STAT       ADJCHISQ       .         .        .        .          .
 34    STAT       P_ACHISQ       .         .        .        .          .
 35    STAT       RLSCHISQ       .         .        .       0.0000      .
 36    STAT       AIC            .         .        .       0.0000      .
 37    STAT       CAIC           .         .        .       0.0000      .
 38    STAT       SBC            .         .        .       0.0000      .
 39    STAT       CENTRALI       .         .        .       1.0000      .
 40    STAT       BB_NONOR       .         .        .        .          .
 41    STAT       BB_NORMD       .         .        .       1.0000      .
 42    STAT       PARSIMON       .         .        .       0.0000      .
 43    STAT       ZTESTWH        .         .        .        .          .
 44    STAT       BOL_RHO1       .         .        .        .          .
 45    STAT       BOL_DEL2       .         .        .       1.0000      .
 46    STAT       CNHOELT        .         .        .        .          .
 47    ESTIM                     2         1        3       1.0000     0.00000
 48    ESTIM                     2         2        4       1.0000     0.00000
 49    ESTIM      beta           2         3        4       0.3907     0.07708
 50    ESTIM                     3         1        1       0.2500     0.00000
 51    ESTIM                     3         2        2       0.2500     0.00000
 52    ESTIM      v_dist         3         3        3       0.3815     0.28556
 53    ESTIM      v_factnu       3         4        4      10.5046     4.58577
```

Figure 13.9. Spleen Data: OUTRAM= Data Set with Final Parameter Estimates

This data set can be used as input to another run of PROC CALIS with the INRAM= option in the PROC CALIS statement. For example, if the iteration limit is exceeded, you can use the RAM data set to start a new run that begins with the final estimates from the last run. Or you can change the data set to add or remove constraints or modify the model in various other ways. The easiest way to change a RAM data set is to use the FSEDIT procedure, but you can also use a DATA step. For example, you could set the variance of the disturbance term to zero, effectively removing the disturbance from the equation, by removing the parameter name v_dist in the _NAME_ variable and setting the value of the estimate to zero in the _ESTIM_ variable:

```
data splram2(type=ram);
   set splram1;
   if _name_='v_dist' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;

proc calis data=spleen inram=splram2 cov stderr;
run;
```

The resulting RAM output is displayed in Figure 13.10.

```
                            The CALIS Procedure
          Covariance Structure Analysis: Maximum Likelihood Estimation

                               RAM Estimates

                                                            Standard
Term Matrix ----Row----- ---Column--- Parameter   Estimate     Error t Value

   1      2 sqrtrose   1 factrose   3 .            1.00000
   1      2 sqrtnucl   2 factnucl   4 .            1.00000
   1      2 factrose   3 factnucl   4 beta         0.40340   0.05078    7.94
   1      3 err_rose   1 err_rose   1 .            0.25000
   1      3 err_nucl   2 err_nucl   2 .            0.25000
   1      3 disturb    3 disturb    3 .                  0
   1      3 factnucl   4 factnucl   4 v_factnu    10.45846   4.56608    2.29
```

Figure 13.10. Spleen Data: RAM Model with INRAM= Data Set

Some Measurement Models

Psychometric test theory involves many kinds of models relating scores on psychological and educational tests to latent variables representing intelligence or various underlying abilities. The following example uses data on four vocabulary tests from Lord (1957). Tests W and X have 15 items each and are administered with very liberal time limits. Tests Y and Z have 75 items and are administered under time pressure. The covariance matrix is read by the following DATA step:

```
data lord(type=cov);
   input _type_ $ _name_ $ w x y z;
   datalines;
n   . 649       .       .       .
cov w 86.3979   .       .       .
cov x 57.7751 86.2632   .       .
cov y 56.8651 59.3177 97.2850   .
cov z 58.8986 59.6683 73.8201 97.8192
;
```

The psychometric model of interest states that W and X are determined by a single common factor F_{WX}, and Y and Z are determined by a single common factor F_{YZ}. The two common factors are expected to have a positive correlation, and it is desired to estimate this correlation. It is convenient to assume that the common factors have unit variance, so their correlation will be equal to their covariance. The error terms for all the manifest variables are assumed to be uncorrelated with each other and with the common factors. The model (labeled here as Model Form D) is as follows.

Model Form D

$$\begin{aligned}
W &= \beta_W F_{WX} + E_W \\
X &= \beta_X F_{WX} + E_X \\
Y &= \beta_Y F_{YZ} + E_Y \\
Z &= \beta_Z F_{YZ} + E_Z \\
\text{Var}(F_{WX}) &= \text{Var}(F_{YZ}) = 1 \\
\text{Cov}(F_{WX}, F_{YZ}) &= \rho \\
\text{Cov}(E_W, E_X) &= \text{Cov}(E_W, E_Y) = \text{Cov}(E_W, E_Z) = \text{Cov}(E_X, E_Y) \\
&= \text{Cov}(E_X, E_Z) = \text{Cov}(E_Y, E_Z) = \text{Cov}(E_W, F_{WX}) \\
&= \text{Cov}(E_W, F_{YZ}) = \text{Cov}(E_X, F_{WX}) = \text{Cov}(E_X, F_{YZ}) \\
&= \text{Cov}(E_Y, F_{WX}) = \text{Cov}(E_Y, F_{YZ}) = \text{Cov}(E_Z, F_{WX}) \\
&= \text{Cov}(E_Z, F_{YZ}) = 0
\end{aligned}$$

The corresponding path diagram is as follows.

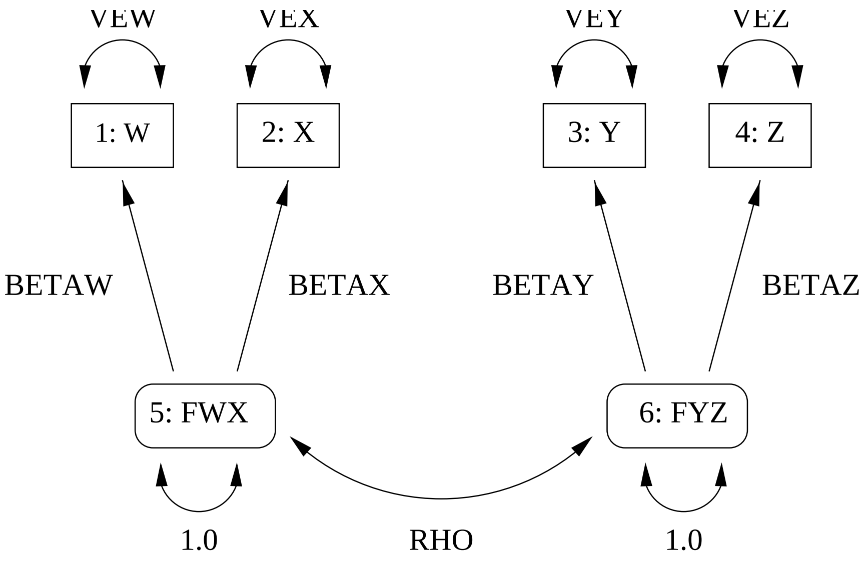
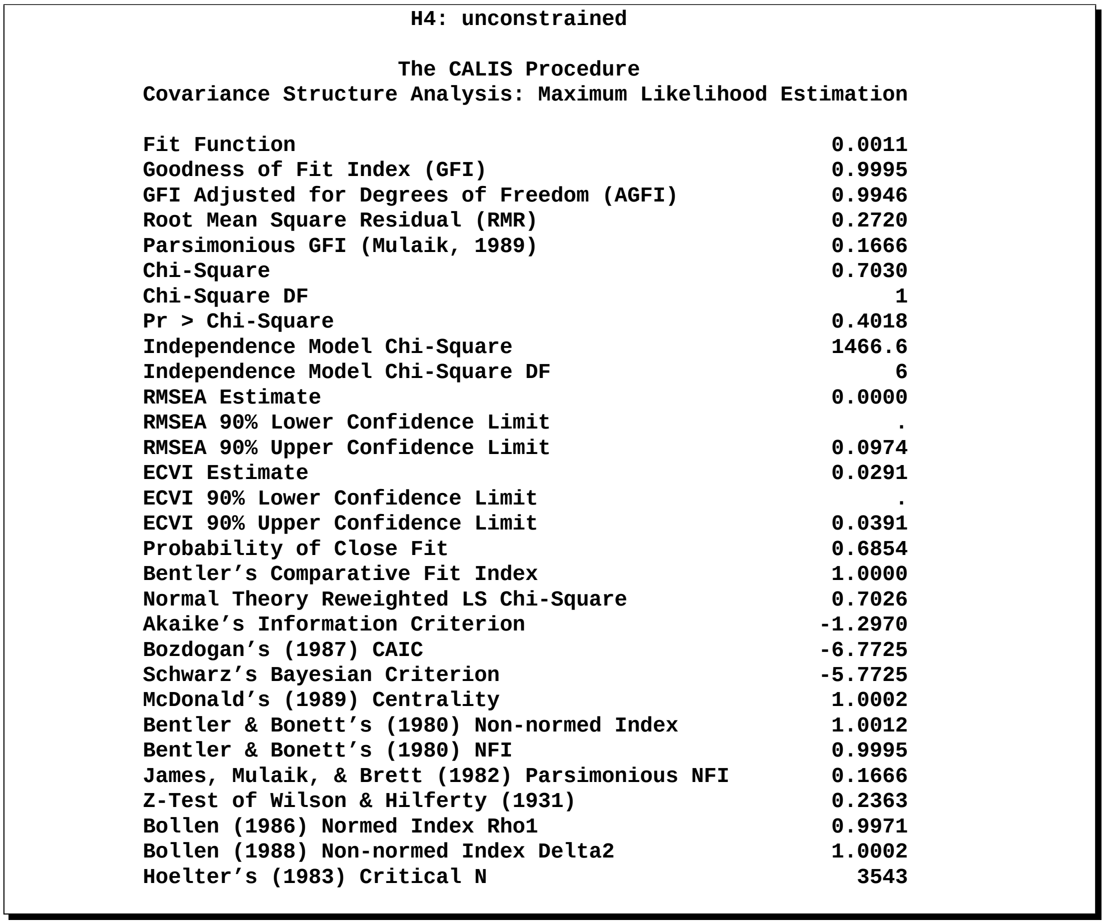


Figure 13.11. Path Diagram: Lord

This path diagram can be converted to a RAM model as follows:

```
   /* 1=w 2=x 3=y 4=z 5=fwx 6=fyz */
title 'H4: unconstrained';
proc calis data=lord cov;
   ram 1 1 5 betaw,
       1 2 5 betax,
       1 3 6 betay,
       1 4 6 betaz,
       2 1 1 vew,
       2 2 2 vex,
       2 3 3 vey,
       2 4 4 vez,
       2 5 5 1,
       2 6 6 1,
       2 5 6 rho;
run;
```

Here are the major results.



```
                     H4: unconstrained

                    The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function                                          0.0011
Goodness of Fit Index (GFI)                           0.9995
GFI Adjusted for Degrees of Freedom (AGFI)            0.9946
Root Mean Square Residual (RMR)                       0.2720
Parsimonious GFI (Mulaik, 1989)                       0.1666
Chi-Square                                            0.7030
Chi-Square DF                                              1
Pr > Chi-Square                                       0.4018
Independence Model Chi-Square                         1466.6
Independence Model Chi-Square DF                           6
RMSEA Estimate                                        0.0000
RMSEA 90% Lower Confidence Limit                           .
RMSEA 90% Upper Confidence Limit                      0.0974
ECVI Estimate                                         0.0291
ECVI 90% Lower Confidence Limit                            .
ECVI 90% Upper Confidence Limit                       0.0391
Probability of Close Fit                              0.6854
Bentler’s Comparative Fit Index                       1.0000
Normal Theory Reweighted LS Chi-Square                0.7026
Akaike’s Information Criterion                       -1.2970
Bozdogan’s (1987) CAIC                               -6.7725
Schwarz’s Bayesian Criterion                         -5.7725
McDonald’s (1989) Centrality                          1.0002
Bentler & Bonett’s (1980) Non-normed Index            1.0012
Bentler & Bonett’s (1980) NFI                         0.9995
James, Mulaik, & Brett (1982) Parsimonious NFI        0.1666
Z-Test of Wilson & Hilferty (1931)                    0.2363
Bollen (1986) Normed Index Rho1                       0.9971
Bollen (1988) Non-normed Index Delta2                 1.0002
Hoelter’s (1983) Critical N                             3543
```

Figure 13.12. Lord Data: Major Results for RAM Model, Hypothesis H4

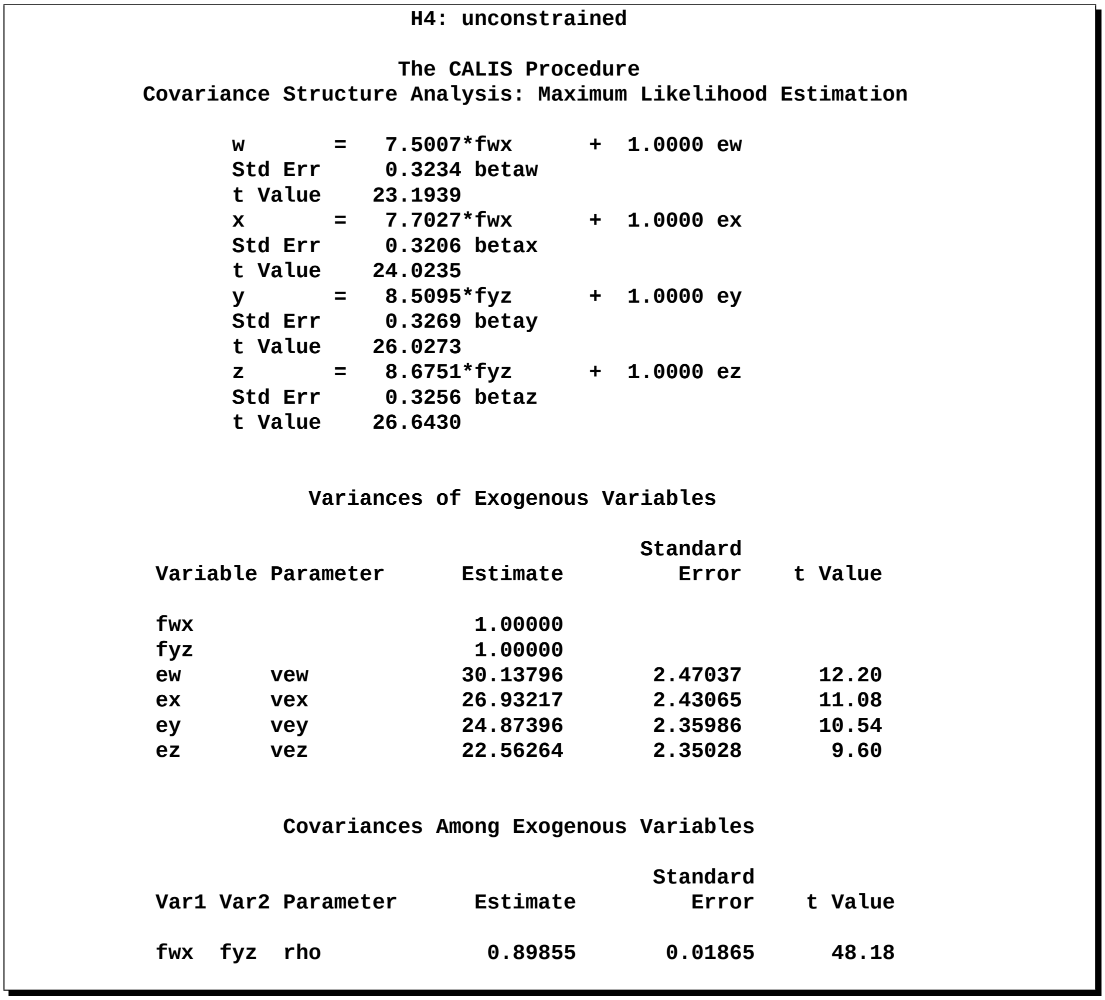
```
                                H4: unconstrained

          Covariance Structure Analysis: Maximum Likelihood Estimation

                                  RAM Estimates

                                                                  Standard
Term  Matrix  --Row--  -Column-  Parameter    Estimate       Error  t Value

   1       2  w     1  F1     5  betaw         7.50066     0.32339    23.19
   1       2  x     2  F1     5  betax         7.70266     0.32063    24.02
   1       2  y     3  F2     6  betay         8.50947     0.32694    26.03
   1       2  z     4  F2     6  betaz         8.67505     0.32560    26.64
   1       3  E1    1  E1     1  vew          30.13796     2.47037    12.20
   1       3  E2    2  E2     2  vex          26.93217     2.43065    11.08
   1       3  E3    3  E3     3  vey          24.87396     2.35986    10.54
   1       3  E4    4  E4     4  vez          22.56264     2.35028     9.60
   1       3  D1    5  D1     5  .             1.00000
   1       3  D2    6  D1     5  rho           0.89855     0.01865    48.18
   1       3  D2    6  D2     6  .             1.00000
```

Figure 13.12. (continued)

The same analysis can be performed with the LINEQS statement. Subsequent analyses are illustrated with the LINEQS statement rather than the RAM statement because it is slightly easier to understand the constraints as written in the LINEQS statement without constantly referring to the path diagram. The LINEQS and RAM statements may yield slightly different results due to the inexactness of the numerical optimization; the discrepancies can be reduced by specifying a more stringent convergence criterion such as GCONV=1E-4 or GCONV=1E-6. It is convenient to create an OUTRAM= data set for use in fitting other models with additional constraints.

```
title 'H4: unconstrained';
proc calis data=lord cov outram=ram4;
   lineqs w=betaw fwx + ew,
          x=betax fwx + ex,
          y=betay fyz + ey,
          z=betaz fyz + ez;
   std fwx fyz=1,
       ew ex ey ez=vew vex vey vez;
   cov fwx fyz=rho;
run;
```

The LINEQS displayed output is as follows.



```
                              H4: unconstrained

                             The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

                  w       =   7.5007*fwx      +  1.0000 ew
                  Std Err     0.3234 betaw
                  t Value    23.1939
                  x       =   7.7027*fwx      +  1.0000 ex
                  Std Err     0.3206 betax
                  t Value    24.0235
                  y       =   8.5095*fyz      +  1.0000 ey
                  Std Err     0.3269 betay
                  t Value    26.0273
                  z       =   8.6751*fyz      +  1.0000 ez
                  Std Err     0.3256 betaz
                  t Value    26.6430


                          Variances of Exogenous Variables

                                                    Standard
            Variable Parameter      Estimate           Error    t Value

            fwx                      1.00000
            fyz                      1.00000
            ew       vew            30.13796         2.47037      12.20
            ex       vex            26.93217         2.43065      11.08
            ey       vey            24.87396         2.35986      10.54
            ez       vez            22.56264         2.35028       9.60


                        Covariances Among Exogenous Variables

                                                     Standard
            Var1 Var2 Parameter      Estimate           Error    t Value

            fwx  fyz  rho             0.89855         0.01865      48.18
```

Figure 13.13. Lord Data: Using LINEQS Statement for RAM Model, Hypothesis H4

In an analysis of these data by Jöreskog and Sörbom (1979, pp. 54–56; Loehlin 1987, pp. 84–87), four hypotheses are considered:

H_1: $\rho = 1$,
$\beta_W = \beta_X$, Var$(E_W) = $ Var(E_X),
$\beta_Y = \beta_Z$, Var$(E_Y) = $ Var(E_Z)

H_2: same as H_1: except ρ is unconstrained

H_3: $\rho = 1$

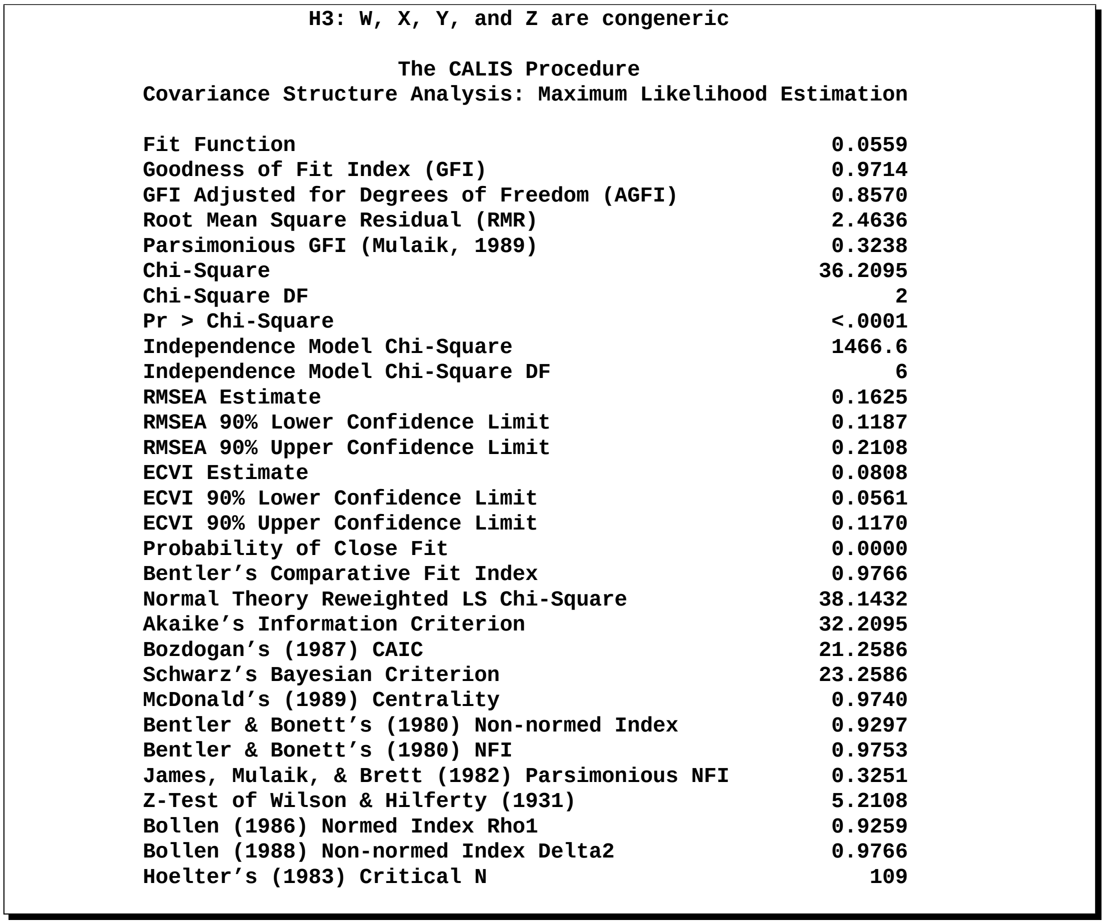
H_4: Model Form D without any additional constraints

The hypothesis H_3 says that there is really just one common factor instead of two; in the terminology of test theory, W, X, Y, and Z are said to be congeneric. The hypothesis H_2 says that W and X have the same true-scores and have equal error variance; such tests are said to be parallel. The hypothesis H_2 also requires Y and Z to be parallel. The hypothesis H_1 says that W and X are parallel tests, Y and Z are parallel tests, and all four tests are congeneric.

It is most convenient to fit the models in the opposite order from that in which they are numbered. The previous analysis fit the model for H_4 and created an OUTRAM= data set called ram4. The hypothesis H_3 can be fitted directly or by modifying the ram4 data set. Since H_3 differs from H_4 only in that ρ is constrained to equal 1, the ram4 data set can be modified by finding the observation for which _NAME_='rho' and changing the variable _NAME_ to a blank value (meaning that the observation represents a constant rather than a parameter to be fitted) and setting the variable _ESTIM_ to the value 1. Both of the following analyses produce the same results:

```
title 'H3: W, X, Y, and Z are congeneric';
proc calis data=lord cov;
   lineqs w=betaw f + ew,
          x=betax f + ex,
          y=betay f + ey,
          z=betaz f + ez;
   std f=1,
       ew ex ey ez=vew vex vey vez;
run;

data ram3(type=ram);
   set ram4;
   if _name_='rho' then
      do;
         _name_=' ';
         _estim_=1;
      end;
run;


proc calis data=lord inram=ram3 cov;
run;
```

The resulting output from either of these analyses is displayed in Figure 13.14.



```
                       H3: W, X, Y, and Z are congeneric

                              The CALIS Procedure
          Covariance Structure Analysis: Maximum Likelihood Estimation

          Fit Function                                              0.0559
          Goodness of Fit Index (GFI)                               0.9714
          GFI Adjusted for Degrees of Freedom (AGFI)                0.8570
          Root Mean Square Residual (RMR)                           2.4636
          Parsimonious GFI (Mulaik, 1989)                           0.3238
          Chi-Square                                               36.2095
          Chi-Square DF                                                  2
          Pr > Chi-Square                                           <.0001
          Independence Model Chi-Square                             1466.6
          Independence Model Chi-Square DF                               6
          RMSEA Estimate                                            0.1625
          RMSEA 90% Lower Confidence Limit                          0.1187
          RMSEA 90% Upper Confidence Limit                          0.2108
          ECVI Estimate                                             0.0808
          ECVI 90% Lower Confidence Limit                           0.0561
          ECVI 90% Upper Confidence Limit                           0.1170
          Probability of Close Fit                                  0.0000
          Bentler’s Comparative Fit Index                           0.9766
          Normal Theory Reweighted LS Chi-Square                   38.1432
          Akaike’s Information Criterion                           32.2095
          Bozdogan’s (1987) CAIC                                   21.2586
          Schwarz’s Bayesian Criterion                             23.2586
          McDonald’s (1989) Centrality                              0.9740
          Bentler & Bonett’s (1980) Non-normed Index                0.9297
          Bentler & Bonett’s (1980) NFI                             0.9753
          James, Mulaik, & Brett (1982) Parsimonious NFI            0.3251
          Z-Test of Wilson & Hilferty (1931)                        5.2108
          Bollen (1986) Normed Index Rho1                           0.9259
          Bollen (1988) Non-normed Index Delta2                     0.9766
          Hoelter’s (1983) Critical N                                  109
```

Figure 13.14. Lord Data: Major Results for Hypothesis H3

```
                    H3: W, X, Y, and Z are congeneric

          Covariance Structure Analysis: Maximum Likelihood Estimation

                 w       =   7.1047*fwx      +  1.0000 ew
                 Std Err     0.3218 betaw
                 t Value    22.0802
                 x       =   7.2691*fwx      +  1.0000 ex
                 Std Err     0.3183 betax
                 t Value    22.8397
                 y       =   8.3735*fyz      +  1.0000 ey
                 Std Err     0.3254 betay
                 t Value    25.7316
                 z       =   8.5106*fyz      +  1.0000 ez
                 Std Err     0.3241 betaz
                 t Value    26.2598


                       Variances of Exogenous Variables

                                                          Standard
           Variable Parameter       Estimate                 Error    t Value

           fwx                       1.00000
           fyz                       1.00000
           ew       vew             35.92087               2.41466      14.88
           ex       vex             33.42397               2.31038      14.47
           ey       vey             27.16980               2.24619      12.10
           ez       vez             25.38948               2.20839      11.50
```

Figure 13.14. (continued)

The hypothesis H_2 requires that several pairs of parameters be constrained to have equal estimates. With PROC CALIS, you can impose this constraint by giving the same name to parameters that are constrained to be equal. This can be done directly in the LINEQS and STD statements or by using PROC FSEDIT or a DATA step to change the values in the ram4 data set:

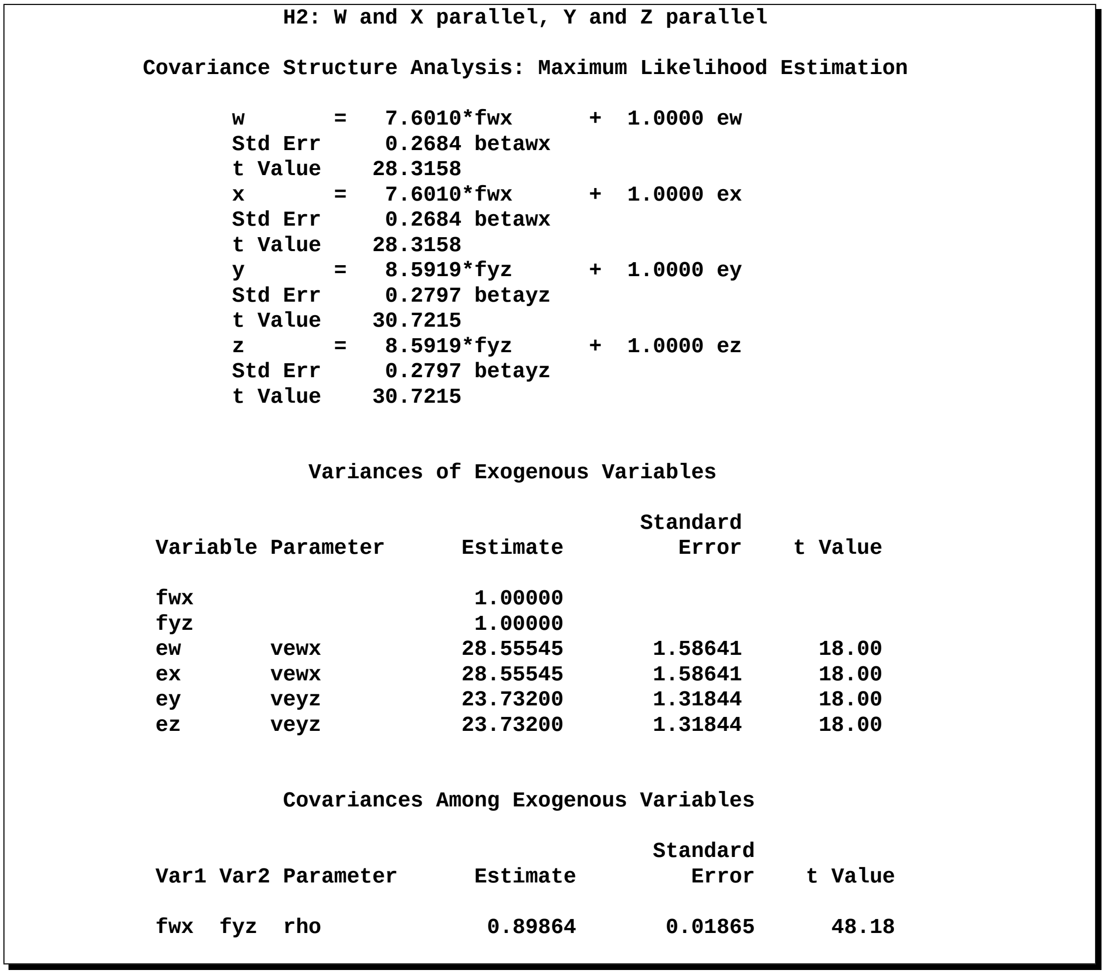
```
title 'H2: W and X parallel, Y and Z parallel';
proc calis data=lord cov;
   lineqs w=betawx fwx + ew,
          x=betawx fwx + ex,
          y=betayz fyz + ey,
          z=betayz fyz + ez;
   std fwx fyz=1,
       ew ex ey ez=vewx vewx veyz veyz;
   cov fwx fyz=rho;
run;
```

```
data ram2(type=ram);
   set ram4;
   if _name_=’betaw’ then _name_=’betawx’;
   if _name_=’betax’ then _name_=’betawx’;
   if _name_=’betay’ then _name_=’betayz’;
   if _name_=’betaz’ then _name_=’betayz’;
   if _name_=’vew’ then _name_=’vewx’;
   if _name_=’vex’ then _name_=’vewx’;
   if _name_=’vey’ then _name_=’veyz’;
   if _name_=’vez’ then _name_=’veyz’;
run;


proc calis data=lord inram=ram2 cov;
run;
```

The resulting output from either of these analyses is displayed in Figure 13.15.

```
                 H2: W and X parallel, Y and Z parallel

                          The CALIS Procedure
         Covariance Structure Analysis: Maximum Likelihood Estimation

         Fit Function                                          0.0030
         Goodness of Fit Index (GFI)                           0.9985
         GFI Adjusted for Degrees of Freedom (AGFI)            0.9970
         Root Mean Square Residual (RMR)                       0.6983
         Parsimonious GFI (Mulaik, 1989)                       0.8321
         Chi-Square                                            1.9335
         Chi-Square DF                                              5
         Pr > Chi-Square                                       0.8583
         Independence Model Chi-Square                         1466.6
         Independence Model Chi-Square DF                           6
         RMSEA Estimate                                        0.0000
         RMSEA 90% Lower Confidence Limit                           .
         RMSEA 90% Upper Confidence Limit                      0.0293
         ECVI Estimate                                         0.0185
         ECVI 90% Lower Confidence Limit                            .
         ECVI 90% Upper Confidence Limit                       0.0276
         Probability of Close Fit                              0.9936
         Bentler’s Comparative Fit Index                       1.0000
         Normal Theory Reweighted LS Chi-Square                1.9568
         Akaike’s Information Criterion                       -8.0665
         Bozdogan’s (1987) CAIC                              -35.4436
         Schwarz’s Bayesian Criterion                        -30.4436
         McDonald’s (1989) Centrality                          1.0024
         Bentler & Bonett’s (1980) Non-normed Index            1.0025
         Bentler & Bonett’s (1980) NFI                         0.9987
         James, Mulaik, & Brett (1982) Parsimonious NFI        0.8322
         Z-Test of Wilson & Hilferty (1931)                   -1.0768
         Bollen (1986) Normed Index Rho1                       0.9984
         Bollen (1988) Non-normed Index Delta2                 1.0021
         Hoelter’s (1983) Critical N                             3712
```

Figure 13.15. Lord Data: Major Results for Hypothesis H2



```
                    H2: W and X parallel, Y and Z parallel

         Covariance Structure Analysis: Maximum Likelihood Estimation

                w       =   7.6010*fwx      +  1.0000 ew
                Std Err     0.2684 betawx
                t Value    28.3158
                x       =   7.6010*fwx      +  1.0000 ex
                Std Err     0.2684 betawx
                t Value    28.3158
                y       =   8.5919*fyz      +  1.0000 ey
                Std Err     0.2797 betayz
                t Value    30.7215
                z       =   8.5919*fyz      +  1.0000 ez
                Std Err     0.2797 betayz
                t Value    30.7215


                      Variances of Exogenous Variables

                                                  Standard
          Variable Parameter      Estimate           Error    t Value

          fwx                      1.00000
          fyz                      1.00000
          ew       vewx           28.55545         1.58641      18.00
          ex       vewx           28.55545         1.58641      18.00
          ey       veyz           23.73200         1.31844      18.00
          ez       veyz           23.73200         1.31844      18.00


                    Covariances Among Exogenous Variables

                                                   Standard
          Var1 Var2 Parameter      Estimate           Error    t Value

          fwx  fyz  rho             0.89864         0.01865      48.18
```

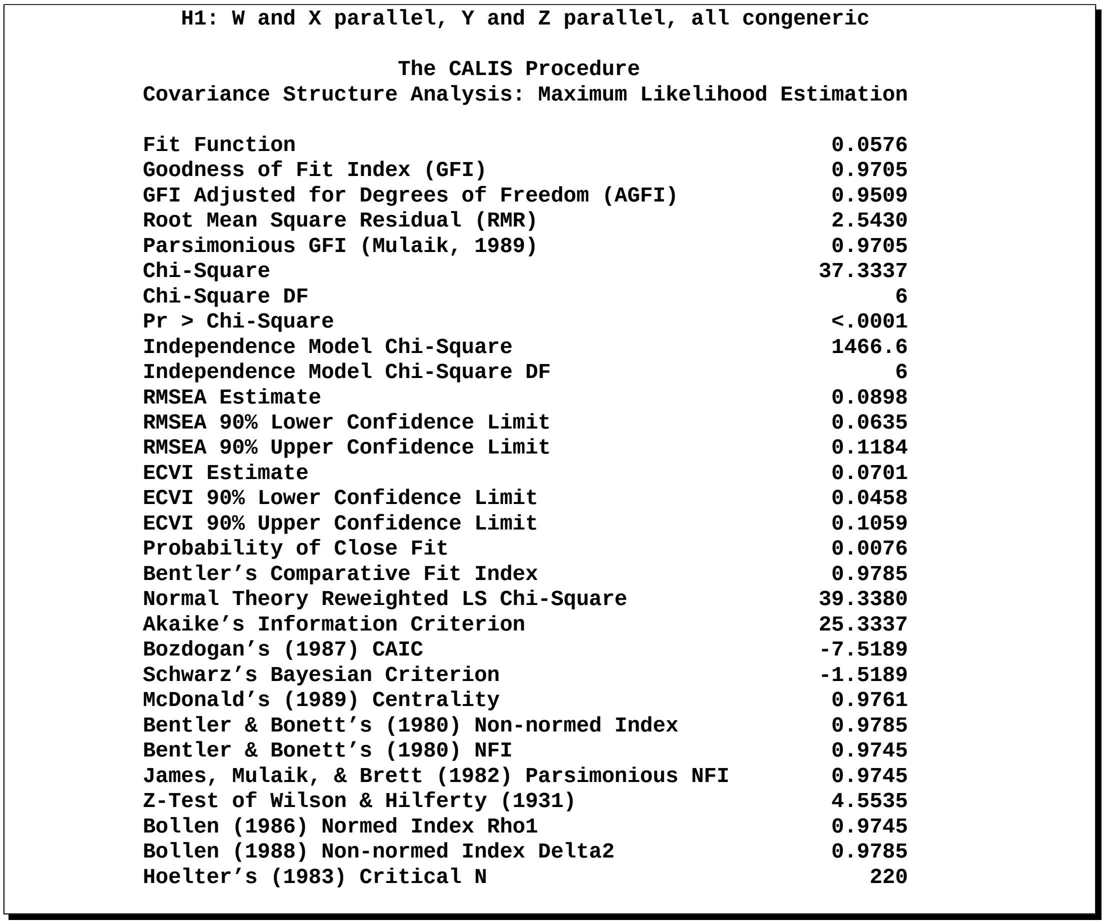
Figure 13.15. (continued)

The hypothesis H_1 requires one more constraint in addition to those in H_2:

```
title 'H1: W and X parallel, Y and Z parallel, all congeneric';
proc calis data=lord cov;
   lineqs w=betawx f + ew,
          x=betawx f + ex,
          y=betayz f + ey,
          z=betayz f + ez;
   std f=1,
       ew ex ey ez=vewx vewx veyz veyz;
run;
```

```
data ram1(type=ram);
   set ram2;
   if _name_='rho' then
      do;
         _name_=' ';
         _estim_=1;
      end;
run;


proc calis data=lord inram=ram1 cov;
run;
```

The resulting output from either of these analyses is displayed in Figure 13.16.



```
              H1: W and X parallel, Y and Z parallel, all congeneric

                                The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                          0.0576
           Goodness of Fit Index (GFI)                           0.9705
           GFI Adjusted for Degrees of Freedom (AGFI)            0.9509
           Root Mean Square Residual (RMR)                       2.5430
           Parsimonious GFI (Mulaik, 1989)                       0.9705
           Chi-Square                                           37.3337
           Chi-Square DF                                              6
           Pr > Chi-Square                                       <.0001
           Independence Model Chi-Square                         1466.6
           Independence Model Chi-Square DF                           6
           RMSEA Estimate                                        0.0898
           RMSEA 90% Lower Confidence Limit                      0.0635
           RMSEA 90% Upper Confidence Limit                      0.1184
           ECVI Estimate                                         0.0701
           ECVI 90% Lower Confidence Limit                       0.0458
           ECVI 90% Upper Confidence Limit                       0.1059
           Probability of Close Fit                              0.0076
           Bentler's Comparative Fit Index                       0.9785
           Normal Theory Reweighted LS Chi-Square               39.3380
           Akaike's Information Criterion                       25.3337
           Bozdogan's (1987) CAIC                               -7.5189
           Schwarz's Bayesian Criterion                         -1.5189
           McDonald's (1989) Centrality                          0.9761
           Bentler & Bonett's (1980) Non-normed Index            0.9785
           Bentler & Bonett's (1980) NFI                         0.9745
           James, Mulaik, & Brett (1982) Parsimonious NFI        0.9745
           Z-Test of Wilson & Hilferty (1931)                    4.5535
           Bollen (1986) Normed Index Rho1                       0.9745
           Bollen (1988) Non-normed Index Delta2                 0.9785
           Hoelter's (1983) Critical N                              220
```

Figure 13.16. Lord Data: Major Results for Hypothesis H1

```
               H1: W and X parallel, Y and Z parallel, all congeneric

            Covariance Structure Analysis: Maximum Likelihood Estimation

                   w        =    7.1862*fwx      +  1.0000 ew
                   Std Err       0.2660 betawx
                   t Value      27.0180
                   x        =    7.1862*fwx      +  1.0000 ex
                   Std Err       0.2660 betawx
                   t Value      27.0180
                   y        =    8.4420*fyz      +  1.0000 ey
                   Std Err       0.2800 betayz
                   t Value      30.1494
                   z        =    8.4420*fyz      +  1.0000 ez
                   Std Err       0.2800 betayz
                   t Value      30.1494


                         Variances of Exogenous Variables

                                                        Standard
             Variable Parameter      Estimate              Error    t Value

             fwx                      1.00000
             fyz                      1.00000
             ew       vewx           34.68865          1.64634      21.07
             ex       vewx           34.68865          1.64634      21.07
             ey       veyz           26.28513          1.39955      18.78
             ez       veyz           26.28513          1.39955      18.78


                       Covariances Among Exogenous Variables

                                                         Standard
             Var1 Var2 Parameter      Estimate              Error    t Value

             fwx  fyz                  1.00000
```

Figure 13.16. (continued)

The goodness-of-fit tests for the four hypotheses are summarized in the following table.

Hypothesis	**Number** of **Parameters**	χ^2	**Degrees of Freedom**	p**-value**	$\hat{\rho}$
H_1	4	37.33	6	0.0000	1.0
H_2	5	1.93	5	0.8583	0.8986
H_3	8	36.21	2	0.0000	1.0
H_4	9	0.70	1	0.4018	0.8986

The hypotheses H_1 and H_3, which posit $\rho = 1$, can be rejected. Hypotheses H_2 and H_4 seem to be consistent with the available data. Since H_2 is obtained by adding four constraints to H_4, you can test H_2 versus H_4 by computing the differences of the chi-square statistics and their degrees of freedom, yielding a chi-square of 1.23

with four degrees of freedom, which is obviously not significant. So hypothesis H_2 is consistent with the available data.

The estimates of ρ for H_2 and H_4 are almost identical, about 0.90, indicating that the speeded and unspeeded tests are measuring almost the same latent variable, even though the hypotheses that stated they measured exactly the same latent variable are rejected.

A Combined Measurement-Structural Model with Reciprocal Influence and Correlated Residuals

To illustrate a more complex model, this example uses some well-known data from Haller and Butterworth (1960). Various models and analyses of these data are given by Duncan, Haller, and Portes (1968), Jöreskog and Sörbom (1988), and Loehlin (1987).

The study is concerned with the career aspirations of high-school students and how these aspirations are affected by close friends. The data are collected from 442 seventeen-year-old boys in Michigan. There are 329 boys in the sample who named another boy in the sample as a best friend. The observations to be analyzed consist of the data from these 329 boys paired with the data from their best friends.

The method of data collection introduces two statistical problems. First, restricting the analysis to boys whose best friends are in the original sample causes the reduced sample to be biased. Second, since the data from a given boy may appear in two or more observations, the observations are not independent. Therefore, any statistical conclusions should be considered tentative. It is difficult to accurately assess the effects of the dependence of the observations on the analysis, but it could be argued on intuitive grounds that since each observation has data from two boys and since it seems likely that many of the boys will appear in the data set at least twice, the effective sample size may be as small as half of the reported 329 observations.

The correlation matrix is taken from Jöreskog and Sörbom (1988).

```
title 'Peer Influences on Aspiration: Haller & Butterworth (1960)';
data aspire(type=corr);
   _type_='corr';
   input _name_ $ riq rpa rses roa rea fiq fpa fses foa fea;
   label riq='Respondent: Intelligence'
         rpa='Respondent: Parental Aspiration'
         rses='Respondent: Family SES'
         roa='Respondent: Occupational Aspiration'
         rea='Respondent: Educational Aspiration'
         fiq='Friend: Intelligence'
         fpa='Friend: Parental Aspiration'
         fses='Friend: Family SES'
         foa='Friend: Occupational Aspiration'
         fea='Friend: Educational Aspiration';
   datalines;
riq   1.      .      .      .      .      .       .      .      .      .
rpa   .1839  1.     .      .      .      .       .      .      .      .
rses  .2220  .0489  1.     .      .      .       .      .      .      .
roa   .4105  .2137  .3240  1.     .      .       .      .      .      .
rea   .4043  .2742  .4047  .6247  1.     .       .      .      .      .
fiq   .3355  .0782  .2302  .2995  .2863  1.      .      .      .      .
fpa   .1021  .1147  .0931  .0760  .0702  .2087   1.     .      .      .
fses  .1861  .0186  .2707  .2930  .2407  .2950  -.0438  1.     .      .
foa   .2598  .0839  .2786  .4216  .3275  .5007   .1988  .3607  1.     .
fea   .2903  .1124  .3054  .3269  .3669  .5191   .2784  .4105  .6404  1.
;
```

The model analyzed by Jöreskog and Sörbom (1988) is displayed in the following path diagram:

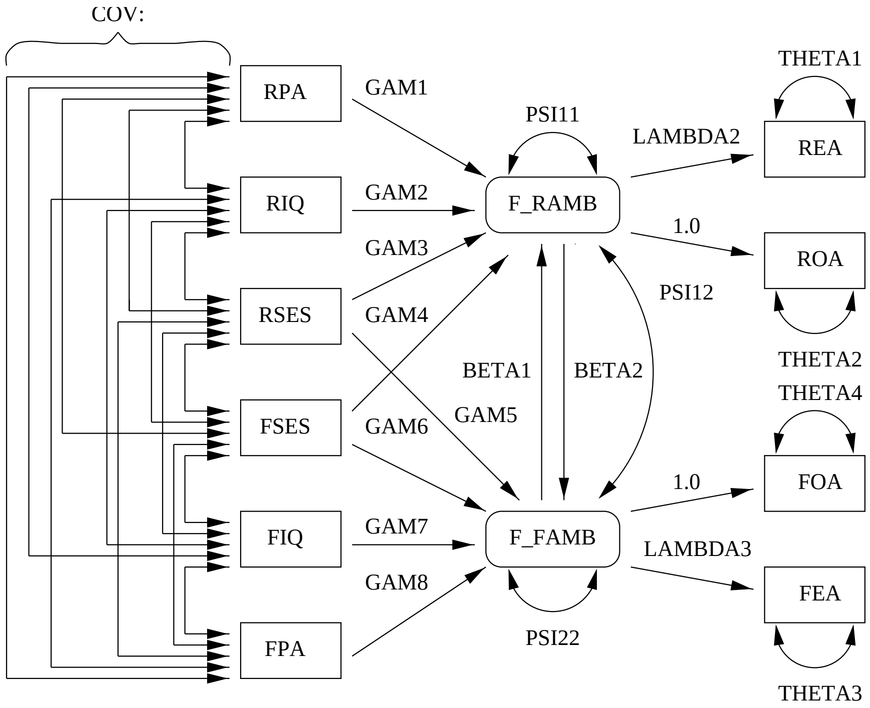


Figure 13.17. Path Diagram: Career Aspiration – Jöreskog and Sörbom

Two latent variables, f_ramb and f_famb, represent the respondent's level of ambition and his best friend's level of ambition, respectively. The model states that the respondent's ambition is determined by his intelligence and socioeconomic status, his perception of his parents' aspiration for him, and his friend's socioeconomic status and ambition. It is assumed that his friend's intelligence and socioeconomic status affect the respondent's ambition only indirectly through his friend's ambition. Ambition is indexed by the manifest variables of occupational and educational aspiration, which are assumed to have uncorrelated residuals. The path coefficient from ambition to occupational aspiration is set to 1.0 to determine the scale of the ambition latent variable.

This model can be analyzed with PROC CALIS using the LINEQS statement as follows, where the names of the parameters correspond to those used by Jöreskog and Sörbom (1988). Since this TYPE=CORR data set does not contain an observation with _TYPE_='N' giving the sample size, it is necessary to specify the degrees of freedom (sample size minus one) with the EDF= option in the PROC CALIS statement.

```
title2 'Joreskog-Sorbom (1988) analysis 1';
proc calis data=aspire edf=328;
   lineqs    /* measurement model for aspiration */
          rea=lambda2 f_ramb + e_rea,
          roa=f_ramb + e_roa,
          fea=lambda3 f_famb + e_fea,
          foa=f_famb + e_foa,
             /* structural model of influences */
          f_ramb=gam1 rpa + gam2 riq + gam3 rses +
             gam4 fses + beta1 f_famb + d_ramb,
          f_famb=gam8 fpa + gam7 fiq + gam6 fses +
             gam5 rses + beta2 f_ramb + d_famb;
   std d_ramb=psi11,
       d_famb=psi22,
       e_rea e_roa e_fea e_foa=theta:;
   cov d_ramb d_famb=psi12,
       rpa riq rses fpa fiq fses=cov:;
run;
```

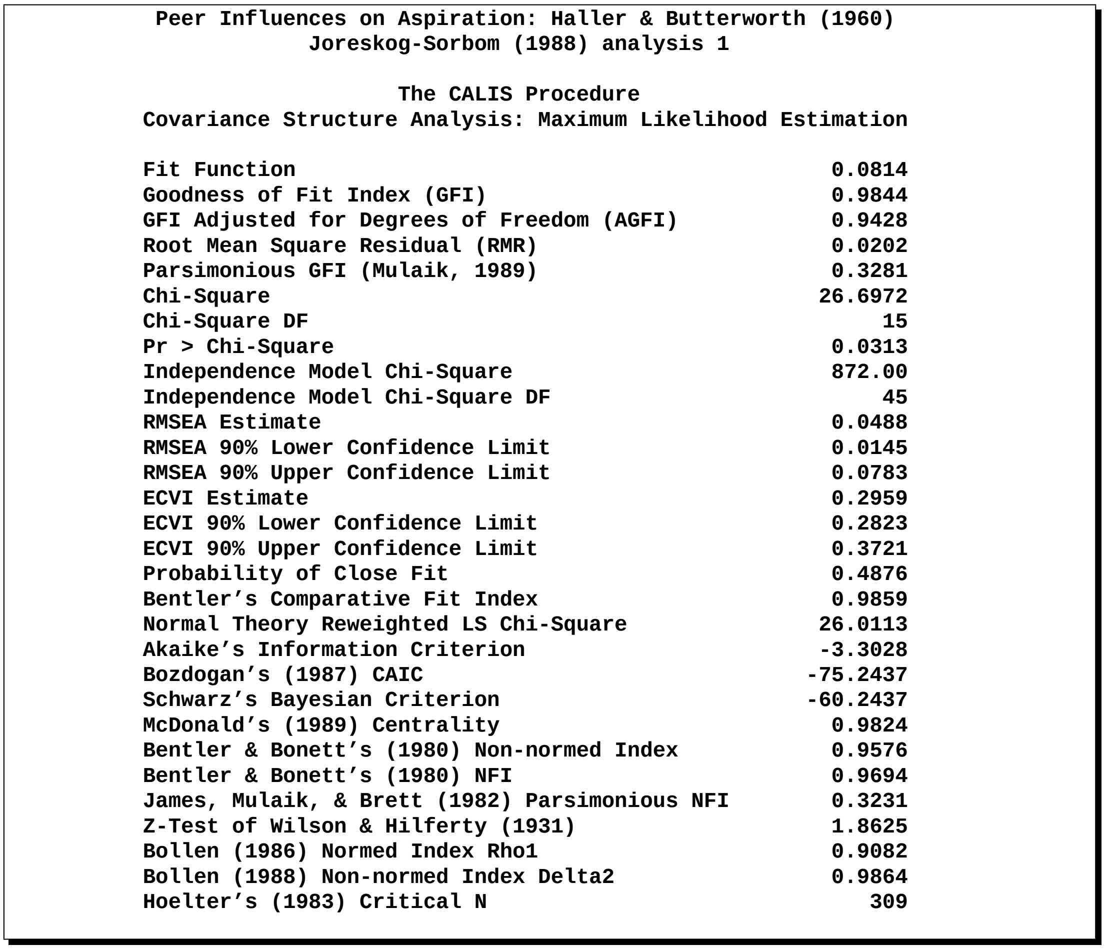
Specify a name followed by a colon to represent a list of names formed by appending numbers to the specified name. For example, in the COV statement, the line

```
rpa riq rses fpa fiq fses=cov:;
```

is equivalent to

```
rpa riq rses fpa fiq fses=cov1-cov15;
```

The results from this analysis are as follows.



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Joreskog-Sorbom (1988) analysis 1

                               The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                          0.0814
           Goodness of Fit Index (GFI)                           0.9844
           GFI Adjusted for Degrees of Freedom (AGFI)            0.9428
           Root Mean Square Residual (RMR)                       0.0202
           Parsimonious GFI (Mulaik, 1989)                       0.3281
           Chi-Square                                           26.6972
           Chi-Square DF                                             15
           Pr > Chi-Square                                       0.0313
           Independence Model Chi-Square                         872.00
           Independence Model Chi-Square DF                          45
           RMSEA Estimate                                        0.0488
           RMSEA 90% Lower Confidence Limit                      0.0145
           RMSEA 90% Upper Confidence Limit                      0.0783
           ECVI Estimate                                         0.2959
           ECVI 90% Lower Confidence Limit                       0.2823
           ECVI 90% Upper Confidence Limit                       0.3721
           Probability of Close Fit                              0.4876
           Bentler's Comparative Fit Index                       0.9859
           Normal Theory Reweighted LS Chi-Square               26.0113
           Akaike's Information Criterion                       -3.3028
           Bozdogan's (1987) CAIC                              -75.2437
           Schwarz's Bayesian Criterion                        -60.2437
           McDonald's (1989) Centrality                          0.9824
           Bentler & Bonett's (1980) Non-normed Index            0.9576
           Bentler & Bonett's (1980) NFI                         0.9694
           James, Mulaik, & Brett (1982) Parsimonious NFI        0.3231
           Z-Test of Wilson & Hilferty (1931)                    1.8625
           Bollen (1986) Normed Index Rho1                       0.9082
           Bollen (1988) Non-normed Index Delta2                 0.9864
           Hoelter's (1983) Critical N                              309
```

Figure 13.18. Career Aspiration Data: J&S Analysis 1

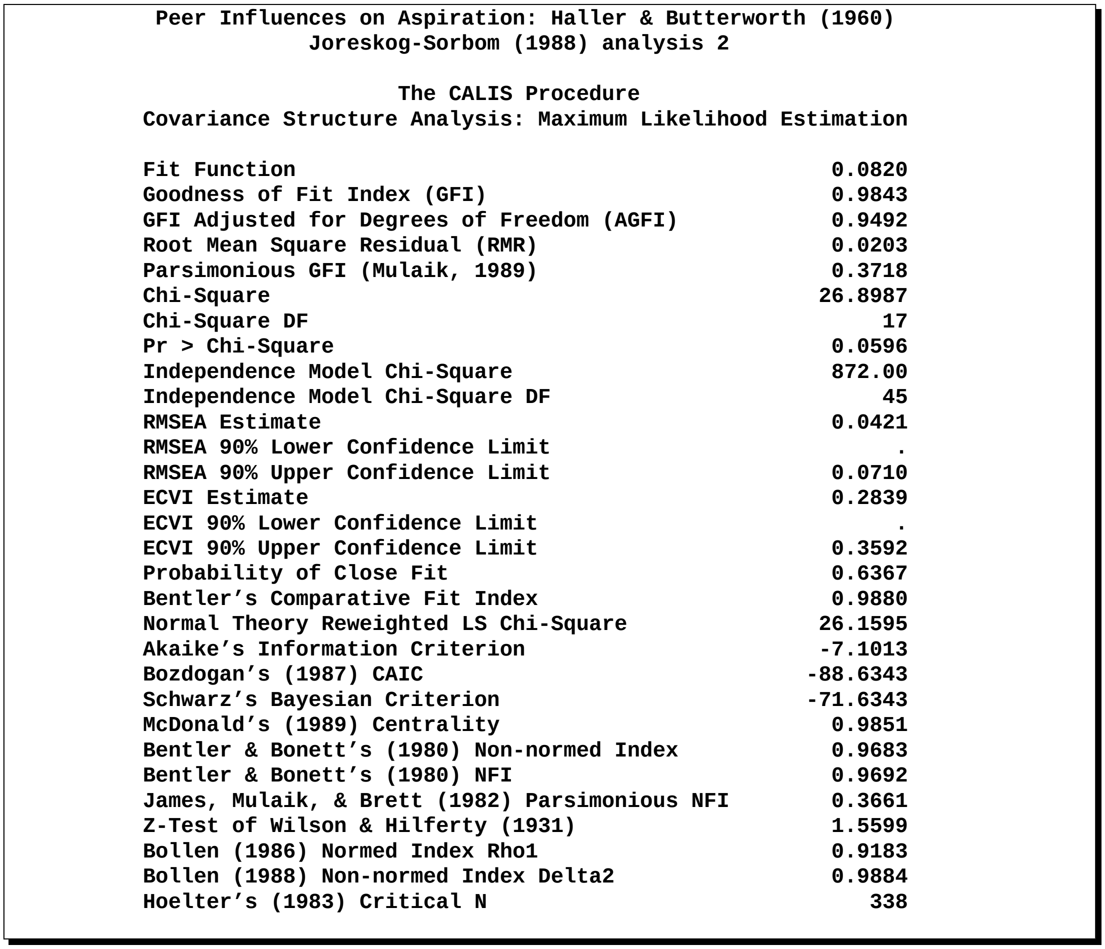
Jöreskog and Sörbom (1988) present more detailed results from a second analysis in which two constraints are imposed:

- The coefficients connecting the latent ambition variables are equal.
- The covariance of the disturbances of the ambition variables is zero.

This analysis can be performed by changing the names beta1 and beta2 to beta and omitting the line from the COV statement for psi12:

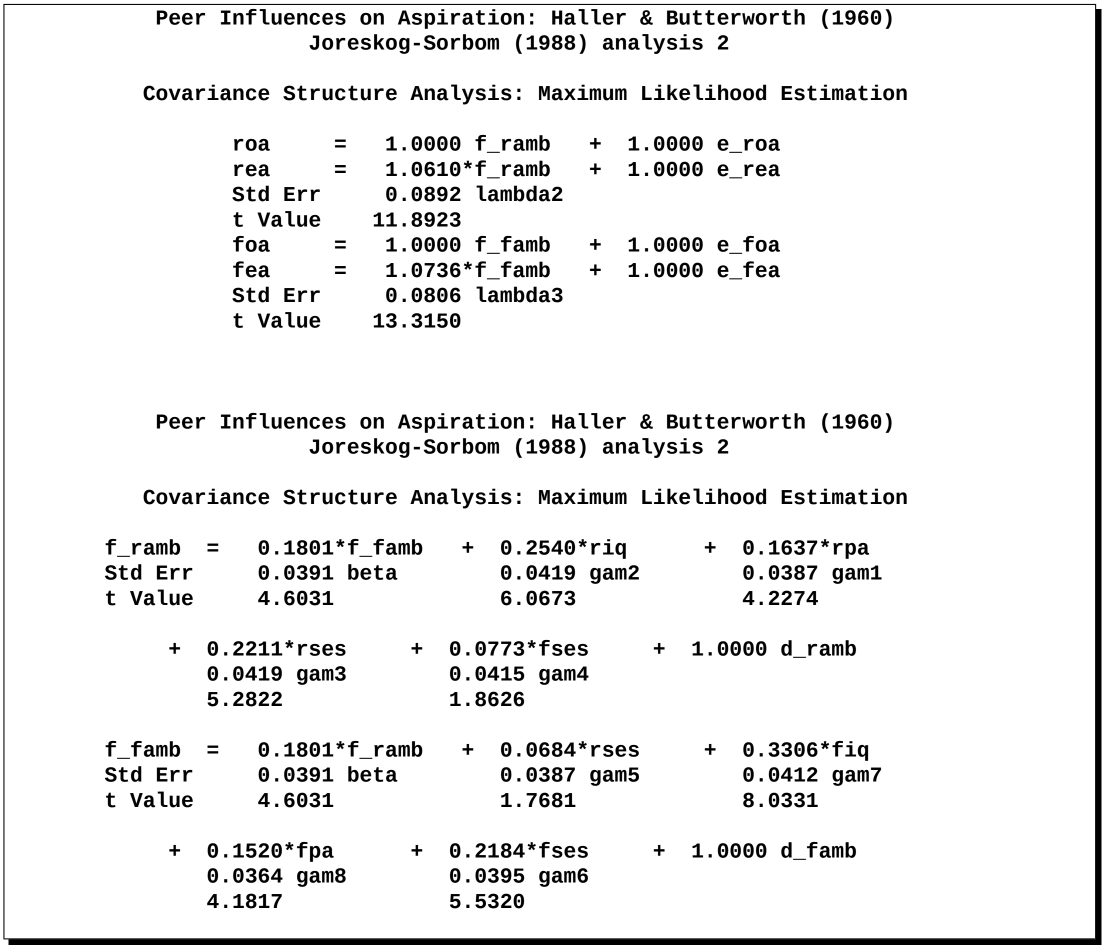
```
title2 ’Joreskog-Sorbom (1988) analysis 2’;
proc calis data=aspire edf=328;
   lineqs    /* measurement model for aspiration */
          rea=lambda2 f_ramb + e_rea,
          roa=f_ramb + e_roa,
          fea=lambda3 f_famb + e_fea,
          foa=f_famb + e_foa,
             /* structural model of influences */
          f_ramb=gam1 rpa + gam2 riq + gam3 rses +
             gam4 fses + beta f_famb + d_ramb,
          f_famb=gam8 fpa + gam7 fiq + gam6 fses +
             gam5 rses + beta f_ramb + d_famb;
   std d_ramb=psi11,
       d_famb=psi22,
       e_rea e_roa e_fea e_foa=theta:;
   cov rpa riq rses fpa fiq fses=cov:;
run;
```

The results are displayed in Figure 13.19.



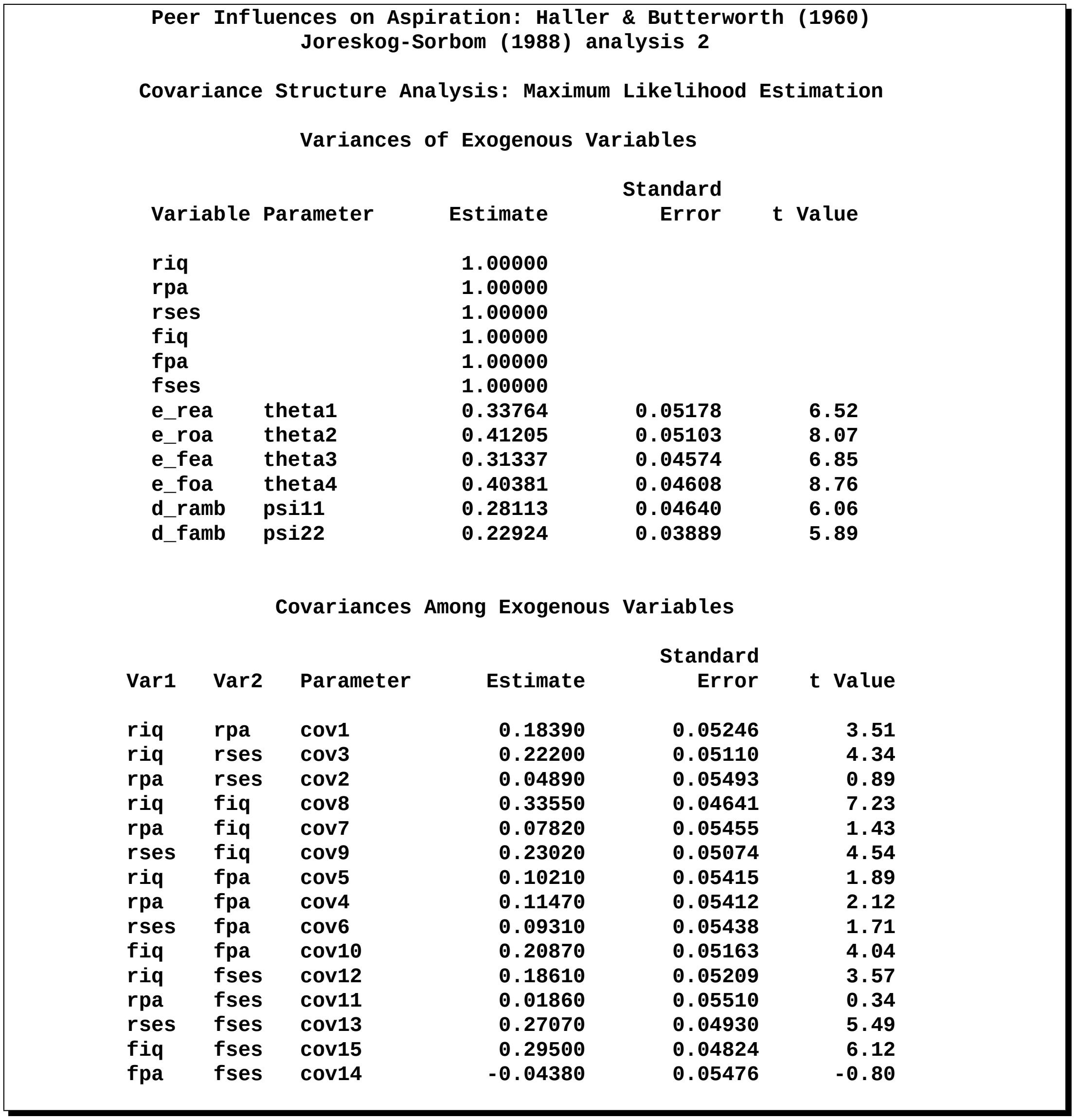
```
 Peer Influences on Aspiration: Haller & Butterworth (1960)
             Joreskog-Sorbom (1988) analysis 2

                    The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function                                          0.0820
Goodness of Fit Index (GFI)                           0.9843
GFI Adjusted for Degrees of Freedom (AGFI)            0.9492
Root Mean Square Residual (RMR)                       0.0203
Parsimonious GFI (Mulaik, 1989)                       0.3718
Chi-Square                                           26.8987
Chi-Square DF                                             17
Pr > Chi-Square                                       0.0596
Independence Model Chi-Square                         872.00
Independence Model Chi-Square DF                          45
RMSEA Estimate                                        0.0421
RMSEA 90% Lower Confidence Limit                           .
RMSEA 90% Upper Confidence Limit                      0.0710
ECVI Estimate                                         0.2839
ECVI 90% Lower Confidence Limit                            .
ECVI 90% Upper Confidence Limit                       0.3592
Probability of Close Fit                              0.6367
Bentler’s Comparative Fit Index                       0.9880
Normal Theory Reweighted LS Chi-Square               26.1595
Akaike’s Information Criterion                       -7.1013
Bozdogan’s (1987) CAIC                              -88.6343
Schwarz’s Bayesian Criterion                        -71.6343
McDonald’s (1989) Centrality                          0.9851
Bentler & Bonett’s (1980) Non-normed Index            0.9683
Bentler & Bonett’s (1980) NFI                         0.9692
James, Mulaik, & Brett (1982) Parsimonious NFI        0.3661
Z-Test of Wilson & Hilferty (1931)                    1.5599
Bollen (1986) Normed Index Rho1                       0.9183
Bollen (1988) Non-normed Index Delta2                 0.9884
Hoelter’s (1983) Critical N                              338
```

Figure 13.19. Career Aspiration Data: J&S Analysis 2



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Joreskog-Sorbom (1988) analysis 2

           Covariance Structure Analysis: Maximum Likelihood Estimation

                  roa      =   1.0000 f_ramb   +  1.0000 e_roa
                  rea      =   1.0610*f_ramb   +  1.0000 e_rea
                  Std Err      0.0892 lambda2
                  t Value     11.8923
                  foa      =   1.0000 f_famb   +  1.0000 e_foa
                  fea      =   1.0736*f_famb   +  1.0000 e_fea
                  Std Err      0.0806 lambda3
                  t Value     13.3150



            Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Joreskog-Sorbom (1988) analysis 2

           Covariance Structure Analysis: Maximum Likelihood Estimation

        f_ramb  =   0.1801*f_famb   +  0.2540*riq      +  0.1637*rpa
        Std Err     0.0391 beta        0.0419 gam2        0.0387 gam1
        t Value     4.6031             6.0673             4.2274

             +  0.2211*rses     +  0.0773*fses     +  1.0000 d_ramb
                0.0419 gam3        0.0415 gam4
                5.2822             1.8626

        f_famb  =   0.1801*f_ramb   +  0.0684*rses     +  0.3306*fiq
        Std Err     0.0391 beta        0.0387 gam5        0.0412 gam7
        t Value     4.6031             1.7681             8.0331

             +  0.1520*fpa      +  0.2184*fses     +  1.0000 d_famb
                0.0364 gam8        0.0395 gam6
                4.1817             5.5320
```

Figure 13.19. (continued)



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Joreskog-Sorbom (1988) analysis 2

          Covariance Structure Analysis: Maximum Likelihood Estimation

                        Variances of Exogenous Variables

                                                           Standard
            Variable Parameter      Estimate                  Error    t Value

            riq                      1.00000
            rpa                      1.00000
            rses                     1.00000
            fiq                      1.00000
            fpa                      1.00000
            fses                     1.00000
            e_rea    theta1          0.33764              0.05178       6.52
            e_roa    theta2          0.41205              0.05103       8.07
            e_fea    theta3          0.31337              0.04574       6.85
            e_foa    theta4          0.40381              0.04608       8.76
            d_ramb   psi11           0.28113              0.04640       6.06
            d_famb   psi22           0.22924              0.03889       5.89


                      Covariances Among Exogenous Variables

                                                           Standard
          Var1   Var2   Parameter      Estimate               Error    t Value

          riq    rpa    cov1            0.18390             0.05246       3.51
          riq    rses   cov3            0.22200             0.05110       4.34
          rpa    rses   cov2            0.04890             0.05493       0.89
          riq    fiq    cov8            0.33550             0.04641       7.23
          rpa    fiq    cov7            0.07820             0.05455       1.43
          rses   fiq    cov9            0.23020             0.05074       4.54
          riq    fpa    cov5            0.10210             0.05415       1.89
          rpa    fpa    cov4            0.11470             0.05412       2.12
          rses   fpa    cov6            0.09310             0.05438       1.71
          fiq    fpa    cov10           0.20870             0.05163       4.04
          riq    fses   cov12           0.18610             0.05209       3.57
          rpa    fses   cov11           0.01860             0.05510       0.34
          rses   fses   cov13           0.27070             0.04930       5.49
          fiq    fses   cov15           0.29500             0.04824       6.12
          fpa    fses   cov14          -0.04380             0.05476      -0.80
```

Figure 13.19. (continued)

The difference between the chi-square values for the two preceding models is 26.8987 - 26.6972= 0.2015 with 2 degrees of freedom, which is far from significant. However, the chi-square test of the restricted model (analysis 2) against the alternative of a completely unrestricted covariance matrix yields a *p*-value of 0.0596, which indicates that the model may not be entirely satisfactory (*p*-values from these data are probably too small because of the dependence of the observations).

Loehlin (1987) points out that the models considered are unrealistic in at least two aspects. First, the variables of parental aspiration, intelligence, and socioeconomic status are assumed to be measured without error. Loehlin adds uncorrelated measurement errors to the model and assumes, for illustrative purposes, that the reliabilities of these variables are known to be 0.7, 0.8, and 0.9, respectively. In practice, these reliabilities would need to be obtained from a separate study of the same or a very similar population. If these constraints are omitted, the model is not identified. However, constraining parameters to a constant in an analysis of a correlation matrix may make the chi-square goodness-of-fit test inaccurate, so there is more reason to be skeptical of the *p*-values. Second, the error terms for the respondent's aspiration are assumed to be uncorrelated with the corresponding terms for his friend. Loehlin introduces a correlation between the two educational aspiration error terms and between the two occupational aspiration error terms. These additions produce the following path diagram for Loehlin's model 1.

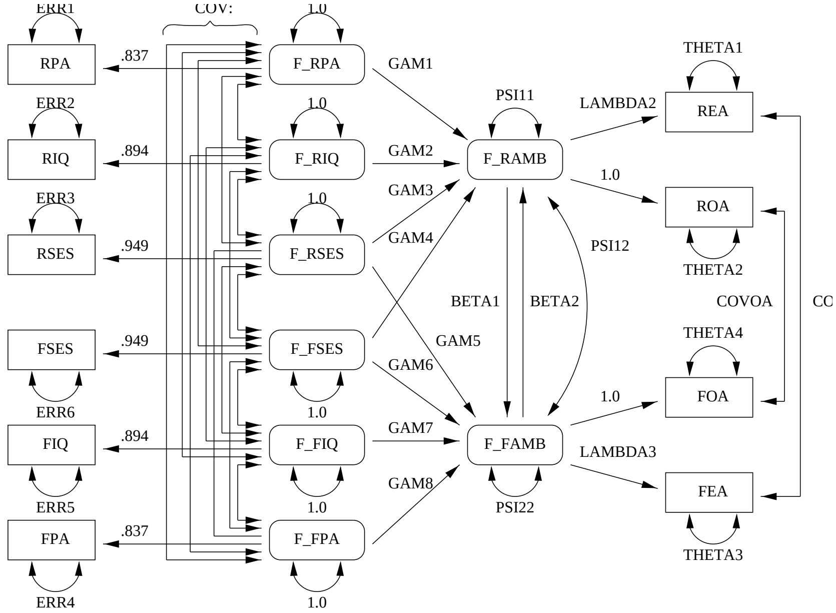
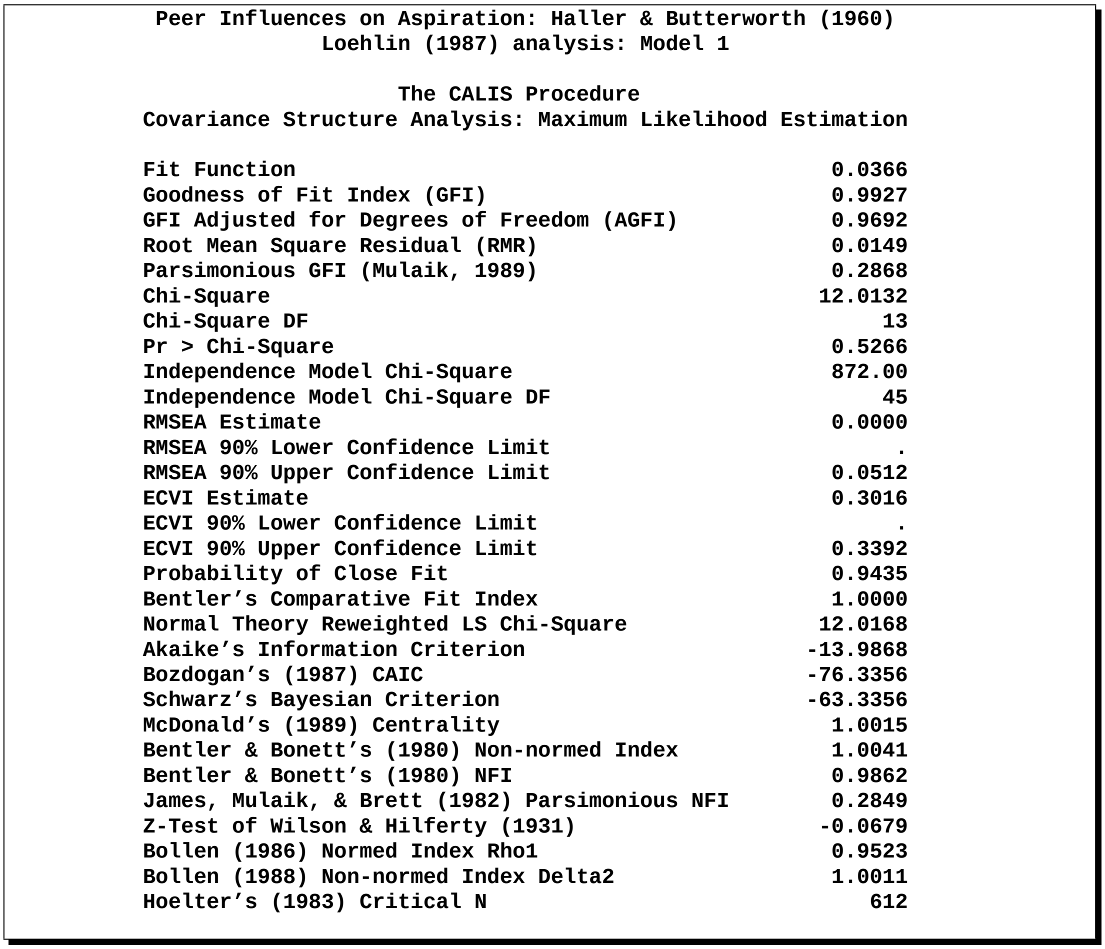


Figure 13.20. Path Diagram: Career Aspiration – Loehlin

The statements for fitting this model are as follows:

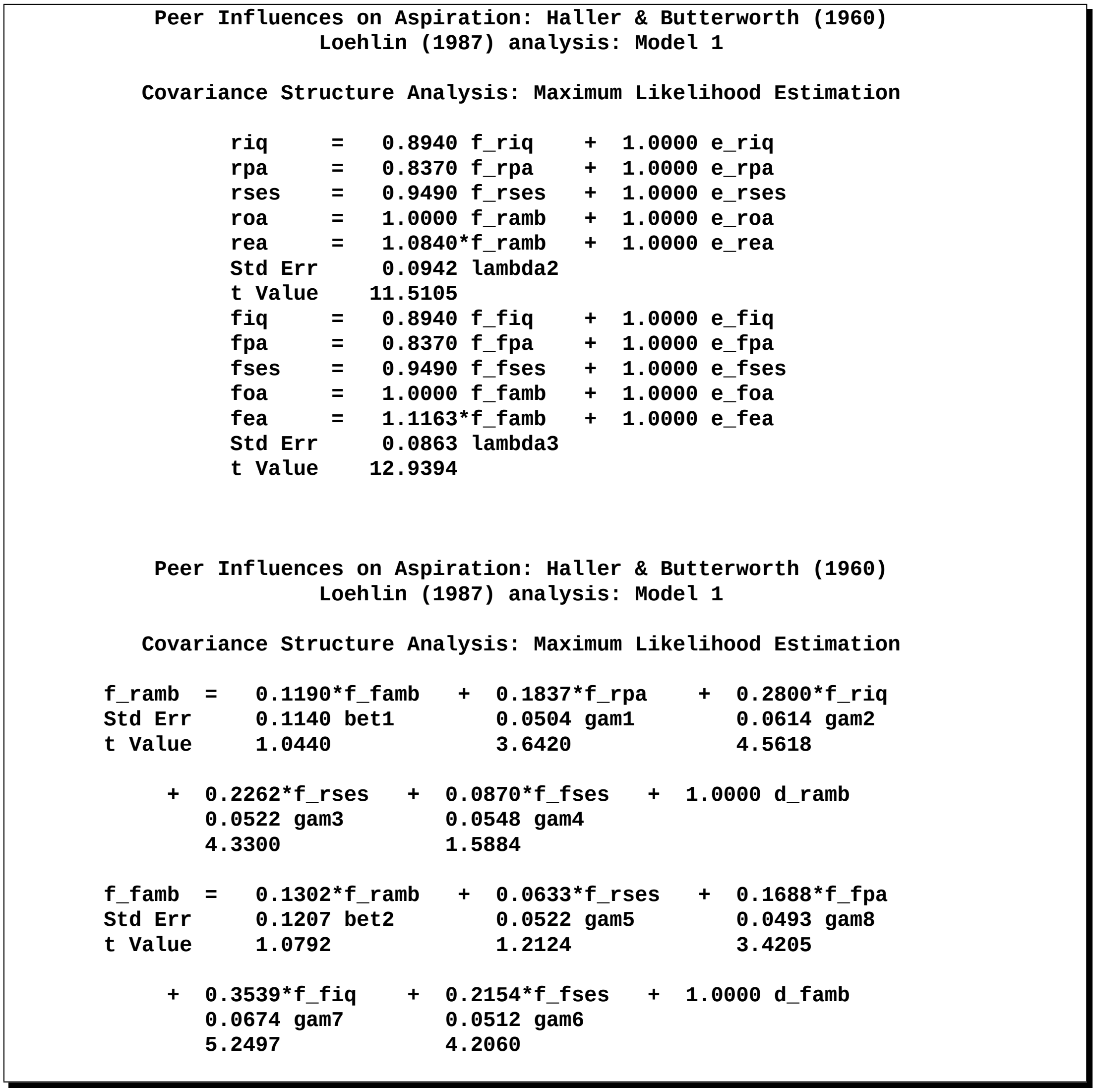
```
title2 ’Loehlin (1987) analysis: Model 1’;
proc calis data=aspire edf=328;
   lineqs    /* measurement model for aspiration */
          rea=lambda2 f_ramb + e_rea,
          roa=f_ramb + e_roa,
          fea=lambda3 f_famb + e_fea,
          foa=f_famb + e_foa,
          /* measurement model for intelligence and environment */
          rpa=.837 f_rpa + e_rpa,
          riq=.894 f_riq + e_riq,
          rses=.949 f_rses + e_rses,
          fpa=.837 f_fpa + e_fpa,
          fiq=.894 f_fiq + e_fiq,
          fses=.949 f_fses + e_fses,
             /* structural model of influences */
          f_ramb=gam1 f_rpa + gam2 f_riq + gam3 f_rses +
             gam4 f_fses + bet1 f_famb + d_ramb,
          f_famb=gam8 f_fpa + gam7 f_fiq + gam6 f_fses +
             gam5 f_rses + bet2 f_ramb + d_famb;
   std d_ramb=psi11,
       d_famb=psi22,
       f_rpa f_riq f_rses f_fpa f_fiq f_fses=1,
       e_rea e_roa e_fea e_foa=theta:,
       e_rpa e_riq e_rses e_fpa e_fiq e_fses=err:;
   cov d_ramb d_famb=psi12,
       e_rea e_fea=covea,
       e_roa e_foa=covoa,
       f_rpa f_riq f_rses f_fpa f_fiq f_fses=cov:;
run;
```

The results are displayed in Figure 13.21.



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 1

                               The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                          0.0366
           Goodness of Fit Index (GFI)                           0.9927
           GFI Adjusted for Degrees of Freedom (AGFI)            0.9692
           Root Mean Square Residual (RMR)                       0.0149
           Parsimonious GFI (Mulaik, 1989)                       0.2868
           Chi-Square                                           12.0132
           Chi-Square DF                                             13
           Pr > Chi-Square                                       0.5266
           Independence Model Chi-Square                         872.00
           Independence Model Chi-Square DF                          45
           RMSEA Estimate                                        0.0000
           RMSEA 90% Lower Confidence Limit                           .
           RMSEA 90% Upper Confidence Limit                      0.0512
           ECVI Estimate                                         0.3016
           ECVI 90% Lower Confidence Limit                            .
           ECVI 90% Upper Confidence Limit                       0.3392
           Probability of Close Fit                              0.9435
           Bentler's Comparative Fit Index                       1.0000
           Normal Theory Reweighted LS Chi-Square               12.0168
           Akaike's Information Criterion                      -13.9868
           Bozdogan's (1987) CAIC                              -76.3356
           Schwarz's Bayesian Criterion                        -63.3356
           McDonald's (1989) Centrality                          1.0015
           Bentler & Bonett's (1980) Non-normed Index            1.0041
           Bentler & Bonett's (1980) NFI                         0.9862
           James, Mulaik, & Brett (1982) Parsimonious NFI        0.2849
           Z-Test of Wilson & Hilferty (1931)                   -0.0679
           Bollen (1986) Normed Index Rho1                       0.9523
           Bollen (1988) Non-normed Index Delta2                 1.0011
           Hoelter's (1983) Critical N                              612
```

Figure 13.21. Career Aspiration Data: Loehlin Model 1



```
           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 1

          Covariance Structure Analysis: Maximum Likelihood Estimation

                 riq     =   0.8940 f_riq    +  1.0000 e_riq
                 rpa     =   0.8370 f_rpa    +  1.0000 e_rpa
                 rses    =   0.9490 f_rses   +  1.0000 e_rses
                 roa     =   1.0000 f_ramb   +  1.0000 e_roa
                 rea     =   1.0840*f_ramb   +  1.0000 e_rea
                 Std Err     0.0942 lambda2
                 t Value    11.5105
                 fiq     =   0.8940 f_fiq    +  1.0000 e_fiq
                 fpa     =   0.8370 f_fpa    +  1.0000 e_fpa
                 fses    =   0.9490 f_fses   +  1.0000 e_fses
                 foa     =   1.0000 f_famb   +  1.0000 e_foa
                 fea     =   1.1163*f_famb   +  1.0000 e_fea
                 Std Err     0.0863 lambda3
                 t Value    12.9394



           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 1

          Covariance Structure Analysis: Maximum Likelihood Estimation

       f_ramb  =   0.1190*f_famb   +  0.1837*f_rpa    +  0.2800*f_riq
       Std Err     0.1140 bet1        0.0504 gam1        0.0614 gam2
       t Value     1.0440             3.6420             4.5618

            +  0.2262*f_rses   +  0.0870*f_fses   +  1.0000 d_ramb
               0.0522 gam3        0.0548 gam4
               4.3300             1.5884

       f_famb  =   0.1302*f_ramb   +  0.0633*f_rses   +  0.1688*f_fpa
       Std Err     0.1207 bet2        0.0522 gam5        0.0493 gam8
       t Value     1.0792             1.2124             3.4205

            +  0.3539*f_fiq    +  0.2154*f_fses   +  1.0000 d_famb
               0.0674 gam7        0.0512 gam6
               5.2497             4.2060
```

Figure 13.21. (continued)

```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 1

           Covariance Structure Analysis: Maximum Likelihood Estimation

                        Variances of Exogenous Variables

                                                      Standard
            Variable Parameter      Estimate             Error    t Value

            f_rpa                    1.00000
            f_riq                    1.00000
            f_rses                   1.00000
            f_fpa                    1.00000
            f_fiq                    1.00000
            f_fses                   1.00000
            e_rea    theta1          0.32707           0.05452       6.00
            e_roa    theta2          0.42307           0.05243       8.07
            e_fea    theta3          0.28715           0.04804       5.98
            e_foa    theta4          0.42240           0.04730       8.93
            e_rpa    err1            0.29584           0.07774       3.81
            e_riq    err2            0.20874           0.07832       2.67
            e_rses   err3            0.09887           0.07803       1.27
            e_fpa    err4            0.29987           0.07807       3.84
            e_fiq    err5            0.19988           0.07674       2.60
            e_fses   err6            0.10324           0.07824       1.32
            d_ramb   psi11           0.25418           0.04469       5.69
            d_famb   psi22           0.19698           0.03814       5.17


                      Covariances Among Exogenous Variables

                                                         Standard
          Var1   Var2   Parameter      Estimate             Error    t Value

          f_rpa  f_riq  cov1            0.24677           0.07519       3.28
          f_rpa  f_rses cov2            0.06184           0.06945       0.89
          f_riq  f_rses cov3            0.26351           0.06687       3.94
          f_rpa  f_fpa  cov4            0.15789           0.07873       2.01
          f_riq  f_fpa  cov5            0.13085           0.07418       1.76
          f_rses f_fpa  cov6            0.11517           0.06978       1.65
          f_rpa  f_fiq  cov7            0.10853           0.07362       1.47
          f_riq  f_fiq  cov8            0.42476           0.07219       5.88
          f_rses f_fiq  cov9            0.27250           0.06660       4.09
          f_fpa  f_fiq  cov10           0.27867           0.07530       3.70
          f_rpa  f_fses cov11           0.02383           0.06952       0.34
          f_riq  f_fses cov12           0.22135           0.06648       3.33
          f_rses f_fses cov13           0.30156           0.06359       4.74
          f_fpa  f_fses cov14          -0.05623           0.06971      -0.81
          f_fiq  f_fses cov15           0.34922           0.06771       5.16
          e_rea  e_fea  covea           0.02308           0.03139       0.74
          e_roa  e_foa  covoa           0.11206           0.03258       3.44
          d_ramb d_famb psi12          -0.00935           0.05010      -0.19
```

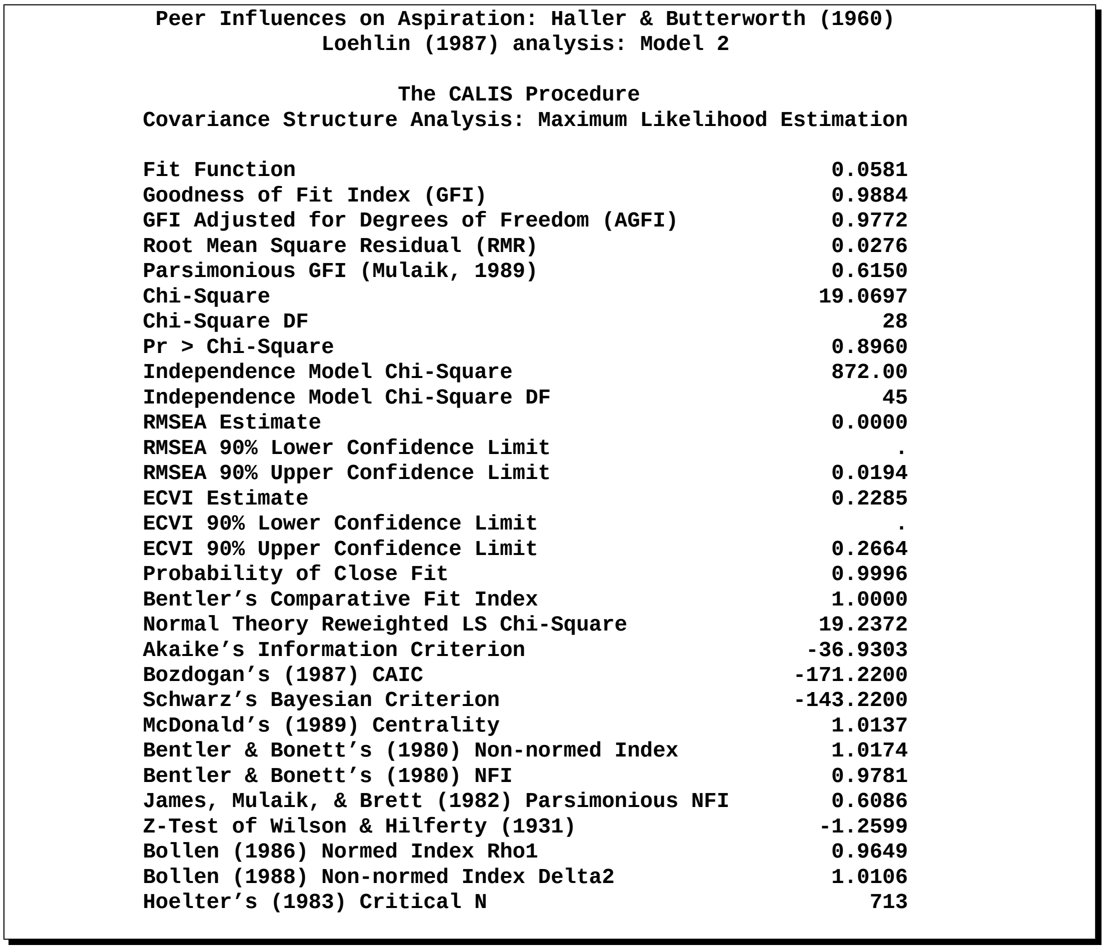
Figure 13.21. (continued)

Since the *p*-value for the chi-square test is 0.5266, this model clearly cannot be rejected. However, Schwarz's Bayesian Criterion for this model (SBC = -63.3356) is somewhat larger than for Jöreskog and Sörbom's (1988) analysis 2 (SBC =-71.6343), suggesting that a more parsimonious model would be desirable.

Since it is assumed that the same model applies to all the boys in the sample, the path diagram should be symmetric with respect to the respondent and friend. In particular, the corresponding coefficients should be equal. By imposing equality constraints on the 15 pairs of corresponding coefficients, this example obtains Loehlin's model 2. The LINEQS model is as follows, where an OUTRAM= data set is created to facilitate subsequent hypothesis tests:

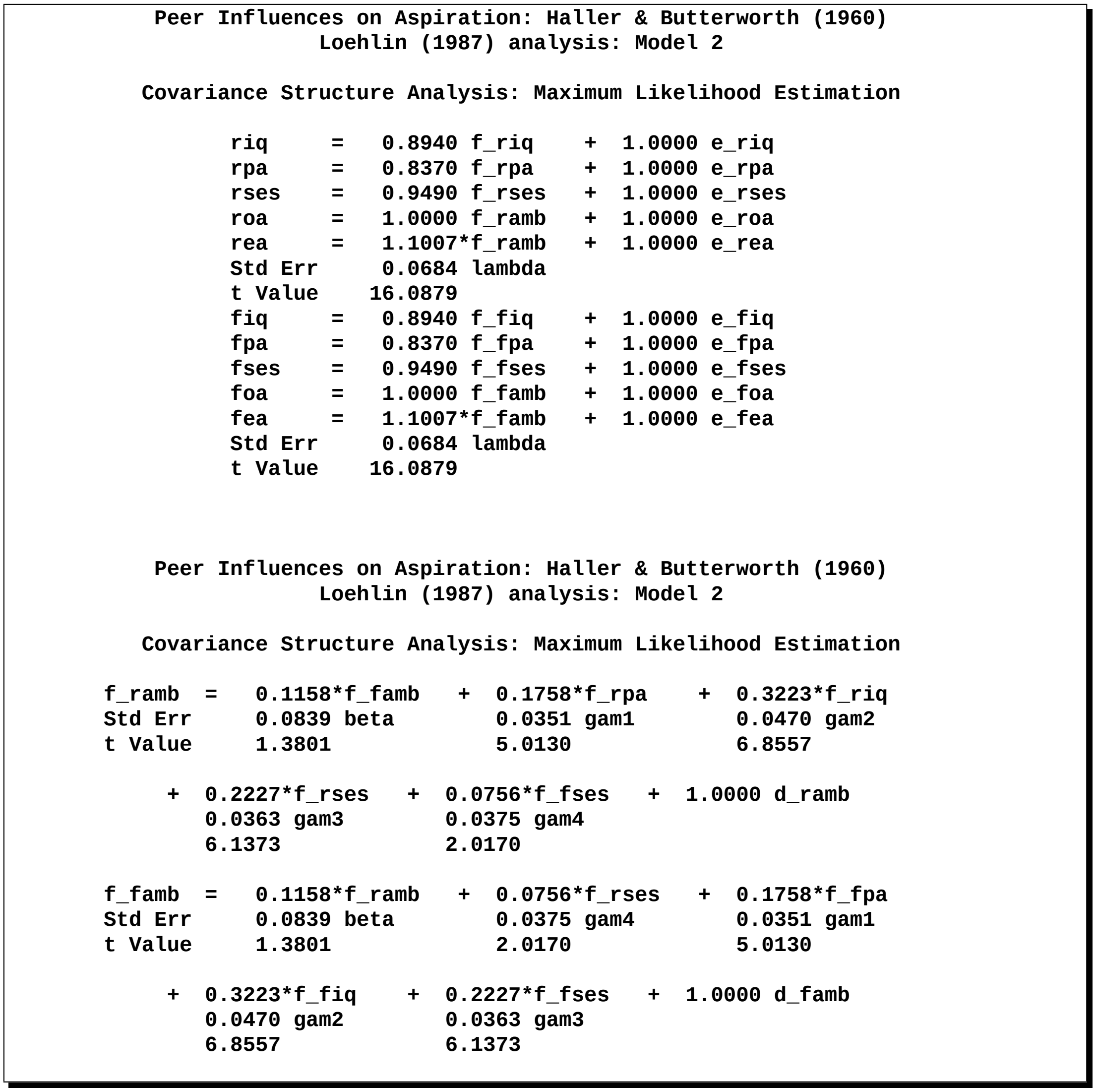
```
title2 'Loehlin (1987) analysis: Model 2';
proc calis data=aspire edf=328 outram=ram2;
   lineqs    /* measurement model for aspiration */
          rea=lambda f_ramb + e_rea,             /* 1 ec! */
          roa=f_ramb + e_roa,
          fea=lambda f_famb + e_fea,
          foa=f_famb + e_foa,
          /* measurement model for intelligence and environment */
          rpa=.837 f_rpa + e_rpa,
          riq=.894 f_riq + e_riq,
          rses=.949 f_rses + e_rses,
          fpa=.837 f_fpa + e_fpa,
          fiq=.894 f_fiq + e_fiq,
          fses=.949 f_fses + e_fses,
             /* structural model of influences */     /* 5 ec! */
          f_ramb=gam1 f_rpa + gam2 f_riq + gam3 f_rses +
             gam4 f_fses + beta f_famb + d_ramb,
          f_famb=gam1 f_fpa + gam2 f_fiq + gam3 f_fses +
             gam4 f_rses + beta f_ramb + d_famb;
   std d_ramb=psi,                               /* 1 ec! */
       d_famb=psi,
       f_rpa f_riq f_rses f_fpa f_fiq f_fses=1,
       e_rea e_fea=thetaea thetaea,              /* 2 ec! */
       e_roa e_foa=thetaoa thetaoa,
       e_rpa e_fpa=errpa1 errpa2,
       e_riq e_fiq=erriq1 erriq2,
       e_rses e_fses=errses1 errses2;
   cov d_ramb d_famb=psi12,
       e_rea e_fea=covea,
       e_roa e_foa = covoa,
       f_rpa f_riq f_rses=cov1-cov3,             /* 3 ec! */
       f_fpa f_fiq f_fses=cov1-cov3,
       f_rpa f_riq f_rses * f_fpa f_fiq f_fses =   /* 3 ec! */
          cov4 cov5 cov6
          cov5 cov7 cov8
          cov6 cov8 cov9;
run;
```

The results are displayed in Figure 13.22.



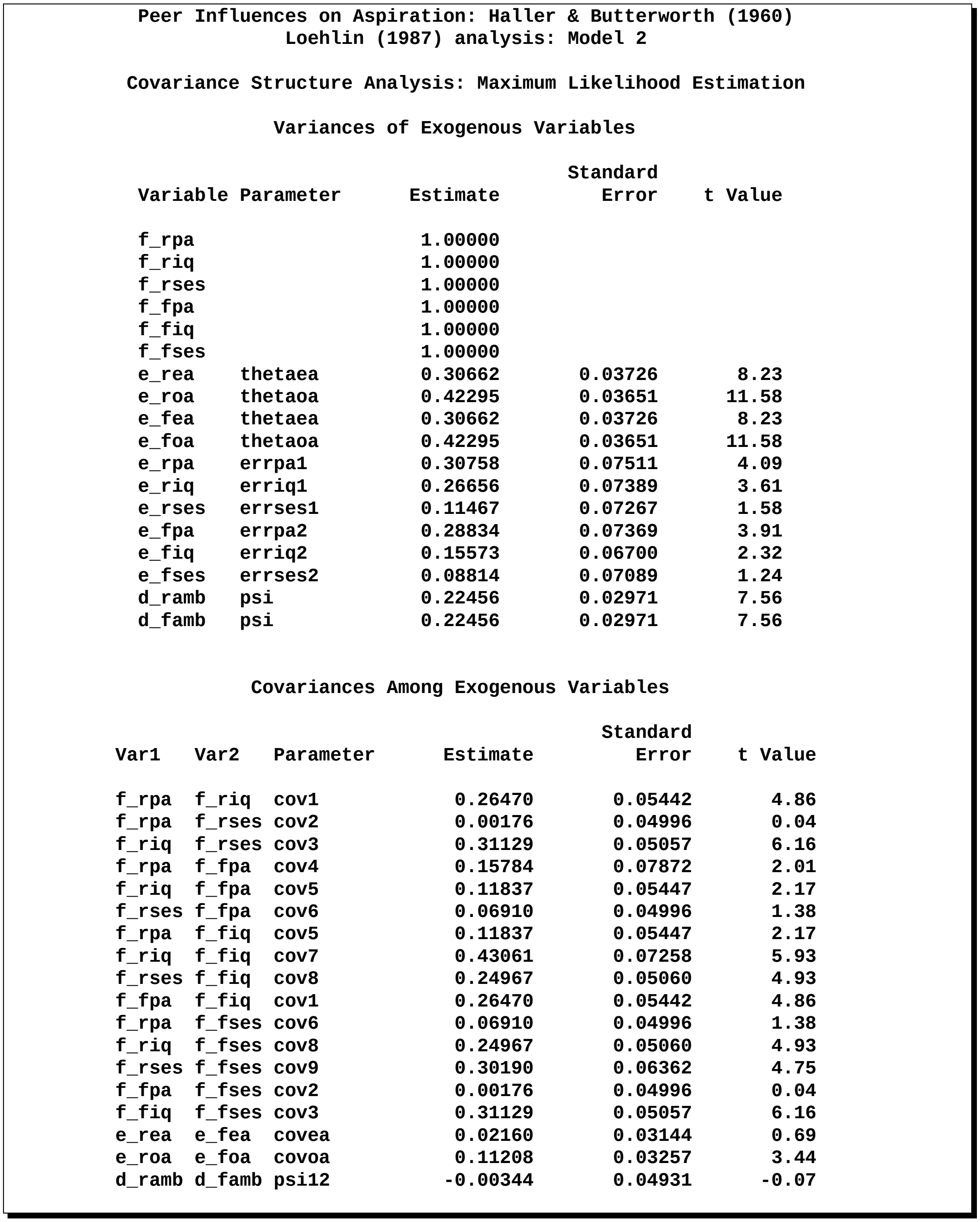
```
 Peer Influences on Aspiration: Haller & Butterworth (1960)
              Loehlin (1987) analysis: Model 2

                    The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function                                          0.0581
Goodness of Fit Index (GFI)                           0.9884
GFI Adjusted for Degrees of Freedom (AGFI)            0.9772
Root Mean Square Residual (RMR)                       0.0276
Parsimonious GFI (Mulaik, 1989)                       0.6150
Chi-Square                                           19.0697
Chi-Square DF                                             28
Pr > Chi-Square                                       0.8960
Independence Model Chi-Square                         872.00
Independence Model Chi-Square DF                          45
RMSEA Estimate                                        0.0000
RMSEA 90% Lower Confidence Limit                           .
RMSEA 90% Upper Confidence Limit                      0.0194
ECVI Estimate                                         0.2285
ECVI 90% Lower Confidence Limit                            .
ECVI 90% Upper Confidence Limit                       0.2664
Probability of Close Fit                              0.9996
Bentler’s Comparative Fit Index                       1.0000
Normal Theory Reweighted LS Chi-Square               19.2372
Akaike’s Information Criterion                      -36.9303
Bozdogan’s (1987) CAIC                             -171.2200
Schwarz’s Bayesian Criterion                       -143.2200
McDonald’s (1989) Centrality                          1.0137
Bentler & Bonett’s (1980) Non-normed Index            1.0174
Bentler & Bonett’s (1980) NFI                         0.9781
James, Mulaik, & Brett (1982) Parsimonious NFI        0.6086
Z-Test of Wilson & Hilferty (1931)                   -1.2599
Bollen (1986) Normed Index Rho1                       0.9649
Bollen (1988) Non-normed Index Delta2                 1.0106
Hoelter’s (1983) Critical N                              713
```

Figure 13.22. Career Aspiration Data: Loehlin Model 2



```
           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 2

          Covariance Structure Analysis: Maximum Likelihood Estimation

                 riq     =   0.8940 f_riq    +  1.0000 e_riq
                 rpa     =   0.8370 f_rpa    +  1.0000 e_rpa
                 rses    =   0.9490 f_rses   +  1.0000 e_rses
                 roa     =   1.0000 f_ramb   +  1.0000 e_roa
                 rea     =   1.1007*f_ramb   +  1.0000 e_rea
                 Std Err     0.0684 lambda
                 t Value    16.0879
                 fiq     =   0.8940 f_fiq    +  1.0000 e_fiq
                 fpa     =   0.8370 f_fpa    +  1.0000 e_fpa
                 fses    =   0.9490 f_fses   +  1.0000 e_fses
                 foa     =   1.0000 f_famb   +  1.0000 e_foa
                 fea     =   1.1007*f_famb   +  1.0000 e_fea
                 Std Err     0.0684 lambda
                 t Value    16.0879



           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 2

          Covariance Structure Analysis: Maximum Likelihood Estimation

       f_ramb  =   0.1158*f_famb   +  0.1758*f_rpa    +  0.3223*f_riq
       Std Err     0.0839 beta        0.0351 gam1        0.0470 gam2
       t Value     1.3801             5.0130             6.8557

            +  0.2227*f_rses   +  0.0756*f_fses   +  1.0000 d_ramb
               0.0363 gam3        0.0375 gam4
               6.1373             2.0170

       f_famb  =   0.1158*f_ramb   +  0.0756*f_rses   +  0.1758*f_fpa
       Std Err     0.0839 beta        0.0375 gam4        0.0351 gam1
       t Value     1.3801             2.0170             5.0130

            +  0.3223*f_fiq    +  0.2227*f_fses   +  1.0000 d_famb
               0.0470 gam2        0.0363 gam3
               6.8557             6.1373
```

Figure 13.22. (continued)



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 2

           Covariance Structure Analysis: Maximum Likelihood Estimation

                        Variances of Exogenous Variables

                                                     Standard
            Variable Parameter      Estimate            Error    t Value

            f_rpa                    1.00000
            f_riq                    1.00000
            f_rses                   1.00000
            f_fpa                    1.00000
            f_fiq                    1.00000
            f_fses                   1.00000
            e_rea    thetaea         0.30662          0.03726       8.23
            e_roa    thetaoa         0.42295          0.03651      11.58
            e_fea    thetaea         0.30662          0.03726       8.23
            e_foa    thetaoa         0.42295          0.03651      11.58
            e_rpa    errpa1          0.30758          0.07511       4.09
            e_riq    erriq1          0.26656          0.07389       3.61
            e_rses   errses1         0.11467          0.07267       1.58
            e_fpa    errpa2          0.28834          0.07369       3.91
            e_fiq    erriq2          0.15573          0.06700       2.32
            e_fses   errses2         0.08814          0.07089       1.24
            d_ramb   psi             0.22456          0.02971       7.56
            d_famb   psi             0.22456          0.02971       7.56


                      Covariances Among Exogenous Variables

                                                        Standard
          Var1   Var2   Parameter      Estimate            Error    t Value

          f_rpa  f_riq  cov1            0.26470          0.05442       4.86
          f_rpa  f_rses cov2            0.00176          0.04996       0.04
          f_riq  f_rses cov3            0.31129          0.05057       6.16
          f_rpa  f_fpa  cov4            0.15784          0.07872       2.01
          f_riq  f_fpa  cov5            0.11837          0.05447       2.17
          f_rses f_fpa  cov6            0.06910          0.04996       1.38
          f_rpa  f_fiq  cov5            0.11837          0.05447       2.17
          f_riq  f_fiq  cov7            0.43061          0.07258       5.93
          f_rses f_fiq  cov8            0.24967          0.05060       4.93
          f_fpa  f_fiq  cov1            0.26470          0.05442       4.86
          f_rpa  f_fses cov6            0.06910          0.04996       1.38
          f_riq  f_fses cov8            0.24967          0.05060       4.93
          f_rses f_fses cov9            0.30190          0.06362       4.75
          f_fpa  f_fses cov2            0.00176          0.04996       0.04
          f_fiq  f_fses cov3            0.31129          0.05057       6.16
          e_rea  e_fea  covea           0.02160          0.03144       0.69
          e_roa  e_foa  covoa           0.11208          0.03257       3.44
          d_ramb d_famb psi12          -0.00344          0.04931      -0.07
```

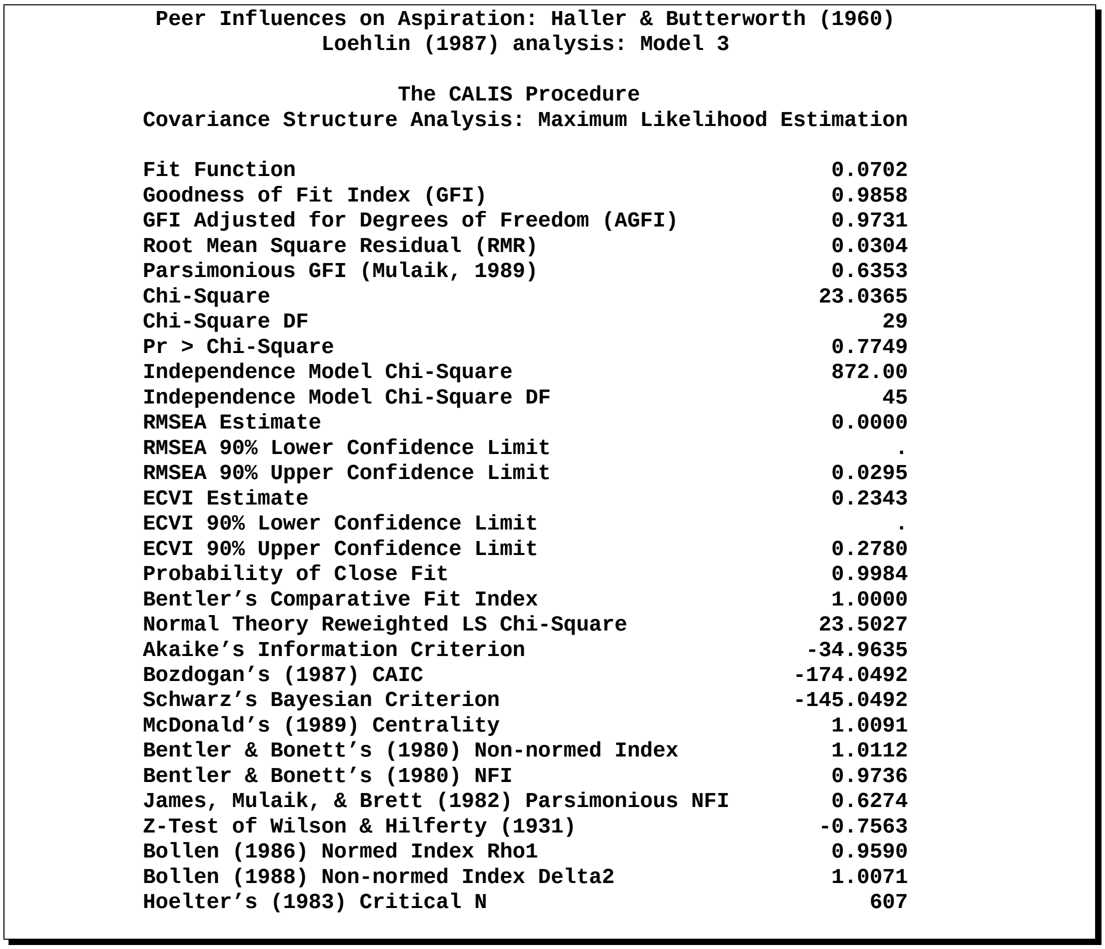
Figure 13.22. (continued)

The test of Loehlin's model 2 against model 1 yields a chi-square of 19.0697 - 12.0132 = 7.0565 with 15 degrees of freedom, which is clearly not significant. Schwarz's Bayesian Criterion (SBC) is also much lower for model 2 (-143.2200) than model 1 (-63.3356). Hence, model 2 seems preferable on both substantive and statistical grounds.

A question of substantive interest is whether the friend's socioeconomic status (SES) has a significant direct influence on a boy's ambition. This can be addressed by omitting the paths from f_fses to f_ramb and from f_rses to f_famb designated by the parameter name gam4, yielding Loehlin's model 3:

```
title2 'Loehlin (1987) analysis: Model 3';
data ram3(type=ram);
   set ram2;
   if _name_='gam4' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;


proc calis data=aspire edf=328 inram=ram3;
run;
```

The output is displayed in Figure 13.23.



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 3

                               The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                            0.0702
           Goodness of Fit Index (GFI)                             0.9858
           GFI Adjusted for Degrees of Freedom (AGFI)              0.9731
           Root Mean Square Residual (RMR)                         0.0304
           Parsimonious GFI (Mulaik, 1989)                         0.6353
           Chi-Square                                             23.0365
           Chi-Square DF                                               29
           Pr > Chi-Square                                         0.7749
           Independence Model Chi-Square                           872.00
           Independence Model Chi-Square DF                            45
           RMSEA Estimate                                          0.0000
           RMSEA 90% Lower Confidence Limit                             .
           RMSEA 90% Upper Confidence Limit                        0.0295
           ECVI Estimate                                           0.2343
           ECVI 90% Lower Confidence Limit                              .
           ECVI 90% Upper Confidence Limit                         0.2780
           Probability of Close Fit                                0.9984
           Bentler's Comparative Fit Index                         1.0000
           Normal Theory Reweighted LS Chi-Square                 23.5027
           Akaike's Information Criterion                        -34.9635
           Bozdogan's (1987) CAIC                               -174.0492
           Schwarz's Bayesian Criterion                         -145.0492
           McDonald's (1989) Centrality                            1.0091
           Bentler & Bonett's (1980) Non-normed Index              1.0112
           Bentler & Bonett's (1980) NFI                           0.9736
           James, Mulaik, & Brett (1982) Parsimonious NFI          0.6274
           Z-Test of Wilson & Hilferty (1931)                     -0.7563
           Bollen (1986) Normed Index Rho1                         0.9590
           Bollen (1988) Non-normed Index Delta2                   1.0071
           Hoelter's (1983) Critical N                                607
```

Figure 13.23. Career Aspiration Data: Loehlin Model 3

The chi-square value for testing model 3 versus model 2 is 23.0365 - 19.0697 = 3.9668 with 1 degree of freedom and a *p*-value of 0.0464. Although the parameter is of marginal significance, the estimate in model 2 (0.0756) is small compared to the other coefficients, and SBC indicates that model 3 is preferable to model 2.

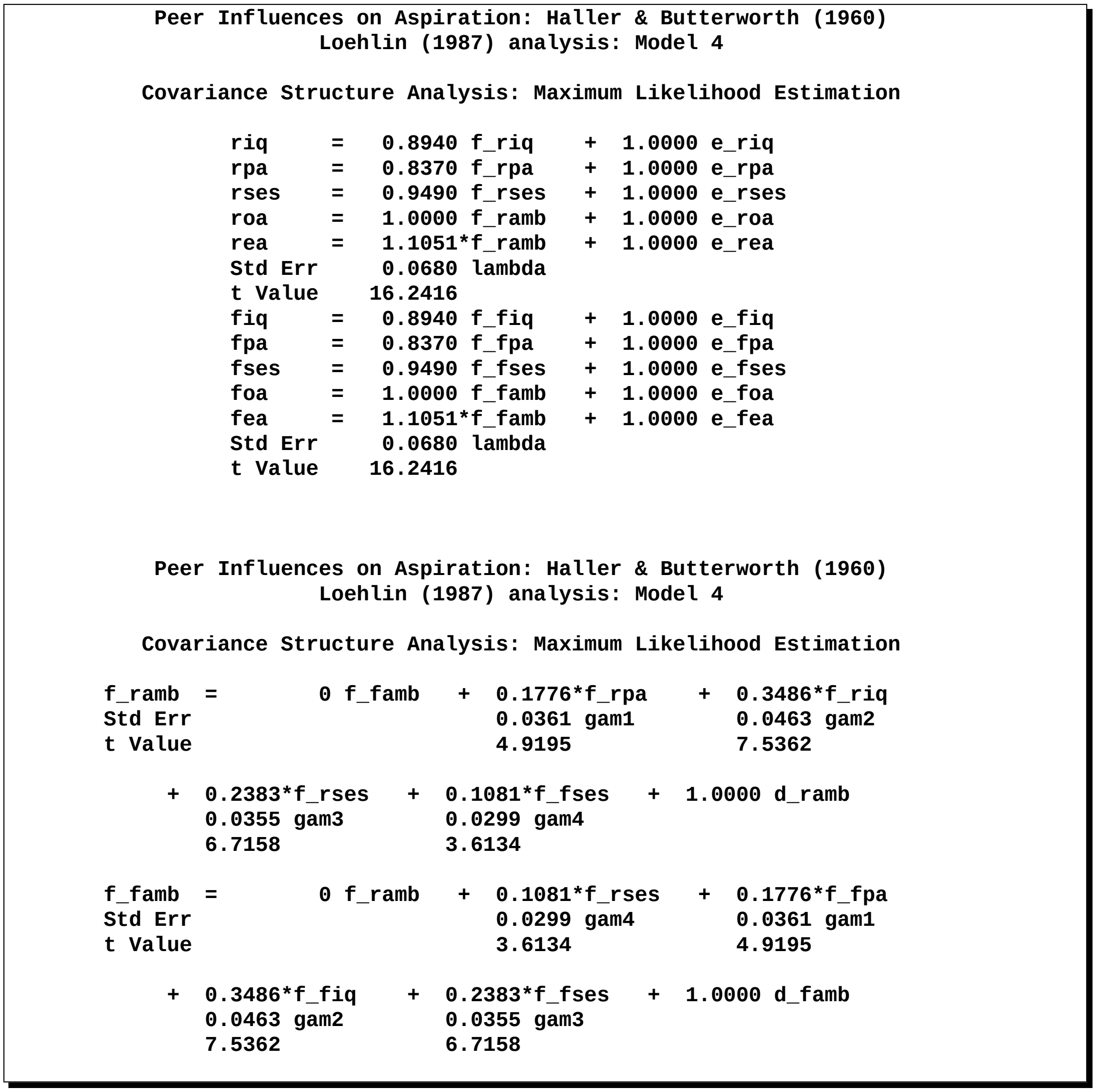
Another important question is whether the reciprocal influences between the respondent's and friend's ambitions are needed in the model. To test whether these paths are zero, set the parameter beta for the paths linking f_ramb and f_famb to zero to obtain Loehlin's model 4:

```
title2 'Loehlin (1987) analysis: Model 4';
data ram4(type=ram);
   set ram2;
   if _name_='beta' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;


proc calis data=aspire edf=328 inram=ram4;
run;
```

The output is displayed in Figure 13.24.

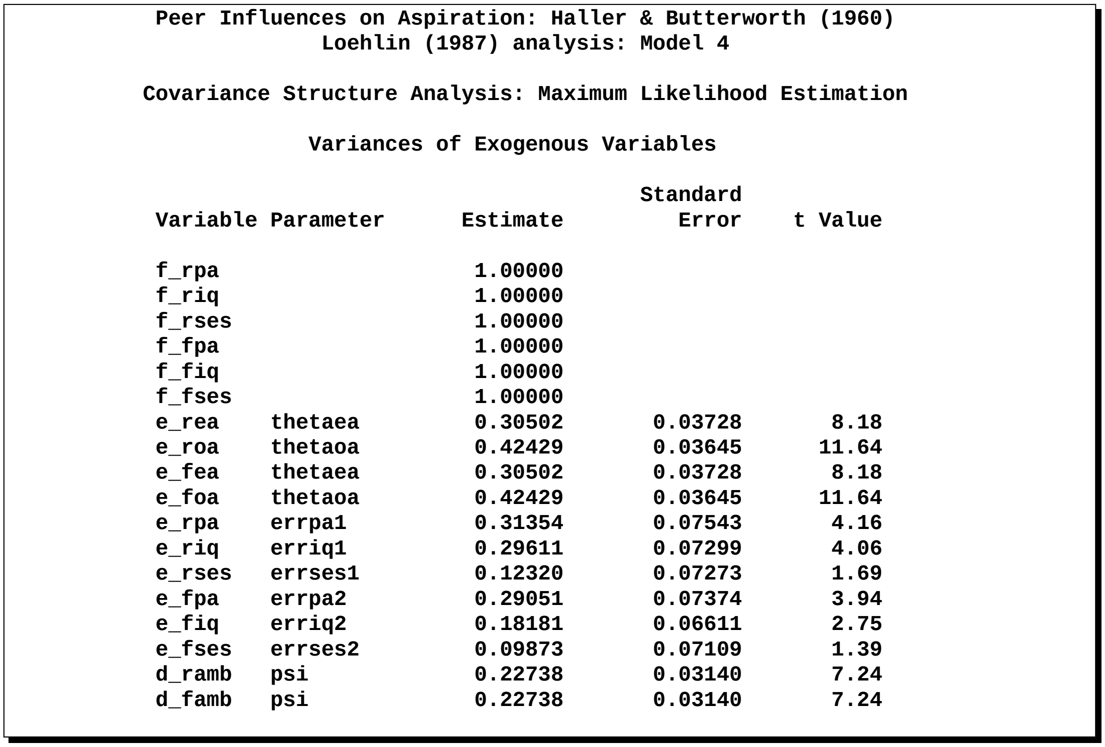
```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 4

                               The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                           0.0640
           Goodness of Fit Index (GFI)                            0.9873
           GFI Adjusted for Degrees of Freedom (AGFI)             0.9760
           Root Mean Square Residual (RMR)                        0.0304
           Parsimonious GFI (Mulaik, 1989)                        0.6363
           Chi-Square                                            20.9981
           Chi-Square DF                                              29
           Pr > Chi-Square                                        0.8592
           Independence Model Chi-Square                          872.00
           Independence Model Chi-Square DF                           45
           RMSEA Estimate                                         0.0000
           RMSEA 90% Lower Confidence Limit                            .
           RMSEA 90% Upper Confidence Limit                       0.0234
           ECVI Estimate                                          0.2281
           ECVI 90% Lower Confidence Limit                             .
           ECVI 90% Upper Confidence Limit                        0.2685
           Probability of Close Fit                               0.9994
           Bentler's Comparative Fit Index                        1.0000
           Normal Theory Reweighted LS Chi-Square                20.8040
           Akaike's Information Criterion                       -37.0019
           Bozdogan's (1987) CAIC                              -176.0876
           Schwarz's Bayesian Criterion                        -147.0876
           McDonald's (1989) Centrality                           1.0122
           Bentler & Bonett's (1980) Non-normed Index             1.0150
           Bentler & Bonett's (1980) NFI                          0.9759
           James, Mulaik, & Brett (1982) Parsimonious NFI         0.6289
           Z-Test of Wilson & Hilferty (1931)                    -1.0780
           Bollen (1986) Normed Index Rho1                        0.9626
           Bollen (1988) Non-normed Index Delta2                  1.0095
           Hoelter's (1983) Critical N                               666
```

Figure 13.24. Career Aspiration Data: Loehlin Model 4



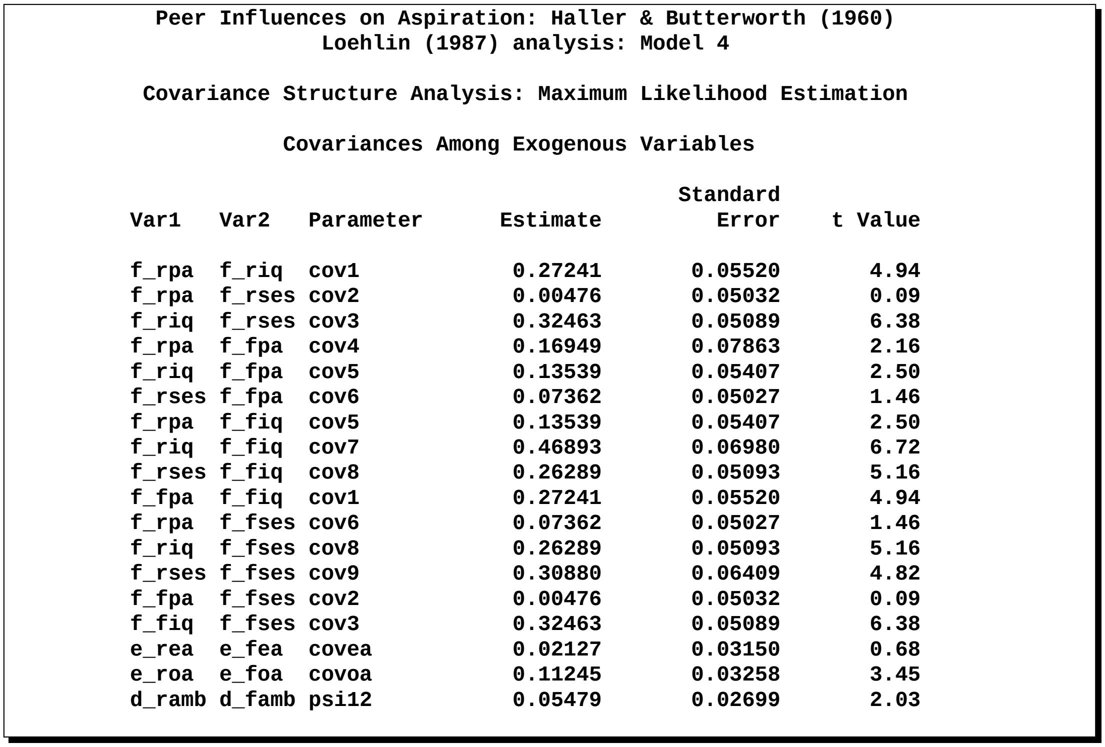
```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 4

           Covariance Structure Analysis: Maximum Likelihood Estimation

                  riq     =   0.8940 f_riq    +  1.0000 e_riq
                  rpa     =   0.8370 f_rpa    +  1.0000 e_rpa
                  rses    =   0.9490 f_rses   +  1.0000 e_rses
                  roa     =   1.0000 f_ramb   +  1.0000 e_roa
                  rea     =   1.1051*f_ramb   +  1.0000 e_rea
                  Std Err     0.0680 lambda
                  t Value    16.2416
                  fiq     =   0.8940 f_fiq    +  1.0000 e_fiq
                  fpa     =   0.8370 f_fpa    +  1.0000 e_fpa
                  fses    =   0.9490 f_fses   +  1.0000 e_fses
                  foa     =   1.0000 f_famb   +  1.0000 e_foa
                  fea     =   1.1051*f_famb   +  1.0000 e_fea
                  Std Err     0.0680 lambda
                  t Value    16.2416



            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 4

           Covariance Structure Analysis: Maximum Likelihood Estimation

        f_ramb  =        0 f_famb   +  0.1776*f_rpa    +  0.3486*f_riq
        Std Err                        0.0361 gam1        0.0463 gam2
        t Value                        4.9195             7.5362

             +  0.2383*f_rses   +  0.1081*f_fses   +  1.0000 d_ramb
                0.0355 gam3        0.0299 gam4
                6.7158             3.6134

        f_famb  =        0 f_ramb   +  0.1081*f_rses   +  0.1776*f_fpa
        Std Err                        0.0299 gam4        0.0361 gam1
        t Value                        3.6134             4.9195

             +  0.3486*f_fiq    +  0.2383*f_fses   +  1.0000 d_famb
                0.0463 gam2        0.0355 gam3
                7.5362             6.7158
```

Figure 13.24. (continued)



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 4

           Covariance Structure Analysis: Maximum Likelihood Estimation

                         Variances of Exogenous Variables

                                                       Standard
            Variable Parameter      Estimate              Error    t Value

            f_rpa                    1.00000
            f_riq                    1.00000
            f_rses                   1.00000
            f_fpa                    1.00000
            f_fiq                    1.00000
            f_fses                   1.00000
            e_rea    thetaea         0.30502            0.03728       8.18
            e_roa    thetaoa         0.42429            0.03645      11.64
            e_fea    thetaea         0.30502            0.03728       8.18
            e_foa    thetaoa         0.42429            0.03645      11.64
            e_rpa    errpa1          0.31354            0.07543       4.16
            e_riq    erriq1          0.29611            0.07299       4.06
            e_rses   errses1         0.12320            0.07273       1.69
            e_fpa    errpa2          0.29051            0.07374       3.94
            e_fiq    erriq2          0.18181            0.06611       2.75
            e_fses   errses2         0.09873            0.07109       1.39
            d_ramb   psi             0.22738            0.03140       7.24
            d_famb   psi             0.22738            0.03140       7.24
```

Figure 13.24. (continued)



```
          Peer Influences on Aspiration: Haller & Butterworth (1960)
                       Loehlin (1987) analysis: Model 4

         Covariance Structure Analysis: Maximum Likelihood Estimation

                    Covariances Among Exogenous Variables

                                                   Standard
    Var1   Var2   Parameter      Estimate            Error    t Value

    f_rpa  f_riq  cov1            0.27241          0.05520       4.94
    f_rpa  f_rses cov2            0.00476          0.05032       0.09
    f_riq  f_rses cov3            0.32463          0.05089       6.38
    f_rpa  f_fpa  cov4            0.16949          0.07863       2.16
    f_riq  f_fpa  cov5            0.13539          0.05407       2.50
    f_rses f_fpa  cov6            0.07362          0.05027       1.46
    f_rpa  f_fiq  cov5            0.13539          0.05407       2.50
    f_riq  f_fiq  cov7            0.46893          0.06980       6.72
    f_rses f_fiq  cov8            0.26289          0.05093       5.16
    f_fpa  f_fiq  cov1            0.27241          0.05520       4.94
    f_rpa  f_fses cov6            0.07362          0.05027       1.46
    f_riq  f_fses cov8            0.26289          0.05093       5.16
    f_rses f_fses cov9            0.30880          0.06409       4.82
    f_fpa  f_fses cov2            0.00476          0.05032       0.09
    f_fiq  f_fses cov3            0.32463          0.05089       6.38
    e_rea  e_fea  covea           0.02127          0.03150       0.68
    e_roa  e_foa  covoa           0.11245          0.03258       3.45
    d_ramb d_famb psi12           0.05479          0.02699       2.03
```

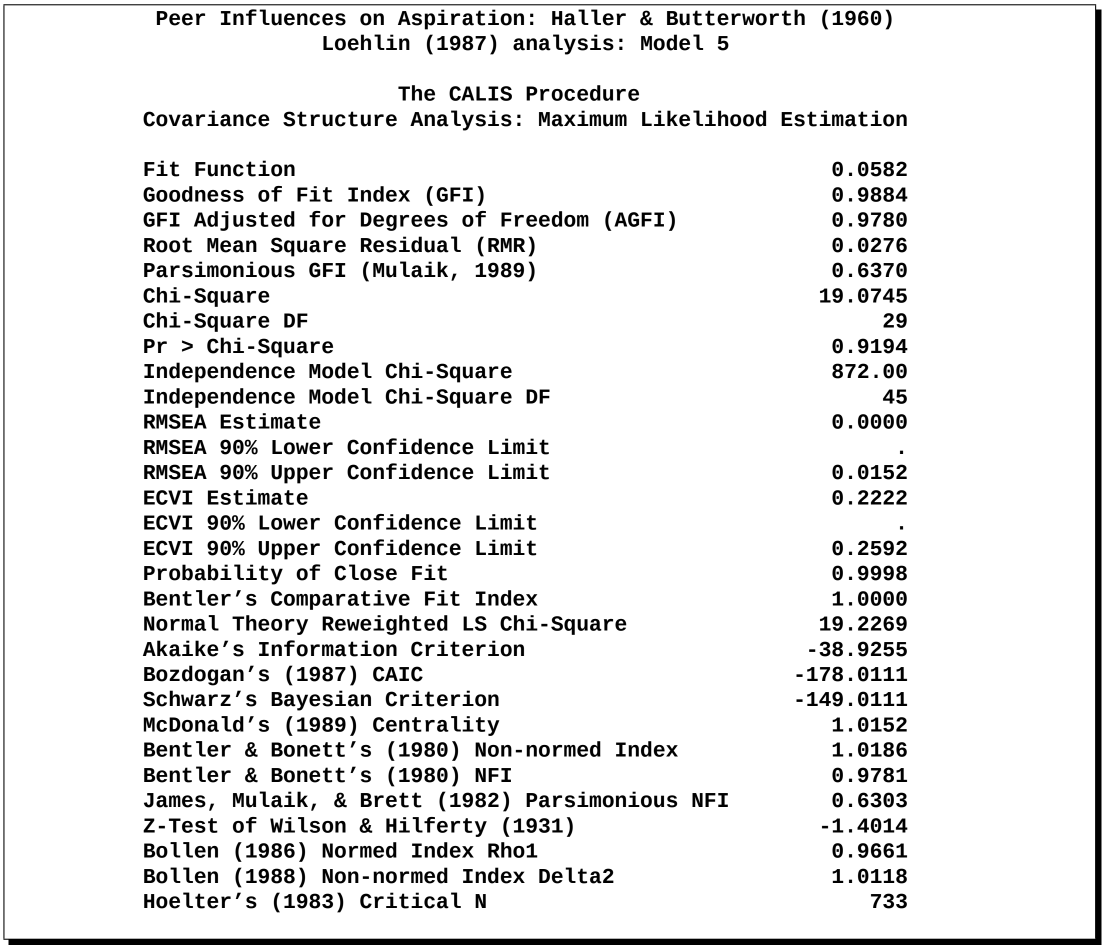
Figure 13.24. (continued)

The chi-square value for testing model 4 versus model 2 is 20.9981 - 19.0697 = 1.9284 with 1 degree of freedom and a *p*-value of 0.1649. Hence, there is little evidence of reciprocal influence.

Loehlin's model 2 has not only the direct paths connecting the latent ambition variables f_ramb and f_famb but also a covariance between the disturbance terms d_ramb and d_famb to allow for other variables omitted from the model that might jointly influence the respondent and his friend. To test the hypothesis that this covariance is zero, set the parameter psi12 to zero, yielding Loehlin's model 5:

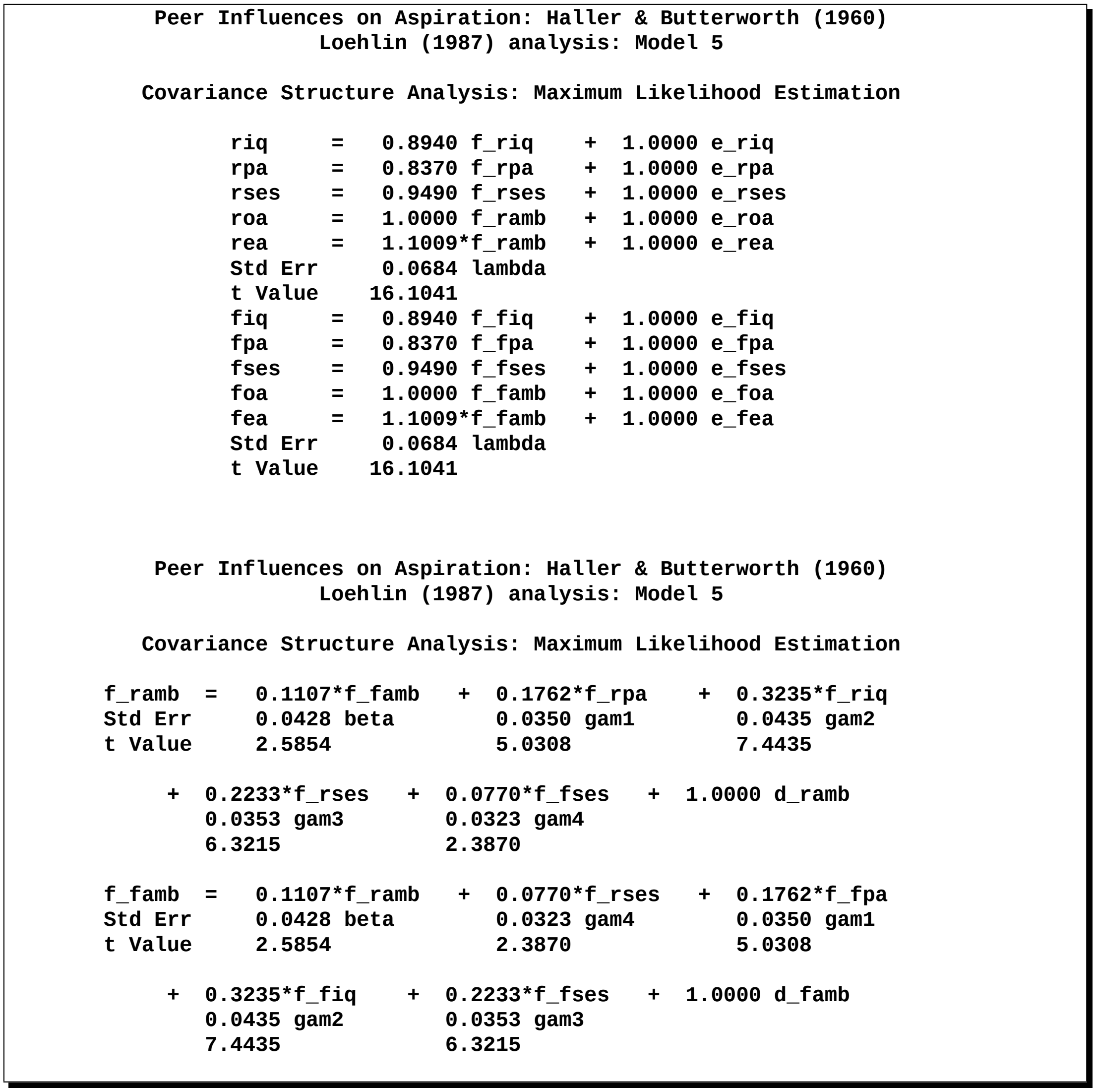
```
title2 'Loehlin (1987) analysis: Model 5';
data ram5(type=ram);
   set ram2;
   if _name_='psi12' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;


proc calis data=aspire edf=328 inram=ram5;
run;
```

The output is displayed in Figure 13.25.



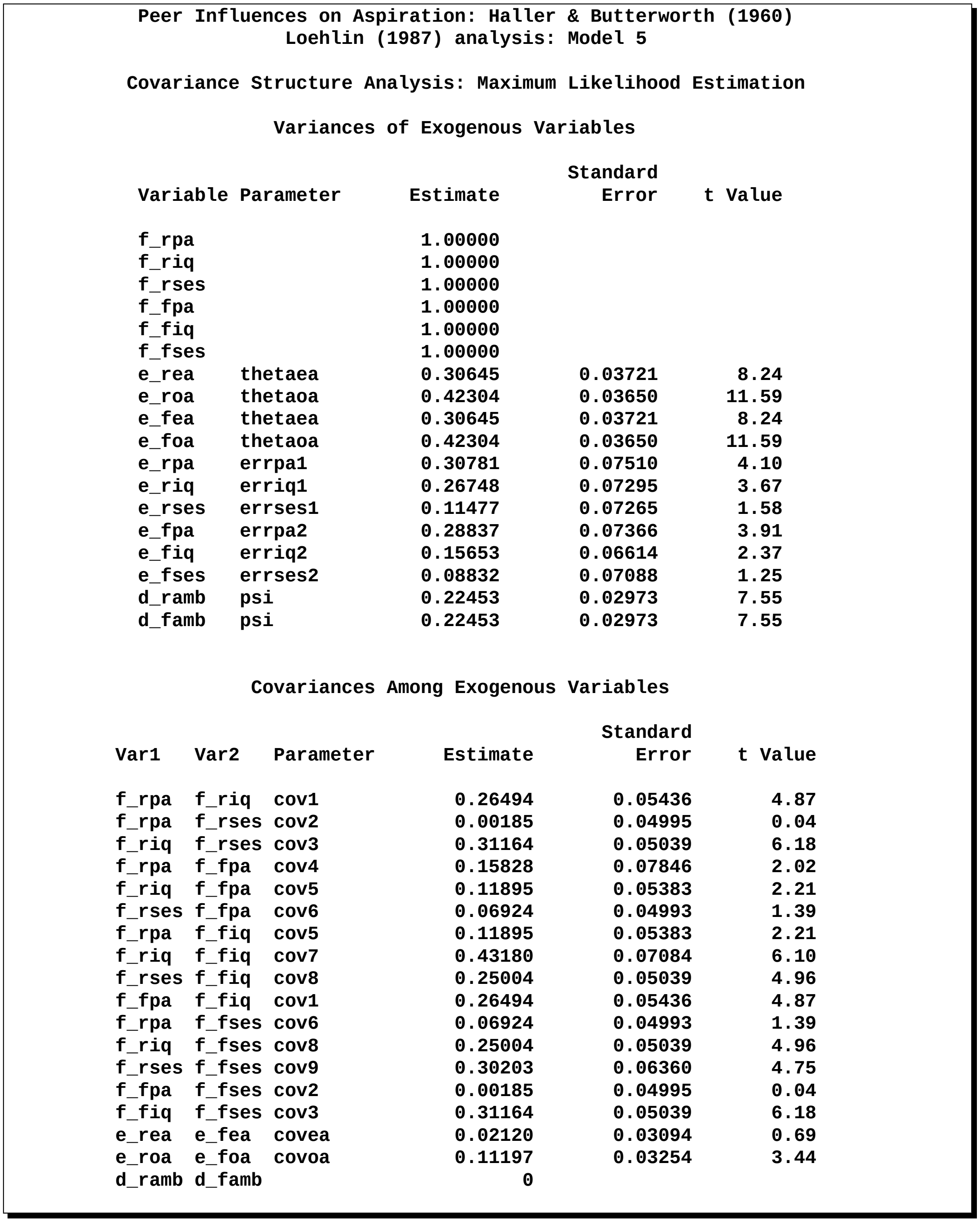
```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 5

                               The CALIS Procedure
           Covariance Structure Analysis: Maximum Likelihood Estimation

           Fit Function                                              0.0582
           Goodness of Fit Index (GFI)                               0.9884
           GFI Adjusted for Degrees of Freedom (AGFI)                0.9780
           Root Mean Square Residual (RMR)                           0.0276
           Parsimonious GFI (Mulaik, 1989)                           0.6370
           Chi-Square                                               19.0745
           Chi-Square DF                                                 29
           Pr > Chi-Square                                           0.9194
           Independence Model Chi-Square                             872.00
           Independence Model Chi-Square DF                              45
           RMSEA Estimate                                            0.0000
           RMSEA 90% Lower Confidence Limit                               .
           RMSEA 90% Upper Confidence Limit                          0.0152
           ECVI Estimate                                             0.2222
           ECVI 90% Lower Confidence Limit                                .
           ECVI 90% Upper Confidence Limit                           0.2592
           Probability of Close Fit                                  0.9998
           Bentler’s Comparative Fit Index                           1.0000
           Normal Theory Reweighted LS Chi-Square                   19.2269
           Akaike’s Information Criterion                          -38.9255
           Bozdogan’s (1987) CAIC                                 -178.0111
           Schwarz’s Bayesian Criterion                           -149.0111
           McDonald’s (1989) Centrality                              1.0152
           Bentler & Bonett’s (1980) Non-normed Index                1.0186
           Bentler & Bonett’s (1980) NFI                             0.9781
           James, Mulaik, & Brett (1982) Parsimonious NFI            0.6303
           Z-Test of Wilson & Hilferty (1931)                       -1.4014
           Bollen (1986) Normed Index Rho1                           0.9661
           Bollen (1988) Non-normed Index Delta2                     1.0118
           Hoelter’s (1983) Critical N                                  733
```

Figure 13.25. Career Aspiration Data: Loehlin Model 5



```
             Peer Influences on Aspiration: Haller & Butterworth (1960)
                          Loehlin (1987) analysis: Model 5

            Covariance Structure Analysis: Maximum Likelihood Estimation

                   riq      =   0.8940 f_riq    +  1.0000 e_riq
                   rpa      =   0.8370 f_rpa    +  1.0000 e_rpa
                   rses     =   0.9490 f_rses   +  1.0000 e_rses
                   roa      =   1.0000 f_ramb   +  1.0000 e_roa
                   rea      =   1.1009*f_ramb   +  1.0000 e_rea
                   Std Err      0.0684 lambda
                   t Value     16.1041
                   fiq      =   0.8940 f_fiq    +  1.0000 e_fiq
                   fpa      =   0.8370 f_fpa    +  1.0000 e_fpa
                   fses     =   0.9490 f_fses   +  1.0000 e_fses
                   foa      =   1.0000 f_famb   +  1.0000 e_foa
                   fea      =   1.1009*f_famb   +  1.0000 e_fea
                   Std Err      0.0684 lambda
                   t Value     16.1041



             Peer Influences on Aspiration: Haller & Butterworth (1960)
                          Loehlin (1987) analysis: Model 5

            Covariance Structure Analysis: Maximum Likelihood Estimation

         f_ramb  =   0.1107*f_famb   +  0.1762*f_rpa    +  0.3235*f_riq
         Std Err     0.0428 beta        0.0350 gam1        0.0435 gam2
         t Value     2.5854             5.0308             7.4435

              +  0.2233*f_rses   +  0.0770*f_fses   +  1.0000 d_ramb
                 0.0353 gam3        0.0323 gam4
                 6.3215             2.3870

         f_famb  =   0.1107*f_ramb   +  0.0770*f_rses   +  0.1762*f_fpa
         Std Err     0.0428 beta        0.0323 gam4        0.0350 gam1
         t Value     2.5854             2.3870             5.0308

              +  0.3235*f_fiq    +  0.2233*f_fses   +  1.0000 d_famb
                 0.0435 gam2        0.0353 gam3
                 7.4435             6.3215
```

Figure 13.25. (continued)



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 5

           Covariance Structure Analysis: Maximum Likelihood Estimation

                        Variances of Exogenous Variables

                                                  Standard
            Variable Parameter      Estimate         Error    t Value

            f_rpa                    1.00000
            f_riq                    1.00000
            f_rses                   1.00000
            f_fpa                    1.00000
            f_fiq                    1.00000
            f_fses                   1.00000
            e_rea    thetaea         0.30645       0.03721       8.24
            e_roa    thetaoa         0.42304       0.03650      11.59
            e_fea    thetaea         0.30645       0.03721       8.24
            e_foa    thetaoa         0.42304       0.03650      11.59
            e_rpa    errpa1          0.30781       0.07510       4.10
            e_riq    erriq1          0.26748       0.07295       3.67
            e_rses   errses1         0.11477       0.07265       1.58
            e_fpa    errpa2          0.28837       0.07366       3.91
            e_fiq    erriq2          0.15653       0.06614       2.37
            e_fses   errses2         0.08832       0.07088       1.25
            d_ramb   psi             0.22453       0.02973       7.55
            d_famb   psi             0.22453       0.02973       7.55


                      Covariances Among Exogenous Variables

                                                     Standard
          Var1   Var2   Parameter      Estimate         Error    t Value

          f_rpa  f_riq  cov1            0.26494       0.05436       4.87
          f_rpa  f_rses cov2            0.00185       0.04995       0.04
          f_riq  f_rses cov3            0.31164       0.05039       6.18
          f_rpa  f_fpa  cov4            0.15828       0.07846       2.02
          f_riq  f_fpa  cov5            0.11895       0.05383       2.21
          f_rses f_fpa  cov6            0.06924       0.04993       1.39
          f_rpa  f_fiq  cov5            0.11895       0.05383       2.21
          f_riq  f_fiq  cov7            0.43180       0.07084       6.10
          f_rses f_fiq  cov8            0.25004       0.05039       4.96
          f_fpa  f_fiq  cov1            0.26494       0.05436       4.87
          f_rpa  f_fses cov6            0.06924       0.04993       1.39
          f_riq  f_fses cov8            0.25004       0.05039       4.96
          f_rses f_fses cov9            0.30203       0.06360       4.75
          f_fpa  f_fses cov2            0.00185       0.04995       0.04
          f_fiq  f_fses cov3            0.31164       0.05039       6.18
          e_rea  e_fea  covea           0.02120       0.03094       0.69
          e_roa  e_foa  covoa           0.11197       0.03254       3.44
          d_ramb d_famb                       0
```

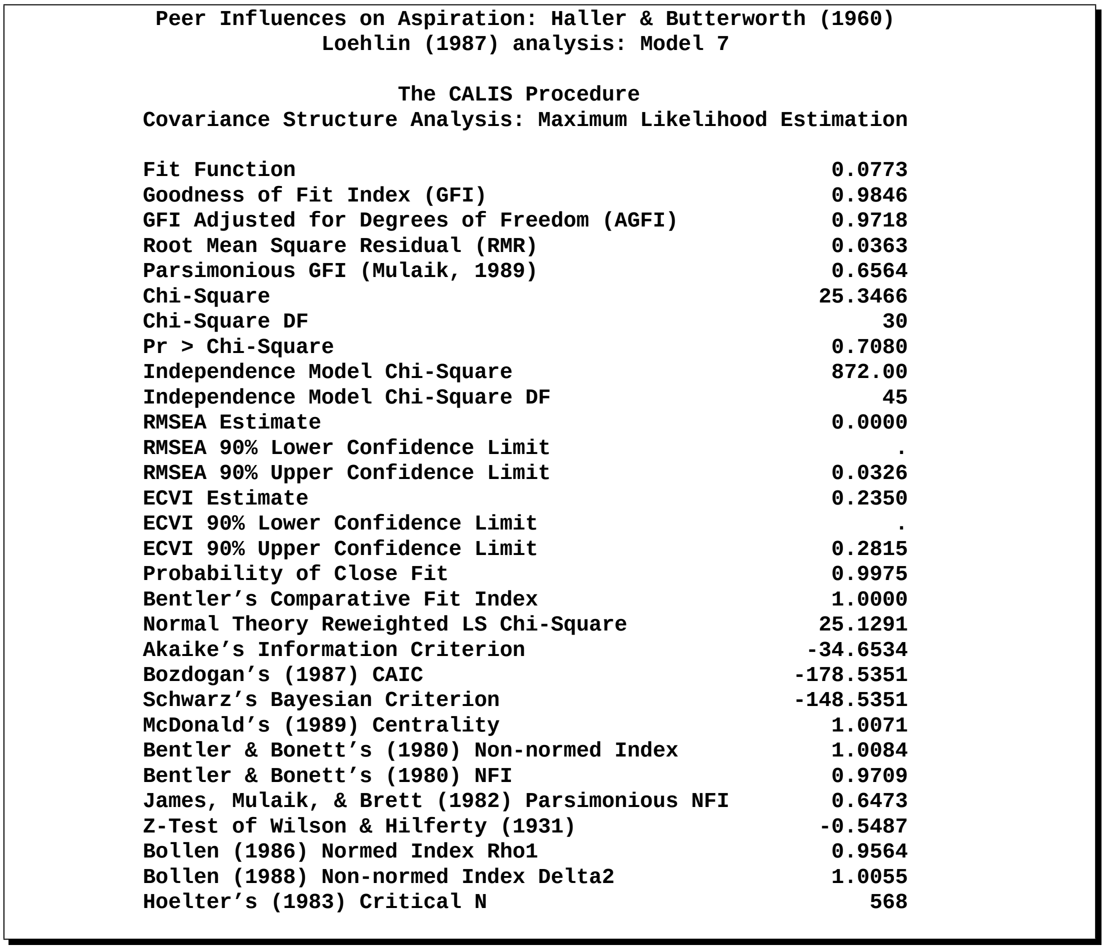
Figure 13.25. (continued)

The chi-square value for testing model 5 versus model 2 is 19.0745 - 19.0697 = 0.0048 with 1 degree of freedom. Omitting the covariance between the disturbance terms, therefore, causes hardly any deterioration in the fit of the model.

These data fail to provide evidence of direct reciprocal influence between the respondent’s and friend’s ambitions or of a covariance between the disturbance terms when these hypotheses are considered separately. Notice, however, that the covariance psi12 between the disturbance terms increases from -0.003344 for model 2 to 0.05479 for model 4. Before you conclude that all of these paths can be omitted from the model, it is important to test both hypotheses together by setting both beta and psi12 to zero as in Loehlin’s model 7:

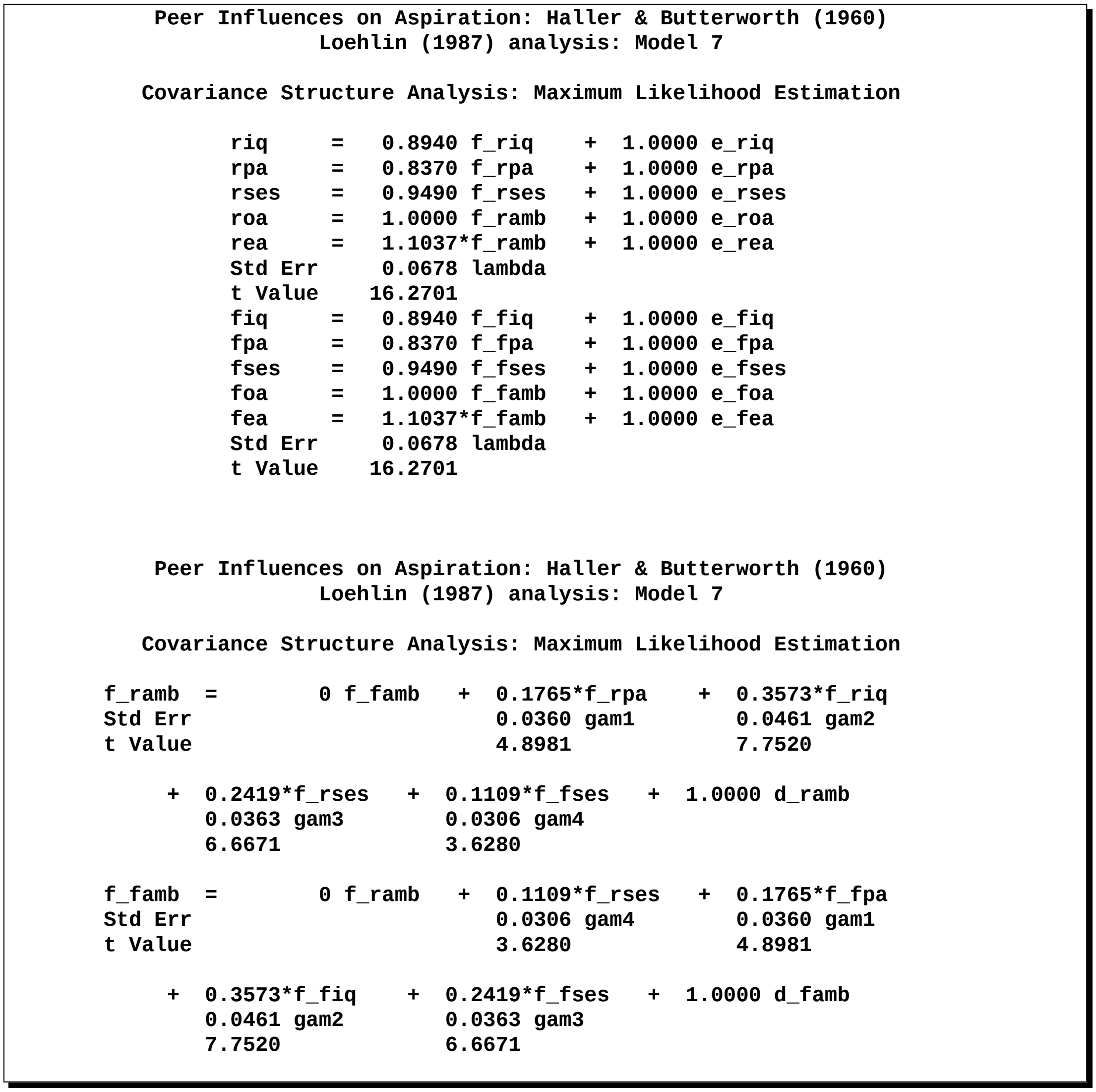
```
title2 'Loehlin (1987) analysis: Model 7';
data ram7(type=ram);
   set ram2;
   if _name_='psi12'|_name_='beta' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;


proc calis data=aspire edf=328 inram=ram7;
run;
```

The relevant output is displayed in Figure 13.26.



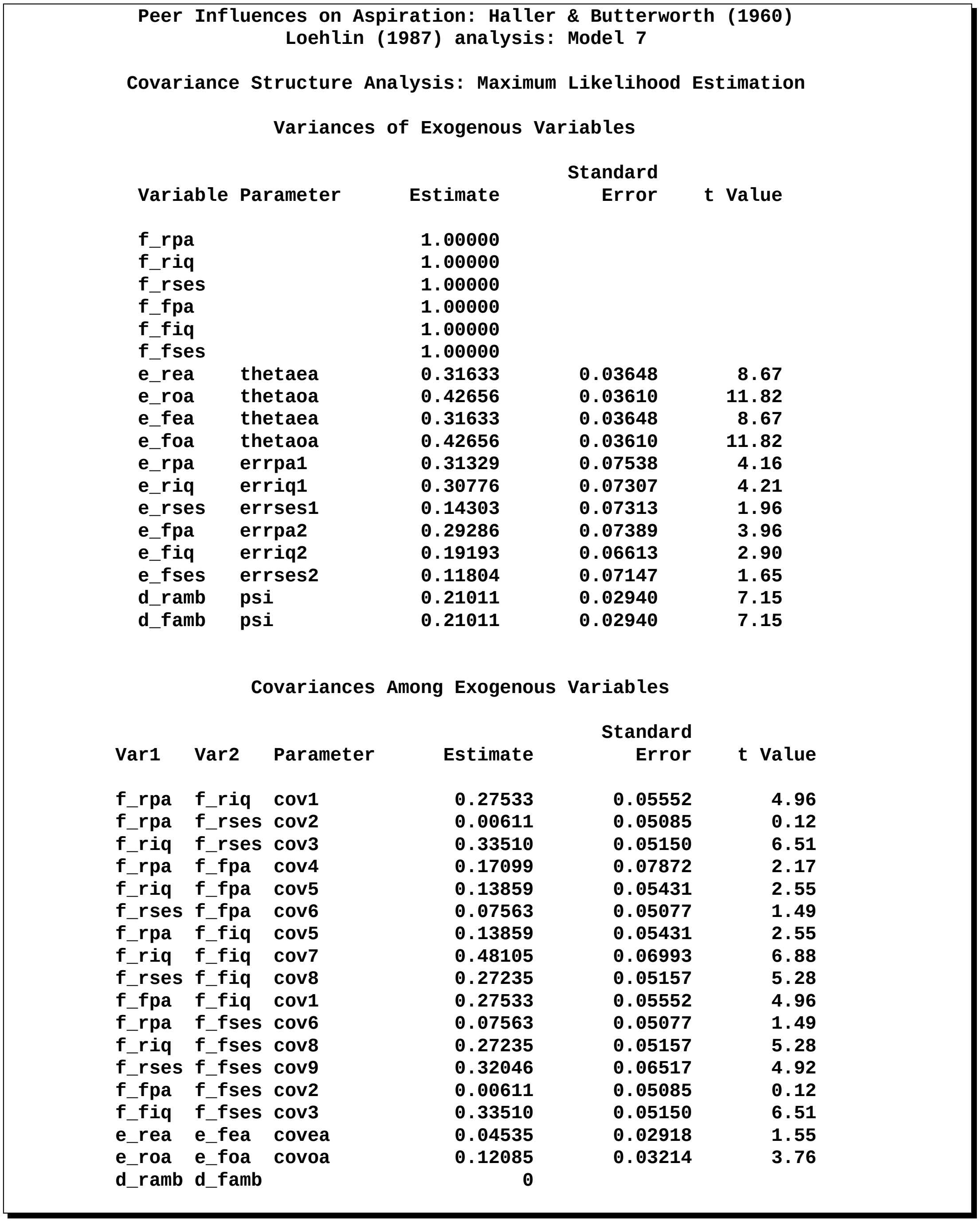
```
 Peer Influences on Aspiration: Haller & Butterworth (1960)
              Loehlin (1987) analysis: Model 7

                    The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function                                          0.0773
Goodness of Fit Index (GFI)                           0.9846
GFI Adjusted for Degrees of Freedom (AGFI)            0.9718
Root Mean Square Residual (RMR)                       0.0363
Parsimonious GFI (Mulaik, 1989)                       0.6564
Chi-Square                                           25.3466
Chi-Square DF                                             30
Pr > Chi-Square                                       0.7080
Independence Model Chi-Square                         872.00
Independence Model Chi-Square DF                          45
RMSEA Estimate                                        0.0000
RMSEA 90% Lower Confidence Limit                           .
RMSEA 90% Upper Confidence Limit                      0.0326
ECVI Estimate                                         0.2350
ECVI 90% Lower Confidence Limit                            .
ECVI 90% Upper Confidence Limit                       0.2815
Probability of Close Fit                              0.9975
Bentler’s Comparative Fit Index                       1.0000
Normal Theory Reweighted LS Chi-Square               25.1291
Akaike’s Information Criterion                      -34.6534
Bozdogan’s (1987) CAIC                             -178.5351
Schwarz’s Bayesian Criterion                       -148.5351
McDonald’s (1989) Centrality                          1.0071
Bentler & Bonett’s (1980) Non-normed Index            1.0084
Bentler & Bonett’s (1980) NFI                         0.9709
James, Mulaik, & Brett (1982) Parsimonious NFI        0.6473
Z-Test of Wilson & Hilferty (1931)                   -0.5487
Bollen (1986) Normed Index Rho1                       0.9564
Bollen (1988) Non-normed Index Delta2                 1.0055
Hoelter’s (1983) Critical N                              568
```

Figure 13.26. Career Aspiration Data: Loehlin Model 7



```
           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 7

          Covariance Structure Analysis: Maximum Likelihood Estimation

                 riq     =   0.8940 f_riq    +  1.0000 e_riq
                 rpa     =   0.8370 f_rpa    +  1.0000 e_rpa
                 rses    =   0.9490 f_rses   +  1.0000 e_rses
                 roa     =   1.0000 f_ramb   +  1.0000 e_roa
                 rea     =   1.1037*f_ramb   +  1.0000 e_rea
                 Std Err     0.0678 lambda
                 t Value    16.2701
                 fiq     =   0.8940 f_fiq    +  1.0000 e_fiq
                 fpa     =   0.8370 f_fpa    +  1.0000 e_fpa
                 fses    =   0.9490 f_fses   +  1.0000 e_fses
                 foa     =   1.0000 f_famb   +  1.0000 e_foa
                 fea     =   1.1037*f_famb   +  1.0000 e_fea
                 Std Err     0.0678 lambda
                 t Value    16.2701



           Peer Influences on Aspiration: Haller & Butterworth (1960)
                        Loehlin (1987) analysis: Model 7

          Covariance Structure Analysis: Maximum Likelihood Estimation

       f_ramb  =        0 f_famb   +  0.1765*f_rpa    +  0.3573*f_riq
       Std Err                        0.0360 gam1        0.0461 gam2
       t Value                        4.8981             7.7520

            +  0.2419*f_rses   +  0.1109*f_fses   +  1.0000 d_ramb
               0.0363 gam3        0.0306 gam4
               6.6671             3.6280

       f_famb  =        0 f_ramb   +  0.1109*f_rses   +  0.1765*f_fpa
       Std Err                        0.0306 gam4        0.0360 gam1
       t Value                        3.6280             4.8981

            +  0.3573*f_fiq    +  0.2419*f_fses   +  1.0000 d_famb
               0.0461 gam2        0.0363 gam3
               7.7520             6.6671
```

Figure 13.26. (continued)



```
            Peer Influences on Aspiration: Haller & Butterworth (1960)
                         Loehlin (1987) analysis: Model 7

           Covariance Structure Analysis: Maximum Likelihood Estimation

                        Variances of Exogenous Variables

                                                        Standard
            Variable Parameter      Estimate               Error    t Value

            f_rpa                    1.00000
            f_riq                    1.00000
            f_rses                   1.00000
            f_fpa                    1.00000
            f_fiq                    1.00000
            f_fses                   1.00000
            e_rea    thetaea         0.31633             0.03648       8.67
            e_roa    thetaoa         0.42656             0.03610      11.82
            e_fea    thetaea         0.31633             0.03648       8.67
            e_foa    thetaoa         0.42656             0.03610      11.82
            e_rpa    errpa1          0.31329             0.07538       4.16
            e_riq    erriq1          0.30776             0.07307       4.21
            e_rses   errses1         0.14303             0.07313       1.96
            e_fpa    errpa2          0.29286             0.07389       3.96
            e_fiq    erriq2          0.19193             0.06613       2.90
            e_fses   errses2         0.11804             0.07147       1.65
            d_ramb   psi             0.21011             0.02940       7.15
            d_famb   psi             0.21011             0.02940       7.15


                      Covariances Among Exogenous Variables

                                                           Standard
          Var1   Var2   Parameter      Estimate               Error    t Value

          f_rpa  f_riq  cov1            0.27533             0.05552       4.96
          f_rpa  f_rses cov2            0.00611             0.05085       0.12
          f_riq  f_rses cov3            0.33510             0.05150       6.51
          f_rpa  f_fpa  cov4            0.17099             0.07872       2.17
          f_riq  f_fpa  cov5            0.13859             0.05431       2.55
          f_rses f_fpa  cov6            0.07563             0.05077       1.49
          f_rpa  f_fiq  cov5            0.13859             0.05431       2.55
          f_riq  f_fiq  cov7            0.48105             0.06993       6.88
          f_rses f_fiq  cov8            0.27235             0.05157       5.28
          f_fpa  f_fiq  cov1            0.27533             0.05552       4.96
          f_rpa  f_fses cov6            0.07563             0.05077       1.49
          f_riq  f_fses cov8            0.27235             0.05157       5.28
          f_rses f_fses cov9            0.32046             0.06517       4.92
          f_fpa  f_fses cov2            0.00611             0.05085       0.12
          f_fiq  f_fses cov3            0.33510             0.05150       6.51
          e_rea  e_fea  covea           0.04535             0.02918       1.55
          e_roa  e_foa  covoa           0.12085             0.03214       3.76
          d_ramb d_famb                       0
```

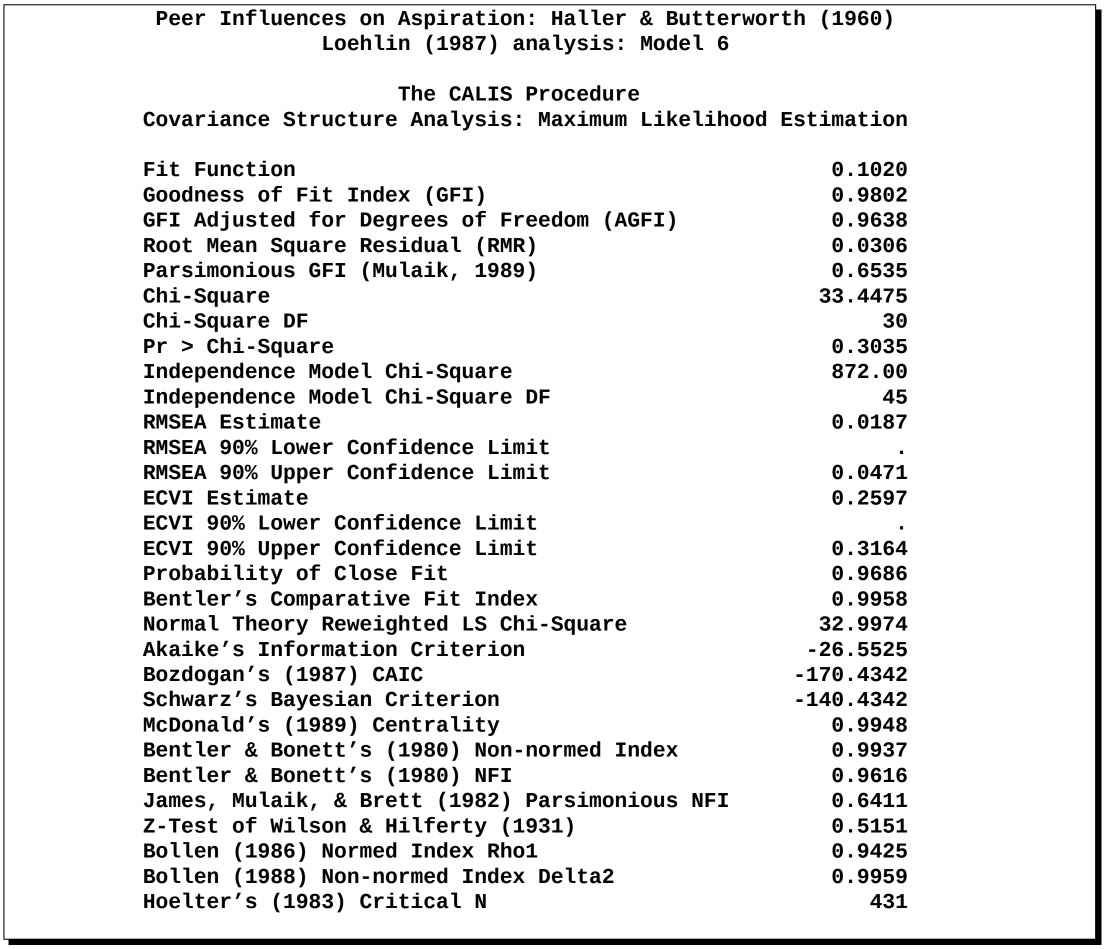
Figure 13.26. (continued)

When model 7 is tested against models 2, 4, and 5, the *p*-values are respectively 0.0433, 0.0370, and 0.0123, indicating that the combined effect of the reciprocal influence and the covariance of the disturbance terms is statistically significant. Thus, the hypothesis tests indicate that it is acceptable to omit either the reciprocal influences or the covariance of the disturbances but not both.

It is also of interest to test the covariances between the error terms for educational (covea) and occupational aspiration (covoa), since these terms are omitted from Jöreskog and Sörbom’s models. Constraining covea and covoa to zero produces Loehlin’s model 6:

```
title2 ’Loehlin (1987) analysis: Model 6’;
data ram6(type=ram);
   set ram2;
   if _name_=’covea’|_name_=’covoa’ then
      do;
         _name_=’ ’;
         _estim_=0;
      end;
run;



proc calis data=aspire edf=328 inram=ram6;
run;
```

The relevant output is displayed in Figure 13.27.



```
          Peer Influences on Aspiration: Haller & Butterworth (1960)
                       Loehlin (1987) analysis: Model 6

                             The CALIS Procedure
         Covariance Structure Analysis: Maximum Likelihood Estimation

         Fit Function                                             0.1020
         Goodness of Fit Index (GFI)                              0.9802
         GFI Adjusted for Degrees of Freedom (AGFI)               0.9638
         Root Mean Square Residual (RMR)                          0.0306
         Parsimonious GFI (Mulaik, 1989)                          0.6535
         Chi-Square                                              33.4475
         Chi-Square DF                                                30
         Pr > Chi-Square                                          0.3035
         Independence Model Chi-Square                            872.00
         Independence Model Chi-Square DF                             45
         RMSEA Estimate                                           0.0187
         RMSEA 90% Lower Confidence Limit                              .
         RMSEA 90% Upper Confidence Limit                         0.0471
         ECVI Estimate                                            0.2597
         ECVI 90% Lower Confidence Limit                               .
         ECVI 90% Upper Confidence Limit                          0.3164
         Probability of Close Fit                                 0.9686
         Bentler's Comparative Fit Index                          0.9958
         Normal Theory Reweighted LS Chi-Square                  32.9974
         Akaike's Information Criterion                         -26.5525
         Bozdogan's (1987) CAIC                                -170.4342
         Schwarz's Bayesian Criterion                          -140.4342
         McDonald's (1989) Centrality                             0.9948
         Bentler & Bonett's (1980) Non-normed Index               0.9937
         Bentler & Bonett's (1980) NFI                            0.9616
         James, Mulaik, & Brett (1982) Parsimonious NFI           0.6411
         Z-Test of Wilson & Hilferty (1931)                       0.5151
         Bollen (1986) Normed Index Rho1                          0.9425
         Bollen (1988) Non-normed Index Delta2                    0.9959
         Hoelter's (1983) Critical N                                 431
```

Figure 13.27. Career Aspiration Data: Loehlin Model 6

The chi-square value for testing model 6 versus model 2 is 33.4476 - 19.0697 = 14.3779 with 2 degrees of freedom and a *p*-value of 0.0008, indicating that there is considerable evidence of correlation between the error terms.

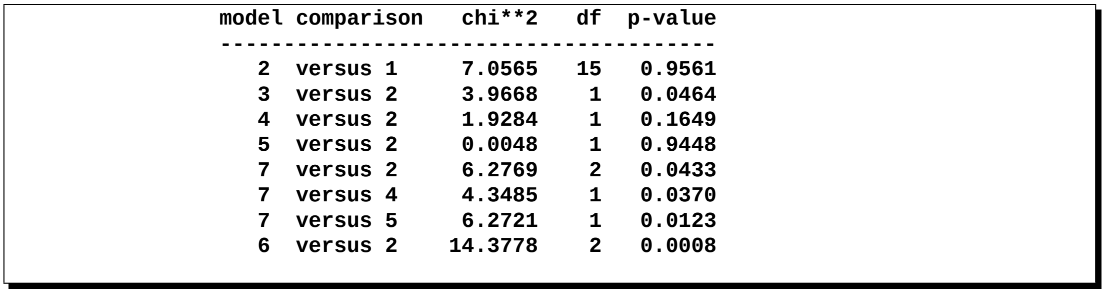
The following table summarizes the results from Loehlin's seven models.

Model	χ^2	**df**	***p*-value**	**SBC**
1. Full model	12.0132	13	0.5266	-63.3356
2. Equality constraints	19.0697	28	0.8960	-143.2200
3. No SES path	23.0365	29	0.7749	-145.0492
4. No reciprocal influence	20.9981	29	0.8592	-147.0876
5. No disturbance correlation	19.0745	29	0.9194	-149.0111
6. No error correlation	33.4475	30	0.3035	-140.4342
7. Constraints from both 4 & 5	25.3466	30	0.7080	-148.5351

For comparing models, you can use a DATA step to compute the differences of the chi-square statistics and *p*-values.

```
data _null_;
   array achisq[7] _temporary_
      (12.0132 19.0697 23.0365 20.9981 19.0745 33.4475 25.3466);
   array adf[7] _temporary_
      (13 28 29 29 29 30 30);
   retain indent 16;
   file print;
   input ho ha @@;
   chisq = achisq[ho] - achisq[ha];
   df = adf[ho] - adf[ha];
   p = 1 - probchi( chisq, df);
   if _n_ = 1 then put
      / +indent ’model comparison   chi**2   df  p-value’
      / +indent ’---------------------------------------’;
   put +indent +3 ho ’ versus ’ ha @18 +indent chisq 8.4 df 5. p 9.4;
datalines;
2 1    3 2    4 2    5 2    7 2    7 4    7 5    6 2
;
```

The DATA step displays the following table in Figure 13.28.



```
model comparison   chi**2   df  p-value
---------------------------------------
   2  versus 1     7.0565   15   0.9561
   3  versus 2     3.9668    1   0.0464
   4  versus 2     1.9284    1   0.1649
   5  versus 2     0.0048    1   0.9448
   7  versus 2     6.2769    2   0.0433
   7  versus 4     4.3485    1   0.0370
   7  versus 5     6.2721    1   0.0123
   6  versus 2    14.3778    2   0.0008
```

Figure 13.28. Career Aspiration Data: Model Comparisons

Although none of the seven models can be rejected when tested against the alternative of an unrestricted covariance matrix, the model comparisons make it clear that there are important differences among the models. Schwarz’s Bayesian Criterion indicates model 5 as the model of choice. The constraints added to model 5 in model 7 can be rejected (*p*=0.0123), while model 5 cannot be rejected when tested against the less-constrained model 2 (*p*=0.9448). Hence, among the small number of models considered, model 5 has strong statistical support. However, as Loehlin (1987, p. 106) points out, many other models for these data could be constructed. Further analysis should consider, in addition to simple modifications of the models, the possibility that more than one friend could influence a boy’s aspirations, and that a boy’s ambition might have some effect on his choice of friends. Pursuing such theories would be statistically challenging.

References

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Bentler, P.M. and Bonett, D.G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bollen, K.A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.

Bollen, K.A. (1989), *Structural Equations with Latent Variables*, New York: John Wiley & Sons, Inc.

Boomsma, A. (1983), *On the Robustness of LISREL (Maximum Likelihood Estimation) against Small Sample Size and Nonnormality*, Amsterdam: Sociometric Research Foundation.

Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in *Testing Structural Equation Models*, ed. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications, Inc.

Duncan, O.D., Haller, A.O., and Portes, A. (1968), "Peer Influences on Aspirations: A Reinterpretation," *American Journal of Sociology*, 74, 119–137.

Fuller, W.A. (1987), *Measurement Error Models*, New York: John Wiley & Sons, Inc.

Haller, A.O., and Butterworth, C.E. (1960), "Peer Influences on Levels of Occupational and Educational Aspiration," *Social Forces*, 38, 289–295.

Hampel F.R., Ronchetti E.M., Rousseeuw P.J., and Stahel W.A. (1986), *Robust Statistics*, New York: John Wiley & Sons, Inc.

Hoelter, J.W. (1983), "The Analysis of Covariance Structures: Goodness-of-Fit Indices," *Sociological Methods and Research*, 11, 325–344.

Huber, P.J. (1981), *Robust Statistics*, New York: John Wiley & Sons, Inc.

James, L.R., Mulaik, S.A., and Brett, J.M. (1982), *Causal Analysis*, Beverly Hills: Sage Publications.

Jöreskog, K.G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, ed. Goldberger, A.S. and Duncan, O.D., New York: Academic Press.

Jöreskog, K.G. and Sörbom, D. (1979), *Advances in Factor Analysis and Structural Equation Models*, Cambridge, MA: Abt Books.

Jöreskog, K.G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago: SPSS.

Keesling, J.W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph.D. dissertation, Chicago: University of Chicago.

Lee, S.Y. (1985), "Analysis of Covariance and Correlation Structures," *Computational Statistics and Data Analysis*, 2, 279–295.

Loehlin, J.C. (1987), *Latent Variable Models*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Lord, F.M. (1957), "A Significance Test for the Hypothesis that Two Variables Measure the Same Trait Except for Errors of Measurement," *Psychometrika*, 22, 207–220.

McArdle, J.J. and McDonald, R.P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R.P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Steiger, J.H. and Lind, J.C. (1980), "Statistically Based Tests for the Number of Common Factors," paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Voss, R.E. (1969), "Response by Corn to NPK Fertilization on Marshall and Monona Soils as Influenced by Management and Meteorological Factors," Ph.D. dissertation, Ames, IA: Iowa State University.

Wiley, D.E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, ed. Goldberger, A.S. and Duncan, O.D., New York: Academic Press.

Wilson, E.B. and Hilferty, M.M. (1931), "The Distribution of Chi-Square," *Proceedings of the National Academy of Sciences*, 17, 694.

Chapter 14
Using the Output Delivery System

Chapter Contents

Chapter 14
Using the Output Delivery System

Overview

In the latest version of SAS software, all SAS/STAT procedures use the Output Delivery System (ODS) to manage their output. This includes managing the form in which the output appears as well as its organization and format. The default for SAS/STAT procedures is to produce the usual SAS listing file. However, by using the features of the Output Delivery System, you can make changes to the format and appearance of your SAS output. For example, you can

- display your output in hypertext markup language (HTML)
- display your output in Rich Text Format (RTF), PDF, or PostScript
- create SAS data sets directly from output tables
- select or exclude individual output objects
- customize the layout, format, and headers of your output

ODS features can provide you with a powerful tool for managing your output. This chapter provides background material and illustrates typical applications of ODS with SAS/STAT software.

For complete documentation on the Output Delivery System, refer to the *SAS Output Delivery System User's Guide*.

The Output Delivery System

Output Objects and ODS Destinations

All SAS procedures produce *output objects* that the Output Delivery System delivers to various *ODS destinations*, according to the default specifications for the procedure or to your own specifications.

All output objects (for example, a table of parameter estimates) consist of two component parts:

- the data component, which consists of the results computed by a SAS procedure
- the template, which contains rules for formatting and displaying the results

When you invoke a SAS procedure, the procedure sends all output to the Output Delivery System. ODS then routes the output to all open destinations. You define the form that the output should take when you specify an ODS destination. Some supported destinations are as follows:

- LISTING destination (the standard SAS listing), which is the default
- HTML destination, for hypertext markup language
- RTF, for inclusion in Microsoft Word
- PDF, PostScript, and PCL, for high-fidelity printers
- OUTPUT destination, SAS data set
- DOCUMENT destination, for modifying and replaying your output without rerunning the procedure that created it

You can activate multiple ODS destinations at the same time, so that a single procedure step can route output to multiple destinations. If you do not supply any ODS statements, ODS delivers all output to the SAS listing, which is the default.

Each output object has an associated template that defines its presentation format. You can modify the presentation of the output by using the TEMPLATE procedure to alter these templates or to create new templates. You can also specify stylistic elements for ODS destinations, such as cell formats and headers, column ordering, colors, and fonts. For detailed information, refer to the chapter titled “The Template Procedure” in the *SAS Output Delivery System User’s Guide*.

Using the Output Delivery System

The ODS statement is a global statement that enables you to provide instructions to the Output Delivery System. You can use ODS statements to specify options for different ODS destinations, select templates to format your output, and select and exclude output. You can also display the names of individual output tables as they are generated.

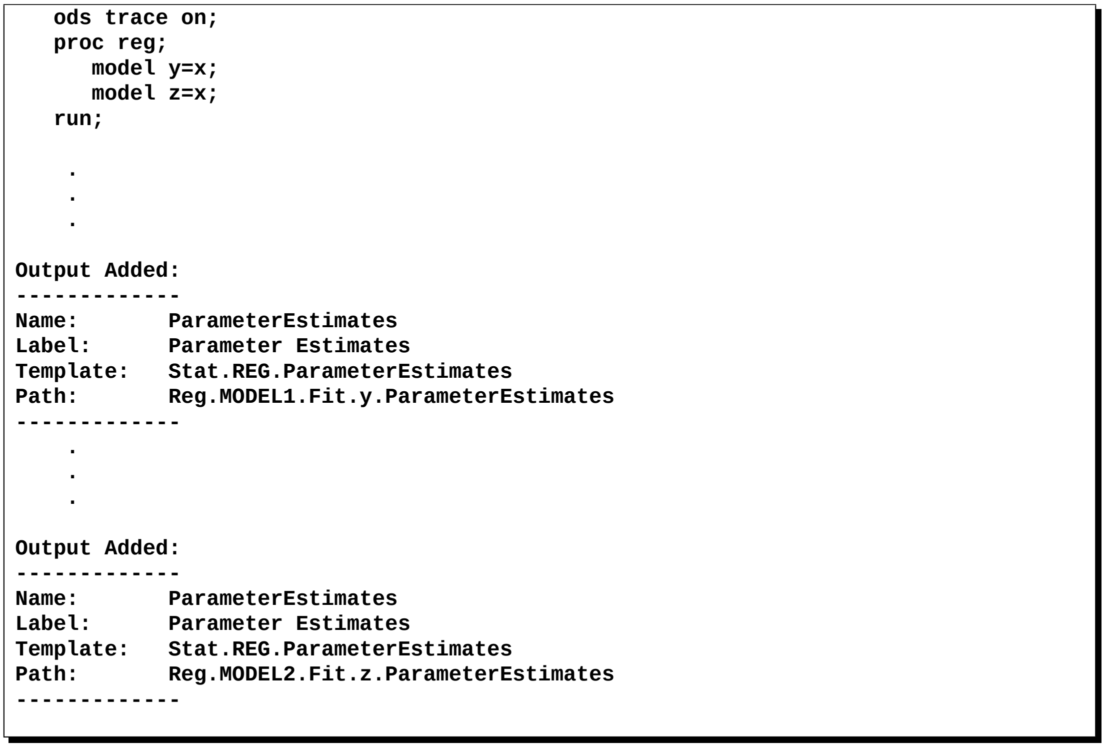
In order to select, exclude, or modify a table, you must first know its name. You can obtain the table names in several ways:

- For any SAS/STAT procedure, you can obtain table names from the individual procedure chapter or from the individual procedure section of the SAS online Help system.
- For any SAS procedure, you can use the SAS Explorer window to view the names of the tables created in your SAS run (see the section “Using ODS with the SAS Explorer” on page 277 for more information).
- For any SAS procedure, you can use the ODS TRACE statement to find the names of tables created in your SAS run. The ODS TRACE statement writes identifying information to the SAS log (or, optionally, to the SAS listing) for each generated output table.

Specify the ODS TRACE ON statement prior to the procedure statements that create the output for which you want information. For example, the following statements write the trace record for the specific tables created in this REG procedure step.

```
ods trace on;
proc reg;
   model y=x;
   model z=x;
run;
ods trace off;
```

By default, the trace record is written to the SAS log, as displayed in Figure 14.1. Alternatively, you can specify the LISTING option, which writes the information, interleaved with the procedure output, to the SAS listing (see Example 14.3).



```
   ods trace on;
   proc reg;
      model y=x;
      model z=x;
   run;

    .
    .
    .

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL1.Fit.y.ParameterEstimates
-------------
    .
    .
    .

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL2.Fit.z.ParameterEstimates
-------------
```

Figure 14.1. Partial Contents of the SAS Log: Result of the ODS TRACE ON Statement

Figure 14.1 displays the trace record, which contains the name of each created table and its associated label, template, and path. The label provides a description of the table. The template name displays the name of the template used to format the table. The path shows the output hierarchy to which the table belongs.

The fully qualified path is given in the trace record. A partially qualified path consists of any part of the full path that begins immediately after a period (`.`) and continues to the end of the full path.

For example, the full path for the parameter estimates for the first model in the preceding regression analysis is

```
Reg.Model1.Fit.y.ParameterEstimates
```

Therefore, partially qualified paths for the table are

```
Model1.fit.y.ParameterEstimates
fit.y.ParameterEstimates
y.ParameterEstimates
```

To refer to a table (in order to select or exclude it from display, for example), specify either the table name or the table's fully or partially qualified path. You may want to use qualified paths when your SAS program creates several tables that have the same name, as in the preceding example. In such a case, you can use a partially qualified path to select a subset of tables, or you can use a fully qualified path to select a particular table.

You specify the tables that ODS selects or excludes with the ODS SELECT or ODS EXCLUDE statement. Suppose that you want to display only the tables of parameter estimates from the preceding regression analysis. You can give any of the following statements (before invoking the REG procedure) to display both tables of parameter estimates. For this example, these statements are equivalent:

```
ods select Reg.Model1.Fit.y.ParameterEstimates
           Reg.Model1.Fit.z.ParameterEstimates;

ods select y.ParameterEstimates z.ParameterEstimates;

ods select ParameterEstimates;
```

The first ODS SELECT statement specifies the full path for both tables. The second statement specifies the partially qualified path for both tables. The third statement specifies the single name "ParameterEstimates," which is shared by both tables.

The Output Delivery System records the specified table names in its internal selection or exclusion list. ODS then processes the output it receives. Note that ODS maintains an overall selection or exclusion list that pertains to all ODS destinations, and it maintains a separate selection or exclusion list for each ODS destination. The list for a specific destination provides the primary filtering step. Restrictions you specify in the overall list are added to the destination-specific lists.

Suppose, for example, that your LISTING exclusion list (that is, the list of tables you want to exclude from the SAS listing) contains the "FitStatistics" table, which you specify with the statement

```
ods listing exclude FitStatistics;
```

Suppose also that your overall selection list (that is, the list of tables you want to select for all destinations) contains the tables “FitStatistics” and “ParameterEstimates,” which you specify with the statement

```
ods select ParameterEstimates FitStatistics;
```

The Output Delivery System then sends only the “ParameterEstimates” and “FitStatistics” tables to all open destinations except the SAS listing. It sends only the “ParameterEstimates” table to the SAS listing because the table “FitStatistics” is excluded from that destination.

Some SAS procedures, such as the REG or the GLM procedure, support run-group processing, which means that a RUN statement does not end the procedure. A QUIT statement explicitly ends such procedures; if you omit the QUIT statement, a PROC or a DATA statement implicitly ends such procedures. When you use the Output Delivery System with procedures that support run-group processing, it is good programming practice to specify a QUIT statement at the end of the procedure. This causes ODS to clear the selection or exclusion list, and you are less likely to encounter unexpected results.

Using ODS with the SAS Explorer

The SAS Explorer enables you to examine the various parts of the SAS System. Figure 14.2 displays the Results window from the SAS Explorer. The Results node retains a running record of your output as it is generated during your SAS session. Figure 14.2 displays the output hierarchy when the preceding statements are executed.

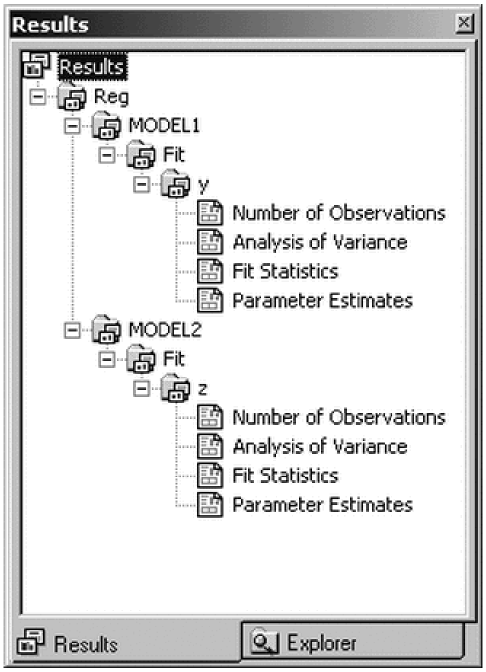


Figure 14.2. The Results Window from the SAS Explorer

When you click on the output table names in the Results window, you link directly to the output in the output window or, if you specify the HTML destination, in an HTML browser. The items on the left-hand side of the Results node are output directories. The items on the right-hand side of the Results node are the names of the actual output objects. You can also use the Explorer to determine names of the templates associated with each output table.

Controlling Output Appearance with Templates

A template is an abstract description of how output should appear when it is formatted. Templates describe several characteristics of the output, including headers, column ordering, style information, justification, and formats. All SAS/STAT procedures have templates, which are stored in the SASHELP library.

You can create or modify a template with the TEMPLATE procedure. For example, you can specify different column headings or different orderings of columns in a table. You can find the template associated with a particular output table or column by using the ODS TRACE statement or the SAS Explorer.

You can display the contents of a template by executing the following statements:

```
proc template;
   source  TemplateName;
run;
```

where *TemplateName* is the name of the template.

Suppose you want to change the way all of the analysis of variance tests are displayed by the GLM procedure. You can redefine the templates that the procedure uses with PROC TEMPLATE. For example, in order to have the “SS” and “MS” columns always displayed with more digits, you can redefine the columns used by the procedure to display them:

```
proc template;
   edit Stat.GLM.SS;
      format=Best16.;
   end;
   edit Stat.GLM.MS;
      format=Best16.;
   end;
run;
```

The BEST*w*. format enables you to display the most information about a value, according to the available field width. The BEST16. format specifies a field width of 16. Refer to the chapter on formats in the *SAS Language Reference: Dictionary* for detailed information.

When you run PROC TEMPLATE to modify or edit a template, the template is stored in your SASUSER library (see Example 14.10). You can then modify the path that ODS uses to look up templates with the ODS PATH statement in order to access these new templates in a later SAS session. This means that you can create a default set of templates to modify the presentation format for all your SAS output. (Note that you can specify the SHOW option in the ODS PATH statement to determine the current path.)

It is important to note the difference between a style template and a column or table template. A column or table template applies to the specific columns or tables that reference the template. For example, the preceding statements that modify the “Stat.GLM.SS” and “Stat.GLM.MS” templates provide an example of modifying specific column templates.

A style template applies to an entire SAS job and can be specified with the STYLE= option in a valid ODS destination, such as HTML, RTF, or PDF. Notice that style definitions do not apply to the LISTING destination, which uses the SAS monospace format. For example, you can specify a style as follows:

```
ods html style=Styles.Brown;
```

A style template controls stylistic elements such as colors, fonts, and presentation attributes. When you use a style template, you ensure that all your output shares a consistent presentation style.

You can also reference style information in table templates for individual headers and data cells. You can modify either type of template with the TEMPLATE procedure. For information on creating your own styles, refer to the *SAS Output Delivery System User’s Guide*.

Interaction between ODS and the NOPRINT Option

Most SAS/STAT procedures support a NOPRINT option that you can use when you want to create an output data set but do not want any displayed output. Typically, you use an OUTPUT statement in addition to the procedure’s NOPRINT option to create a data set and suppress displayed output.

You can also use the Output Delivery System to create output data sets by using the ODS OUTPUT statement. However, if you specify the NOPRINT option, the procedure may not send any output to the Output Delivery System. Therefore, when you want to create output data sets through ODS (using the ODS OUTPUT statement), and you want to suppress the display of all output, specify

```
ods select none;
```

or close the active ODS destinations by entering the command

```
ods  DestinationName close;
```

where *DestinationName* is the name of the active ODS destination (for example, ODS HTML CLOSE).

Note: The ODS statement does not instruct a procedure to generate output: instead, it specifies how the Output Delivery System should manage the table once it is created. You must ensure that the proper options are in effect. For example, the following code does not create the requested data set Parms.

```
proc glm;
   ods output ParameterEstimates=Parms;
   class x;
   model y=x;
   run;
quit;
```

When you execute these statements, the following message is displayed in the log:

```
WARNING: Output 'ParameterEstimates' was not created.
```

The data set Parms is not created because the table of parameter estimates is generated only when the SOLUTION option is specified in the MODEL statement in the GLM procedure.

Compatibility Issues with Version 6 Prototypes

- In Version 6, the MIXED and GENMOD procedures use a prototype of the Output Delivery System. This prototype provides the MAKE statement in order to create data sets from output tables, and this statement remains supported in these procedures. However, the new mechanism to create SAS data sets from output tables is the ODS OUTPUT statement for all procedures.
- The Version 6 prototype of the ODS output hierarchy is stored in a SAS catalog. The latest version of SAS software has a more flexible item-store file type used to store templates and ODS output.
- The Version 6 prototype ODS uses two macro variables (_DISK_ and _PRINT_) to regulate the saving of an output hierarchy. The latest version of SAS software uses the global ODS statement to accomplish this task.
- The Version 6 PROC TEMPLATE and PROC OUTPUT syntax is not compatible with the latest version of SAS software.

Examples

The following examples display typical uses of the Output Delivery System.

Example 14.1. Creating HTML Output with ODS

This example demonstrates how you can use the ODS HTML statement to display your output in hypertext markup language (HTML).

The following statements create the data set scores, which contains the golf scores for boys and girls in a physical education class. The TTEST procedure is then invoked to compare the scores.

The ODS HTML statement specifies the name of the file to contain the HTML output.

```
data scores;
   input Gender $ Score @@;
   datalines;
f 75  f 76  f 80  f 77  f 80  f 77  f 73
m 82  m 80  m 85  m 85  m 78  m 87  m 82
;
run;

ods html body='ttest.htm';

title 'Comparing Group Means';
proc ttest ;
   class Gender;
   var Score;
run;
ods html close;
```

By default, the SAS listing receives all output generated during your SAS run. In this example, the ODS HTML statement opens the HTML destination, and both destinations receive the generated output. Output 14.1.1 displays the results as they are rendered in the SAS listing.

Note that you must specify the following statement before you can view your output in a browser.

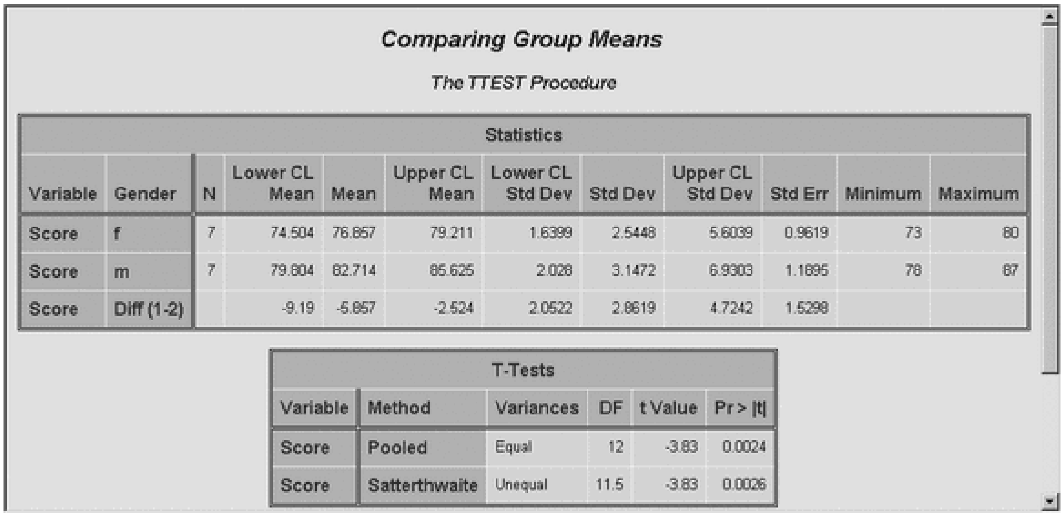
```
ods html close;
```

If you do not close the HTML destination, your HTML file may contain no output, or you may experience other unexpected results. Output 14.1.2 displays the file ttest.htm, which is specified in the preceding ODS HTML statement.

Output 14.1.1. Results for PROC TTEST: SAS Listing Procedure Output

```
                                  Comparing Group Means

                                   The TTEST Procedure

                                       Statistics

                                 Lower CL          Upper CL  Lower CL          Upper CL
Variable  Class            N       Mean    Mean      Mean   Std Dev  Std Dev   Std Dev  Std Err  Minimum  Maximum

Score     f                7     74.504  76.857    79.211    1.6399   2.5448    5.6039   0.9619       73       80
Score     m                7     79.804  82.714    85.625     2.028   3.1472    6.9303   1.1895       78       87
Score     Diff (1-2)              -9.19  -5.857    -2.524    2.0522   2.8619    4.7242   1.5298


                                         T-Tests

                     Variable    Method           Variances      DF    t Value    Pr > |t|

                     Score       Pooled           Equal          12      -3.83      0.0024
                     Score       Satterthwaite    Unequal      11.5      -3.83      0.0026


                                  Equality of Variances

                       Variable    Method      Num DF    Den DF    F Value    Pr > F

                       Score       Folded F         6         6       1.53    0.6189
```

Output 14.1.2. Results for PROC TTEST: HTML Procedure Output



Comparing Group Means

The TTEST Procedure

Statistics											
Variable	Gender	N	Lower CL Mean	Mean	Upper CL Mean	Lower CL Std Dev	Std Dev	Upper CL Std Dev	Std Err	Minimum	Maximum
Score	f	7	74.504	76.857	79.211	1.6399	2.5448	5.6039	0.9619	73	80
Score	m	7	79.804	82.714	85.625	2.028	3.1472	6.9303	1.1895	78	87
Score	Diff (1-2)		-9.19	-5.857	-2.524	2.0522	2.8619	4.7242	1.5298		

T-Tests					
Variable	Method	Variances	DF	t Value	Pr > |t|
Score	Pooled	Equal	12	-3.83	0.0024
Score	Satterthwaite	Unequal	11.5	-3.83	0.0026

Example 14.2. Creating HTML Output with a Table of Contents

The following example uses ODS to display the output in HTML with a table of contents.

The data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14.

```
data pr;
   input Person Gender $ y1 y2 y3 y4;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
   datalines;
 1   F   21.0    20.0    21.5    23.0
 2   F   21.0    21.5    24.0    25.5
 3   F   20.5    24.0    24.5    26.0
 4   F   23.5    24.5    25.0    26.5
 5   F   21.5    23.0    22.5    23.5
 6   F   20.0    21.0    21.0    22.5
 7   F   21.5    22.5    23.0    25.0
 8   F   23.0    23.0    23.5    24.0
 9   F   20.0    21.0    22.0    21.5
10   F   16.5    19.0    19.0    19.5
11   F   24.5    25.0    28.0    28.0
12   M   26.0    25.0    29.0    31.0
13   M   21.5    22.5    23.0    26.5
14   M   23.0    22.5    24.0    27.5
15   M   25.5    27.5    26.5    27.0
16   M   20.0    23.5    22.5    26.0
17   M   24.5    25.5    27.0    28.5
18   M   22.0    22.0    24.5    26.5
19   M   24.0    21.5    24.5    25.5
20   M   23.0    20.5    31.0    26.0
21   M   27.5    28.0    31.0    31.5
22   M   23.0    23.0    23.5    25.0
23   M   21.5    23.5    24.0    28.0
24   M   17.0    24.5    26.0    29.5
25   M   22.5    25.5    25.5    26.0
26   M   23.0    24.5    26.0    30.0
27   M   22.0    21.5    23.5    25.0
run;

ods html body='mixed.htm'
         contents='mixedc.htm'
         frame='mixedf.htm';

proc mixed data=pr method=ml covtest asycov;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=un subject=Person r;
run;
ods html close;
```

The ODS HTML statement specifies three files. The BODY= argument specifies the file to contain the output generated from the statements that follow. The BODY= argument is required.

The CONTENTS= option specifies a file to contain the table of contents. The

FRAME= option specifies a file to contain both the table of contents and the output. You open the FRAME= file in your browser to view the table of contents together with the generated output (see Output 14.2.1). Note that if you specify the ODS HTML statement with only the BODY= argument, no table of contents is created.

The MIXED procedure is invoked to fit the specified model. The resulting output is displayed in Output 14.2.1.

Output 14.2.1. HTML Output from the MIXED Procedure

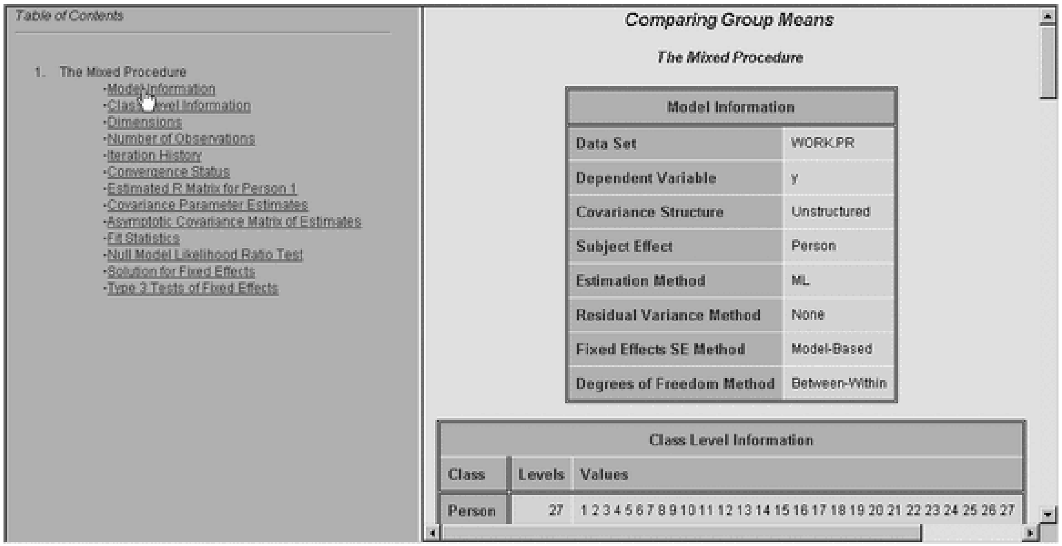


Table of Contents

1. The Mixed Procedure
   - Model Information
   - Class Level Information
   - Dimensions
   - Number of Observations
   - Iteration History
   - Convergence Status
   - Estimated R Matrix for Person 1
   - Covariance Parameter Estimates
   - Asymptotic Covariance Matrix of Estimates
   - Fit Statistics
   - Null Model Likelihood Ratio Test
   - Solution for Fixed Effects
   - Type 3 Tests of Fixed Effects

Comparing Group Means

The Mixed Procedure

Model Information	
Data Set	WORK.PR
Dependent Variable	y
Covariance Structure	Unstructured
Subject Effect	Person
Estimation Method	ML
Residual Variance Method	None
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

Class Level Information		
Class	Levels	Values
Person	27	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27

The table of contents displayed in Output 14.2.1 contains the descriptive label for each output table produced in the MIXED procedure step. You can select any label in the table of contents and the corresponding output will be displayed in the right-hand side of the browser window.

Example 14.3. Determining the Names of ODS Tables

In order to select or exclude a table, or to render it as a SAS data set, you must first know its name. You can obtain the table names in several ways (see the section "Using the Output Delivery System" beginning on page 274 for more information).

This example uses the ODS TRACE statement with the LISTING option to obtain the names of the created output objects. By default, the ODS TRACE statement writes its information to the SAS log. However, you can specify the LISTING option to have the information interleaved with the procedure output in the SAS listing.

Suppose that you perform a randomized trial on rats that have been exposed to a carcinogen. You divide them into two groups and give each group a different treatment. In the following example, interest lies in whether the survival distributions differ between the two treatments. The data set Exposed contains four variables: Days (survival time in days from treatment to death), Status (censoring indicator variable: 0 if censored and 1 if not censored), Treatment (treatment indicator), and Sex (gender: F if female and M if male).

```
data Exposed;
  input Days Status Treatment Sex $ @@;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
255   1   1   M   171   1   1   F
224   0   1   F   325   1   1   M
225   1   1   F   325   1   1   M
287   1   1   M   217   1   1   F
319   1   1   M   255   1   1   F
264   1   1   M   256   1   1   F
237   0   2   F   291   1   2   M
156   1   2   F   323   1   2   M
270   1   2   M   253   1   2   M
257   1   2   M   206   1   2   F
242   1   2   M   206   1   2   F
157   1   2   F   237   1   2   M
249   1   2   M   211   1   2   F
180   1   2   F   229   1   2   F
226   1   2   F   234   1   2   F
268   0   2   M   209   1   2   F
;
ods trace on / listing;

proc lifetest data=Exposed;
   time Days*Status(0);
   strata Treatment;
run;

ods trace off;
```

The purpose of these statements is to obtain the names of the ODS tables produced in this PROC LIFETEST run. The ODS TRACE ON statement writes the trace record of ODS output tables. The LISTING option specifies that the information is interleaved with the output and written to the SAS listing.

The LIFETEST procedure is invoked to perform the analysis, the SAS listing receives the procedure output and the trace record, and the trace is then disabled with the OFF option.

Output 14.3.1. The ODS Trace: Interleaved with LIFETEST Results: Partial Results

```
                                    The LIFETEST Procedure

Output Added:
-------------
Name:       ProductLimitEstimates
Label:      Product-Limit Estimates
Template:   Stat.Lifetest.ProductLimitEstimates
Path:       Lifetest.Stratum1.ProductLimitEstimates
-------------


                                   Stratum 1: Treatment = 1

                                Product-Limit Survival Estimates

                                                  Survival
                                                  Standard     Number      Number
                 Days     Survival    Failure      Error       Failed       Left

                0.000       1.0000           0           0        0          20
              171.000       0.9500      0.0500      0.0487        1          19
              179.000       0.9000      0.1000      0.0671        2          18
              217.000       0.8500      0.1500      0.0798        3          17
              224.000*           .           .           .        3          16
              225.000       0.7969      0.2031      0.0908        4          15
              255.000            .           .           .        5          14
              255.000       0.6906      0.3094      0.1053        6          13
              256.000            .           .           .        7          12
              256.000            .           .           .        8          11
              256.000            .           .           .        9          10
              256.000       0.4781      0.5219      0.1146       10           9
              262.000       0.4250      0.5750      0.1135       11           8
              264.000       0.3719      0.6281      0.1111       12           7
              287.000       0.3188      0.6813      0.1071       13           6
              319.000            .           .           .       14           5
              319.000       0.2125      0.7875      0.0942       15           4
              325.000            .           .           .       16           3
              325.000       0.1063      0.8938      0.0710       17           2
              355.000       0.0531      0.9469      0.0517       18           1
              378.000*           .           .           .       18           0

                  NOTE: The marked survival times are censored observations.


                           Summary Statistics for Time Variable Days

Output Added:
-------------
Name:       Quartiles
Label:      Quartiles
Template:   Stat.Lifetest.Quartiles
Path:       Lifetest.Stratum1.TimeSummary.Quartiles
-------------


                                       Quartile Estimates

                                       Point     95% Confidence Interval
                          Percent    Estimate      [Lower       Upper)

                               75     319.000     262.000     325.000
                               50     256.000     255.000     319.000
                               25     255.000     217.000     256.000
```

As you can see in Output 14.3.1, the ODS TRACE ON statement writes the name, label, template, and path name of each generated ODS table. For more information on names, labels, and qualified path names, see the discussion in the section "Using the Output Delivery System" beginning on page 274.

The information obtained with the ODS TRACE ON statement enables you to request output tables by name. The examples that follow demonstrate how you can use this information to select, exclude, or create data sets from particular output tables.

Example 14.4. Selecting ODS Tables for Display

You can use the ODS SELECT statement to deliver only the desired tables to ODS destinations. In the following example, the GLM procedure is used to perform an analysis on an unbalanced two-way experimental design.

```
data twoway;
title "Unbalanced Two-way Design";
   input Treatment Block y @@;
   datalines;
1 1 17   1 1 28   1 1 19   1 1 21   1 1 19
1 2 43   1 2 30   1 2 39   1 2 44   1 2 44
1 3 16
2 1 21   2 1 21   2 1 24   2 1 25
2 2 39   2 2 45   2 2 42   2 2 47
2 3 19   2 3 22   2 3 16
3 1 22   3 1 30   3 1 33   3 1 31
3 2 46
3 3 26   3 3 31   3 3 26   3 3 33   3 3 29   3 3 25
;

proc glm data=twoway;
   class Treatment Block;
   model y = Treatment | Block;
   means Treatment;
   lsmeans Treatment;

ods select ModelANOVA Means;
ods trace on;
ods show;
run;
```

In the preceding statements, the GLM procedure is invoked to produce the output. The ODS SELECT statement specifies that only the two tables "ModelANOVA" and "Means" are to be delivered to the ODS destinations. In this example, no ODS destinations are explicitly opened. Therefore, only the default SAS listing receives the procedure output. The ODS SHOW statement displays the current overall selection list in the SAS log. The ODS TRACE statement writes the trace record of the ODS output objects to the SAS log.

Output 14.4.1 displays the results of the ODS SHOW statement, which writes the current overall selection list to the SAS log.

Output 14.4.1. Results of the ODS SHOW Statement

```
     ods select ModelANOVA Means;
     ods show;
Current OVERALL select list is:
1. ModelANOVA
2. Means
```

Partial results of the ODS TRACE statement, which are written to the SAS log, are displayed in Output 14.4.2. Note that there are two tables having the name “ModelANOVA,” which are the “Type I Model Anova” and the “Type III Model Anova” tables. Similarly, there are two ODS tables having the name “Means,” which are the “Means” and the “LS-means” tables.

Output 14.4.2. The ODS TRACE: Partial Contents of the SAS Log

```
Output Added:
-------------
Name:       ClassLevels
Label:      Class Levels
Template:   STAT.GLM.ClassLevels
Path:       GLM.Data.ClassLevels
-------------
            .
            .
            .
            .
Output Added:
-------------
Name:       ModelANOVA
Label:      Type I Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.ANOVA.y.ModelANOVA
-------------

Output Added:
-------------
Name:       ModelANOVA
Label:      Type III Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.ANOVA.y.ModelANOVA
-------------
NOTE: Means from the MEANS statement are not adjusted for other
terms in the model.  For adjusted means, use the LSMEANS statement.

Output Added:
-------------
Name:       Means
Label:      Means
Template:   stat.GLM.Means
Path:       GLM.Means.Treatment.Means
-------------

Output Added:
-------------
Name:       Means
Label:      Means
Template:   stat.GLM.LSMeans
Path:       GLM.LSMEANS.Treatment.Means
```

In the following statements, the ODS SHOW statement writes the current overall selection list to the SAS log. The QUIT statement ends the GLM procedure. The second ODS SHOW statement writes the selection list to the log after PROC GLM terminates. The ODS selection list is reset to ALL, by default, when a procedure terminates. For more information on ODS exclusion and selection lists, see the section "Using the Output Delivery System" beginning on page 274.

```
ods show;
quit;
ods show;
```

The results of the statements are displayed in Output 14.4.3. Before the GLM procedure terminates, the ODS selection list includes only the two tables, "ModelANOVA" and "Means."

Output 14.4.3. The ODS Selection List: Before and After PROC GLM Terminates

```
       ods show;

   Current OVERALL select list is:
   1. ModelANOVA
   2. Means


       quit;

NOTE: There were 33 observations read from the dataset WORK.TWOWAY.

       ods show;

   Current OVERALL select list is: ALL
```

The GLM procedure supports interactive run-group processing. Before the QUIT statement is executed, PROC GLM is active and the ODS selection list remains at its previous setting before PROC GLM was invoked. After the QUIT statement, when PROC GLM is no longer active, the selection list is reset to deliver all output tables.

The entire displayed output consists of the four selected tables (two "ModelANOVA" tables and two "Means" tables), as displayed in Output 14.4.4 and Output 14.4.5.

Output 14.4.4. The ModelANOVA Tables from PROC GLM

```
                                  Unbalanced Two-way Design

                                     The GLM Procedure

Dependent Variable: y

        Source                                   DF       Type I SS     Mean Square    F Value    Pr > F

        Treatment                                 2        8.060606        4.030303       0.24    0.7888
        Block                                     2     2621.864124     1310.932062      77.95    <.0001
        Treatment*Block                           4       32.684361        8.171090       0.49    0.7460


        Source                                   DF     Type III SS     Mean Square    F Value    Pr > F

        Treatment                                 2      266.130682      133.065341       7.91    0.0023
        Block                                     2     1883.729465      941.864732      56.00    <.0001
        Treatment*Block                           4       32.684361        8.171090       0.49    0.7460
```

Output 14.4.5. The Means Tables from PROC GLM

```
                          Unbalanced Two-way Design

                              The GLM Procedure

                Level of             --------------y--------------
                Treatment      N            Mean          Std Dev

                1             11      29.0909091       11.5104695
                2             11      29.1818182       11.5569735
                3             11      30.1818182        6.3058414



                          Unbalanced Two-way Design

                              The GLM Procedure
                             Least Squares Means

                          Treatment         y LSMEAN

                          1               25.6000000
                          2               28.3333333
                          3               34.4444444
```

Example 14.5. Excluding ODS Tables from Display

The following example demonstrates how you can use the ODS EXCLUDE statement to exclude particular tables from ODS destinations. This example also creates a SAS data set from the excluded table.

The data are from Hemmerle and Hartley (1973). The response variable consists of measurements from an oven experiment, and the model contains a fixed effect a and random effects b and a*b.

```
data hh;
   input a b y @@;
   datalines;
1 1 237   1 1 254    1 1 246
1 2 178   1 2 179
2 1 208   2 1 178    2 1 187
2 2 146   2 2 145    2 2 141
3 1 186   3 1 183
3 2 142   3 2 125    3 2 136
;
ods html body='mixed.htm'
         contents='mixedc.htm'
         frame='mixedf.htm';

ods exclude ParmSearch(persist);
ods show;
```

The ODS HTML statement specifies the filenames to contain the output generated from the statements that follow.

The ODS EXCLUDE statement excludes the table “ParmSearch” from display. Although the table is excluded from the displayed output, the information contained in the “ParmSearch” table is graphically summarized in a later step.

The PERSIST option in the ODS EXCLUDE statement excludes the table for the entire SAS session or until you execute an ODS SELECT statement or an ODS EXCLUDE NONE statement. If you omit the PERSIST option, the exclusion list is cleared when the procedure terminates. The resulting exclusion list is displayed in Output 14.5.1.

Output 14.5.1. Results of the ODS SHOW Statement: Before PROC MIXED

```
          ods exclude ParmSearch(persist);
          ods show;
    Current OVERALL exclude list is:
    1. ParmSearch(PERSIST)
```

The following ODS OUTPUT statement outputs the table “ParmSearch” to a SAS data set called parms. The MIXED procedure is invoked and the model is fit. All output from the MIXED procedure, except the “ParmSearch” table, is delivered to the HTML destination and the SAS listing. The ODS SHOW statement again displays the overall current exclusion list.

```
ods output ParmSearch=parms;
proc mixed data=hh asycov mmeq mmeqsol covtest;
   class a b;
   model y = a / outp=predicted;
   random b a*b;
   lsmeans a;
   parms (17 to 20 by 0.1) (.3 to .4 by .005) (1.0);
run;

ods show;
```

The results of the ODS SHOW statement, given after the MIXED procedure has terminated, are displayed in Output 14.5.2.

Output 14.5.2. Results of the ODS SHOW Statement: After PROC MIXED

```
   proc mixed data=hh asycov mmeq mmeqsol covtest;
      class a b;
      model y = a / outp=predicted;
      random b a*b;
      lsmeans a;
      parms (17 to 20 by 0.1) (.3 to .4 by .005) (1.0);
   run;

   ods show;

Current OVERALL exclude list is:
1. ParmSearch(PERSIST)
```

Normally the ODS exclusion list is cleared at the conclusion of a procedure (for more information on ODS exclusion and selection lists, see the section “Using the Output Delivery System” on page 274). However, the PERSIST option in the preceding ODS EXCLUDE statement specifies that the “ParmSearch” table should remain in the exclusion list until the list is explicitly cleared (that is, when the ODS EXCLUDE NONE statement or an ODS SELECT statement is encountered). Output 14.5.2 shows that the exclusion list remains in effect after PROC MIXED terminates.

The PERSIST option is useful when you want to exclude the same table in further analyses during your SAS session.

The “ParmSearch” table is contained in the parms data set (as specified in the ODS OUTPUT statement). The information is plotted with the G3D procedure in the following step:

```
proc g3d data=parms;
   plot CovP1*CovP2 = ResLogLike /
              ctop=red cbottom=blue caxis=black;
run;

ods html close;
```

The MIXED procedure output resulting from the preceding statements is displayed in Output 14.5.3. The table of contents shows the names for all of the output tables. The "ParmSearch" table is not listed in the table of contents because of the preceding ODS EXCLUDE statement.

Output 14.5.3. HTML Output from the Mixed Procedure

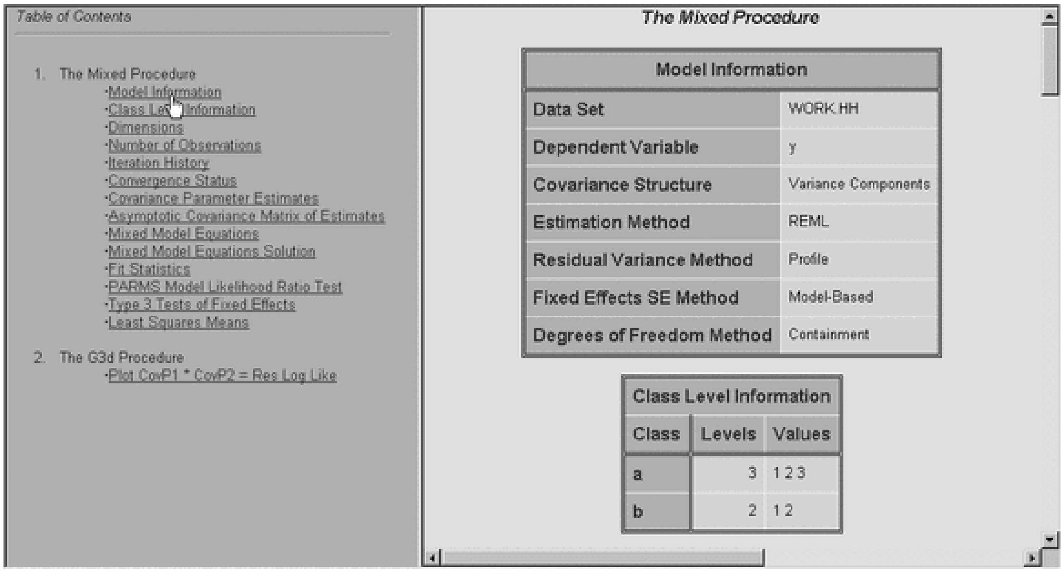


Table of Contents

1. The Mixed Procedure
   - Model Information
   - Class Level Information
   - Dimensions
   - Number of Observations
   - Iteration History
   - Convergence Status
   - Covariance Parameter Estimates
   - Asymptotic Covariance Matrix of Estimates
   - Mixed Model Equations
   - Mixed Model Equations Solution
   - Fit Statistics
   - PARMS Model Likelihood Ratio Test
   - Type 3 Tests of Fixed Effects
   - Least Squares Means
2. The G3d Procedure
   - Plot CovP1 * CovP2 = Res Log Like

The Mixed Procedure

Model Information	
Data Set	WORK.HH
Dependent Variable	y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Class Level Information		
Class	Levels	Values
a	3	1 2 3
b	2	1 2

The results of the G3D procedure is displayed in Output 14.5.4. The large amount of information contained in the table, which is excluded from display, can be summarized with a single plot.

Output 14.5.4. Plot of the ParmSearch Data Set

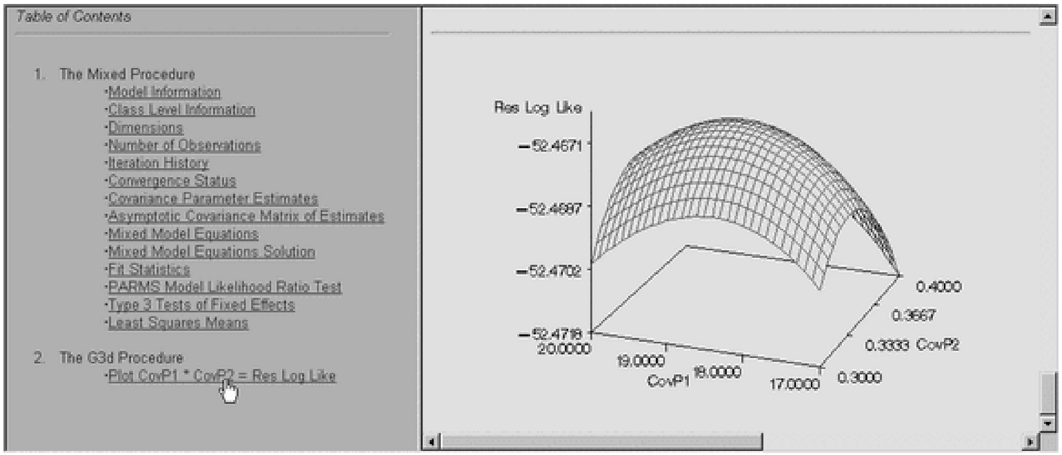


Table of Contents

1. The Mixed Procedure
   - Model Information
   - Class Level Information
   - Dimensions
   - Number of Observations
   - Iteration History
   - Convergence Status
   - Covariance Parameter Estimates
   - Asymptotic Covariance Matrix of Estimates
   - Mixed Model Equations
   - Mixed Model Equations Solution
   - Fit Statistics
   - PARMS Model Likelihood Ratio Test
   - Type 3 Tests of Fixed Effects
   - Least Squares Means
2. The G3d Procedure
   - Plot CovP1 * CovP2 = Res Log Like

Example 14.6. Creating an Output Data Set from an ODS Table

The ODS OUTPUT statement creates SAS data sets from ODS tables. In the following example, the GENMOD procedure is invoked to perform Poisson regression and part of the resulting procedure output is written to a SAS data set.

Suppose the following insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels).

```
data insure;
   input n c car$ age;
   ln = log(n);
   datalines;
500   42  small  1
1200  37  medium 1
100    1  large  1
400  101  small  2
500   73  medium 2
300   14  large  2
;
```

In the data set insure, the variable n represents the number of insurance policyholders and the variable c represents the number of insurance claims. The variable car represents the type of car involved (classified into three groups) and the variable age is the age group of a policyholder (classified into two groups).

In the statements that follow, PROC GENMOD performs a Poisson regression analysis of these data with a log link function. Assume that the number of claims c has a Poisson probability distribution and that its mean, μ_i, is related to the factors car and age.

Determining the Names of the ODS Tables

The purpose of the following statements is to obtain the names of the output tables produced in this PROC GENMOD run. The ODS TRACE statement lists the trace record, and the SAS listing destination is closed so that no output is displayed.

```
ods trace on;
ods listing close;

proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       obstats;
run;
ods trace off;
```

Output 14.6.1. The ODS TRACE: Partial Contents of the SAS Log

```
   ods trace on;
   ods listing close;

   proc genmod data=insure;
      class car age;
      model c = car age / dist   = poisson
                          link   = log
                          offset = ln
                          obstats;
   run;


Output Added:
-------------
Name:       ModelInfo
Label:      Model Information
Template:   Stat.Genmod.ModelInfo
Path:       Genmod.ModelInfo
-------------
         .
         .
         .
         .

NOTE: Algorithm converged.

Output Added:
-------------
Name:       ParameterEstimates
Label:      Analysis Of Parameter Estimates
Template:   stat.genmod.parameterestimates
Path:       Genmod.ParameterEstimates
-------------
NOTE: The scale parameter was held fixed.

Output Added:
-------------
Name:       ObStats
Label:      Observation Statistics
Template:   Stat.Genmod.Obstats
Path:       Genmod.ObStats
-------------
```

By default, the trace record is written to the SAS log, as displayed in Output 14.6.1. Note that you can alternatively specify that the information be interleaved with the procedure output in the SAS listing (see Example 14.3).

Creating the Output Data Set

In the statements that follow, the ODS OUTPUT statement writes the ODS table “ObStats” to a SAS data set called myObStats. All of the usual data set options, such as the KEEP= or RENAME= option, can be used in the ODS OUTPUT statement. Thus, to create the myObStats data set so that it contains only certain variables of the “ObStats” table, you can use the data set options as follows.

```
ods output ObStats=myObStats
              (keep=car age pred
                  rename=(pred=PredictedValue));

proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       obstats;
run;
```

The KEEP= option in the ODS OUTPUT statement specifies that only the variables car, age, and pred are written to the data set, and the pred variable is renamed to PredictedValue. The GENMOD procedure is again invoked. In order to limit the amount of displayed output, the SAS listing destination remains closed. When a destination is closed, it remains closed until it is explicitly reopened.

In the following statements, the output data set myObStats is sorted, and the SAS listing is reopened for output. The results are displayed in Output 14.6.2.

```
proc sort data=myObStats;
   by descending PredictedValue;
run;

ods listing;
proc print data=myObStats noobs;
title 'Values of Car, Age, and the Predicted Values';
run;
```

Output 14.6.2. The ObStats Table Created as a SAS Data Set

```
                  Values of Car, Age, and the Predicted Values

                                             Predicted
                           car       age       Value

                          small       2       107.2011
                          medium      2      67.025444
                          medium      1      42.974556
                          small       1      35.798902
                          large       2      13.773459
                          large       1      1.2265414
```

Example 14.7. Creating an Output Data Set: Subsetting the Data

This example demonstrates how you can create an output data set with the ODS OUTPUT statement and also uses data set selection keywords to limit the output that ODS writes to a SAS data set.

The following data set, called Color, contains the eye and hair color of children from two different regions of Europe. The data are recorded as cell counts, where the variable Count contains the number of children exhibiting each of the 15 eye and hair color combinations.

```
data Color;
   input Region Eyes $ Hair $ Count @@;
      label Eyes  ='Eye Color'
            Hair  ='Hair Color'
            Region='Geographic Region';
datalines;
1 blue  fair   23  1 blue  red     7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red     7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red     5  1 brown medium 41  1 brown dark   40
1 brown black   3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black   6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

In the statements that follow, the SAS listing is closed. The ODS OUTPUT statement creates the “ChiSq” table as a SAS data set called myStats. Note that you can obtain the names of the tables created by any SAS/STAT procedure in the individual procedure chapter or from the individual procedure section of the SAS online Help system. You can also determine the names of tables with the ODS TRACE statement (see Example 14.3 and Example 14.6).

The DROP= data set option excludes variables from the new data set. The WHERE= data set option selects particular observations for output to the new data set myStats, those that begin with 'Chi' or 'Like'.

```
ods listing close;

ods output ChiSq=myStats
           (drop=Table
            where=(Statistic =: 'Chi' or
                   Statistic =: 'Like'));
```

In the following statements, the Color data set is first sorted by the Region variable. The FREQ procedure is invoked to create and analyze a crosstabulation table from the two categorical variables Eyes and Hair, for each value of the variable Region.

No ODS destinations are open until the ODS LISTING statement is encountered just prior to the invocation of the PRINT procedure.

```
proc sort data=Color;
   by Region;
run;
```

```
proc freq data=Color order=data;
   weight Count;
   tables Eyes*Hair / testp=(30 12 30 25 3);
   by Region;
   title 'Hair Color of European Children';
run;

ods listing;
proc print data=myStats;
run;
```

Output 14.7.1 displays the output resulting from the previous statements.

Output 14.7.1. Output Data Set from PROC FREQ and ODS

```
                     Hair Color of European Children

  Obs     Region     Statistic                        DF         Value       Prob

   1         1       Chi-Square                        8       12.6331     0.1251
   2         1       Likelihood Ratio Chi-Square       8       14.1503     0.0779
   3         2       Chi-Square                        8       18.2839     0.0192
   4         2       Likelihood Ratio Chi-Square       8       23.3021     0.0030
```

Example 14.8. RUN Group Processing

This example demonstrates how you can write multiple tables to a single data set using the PERSIST= option in the ODS OUTPUT statement. The PERSIST= option maintains ODS settings across RUN statements for procedures that support run-group processing. In the following analysis, the REG procedure is invoked and the covariance matrix of the estimates is output for two different models.

Consider the following population growth trends. The population of the United States from 1790 to 1970 is fit to linear and quadratic functions of time. Note that the quadratic term, YearSq, is created in the DATA step; this is done since polynomial effects such as Year*Year cannot be specified in the MODEL statement in PROC REG. The data are as follows:

```
title1 'Concatenating Two Tables into One Data Set';
title2 'US Population Study';
data USPopulation;
   input Population @@;
   retain Year 1780;
   Year=Year+10;
   YearSq=Year*Year;
   Population=Population/1000;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
;
```

In the following statements, the REG procedure is invoked and the ODS OUTPUT statement with the PERSIST= option creates a data set with the COVB matrix (the covariance matrix of the estimates).

```
proc reg data=USPopulation;
   ods output covb(persist=run)=Bmatrix;
   var YearSq;
   model Population = Year / covb ;
run;
```

The MODEL statement defines the regression model, and the COVB matrix is requested. The RUN statement executes the REG procedure and the model is fit, producing a covariance matrix of the estimates with two rows and two columns.

Output 14.8.1. Regression Results for the Model Population

```
                     Concatenating Two Output Tables into One Data Set
                                    US Population Study

                                     The REG Procedure
                                      Model: MODEL1
                               Dependent Variable: Population

                                   Analysis of Variance

                                              Sum of           Mean
          Source                   DF        Squares         Square    F Value    Pr > F

          Model                     1          66336          66336     201.87    <.0001
          Error                    17     5586.29253      328.60544
          Corrected Total          18          71923


                       Root MSE             18.12748    R-Square     0.9223
                       Dependent Mean       69.76747    Adj R-Sq     0.9178
                       Coeff Var            25.98271


                                   Parameter Estimates

                                     Parameter       Standard
              Variable      DF        Estimate          Error    t Value    Pr > |t|

              Intercept      1     -1958.36630      142.80455     -13.71      <.0001
              Year           1         1.07879        0.07593      14.21      <.0001
```

Output 14.8.2. CovB Matrix for the Model Population

```
                Concatenating Two Output Tables into One Data Set
                               US Population Study

                                The REG Procedure
                                  Model: MODEL1
                         Dependent Variable: Population

                             Covariance of Estimates

                  Variable          Intercept              Year

                  Intercept      20393.138485      -10.83821461
                  Year           -10.83821461      0.0057650078
```

In the next step, the YearSq variable is added to the model and the model is again fit, producing a covariance matrix of the estimates with three rows and three columns.

```
add YearSq;
print;
run;
```

The results of the regression are displayed in Output 14.8.3.

Output 14.8.3. Regression Results for the Model Population

```
                Concatenating Two Output Tables into One Data Set
                               US Population Study

                                The REG Procedure
                                 Model: MODEL1.1
                         Dependent Variable: Population

                              Analysis of Variance

                                         Sum of           Mean
Source                        DF        Squares         Square    F Value    Pr > F

Model                          2          71799          35900    4641.72    <.0001
Error                         16      123.74557        7.73410
Corrected Total               18          71923


             Root MSE              2.78102    R-Square     0.9983
             Dependent Mean       69.76747    Adj R-Sq     0.9981
             Coeff Var             3.98613


                               Parameter Estimates

                           Parameter       Standard
    Variable      DF        Estimate          Error    t Value    Pr > |t|

    Intercept      1           20450      843.47533      24.25      <.0001
    Year           1       -22.78061        0.89785     -25.37      <.0001
    YearSq         1         0.00635     0.00023877      26.58      <.0001
```

Output 14.8.4. CovB Matrix for the Model Population

```
                Concatenating Two Output Tables into One Data Set
                               US Population Study

                               The REG Procedure
                                Model: MODEL1.1
                         Dependent Variable: Population

                            Covariance of Estimates

         Variable          Intercept              Year            YearSq

         Intercept      711450.62602      -757.2493826      0.2013282694
         Year           -757.2493826      0.8061328943      -0.000214361
         YearSq         0.2013282694      -0.000214361      5.7010894E-8
```

The PERSIST=RUN option maintains the ODS selection list across RUN statements for procedures that support run-group processing. If the PERSIST=RUN option is omitted, the selection list is cleared when the RUN statement is encountered and only the first COVB matrix is selected. Because the PERSIST=RUN option is specified, the selection list remains in effect throughout the REG procedure step. This ensures that each of the COVB matrices is selected and output.

```
proc print;
run;
```

Output 14.8.5. Results of the ODS OUTPUT Statement: Specifying the PERSIST Option

```
                           The COVB Matrix Data Set, Using the PERSIST option
                           Concatenating Two Output Tables into One Data Set

          Obs    _Run_     Model       Dependent     Variable        Intercept            Year          YearSq

            1        1    MODEL1      Population    Intercept    20393.138485    -10.83821461               .
            2        1    MODEL1      Population    Year         -10.83821461    0.0057650078               .
            3        2    MODEL1.1    Population    Intercept    711450.62602    -757.2493826    0.2013282694
            4        2    MODEL1.1    Population    Year         -757.2493826    0.8061328943    -0.000214361
            5        2    MODEL1.1    Population    YearSq       0.2013282694    -0.000214361    5.7010894E-8
```

Note that the two COVB matrices do not have the same variables. In previous versions of SAS, the MATCH_ALL option along with a subsequent DATA step was needed to correctly get all of the variables in one data set. Now, the MATCH_ALL option is only needed if you want to make separate data sets for each table.

Example 14.9. Using the TEMPLATE Procedure to Customize Output

You can use the TEMPLATE procedure to modify the appearance of your displayed ODS tables. The following example, similar to that given in Olinger and Tobias (1998), creates output data sets using the ODS OUTPUT statement, modifies a template using PROC TEMPLATE, and displays the output data sets using the modified template.

The data set comes from a preclinical drug experiment (Cole and Grizzle 1966). In order to study the effect of two different drugs on histamine levels in the blood, researchers administer the drugs to 13 animals, and the levels of histamine in the animals' blood is measured after 0, 1, 3, and 5 minutes. The response variable is the logarithm of the histamine level. The following statements create a SAS data set named Histamine that contains the experimental data.

```
title1 "Histamine Study";
data Histamine;
   input Drug $12. Depleted $ hist0 hist1 hist3 hist5;
   logHist0 = log(hist0); logHist1 = log(Hist1);
   logHist3 = log(hist3); logHist5 = log(Hist5);
   datalines;
Morphine     N  .04  .20  .10  .08
Morphine     N  .02  .06  .02  .02
Morphine     N  .07 1.40  .48  .24
Morphine     N  .17  .57  .35  .24
Morphine     Y  .10  .09  .13  .14
Morphine     Y  .07  .07  .06  .07
Morphine     Y  .05  .07  .06  .07
Trimethaphan N  .03  .62  .31  .22
Trimethaphan N  .03 1.05  .73  .60
Trimethaphan N  .07  .83 1.07  .80
Trimethaphan N  .09 3.13 2.06 1.23
Trimethaphan Y  .10  .09  .09  .08
Trimethaphan Y  .08  .09  .09  .10
Trimethaphan Y  .13  .10  .12  .12
Trimethaphan Y  .06  .05  .05  .05
;
```

In the analysis that follows, the GLM procedure is invoked to perform a repeated measures analysis, naming the drug and depletion status as between-subject factors in the MODEL statement and naming post-administration measurement time as the within-subject factor. For more information on this study and its analysis, refer to Example 32.7 in Chapter 32, "The GLM Procedure."

The following ODS statement requests that two ODS tables be written to SAS data sets called HistWithin and HistBetween. The SAS listing is closed so that no output is displayed. The GLM procedure is invoked and the model is fit.

```
ods output MultStat                         = HistWithin
           BetweenSubjects.ModelANOVA   = HistBetween;

ods listing close;
```

```
proc glm data=Histamine;
   class Drug Depleted;
   model LogHist0--LogHist5 = Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
run;
quit;
```

All of the multivariate test results appear in the HistWithin data set. This is because all multivariate test tables are named “MultStat,” although they occur in different directories in the output directory hierarchy.

Note that, even though there are also other tables named “ModelANOVA,” the preceding ODS OUTPUT statement ensures that only the between-subject ANOVA appears in the HistBetween data set. The specific table is selected because of the additional specification of the partial path (“BetweenSubjects”) in which it occurs. For more information on names and qualified path names, see the discussion in the section “Using the Output Delivery System” beginning on page 274.

In the following statements, a new data set, temp1, is created to contain the two data sets output in the preceding GLM run. They are displayed with no further processing.

```
ods listing;
title2 ’Listing of Raw Data Sets’;
data temp1;
   set HistBetween HistWithin;
run;
proc print;
run;
```

Output 14.9.1. Listing of the Raw Data Sets: Histamine Study

```
                                          Histamine Study
                                      Listing of Raw Data Sets

                            Hypothesis
Obs       Dependent            Type       Source              DF            SS             MS     FValue     ProbF

  1    BetweenSubjects          3      Drug                 1     5.99336243     5.99336243       2.71    0.1281
  2    BetweenSubjects          3      Depleted             1    15.44840703    15.44840703       6.98    0.0229
  3    BetweenSubjects          3      Drug*Depleted        1     4.69087508     4.69087508       2.12    0.1734
  4    BetweenSubjects          3      Error               11    24.34683348     2.21334850        _         _
  5                             .                           .      .              .              24.03    0.0001
  6                             .                           .      .              .              24.03    0.0001
  7                             .                           .      .              .              24.03    0.0001
  8                             .                           .      .              .              24.03    0.0001
  9                             .                           .      .              .               5.78    0.0175
 10                             .                           .      .              .               5.78    0.0175
 11                             .                           .      .              .               5.78    0.0175
 12                             .                           .      .              .               5.78    0.0175
 13                             .                           .      .              .              21.31    0.0002
 14                             .                           .      .              .              21.31    0.0002
 15                             .                           .      .              .              21.31    0.0002
 16                             .                           .      .              .              21.31    0.0002
 17                             .                           .      .              .              12.48    0.0015
 18                             .                           .      .              .              12.48    0.0015
 19                             .                           .      .              .              12.48    0.0015
 20                             .                           .      .              .              12.48    0.0015


Obs    Hypothesis                Error          Statistic                     Value      NumDF      DenDF

  1                                                                        .              .          .
  2                                                                        .              .          .
  3                                                                        .              .          .
  4                                                                        .              .          .
  5    Time                  Error SSCP Matrix    Wilks' Lambda             0.11097706        3          9
  6    Time                  Error SSCP Matrix    Pillai's Trace            0.88902294        3          9
  7    Time                  Error SSCP Matrix    Hotelling-Lawley Trace    8.01087137        3          9
  8    Time                  Error SSCP Matrix    Roy's Greatest Root       8.01087137        3          9
  9    Time_Drug             Error SSCP Matrix    Wilks' Lambda             0.34155984        3          9
 10    Time_Drug             Error SSCP Matrix    Pillai's Trace            0.65844016        3          9
 11    Time_Drug             Error SSCP Matrix    Hotelling-Lawley Trace    1.92774470        3          9
 12    Time_Drug             Error SSCP Matrix    Roy's Greatest Root       1.92774470        3          9
 13    Time_Depleted         Error SSCP Matrix    Wilks' Lambda             0.12339988        3          9
 14    Time_Depleted         Error SSCP Matrix    Pillai's Trace            0.87660012        3          9
 15    Time_Depleted         Error SSCP Matrix    Hotelling-Lawley Trace    7.10373567        3          9
 16    Time_Depleted         Error SSCP Matrix    Roy's Greatest Root       7.10373567        3          9
 17    Time_Drug_Depleted    Error SSCP Matrix    Wilks' Lambda             0.19383010        3          9
 18    Time_Drug_Depleted    Error SSCP Matrix    Pillai's Trace            0.80616990        3          9
 19    Time_Drug_Depleted    Error SSCP Matrix    Hotelling-Lawley Trace    4.15915732        3          9
 20    Time_Drug_Depleted    Error SSCP Matrix    Roy's Greatest Root       4.15915732        3          9
```

In order to reduce the amount of information displayed in Output 14.9.1, this example creates the following data set, HistTests. Only the observations from the raw data sets that are needed for interpretation are included. The variable Hypothesis in the HistWithin data set is renamed to Source, and the NumDF variable is renamed DF.

The renamed variables correspond to the variable names found in the HistBetween data set.

```
data HistTests;
   set HistBetween(where =(Source    ^= "Error"))
       HistWithin (rename=(Hypothesis =  Source NumDF=DF)
                   where =(Statistic  = "Hotelling-Lawley Trace"));
run;
proc print ;
title2 'Listing of Selections from the Raw Data Sets';
run;
```

Output 14.9.2. Listing of Selections from the Raw Data Sets: Histamine Study

```
                          Listing of Selections from the Raw Data Sets

                               Hypothesis
   Obs       Dependent            Type       Source                  DF             SS               MS

    1     BetweenSubjects           3       Drug                     1      5.99336243       5.99336243
    2     BetweenSubjects           3       Depleted                 1     15.44840703      15.44840703
    3     BetweenSubjects           3       Drug*Depleted            1      4.69087508       4.69087508
    4                               .       Time                     3       .                .
    5                               .       Time_Drug                3       .                .
    6                               .       Time_Depleted            3       .                .
    7                               .       Time_Drug_Depleted       3       .                .


   Obs     FValue     ProbF          Error                Statistic                Value     DenDF

    1        2.71    0.1281                                                          .           .
    2        6.98    0.0229                                                          .           .
    3        2.12    0.1734                                                          .           .
    4       24.03    0.0001    Error SSCP Matrix    Hotelling-Lawley Trace      8.01087137       9
    5        5.78    0.0175    Error SSCP Matrix    Hotelling-Lawley Trace      1.92774470       9
    6       21.31    0.0002    Error SSCP Matrix    Hotelling-Lawley Trace      7.10373567       9
    7       12.48    0.0015    Error SSCP Matrix    Hotelling-Lawley Trace      4.15915732       9
```

The amount of information contained in the HistTests is appropriate for interpreting the analysis (Output 14.9.2). However, you can further modify the presentation of the data by applying a template to this combined test data. A template specifies how data should be displayed. The output from previous ODS TRACE ON statements (for example, Output 14.4.2) shows that each table has an associated template as well as a name. In particular, the template associated with PROC GLM’s ANOVA table is called “Stat.GLM.Tests”.

You can use the “Stat.GLM.Tests” template to display the SAS data set HistTests, as follows:

```
data _null_;
title2 ’Listing of the Selections, Using a Standard Template’;
   set HistTests;
   file print ods=(template=’Stat.GLM.Tests’);
   put _ods_;
run;
```

The ODS= option in the FILE statement enables you to use the DATA step to display a data set as a table. You do this by specifying data columns and associated attributes, such as the template specification.

The PUT statement contains the _ODS_ keyword. The keyword instructs the PUT statement to send the data values for all columns (as defined in the ODS= option in the FILE statement) to the open ODS destinations. For more information on using ODS in the DATA step, refer to the *SAS Output Delivery System User’s Guide*.

Output 14.9.3. Listing of the Data Sets Using a Standard Template

```
                                      Histamine Study
                         Listing of the Selections, Using a Standard Template
   Source                        DF              SS     Mean Square    F Value    Pr > F

   Drug                           1      5.99336243      5.99336243       2.71    0.1281
   Depleted                       1     15.44840703     15.44840703       6.98    0.0229
   Drug*Depleted                  1      4.69087508      4.69087508       2.12    0.1734
   Time                           3       .               .              24.03    0.0001
   Time_Drug                      3       .               .               5.78    0.0175
   Time_Depleted                  3       .               .              21.31    0.0002
   Time_Drug_Depleted             3       .               .              12.48    0.0015
```

The data set contains the appropriate information, and it is presented in an easily understandable format, using the “Stat.GLM.Tests” template.

Customizing Your Output

Suppose that you now want to modify the template used to format the ANOVA tables in order to emphasize significant effects. The following statements provide an example of how you can use the TEMPLATE procedure to

- redefine the format for the “SS” and “Mean Square” columns
- include the table title and footnote in the body of the table
- translate the missing values for “SS” and “Mean Square” in the rows corresponding to multivariate tests to asterisks (to refer to the footnote)
- add a column depicting the level of significance

For detailed information on using the TEMPLATE procedure, refer to the chapter titled “The Template Procedure” in the *SAS Output Delivery System User’s Guide*.

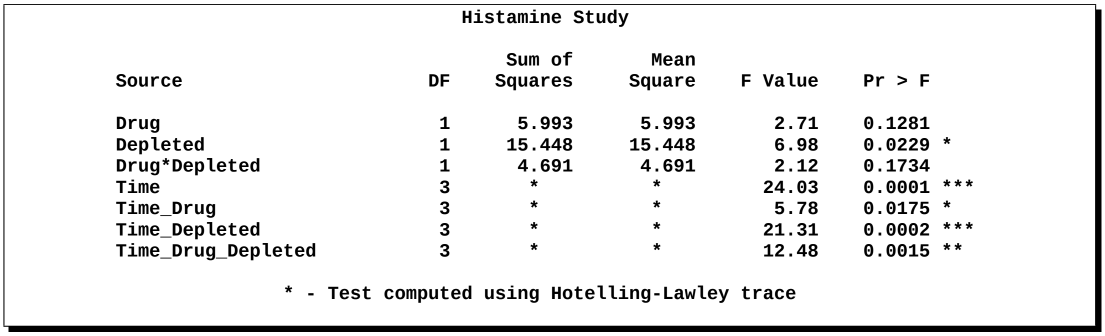
```
proc template;
   define table CombinedTests;
      parent=Stat.GLM.Tests;

      header "#Histamine Study##";
      footer "#* - Test computed using Hotelling-Lawley trace";

      column Source DF SS MS FValue ProbF Star;

      define SS;
         parent = Stat.GLM.SS;
         format = D7.3;
         translate _val_ = . into ’   *’;
      end;
      define MS;
         parent = Stat.GLM.MS;
         format = D7.3;
         translate _val_ = . into ’   *’;
      end;
      define Star;
         compute as ProbF;
         translate _val_ >  0.05  into "",
                   _val_ >  0.01  into "*",
                   _val_ >  0.001 into "**",
                   _val_ <= 0.001 into "***";
         pre_space=1 width=3 just=l;
      end;
   end;
run;
```

The D*w.s* format, used in the preceding statements to redefine the “SS” and “Mean Square” columns, writes numbers in similar ranges with the same number of decimal places. In the format specification, *w* represents the width of the field and *s* represents the number of significant digits. Refer to the chapter on formats in the *SAS Language Reference: Dictionary* for detailed information.

The following statements display the HistTests data set using the customized template. The results are displayed in Output 14.9.4.

```
data _null_;
title2 ’Listing of the Selections, Using a Customized Template’;
   set HistTests;
   file print ods=(template=’CombinedTests’);
   put _ods_;
run;
```

Output 14.9.4. Listing of the Data Sets Using a Customized Template: Histamine Study



```
                                  Histamine Study

                                            Sum of       Mean
        Source                     DF      Squares     Square     F Value     Pr > F

        Drug                        1        5.993      5.993        2.71     0.1281
        Depleted                    1       15.448     15.448        6.98     0.0229 *
        Drug*Depleted               1        4.691      4.691        2.12     0.1734
        Time                        3        *          *           24.03     0.0001 ***
        Time_Drug                   3        *          *            5.78     0.0175 *
        Time_Depleted               3        *          *           21.31     0.0002 ***
        Time_Drug_Depleted          3        *          *           12.48     0.0015 **

                          * - Test computed using Hotelling-Lawley trace
```

Example 14.10. Creating HTML Output, Linked within a Single Analysis

This example demonstrates how you can use ODS to provide links between different parts of your HTML procedure output.

Suppose that you are analyzing a 4×4 factorial experiment for an industrial process, testing for differences in the number of defective products manufactured by different machines using different sources of raw material. The data set Experiment is created as follows.

```
data Experiment;
   do Supplier = 'A','B','C','D';
      do Machine = 1 to 4;
         do rep = 1 to 5;
            input Defects @@;
            output;
            end;
         end;
      end;
   datalines;
 2  6  3  3  6  8  6  6  4  4  4  2  4  0  4  5  5  7  8  5
13 12 12 11 12 16 15 14 14 13 11 10 12 12 10 13 13 14 15 12
 2  6  3  6  6  6  4  4  6  6  0  3  2  0  2  4  6  7  6  4
20 19 18 21 22 22 24 23 20 20 17 19 18 16 17 23 20 20 22 21
;
```

Suppose that you are interested in fitting a model to determine the effect that the supplier of raw material and machine type have on the number of defects in the products. If the *F* test for a factor is significant, you would like to follow up with a multiple comparisons procedure. Thus, the tables of interest are the model ANOVA and the multiple comparisons output.

The following statements demonstrate how you can link a row of the ANOVA table to the corresponding multiple comparisons table. This is done by altering the display of values (inserting links) in the "Source" column of the ANOVA table. The links are inserted by using the TEMPLATE procedure.

```
proc template;
   edit Stat.GLM.Tests;
      edit Source;
         translate _val_ = "Supplier" into
              ('<a href="#IDX6">' || _val_ || '</a>'),
                   _val_ = "Machine"  into
              ('<a href="#IDX8">' || _val_ || '</a>');
         end;
      end;
run;
```

In order to determine the value to use in the HTML anchor link (<A HREF="# ">), you can run the analysis once and view information on your output in the Results node of the SAS Explorer. The anchor name IDX6 is given to the table "ANOVA.Means.Supplier.Defects.MCLines.Tukey.MCLines" (the anchor name is automatically generated in the SAS run). The statements create the Supplier label as a link that, when clicked, opens the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Supplier variable.

The IDX8 anchor name is given to the table "ANOVA.Means.Machine.Defects.MCLines.Tukey.MCLines." The statements create the Machine label as a link that, when clicked, opens the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Machine variable.

The following statements specify that ODS close the SAS listing destination and open the HTML destination. ODS writes the HTML output to the file anovab.htm.

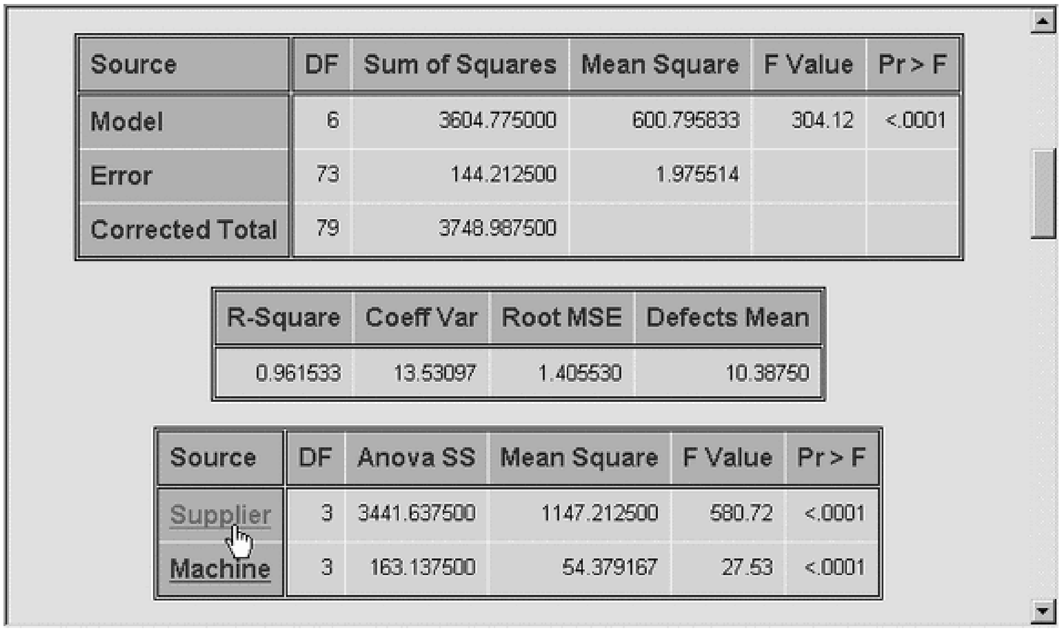
```
ods listing close;
ods html body='anovab.htm';
```

Since this is a balanced experiment, the ANOVA procedure computes the appropriate analysis, performed with the following statements:

```
proc anova data=Experiment;
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
quit;

ods html close;
```

The output from the ANOVA procedure is displayed in Output 14.10.1.

Output 14.10.1. HTML Output from the ANOVA Procedure: Linked Output



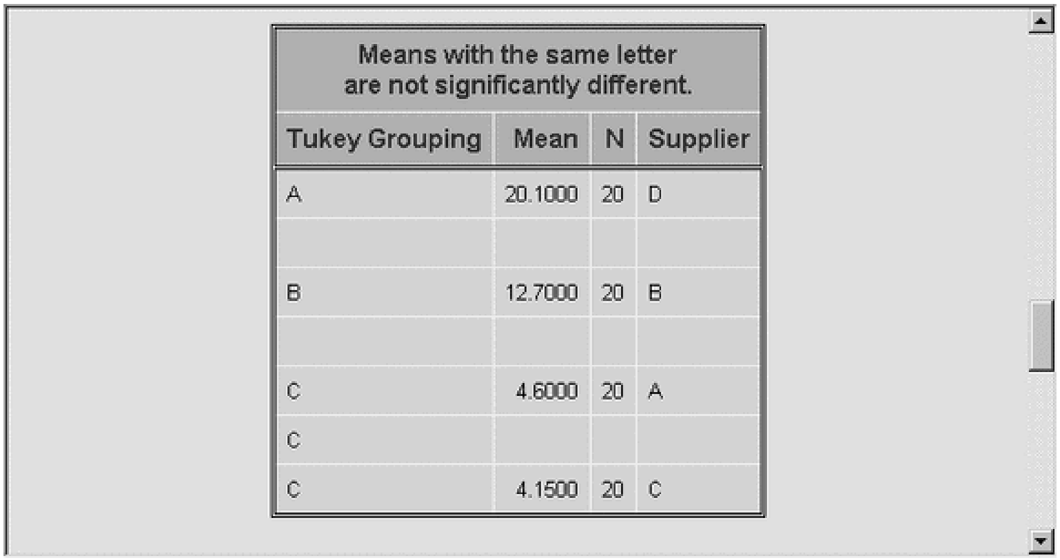
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	3604.775000	600.795833	304.12	<.0001
Error	73	144.212500	1.975514		
Corrected Total	79	3748.987500			

R-Square	Coeff Var	Root MSE	Defects Mean
0.961533	13.53097	1.405530	10.38750

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Supplier	3	3441.637500	1147.212500	580.72	<.0001
Machine	3	163.137500	54.379167	27.53	<.0001

The ANOVA procedure uses the “Stat.GLM.Tests” template to format the ANOVA table. The underlined text displayed in Output 14.10.1 shows the links in the table cells labeled as ‘Supplier’ and ‘Machine.’ Because of the modifications in the preceding statements, the Supplier table listing contains the HTML anchor reference to the tag IDX6. When you click on the ‘Supplier’ link, the appropriate multiple comparison table opens in your browser (Output 14.10.2). The links corresponding to the Machine variable operate similarly.

Output 14.10.2. Linked Output: Multiple Comparison Table from PROC ANOVA



Means with the same letter are not significantly different.			
Tukey Grouping	Mean	N	Supplier
A	20.1000	20	D
B	12.7000	20	B
C	4.6000	20	A
C			
C	4.1500	20	C

Example 14.11. Creating HTML Output, Linked between Analyses

The following example demonstrates how you can use ODS to create links between different types of analyses.

The data in the following example are selected from a larger experiment on the use of drugs in the treatment of leprosy (Snedecor and Cochran 1967, p. 422). Variables in the study are

drug	– two antibiotics (‘a’ and ‘d’) and a control (‘f’)
PreTreatment	– a pretreatment score of leprosy bacilli
PostTreatment	– a posttreatment score of leprosy bacilli

The data set is created as follows:

```
data drugtest;
   input drug $ PreTreatment PostTreatment @@;
   datalines;
a 11  6  a  8  0  a  5  2  a 14  8  a 19 11
a  6  4  a 10 13  a  6  1  a 11  8  a  3  0
d  6  0  d  6  2  d  7  3  d  8  1  d 18 18
d  8  4  d 19 14  d  8  9  d  5  1  d 15  9
f 16 13  f 13 10  f 11 18  f  9  5  f 21 23
f 16 12  f 12  5  f 12 16  f  7  1  f 12 20
;
```

The ODS HTML statement opens the HTML destination, specifies the body file name, requests that a table of contents be generated for the output, and specifies the file name of the frame to contain the body and table of contents. The NOGTITLE option in the ODS HTML statement specifies that titles are not to be included as an integral part of any generated graphics. For all graphics contained in the specified body file, titles appear in the body file and are external to graphics.

```
ods html body='glmb.htm'
         contents='glmc.htm'
         frame='glmf.htm'
         nogtitle;

ods output LSMeans=lsmeans;
```

The ODS OUTPUT statement writes the table of “LS-means” to the data set called lsmeans.

The GLM procedure is invoked to perform an analysis of covariance and compute LS-means for the variable drug.

```
proc glm;
   class drug;
   model PostTreatment = drug | PreTreatment / solution;
   lsmeans drug / stderr pdiff;
quit;
```

The following steps demonstrate how you can create links to connect the results of different analyses. In this example, the table of “LS-means” is graphically summarized with the GCHART procedure. In the steps that follow, each part of the resulting chart is linked to a plot that displays the relationship between the PostTreatment response variable and the PreTreatment variable.

The following DATA step creates a new variable called drugclick that matches each drug value with an HTML file. The variable drugclick is used in the subsequent GCHART procedure run. The variable provides the connection information for linking the two parts of the analysis together. The files referred to in these statements are created in a later step.

```
data lsmeans;
   set lsmeans;
   if drug='a' then drugclick='href=drug1.htm';
   if drug='d' then drugclick='href=drug2.htm';
   if drug='f' then drugclick='href=drug3.htm';
run;
```

The following GOPTIONS and AXIS statements specify settings for the GCHART procedure. PROC GCHART is invoked, and the HBAR statement requests a horizontal bar chart for the variable drug. The length of the bars represent the value of the lsmean variable. The HTML option specifies the variable drugclick as the HTML linking variable to use. The FOOTNOTE1 and FOOTNOTE2 statements provide text that indicates how to use the links on the graph.

```
goptions ftext=swissb hsize=5.5in vsize=3.5in
         border cback=white;
axis1 minor=none label=(angle=90 rotate=0);
axis2 minor=none;

title f=swiss 'Chart of LS-means for Drug Type';
proc gchart data=lsmeans;
   hbar drug / sumvar=lsmean type=mean
               frame cframe=ligr
               gaxis=axis1 raxis=axis2
               html=drugclick;
footnote1 j=l 'click on the bar to see a plot of PostTreatment';
footnote2 j=l 'versus PreTreatment for the corresponding drug';
format lsmean 6.3;
run;

footnote;
ods html close;
run;
```

The preceding statements create a chart that summarizes the information from PROC GLM and that contains links to a second graphic analysis (using the variable drugclick and the HTML option in PROC GCHART).

The following statements provide that second analysis. The three files referred to by the drugclick variable are created as follows.

```
ods html body='drug1.htm'
         newfile=page;

symbol1 c=white v=dot i=r;
title 'Plot of PostTreatment versus PreTreatment';
proc gplot data=drugtest uniform;
   plot PostTreatment*PreTreatment / frame cframe=ligr;
   by drug notsorted;
   footnote;
run;

ods html close;
```

The NEWFILE option in the ODS HTML statement specifies that a new HTML file be created for each page of output. Note that page breaks occur only when a procedure explicitly starts a new page. The NEWFILE option also increments the filename for each new HTML file created, with the first filename corresponding to that given in the BODY= option, drug1.htm.

The GPLOT procedure is invoked, producing a plot of the variable PostTreatment versus the variable PreTreatment for each value of the drug variable. Thus, three plots are created, and each plot is contained in a separate HTML file. The files are named drug1.htm, drug2.htm, and drug3.htm. The filenames match those filenames specified as values of the drugclick variable.

Output 14.11.1. Output from PROC GLM

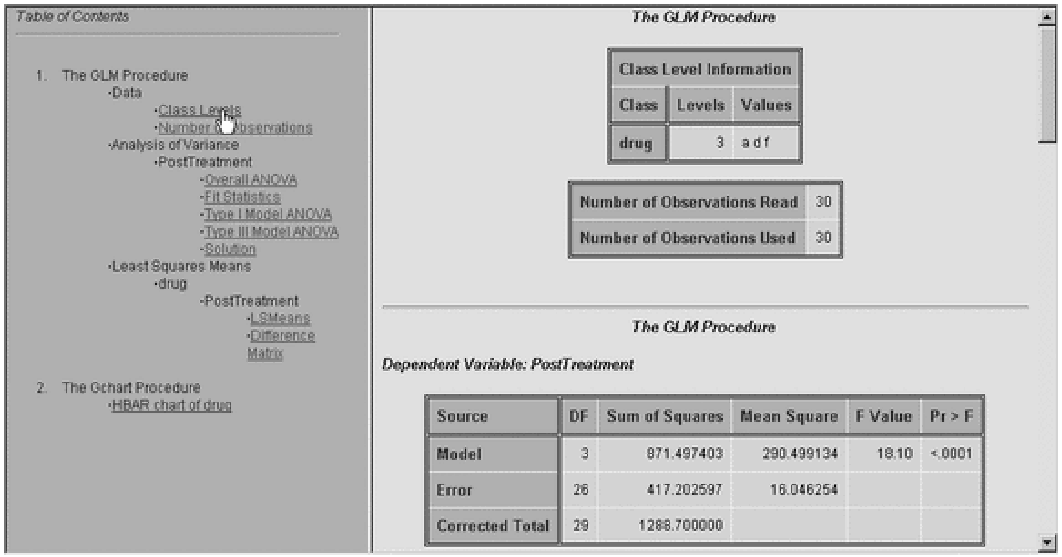


Table of Contents

1. The GLM Procedure
   - Data
     - Class Levels
     - Number of Observations
   - Analysis of Variance
     - PostTreatment
       - Overall ANOVA
       - Fit Statistics
       - Type I Model ANOVA
       - Type III Model ANOVA
       - Solution
   - Least Squares Means
     - drug
       - PostTreatment
         - LSMeans
         - Difference Matrix
2. The Gchart Procedure
   - HBAR chart of drug

The GLM Procedure

Class Level Information		
Class	Levels	Values
drug	3	a d f

Number of Observations Read	30
Number of Observations Used	30

The GLM Procedure

Dependent Variable: PostTreatment

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	871.497403	290.499134	18.10	<.0001
Error	26	417.202597	16.046254		
Corrected Total	29	1288.700000			

Output 14.11.2. Bar Chart of LS-means by Drug Type: Linked Output

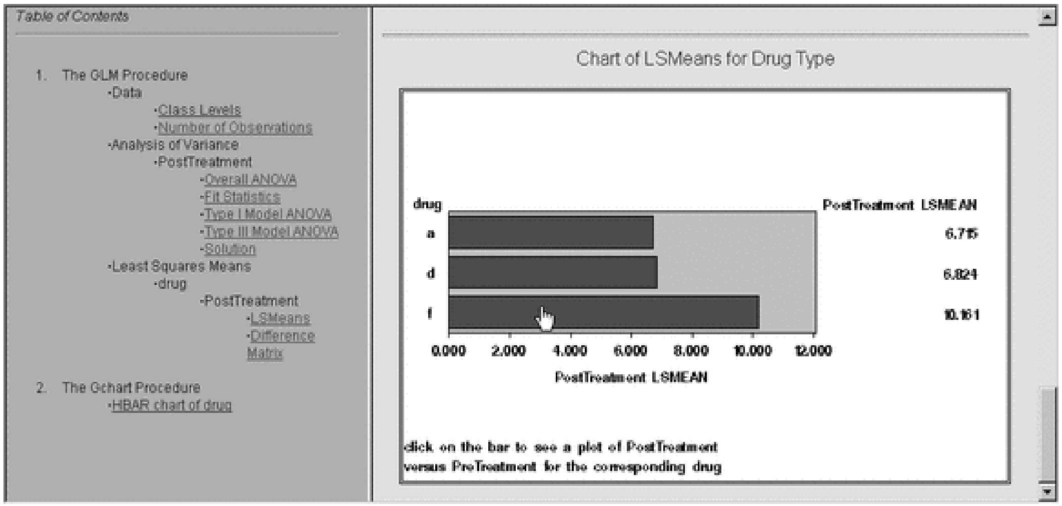


Table of Contents

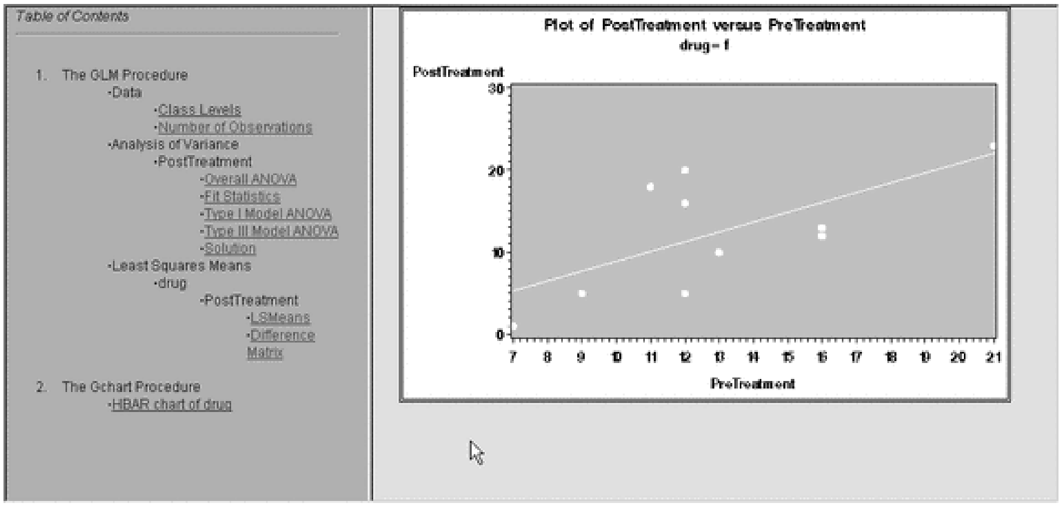
1. The GLM Procedure
   - Data
     - Class Levels
     - Number of Observations
   - Analysis of Variance
     - PostTreatment
       - Overall ANOVA
       - Fit Statistics
       - Type I Model ANOVA
       - Type III Model ANOVA
       - Solution
   - Least Squares Means
     - drug
       - PostTreatment
         - LSMeans
         - Difference Matrix
2. The Gchart Procedure
   - HBAR chart of drug

Chart of LSMeans for Drug Type

click on the bar to see a plot of PostTreatment versus PreTreatment for the corresponding drug

The graphic in Output 14.11.2 displays the difference in lsmeans for each drug type. When you click on a bar that represents a value of the variable drug, the browser opens the plot of PostTreatment versus PostTreatment that corresponds to that value of the variable drug. Output 14.11.3 displays the plot corresponding to the drug type ‘f’. You can view this graphic by clicking on the bottom bar in the bar chart in Output 14.11.2.

Output 14.11.3. Plot of PostTreatment versus PreTreatment for Drug Type 'f': Linked Output



References

Cole, J. W. and Grizzle, J. E. (1966), "Applications of Multivariate Analysis of Variance to Repeated Measures Experiments," *Biometrics*, 22, 810–828.

Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Olinger, C. R. and Tobias, R. D. (1998), "It Chops, It Dices, It Makes Julienne Slices! ODS for Data Analysis Output As-You-Like-It in Version 7," *Proceedings of the Twenty-third Annual SAS Users Group International Conference*.

Pothoff, R. F. and Roy, S. N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Snedecor, G. W. and Cochran, W. G. (1967), *Statistical Methods*, Sixth Edition, Ames, IA: Iowa State University Press.

Chapter 15
Statistical Graphics Using ODS (Experimental)

Chapter Contents

Chapter 15
Statistical Graphics Using ODS (Experimental)

Overview

Graphics are indispensable for modern statistical analysis. They enrich the analysis by revealing patterns, identifying differences, and expressing uncertainty that would not be readily apparent in tabular output. Effective graphics also add visual clarity to an analytical presentation, and they provoke questions that would not otherwise be raised, stimulating deeper investigation.

In SAS 9.1, a number of SAS/STAT procedures have been modified to use an experimental extension to the Output Delivery System (ODS) that enables them to create statistical graphics as automatically as tables. This facility is referred to as *ODS Statistical Graphics* (or *ODS Graphics* for short), and it is invoked when you provide the experimental ODS GRAPHICS statement prior to your procedure statements. Any procedures that use ODS Graphics then create graphics, either by default or when you specify procedure options for requesting specific graphs.

With ODS Graphics, a procedure creates the graphs that are most commonly needed for a particular analysis. In many cases, graphs are automatically enhanced with useful statistical information or metadata, such as sample sizes and p-values, which are displayed in an inset box. Using ODS Graphics eliminates the need to save numerical results in an output data set, manipulate them with a DATA step program, and display them with a graphics procedure.

The SAS/STAT procedures that use ODS Graphics in SAS 9.1 are listed on page 348. The plots produced by each procedure and any corresponding options are described in the procedure chapter. See the “ODS Graphics” subsection in the “Details” section of each procedure chapter for additional information.

In many ways, creating graphics with ODS is analogous to creating tables with ODS. You use

- procedure options and defaults to determine which graphs are created
- ODS destination statements (such as ODS HTML) to specify the output destination for graphics

Additionally, you can use

- graph names in ODS SELECT and ODS EXCLUDE statements to select or exclude graphs from your output
- ODS styles to control the general appearance and consistency of *all graphs*
- ODS templates to control the layout and details of *individual graphs*. A default template is provided by SAS for each graph.

In SAS 9.1, the ODS destinations that support ODS Graphics include HTML, LATEX, PRINTER, and RTF. These are discussed on page 326.

Both tables and graphs are saved in the ODS output file produced for a destination. However, individual graphs can also be saved in files, which are produced in a specific graphics image file type, such as GIF or PostScript. This enables you to access individual graphs for inclusion in a document. For example, you can save graphs in PostScript files to include in a paper that you are writing with $\LaTeX$. Likewise, you can save graphs in GIF files to include in an HTML document. With the HTML destination, you can also request an image map format that supports tool tip displays, which appear when you move a mouse over certain features of the graph.

In common applications of procedures that use ODS Graphics, the default graphs should suffice. However, when modifications become necessary, you can customize a particular graph by changing its template, or you can make consistent changes to all your graphs by selecting a different ODS style or by modifying an existing ODS style definition:

- As with table definitions, you can access graph template definitions and modify them with the TEMPLATE procedure. Graph template definitions are written in an experimental graph template language, which has been added to the TEMPLATE procedure in SAS 9.1. This language includes statements for specifying plot types (such as scatter plots and histograms), plot layouts, and text elements (such as titles and insets). It also provides support for built-in computations (such as histogram binning) and evaluation of computational expressions. Options are available for specifying colors, marker symbols, and other aspects of plot features.
- ODS style definitions include a number of graph elements that correspond to general features of statistical graphics, such as titles and fitted lines. The attributes of these elements, such as fonts and colors, provide the defaults for options in graph templates provided by SAS. Consequently, you can change all of your graphs in a consistent manner by simply selecting a different style. For example, by specifying the “Journal” style, you can create gray-scale graphs and tables that are suitable for publication in statistical journals.

Note: Statistical graphics created with ODS are experimental in this release, meaning that both their appearance and their syntax are subject to change in a future release.

This chapter illustrates the use of ODS Graphics, and it provides general information on managing your graphics. If you are unfamiliar with ODS, you will find it helpful to read Chapter 14, “Using the Output Delivery System.” For complete documentation on the Output Delivery System, refer to the *SAS Output Delivery System User’s Guide*.

How to Use This Chapter

If you are trying out ODS Graphics for the first time, begin by reading the section "Getting Started" on page 321, which provides the essentials. Additional examples are given in the chapters for procedures that use ODS Graphics in SAS 9.1.

To take full advantage of ODS Graphics, you will need to learn more about ODS destinations, output files, and image file types for graphics, as well as ways to access and include individual graphs in reports and presentations. This is explained in the section "Managing Your Graphics" on page 326, the section "Graphics Image Files" on page 334, and the section "Examples" beginning on page 352.

If you need to customize a graph by modifying its template, read the section "Customizing Graphics with Templates" on page 338 and the series of examples beginning on page 363.

If you need to customize a style definition read the section "Styles for Graphics" on page 344 and the series of examples beginning on page 374.

Getting Started

This section introduces the use of ODS Graphics with two simple examples, which illustrate how the ODS GRAPHICS statement and an ODS destination statement are required to produce graphics. In the first example, no procedure options are required; basic graphics are produced by default. In the second example, procedure options are used to request specific plots.

Using the ODS GRAPHICS Statement

This example is taken from the "Getting Started" section of Chapter 61, "The REG Procedure." It illustrates a situation in which only the ODS GRAPHICS statement and a supported ODS destination are needed to create graphical displays.

The following data are from a study of 19 children. The variables Height, Weight, and Age are measured for each child.

```
data Class;
   input Name $ Height Weight Age @@;
   datalines;
Alfred  69.0 112.5 14  Alice  56.5  84.0 13  Barbara 65.3  98.0 13
Carol   62.8 102.5 14  Henry  63.5 102.5 14  James   57.3  83.0 12
Jane    59.8  84.5 12  Janet  62.5 112.5 15  Jeffrey 62.5  84.0 13
John    59.0  99.5 12  Joyce  51.3  50.5 11  Judy    64.3  90.0 14
Louise  56.3  77.0 12  Mary   66.5 112.0 15  Philip  72.0 150.0 16
Robert  64.8 128.0 12  Ronald 67.0 133.0 15  Thomas  57.5  85.0 11
William 66.5 112.0 15
;
```

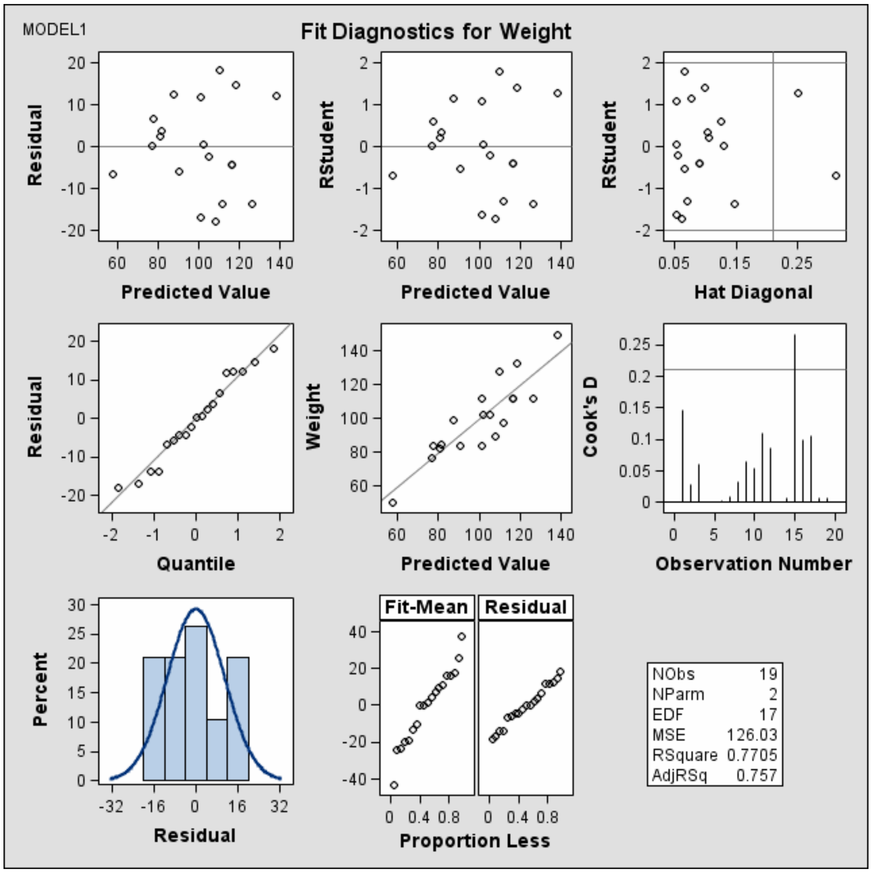
The following statements invoke the REG procedure and fit a simple linear regression model in which Weight is the response variable and Height is the independent variable.

```
ods html;
ods graphics on;

proc reg data = Class;
   model Weight = Height;
run;
quit;

ods graphics off;
ods html close;
```

The ODS HTML statement specifies an HTML destination for the output. Note that the LISTING destination is not supported by ODS Graphics in SAS 9.1. For a discussion of ODS destinations that are supported, see page 326.

The ODS GRAPHICS statement is specified to request ODS Graphics in addition to the usual tabular output. Here, the graphical output consists of a fit diagnostics panel, a residual plot, and a fit plot; these are shown in Figure 15.1, Figure 15.2, and Figure 15.3, respectively.

The ODS GRAPHICS OFF statement disables ODS Graphics, and the ODS HTML CLOSE statement closes the HTML destination.



```
NObs        19
NParm        2
EDF         17
MSE     126.03
RSquare 0.7705
AdjRSq   0.757
```

Figure 15.1. Fit Diagnostics Panel

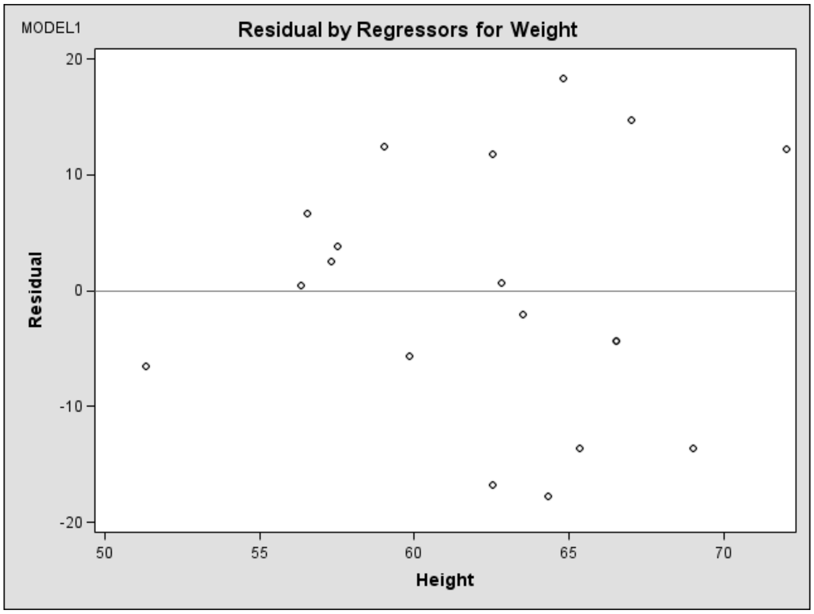
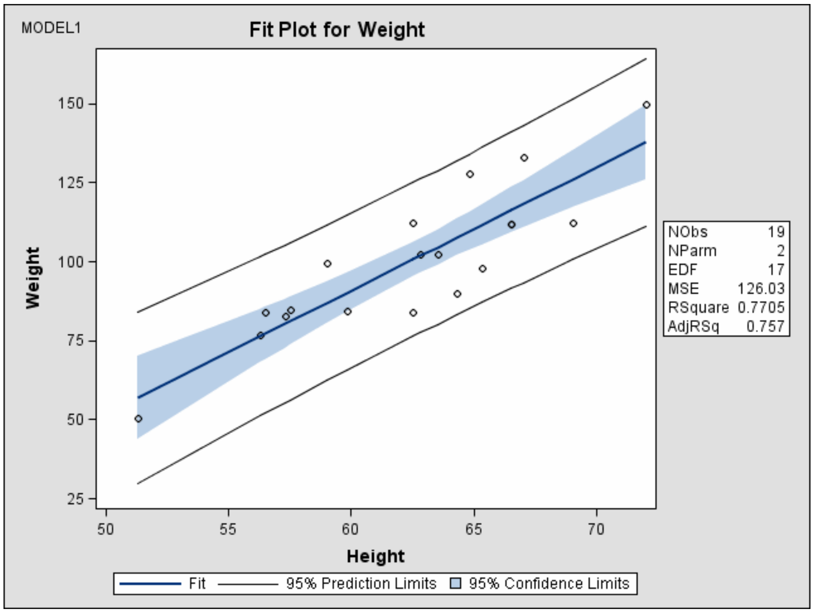


Figure 15.2. Residual Plot



NObs	19
NParm	2
EDF	17
MSE	126.03
RSquare	0.7705
AdjRSq	0.757

Figure 15.3. Fit Plot

Note: ODS Graphics are produced completely independently of both line printer plots and traditional high resolution graphics requested with the PLOT statement in PROC REG. Traditional high resolution graphics are saved in graphics catalogs and controlled by the GOPTIONS statement. In contrast, ODS Graphics are produced in ODS output (not graphics catalogs) and their appearance and layout are controlled by ODS styles and templates. In SAS 9.1 both line printer plots and traditional high resolution graphics requested with the PLOT statement continue to be available and are unaffected by the ODS GRAPHICS statement.

For more information about ODS Graphics available in the REG procedure, see the "ODS Graphics" section on page 3922 in Chapter 61, "The REG Procedure."

A sample program named odsgr01.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Using the ODS GRAPHICS Statement and Procedure Options

This example is taken from the "Getting Started" section of Chapter 36, "The KDE Procedure." Here, new procedure options are used to request graphical displays in addition to the ODS GRAPHICS statement.

The following statements simulate 1,000 observations from a bivariate normal density with means $(0, 0)$, variances $(10, 10)$, and covariance 9.

```
data bivnormal;
   seed = 1283470;
   do i = 1 to 1000;
      z1 = rannor(seed);
      z2 = rannor(seed);
      z3 = rannor(seed);
      x = 3*z1+z2;
      y = 3*z1+z3;
      output;
   end;
   drop seed;
run;
```

The following statements request a bivariate kernel density estimate for the variables x and y.

```
ods html;
ods graphics on;

proc kde data = bivnormal;
   bivar x y / plots = contour surface;
run;

ods graphics off;
ods html close;
```

A contour plot and a surface plot of the estimate are displayed in Figure 15.4 and Figure 15.5, respectively. These graphical displays are requested by specifying the ODS GRAPHICS statement prior to the procedure statements and the experimental PLOTS= option in the BIVAR statement. For more information about the graphics available in the KDE procedure, see the "ODS Graphics" section on page 2009 in Chapter 36, "The KDE Procedure."

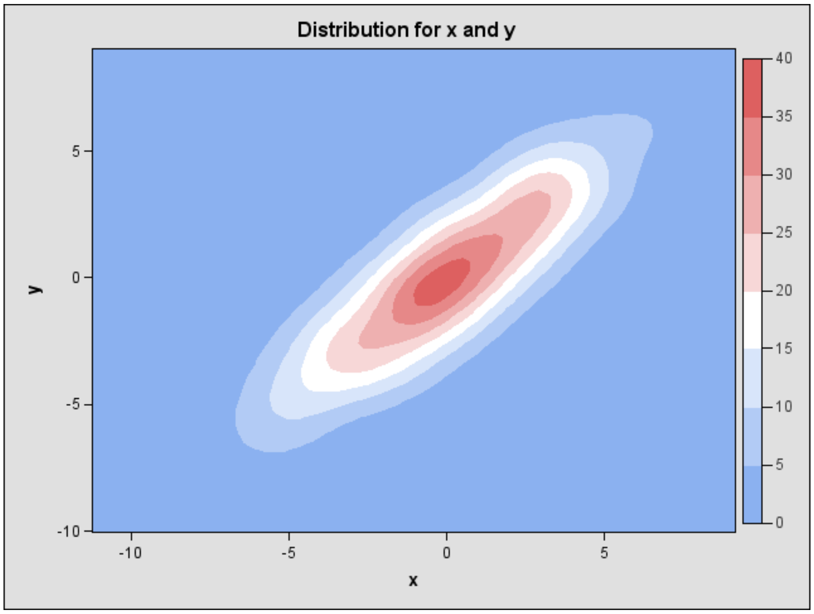


Figure 15.4. Contour Plot of Estimated Density

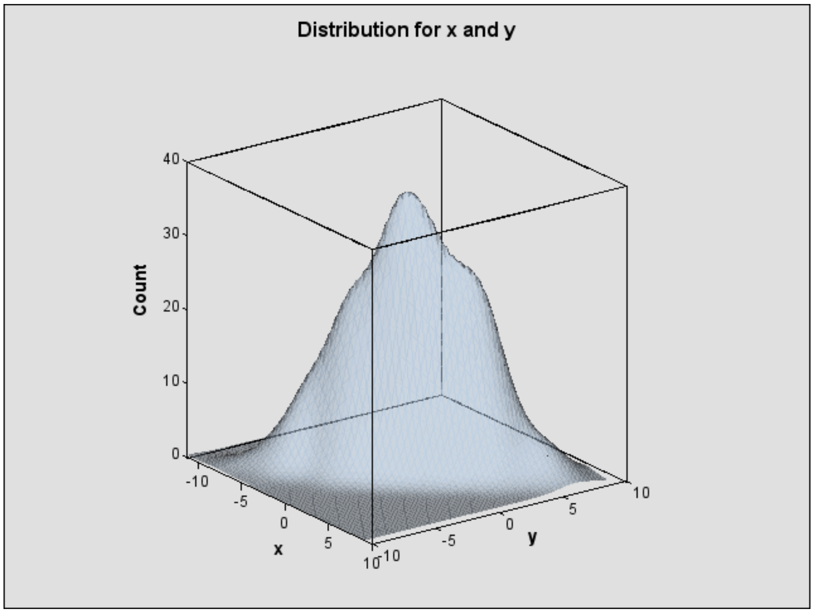


Figure 15.5. Surface Plot of Estimated Density

A sample program named odsgr02.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Managing Your Graphics

This section describes techniques for managing your graphics:

- specifying an ODS destination for graphics
- viewing your graphs in the SAS windowing environment
- referring to graphs by name when using ODS
- selecting and excluding graphs from your output
- modifying the appearance of all your graphs with styles

Specifying an ODS Destination for Graphics

Whenever you use ODS Graphics you must specify a valid ODS destination. The examples in "Getting Started" illustrate how to specify an HTML destination. Other destinations are specified in a similar way. For example, you can specify an RTF destination with the following statements.

```
ods rtf;
ods graphics on;

   ...SAS statements...

ods graphics off;
ods rtf close;
```

The supported ODS destinations are shown in Table 15.1.

Table 15.1. Destinations Supported by ODS Graphics

Destination	**Destination Family**	**Viewer**
DOCUMENT		Not Applicable
HTML	MARKUP	Browser
LATEX	MARKUP	Ghostview
PCL	PRINTER	Ghostview
PDF	PRINTER	Acrobat
PS	PRINTER	Ghostview
RTF		Microsoft Word

Note: In SAS 9.1 the LISTING destination does not support ODS Graphics. You must specify a supported ODS destination in order to produce ODS Graphics, as illustrated by all the examples in this chapter.

Specifying a File for ODS Output

You can specify a file name for your output with the FILE= option in the ODS destination statement, as in the following example:

```
ods html file = "test.htm";
```

The output is written to the file test.htm, which is saved in the SAS current folder. At startup, the SAS current folder is the same directory in which you start your SAS session. If you are running SAS with the windowing environment in the Windows operating system, then the current folder is displayed in the status line at the bottom of the main SAS window, as shown in Figure 15.6.

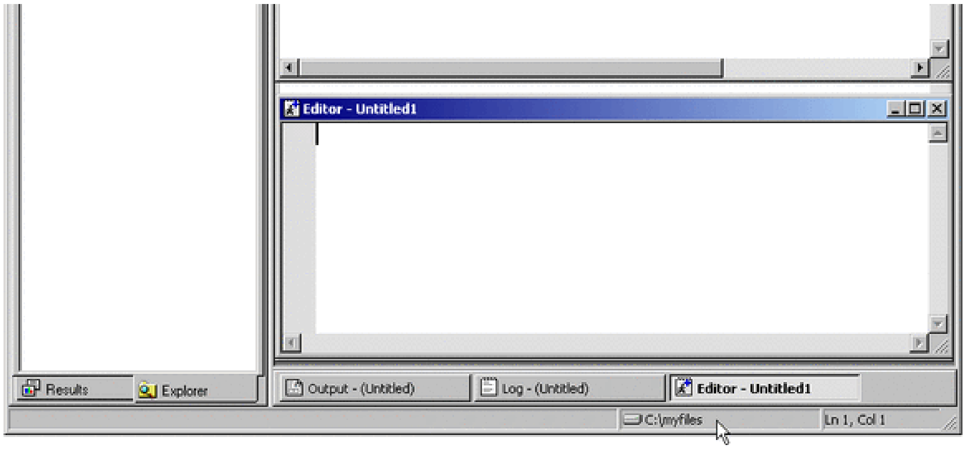


Figure 15.6. Current Folder (Right Bottom)

If you do not specify a file name for your output, then SAS provides a default file, which depends on the ODS destination. This file is saved in the SAS current folder. You can always check the SAS log to verify the name of the file in which your output is saved. For example, suppose you specify the following statement at startup:

```
ods html;
```

Then the following message is displayed in the SAS log:

```
NOTE: Writing HTML Body file: sashtml.htm
```

The default file names for each destination are specified in the SAS Registry. For more information, refer to the SAS Companion for your operating system.

Viewing Your Graphs in the SAS Windowing Environment

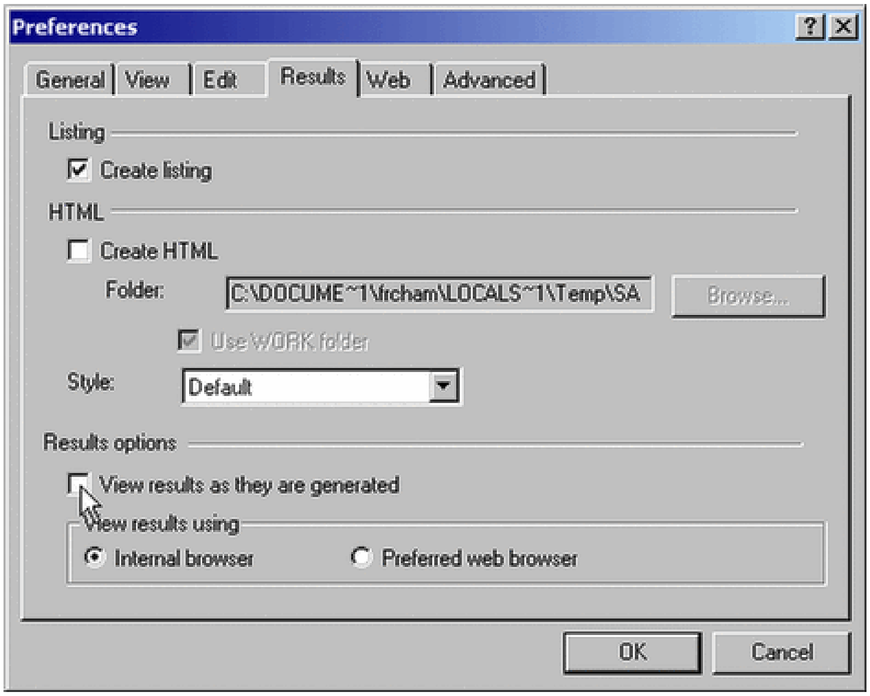
The mechanism for viewing graphics created with ODS can vary depending on your operating system, which viewers are installed on your computer, and the ODS destination you have selected.

If you are using the SAS windowing environment in the Windows operating system and you specify an HTML destination, then by default the results are displayed in the SAS Results Viewer as they are being generated. Depending on your configuration, this may also apply to the PDF and RTF destinations.* For information about the

*If you are using the LATEX or the PS destinations you must use a PostScript viewer, such as Ghostview.

windowing environment in a different operating system, refer to the SAS Companion for that operating system.

If you do not want to view the results as they are being generated, then select **Tools** → **Options** → **Preferences...** from the menu at the top of the main SAS window. Then in the **Results** tab disable **View results as they are generated**, as shown in Figure 15.7.



Listing

☑ Create listing

HTML

□ Create HTML

Folder: C:\DOCUME~1\frcham\LOCALS~1\Temp\SA

☑ Use WORK folder

Style: Default

Results options

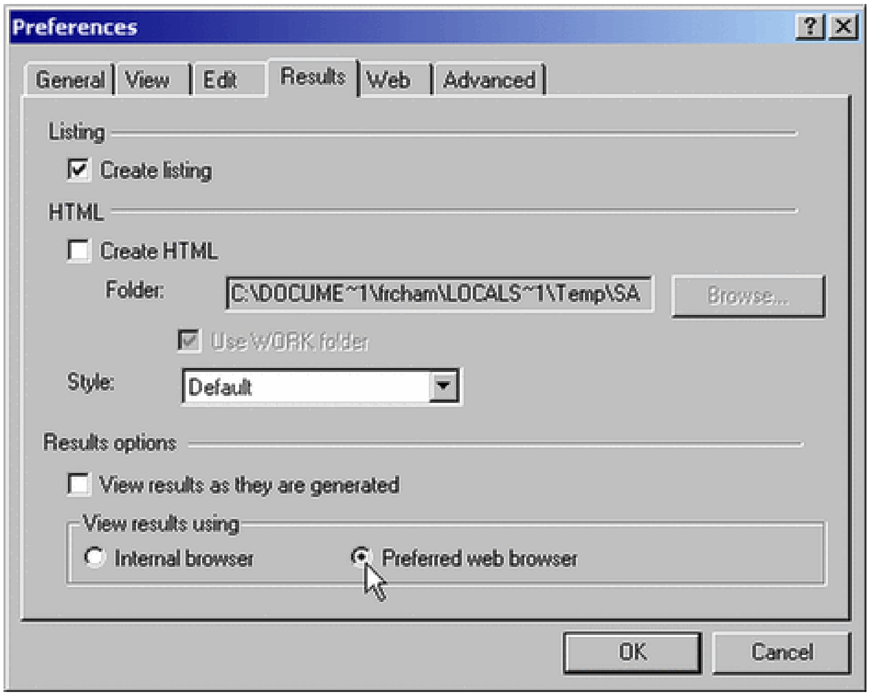
□ View results as they are generated

View results using

● Internal browser ○ Preferred web browser

Figure 15.7. Disabling View of Results as Generated

You can change the default to use an external viewer instead of the Results Viewer. Select **Tools** → **Options** → **Preferences...** from the menu at the top of the main SAS window. Then in the **Results** tab select **Preferred web browser**, as shown in Figure 15.8. Your results will then be displayed in the default viewer that is configured in your computer for the corresponding destination.



Listing

☑ Create listing

HTML

□ Create HTML

Folder: C:\DOCUME~1\frcham\LOCALS~1\Temp\SA

☑ Use WORK folder

Style: Default

Results options

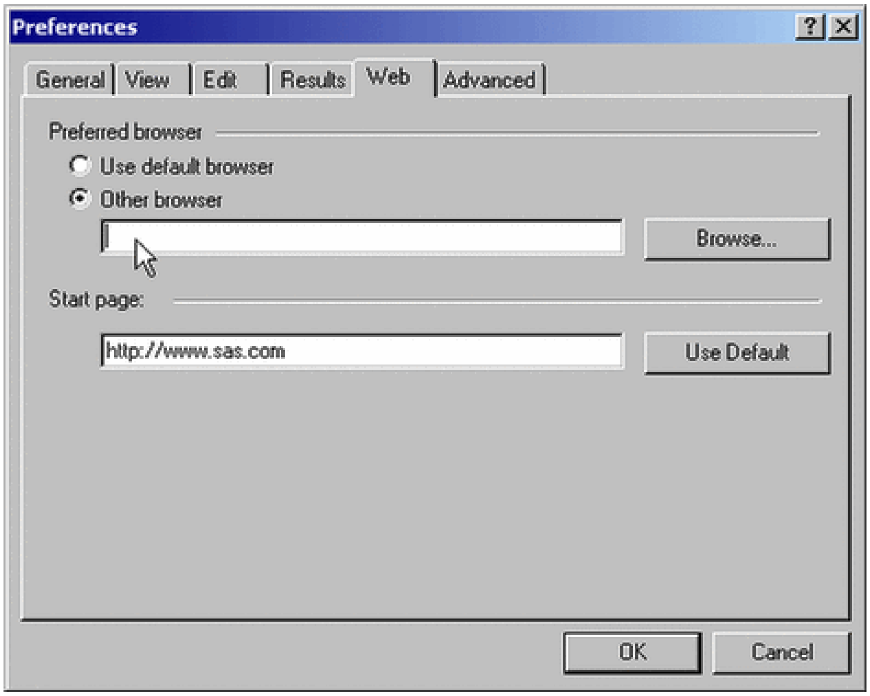
□ View results as they are generated

View results using

○ Internal browser ● Preferred web browser

Figure 15.8. Selecting an External Browser

You can also choose which browser to use for HTML output. Select **Tools** → **Options** → **Preferences. . .** from the menu at the top of the main SAS window. Then in the **Web** tab select **Other browser**, and type (or browse) the path of your preferred browser, as shown in Figure 15.9.



Preferred browser

○ Use default browser

● Other browser

Start page:

http://www.sas.com

Figure 15.9. Changing the Default External Browser

Referring to Graphs by Name

Procedures assign a name to each graph they create with ODS Graphics. This enables you to refer to ODS graphs in the same way that you refer to ODS tables (see the "Using the Output Delivery System" section on page 274 in Chapter 14, "Using the Output Delivery System"). You can determine the names of graphs in several ways:

- You can look up graph names in the "ODS Graphics" section of chapters for procedures that use ODS Graphics. See, for example, the "ODS Graphics" section on page 3922 in Chapter 61, "The REG Procedure."
- You can use the Results window to view the names of ODS graphs created in your SAS session. See the section "Using ODS with the SAS Explorer" on page 277 for more information.
- You can use the ODS TRACE ON statement to list the names of graphs created by your SAS session. This statement adds identifying information in the SAS log (or, optionally, in the SAS listing) for each graph that is produced. See page 330 for an example, and the "Using the Output Delivery System" section on page 274 for more information.

Note that the graph name is not the same as the name of the file containing the graph (see page 335).

Selecting and Excluding Graphs

You can use graph names to specify which ODS graphs are displayed with the ODS SELECT and ODS EXCLUDE statements. See the section "Using the Output Delivery System" on page 274 for information on how to use these statements.

Example

This example revisits the analysis described in the section "Using the ODS GRAPHICS Statement and Procedure Options" on page 324.

To determine which output objects are created by ODS, you specify the ODS TRACE ON statement prior to the procedure statements.

```
ods trace on;

ods html;
ods graphics on;

proc kde data = bivnormal;
   bivar x y / plots = contour surface;
run;

ods graphics off;
ods html close;

ods trace off;
```

Figure 15.10 displays the trace record, which is added to the SAS log. By default, the KDE procedure creates table objects named “Inputs” and “Controls,” and it creates graph objects named “Contour” and “SurfacePlot.” In addition to the name, the trace record provides the label, template, and path for each output object. Graph templates are distinguished from table templates by a naming convention that uses the procedure name in the second level and the word “Graphics” in the third level. For example, the fully qualified template name for the surface plot created by PROC KDE, as shown in Figure 15.10, is

```
Stat.KDE.Graphics.HistSurface
```

```
Output Added:
-------------
Name:       Inputs
Template:   Stat.KDE.Inputs
Path:       KDE.Bivar1.x_y.Inputs
-------------

Output Added:
-------------
Name:       Controls
Template:   Stat.KDE.Controls
Path:       KDE.Bivar1.x_y.Controls
-------------
WARNING: Statistical graphics displays created with ODS are experimental in
         this release.

Output Added:
-------------
Name:       Contour
Label:      Contour Plot
Template:   Stat.KDE.Graphics.ContourScatter
Path:       KDE.Bivar1.x_y.Contour
-------------

Output Added:
-------------
Name:       SurfacePlot
Label:      Density Surface
Template:   Stat.KDE.Graphics.HistSurface
Path:       KDE.Bivar1.x_y.SurfacePlot
-------------
```

Figure 15.10. ODS Trace Record in SAS Log

Note that you can specify the LISTING option in the ODS TRACE ON statement to write the trace record to the LISTING destination:

```
ods trace on / listing;
```

The following statements use the ODS SELECT statement to specify that only the two graph objects named “Contour” and “SurfacePlot” are to be included in the HTML output.

```
ods html;
ods graphics on;

ods select Contour SurfacePlot;

proc kde data = bivnormal;
   bivar x y / plots = contour surface;
run;

ods graphics off;
ods html close;
```

A sample program named odsgr02.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Specifying Styles for Graphics

ODS styles control the overall look of your output. A style definition provides formatting information for specific visual aspects of your SAS output. For ODS tables this information typically includes a list of font definitions (each font defines a family, size, weight, and style) and a list of colors, which are associated with common areas of printed output, including titles, footnotes, by-groups, table headers, and table cells.

Starting with SAS 9, ODS styles also include graphical appearance information such as line and marker properties in addition to font and color information. Furthermore, in SAS 9.1, ODS styles include graphics appearance informats for common elements of statistical graphics created with ODS Graphics. These elements include fitted lines, confidence and prediction bands, and outliers.

For more information about styles, refer to the "TEMPLATE Procedure: Creating a Style Definition" in the *SAS Output Delivery System User's Guide*.

Specifying a Style

You can specify a style using the STYLE= option in a valid ODS destination,* such as HTML, PDF, RTF, or PRINTER. Each style produces output with the same content, but a somewhat different visual appearance. For example, the following statement request output using the "Journal" style.

```
ods html style = Journal;
```

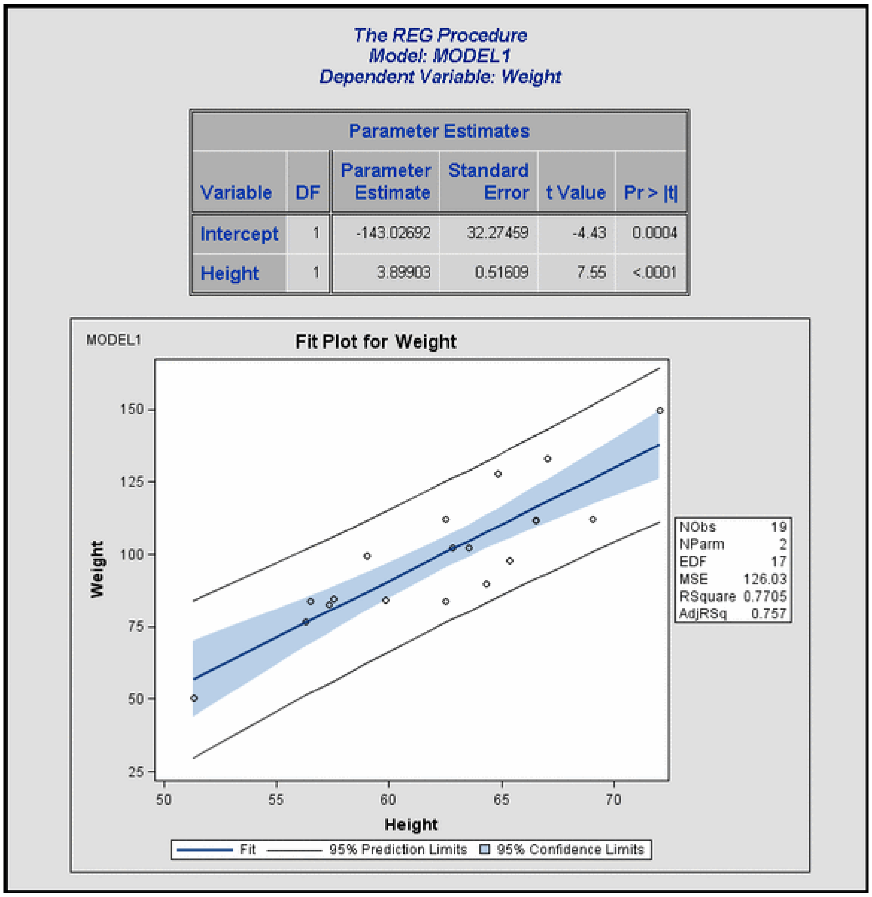
*Style definitions do not apply to the LISTING destination, which uses the SAS monospace format by default for output tables. The LISTING destination is not a valid destination for ODS Graphics in SAS 9.1.

Any SAS-supplied or user-defined style can be used for ODS Graphics. However, of the SAS-supplied styles for SAS 9.1, four are specifically designed and recommended for use with ODS Graphics:

- Analysis
- Default
- Journal
- Statistical

Figure 15.11 and Figure 15.12 illustrate the difference between the “Default” and the “Journal” styles for the HTML destination. Note that the appearance of tables and graphics is coordinated within a particular style. This is also illustrated in the series of examples starting with Example 15.11.

For more information about styles for ODS Graphics, see the section “Styles for Graphics” on page 344 or refer to the “ODS Statistical Graphics and ODS Styles: Usage and Reference (Experimental)” at
http://support.sas.com/documentation/onlinedoc/base/.

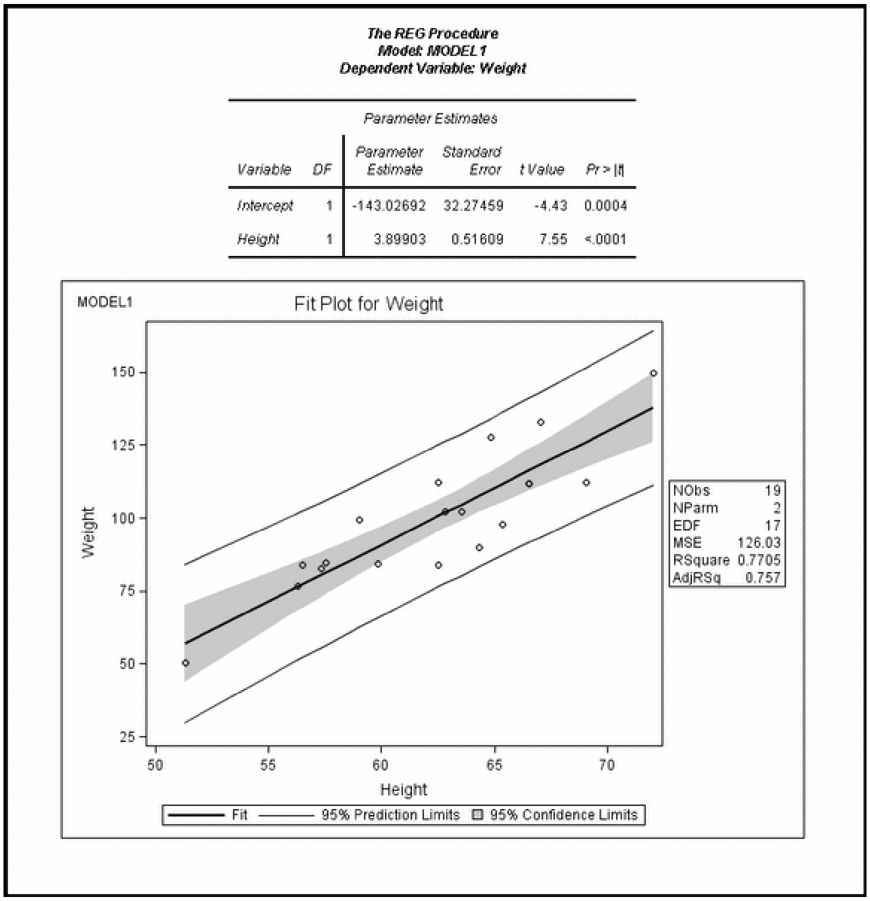


The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

```
NObs        19
NParm        2
EDF         17
MSE     126.03
RSquare 0.7705
AdjRSq   0.757
```

Figure 15.11. HTML Output with Default Style



The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Parameter Estimates

Variable	*DF*	*Parameter Estimate*	*Standard Error*	*t Value*	*Pr > |t|*
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

Figure 15.12. HTML Output with Journal Style

Graphics Image Files

Accessing your graphs as individual image files is useful when you want to include them in various types of documents. The default image file type depends on the ODS destination, but there are other supported image file types that you can specify. You can also specify the names for your graphics image files and the directory in which you want to save them.

This section describes the image file types supported by ODS Graphics, and it explains how to name and save graphics image files.

Describing Supported Image File Types

If you are using an HTML or a LATEX destination, your graphs are individually produced in a specific image file type, such as GIF or PostScript.

If you are using a destination in the PRINTER family or the RTF destination, the graphs are contained in the ODS output file and cannot be accessed as individual image files. However, you can open an RTF output file in Microsoft Word and then copy and paste the graphs into another document, such as a Microsoft PowerPoint presentation; this is illustrated in Example 15.3.

Table 15.2 shows the various ODS destinations supported by ODS Graphics, the viewer that is appropriate for displaying graphs in each destination, and the image file types supported for each destination.

Table 15.2. Destinations and Image File Types Supported by ODS Graphics

Destination	**Destination Family**	**Viewer**	**Image File Types**
DOCUMENT		Not Applicable	Not Applicable
HTML	MARKUP	Browser	GIF (default), JPEG, PNG
LATEX	MARKUP	Ghostview	PostScript (default), EPSI, GIF, JPEG, PNG
PCL	PRINTER	Ghostview	Contained in PostScript file
PDF	PRINTER	Acrobat	Contained in PDF file
PS	PRINTER	Ghostview	Contained in PostScript file
RTF		Microsoft Word	Contained in RTF file

Note: In SAS 9.1 the LISTING destination does not support ODS Graphics. You must specify a supported ODS destination in order to produce ODS Graphics, as illustrated by all the examples in this chapter.

Naming Graphics Image Files

The names of graphics image files are determined by a *base file name*, an *index counter*, and an *extension*. By default, the base file name is the ODS graph name (see page 330). The index counter is set to zero when you begin a SAS session, and it is increased by one after you create a graph, independently of the graph type or the SAS procedure that creates it. The extension indicates the image file type.

For instance, if you run the example on page 324 at the beginning of a SAS session, the two graphics image files created are Contour0.gif and SurfacePlot1.gif. If you immediately rerun this example, then ODS creates the same graphs in different image files named Contour2.gif and SurfacePlot3.gif.

You can specify the RESET option in the ODS GRAPHICS statement to reset the index counter to zero. This is useful to avoid duplication of graphics image files if you are rerunning a SAS program in the same session.

```
   ods graphics on / reset;
```

Note: The index counter is initialized to zero at the beginning of your SAS session or if you specify the RESET option in the ODS GRAPHICS statement. Graphics image files with the same name are overwritten.

You can specify a base file name for all your graphics image files with the IMAGENAME= option in the ODS GRAPHICS statement. For example:

```
   ods graphics on / imagename = "MyName";
```

You can also specify

```
ods graphics on / imagename = "MyName" reset;
```

With the preceding statement, the graphics image files are named MyName0, MyName1, and so on.

You can specify the image file type for the HTML or LATEX destinations with the IMAGEFMT= option in the ODS GRAPHICS statement. For example:

```
ods graphics on / imagefmt = png;
```

For more information, see the "ODS GRAPHICS Statement" section on page 349.

Saving Graphics Image Files

Knowing where your graphics image files are saved and how they are named is particularly important if you are running in batch mode, if you have disabled the SAS Results Viewer (see page 327), or if you plan to access the files for inclusion in a document. The following discussion assumes you are running SAS under the Windows operating system. If you are running on a different operating system, refer to the SAS Companion for your operating system.

Your graphics image files are saved by default in the SAS current folder. If you are using the SAS windowing environment, the current folder is displayed in the status line at the bottom of the main SAS window (see also page 327). If you are running your SAS programs in batch mode, the graphs are saved by default in the same directory where you started your SAS session.

For instance, suppose the SAS current folder is C:\myfiles. If you specify the ODS GRAPHICS statement, then your graphics image files are saved in the directory C:\myfiles. Unlike traditional high resolution graphics created with SAS/GRAPH, ODS Graphics are not temporarily stored in a catalog in your Work directory.

With the HTML and the LATEX destinations, you can specify a directory for saving your graphics image files. With the PRINTER and RTF destinations, you can only specify a directory for your output file. The remainder of this discussion provides details for each destination type.

HTML Destination

If you are using the HTML destination, the individual graphs are created as GIF files by default. You can use the PATH= and GPATH= options in the ODS HTML statement to specify the directory where your HTML and graphics files are saved, respectively. This also gives you more control over your graphs. For example, if you want to save your HTML file named test.htm in the C:\myfiles directory, but you want to save your graphics image files in C:\myfiles\gif, then you specify

```
ods html path  = "C:\myfiles"
         gpath = "C:\myfiles\gif"
         file  = "test.htm";
```

When you specify the URL= suboption with the GPATH= option, SAS creates relative paths for the links and references to the graphics image files in the HTML file. This is useful for building output files that are easily moved from one location to another. For example, the following statements create a relative path to the gif directory in all the links and references contained in test.htm.

```
ods html path  = "C:\myfiles"
         gpath = "C:\myfiles\gif" (url="gif/")
         file  = "test.htm";
```

If you do not specify the URL= suboption, SAS creates absolute paths that are hard-coded in the HTML file; these may cause broken links if you move the files. For more information, refer to the ODS HTML statement in the “Dictionary of ODS Language Statements” (*SAS Output Delivery System User’s Guide*).

LATEX Destination

LaTeX is a document preparation system for high-quality typesetting. The experimental ODS LATEX statement produces output in the form of a LaTeX source file that is ready to compile in LaTeX.

When you request ODS Graphics for a LATEX destination, ODS creates the requested graphs as PostScript files by default, and the LaTeX source file includes references to these image graphics files. You can compile the LaTeX file or you can access the individual PostScript files to include your graphs in a different LaTeX document, such as a paper that you are writing.

You can specify the PATH= and GPATH= options in the ODS LATEX statement, as explained previously for the ODS HTML statement. See Example 15.4 for an illustration.

The ODS LATEX statement is an alias for the ODS MARKUP statement using the TAGSET=LATEX option. For more information, refer to the ODS MARKUP statement in the “Dictionary of ODS Language Statements” (*SAS Output Delivery System User’s Guide*).

If you are using a LATEX destination with the default PostScript image file type, your ODS graphs are created in gray-scale, regardless of the style you are using. When you use this destination, it is recommended that you use the “Journal” style to obtain high quality graphics. For more information about styles, see the “Specifying Styles for Graphics” section on page 332.

To create color graphics using a LATEX destination, specify JPEG, GIF, or PNG with the IMAGEFMT= option in the ODS GRAPHICS statement. If you specify GIF you can use a distiller to obtain a PostScript or a PDF file. If you specify JPEG you may need to include the **DeclareGraphicsExtensions** and the **DeclareGraphicsRule** commands in the preamble of your LaTeX file. For more information, refer to the LaTeX documentation for the **graphicx** package.

PRINTER and RTF Destinations

If you are using a destination in the PRINTER family (PCL, PDF, PS) or the RTF destination, the graphs are contained in the output file and cannot be accessed as individual graphics image files. You can specify the path where the output file is to be saved using the FILE= option of the ODS destination statement. For example, suppose that you specify

```
ods pdf file = "test.pdf";
```

Then your ODS output is saved as the PDF file test.pdf in the SAS current folder (for example, in C:\myfiles).

You can specify a full path name for your output with the FILE= option. For instance to save your PDF file to the directory C:\temp you specify

```
ods pdf file = "C:\temp\test.pdf";
```

You can always check the SAS log to verify where your output is saved. For example, the preceding statement would result in the following log message:

```
NOTE: Writing ODS PDF output to DISK destination
   "C:\temp\test.pdf", printer "PDF".
```

Customizing Graphics with Templates

This section describes how to locate templates for ODS Graphics, and how to display, edit, and save these templates. It also provides an overview of the graph template language. Before presenting these details, a review of the TEMPLATE procedure terminology is helpful.

A *template definition* is a set of SAS statements that can be run with the TEMPLATE procedure to create a compiled template. Two common types of template definitions are *table definitions* and *style definitions*. A table definition describes how to display the output for an output object that is to be rendered as a table, and a style definition provides formatting information for specific visual aspects of your SAS output.

A third type of template definition is a *graph template definition* (or *graph definition* for short), which controls the layout and details of graphs produced with ODS Graphics. Graph definitions begin with a DEFINE STATGRAPH statement and end with an END statement.

A *template store* is a member of a SAS data library that stores compiled templates created by the TEMPLATE procedure. Default templates supplied by SAS are saved in the Sashelp.Tmplmst template store.

In common applications of ODS Graphics, it should not be necessary to modify the default template for each graph, which is supplied by SAS. However, when customization is necessary, you can modify the default template with the graph template language in the TEMPLATE procedure.

If you are using the SAS windowing environment, the easiest way to display, edit, and save your templates is by using the Templates window. For detailed information about managing templates, refer to the "TEMPLATE Procedure: Managing Template Stores" in the *SAS Output Delivery System User's Guide*.

For details concerning the syntax of the graph template language, refer to the "TEMPLATE Procedure: Creating ODS Statistical Graphics Output (Experimental)" at http://support.sas.com/documentation/onlinedoc/base/.

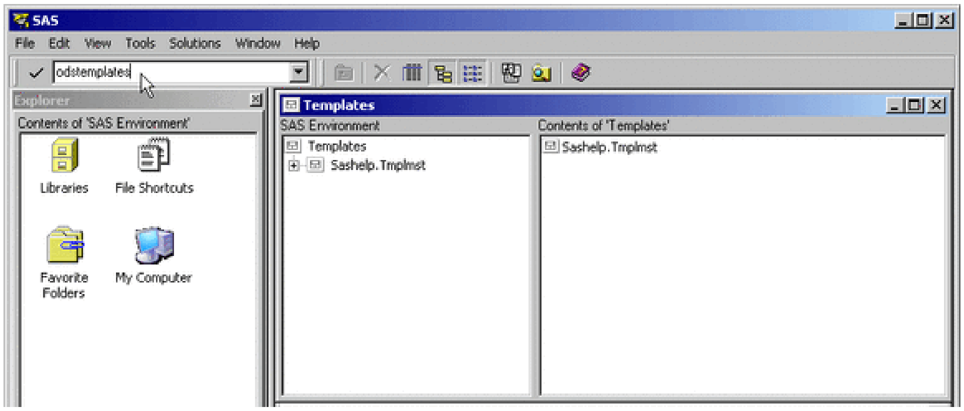
Locating Templates

The first step in customizing a graph is to determine which template was used to create the original graph. The easiest way to do this is to specify the ODS TRACE ON statement prior to the procedure statements that created the graph. The fully qualified template name is displayed in the SAS log. This is illustrated in Example 15.7 and the section "Using the Output Delivery System" on page 274. Note that the ODS TRACE ON statement applies to graphs just as it does to tables.

Displaying Templates

Once you have found the fully qualified name of a template, you can display its definition using one of these methods:

- Open the Templates window by typing **odstemplates** (or **odst** for short) in the command line, as shown in Figure 15.13. If you expand the **Sashelp.Tmplmst** icon, you can browse all the available templates and double-click on any template icon to display its definition. This is illustrated in Example 15.7.



```
odstemplates
```

Figure 15.13. Requesting the Templates Window in the Command Line

- Use the SOURCE statement in PROC TEMPLATE to display a template definition in the SAS log. For example, the following statements display the default definition of the residual Q-Q plot in PROC ROBUSTREG.

```
proc template;
   source Stat.Robustreg.Graphics.ResidualQQPlot;
run;
```

Editing Templates

You can modify the format and appearance of a particular graph by modifying its template. There are several ways to edit a template definition:

- Find the template icon in the Templates window, right-click on the icon, and select **Edit** from the pull-down menu. This opens a Template Editor window in which you can edit the template definition. This approach is illustrated in Example 15.7.
- Find the template icon in the Templates window and double-click on the template icon to display the template definition. Copy and paste the template definition into the Program Editor.
- Use the SOURCE statement with the FILE= option in PROC TEMPLATE. This writes the template definition to a file that you can modify. For example:

  ```
  proc template;
     source Stat.Robustreg.Graphics.ResidualQQPlot /
            file = "qqtpl.sas";
  run;
  ```

  By default the file is saved in the SAS current folder. Note that with this approach you have to add a PROC TEMPLATE statement before the template definition statements and a RUN statement at the end before submitting your modified definition.

Note: Graph definitions are self-contained and do not support parenting as do table definitions. For more information about graph definitions and the graph template language see the "Introducing the Template Language for Graphics" section on page 342.

Saving Customized Templates

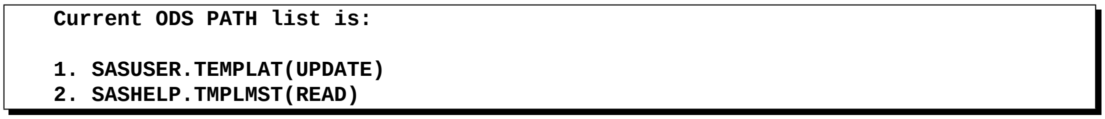
After you edit the template definition you can submit your PROC TEMPLATE statements as you would for any other SAS program:

- If you are using the Template Editor window, select **Submit** from the **Run** menu. For example, see Example 15.7.
- Alternatively, submit your PROC TEMPLATE statements in the Program Editor.

ODS automatically saves the compiled template in the first template store that it can update, according to the currently defined ODS path. If you have not changed the ODS path, then the modified template is saved in the Sasuser.Templat template store. You can display the current ODS path with the following statement.

```
ods path show;
```

By default, the result of this statement is displayed in the SAS log, as illustrated in Figure 15.14.



```
Current ODS PATH list is:

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)
```

Figure 15.14. Result of ODS PATH SHOW Statement

Using Customized Templates

When you create ODS output (either graphs or tables) with a SAS program, ODS searches sequentially through each element of the ODS PATH list for the first template that matches the ODS name of each output object requested. This template is used to produce the output object. If you have not changed the default ODS path, then the first template store searched is Sasuser.Templat, followed by Sashelp.Tmplmst.

Note that you can have templates with the same name in different template stores. The template that is used is the first one found in the ODS path.

The ODS PATH statement specifies which locations to search for definitions that were created by PROC TEMPLATE, as well as the order in which to search for them. You can change the default path by specifying different locations in the ODS PATH statement. For example, the following statement changes the default ODS path so that the first template store searched is Work.Mypath.

```
ods path work.mypath(update) sashelp.tmplmst(read);
```

The UPDATE option provides update access as well as read access to Work.Mypath. The READ option provides read-only access to Sashelp.Tmplmst.

For more information, refer to the ODS PATH Statement in the “Dictionary of ODS Language Statements” (*SAS Output Delivery System User’s Guide*).

Reverting to Default Templates

Customized templates are stored in Sasuser.Templat or in user-defined template stores. The default templates provided by SAS are saved in the read-only template store Sashelp.Tmplmst. Consequently, if you have modified any of the default templates and you want to create ODS Graphics with the original default templates, one way to do so is by changing your ODS path as follows.

```
ods path sashelp.tmplmst(read) sasuser.templat(update);
```

A second approach, which is highly recommended, is to save all your customized templates in a user-defined template store (for example Work.Mypath). Then you can reset the default ODS path with the ODS PATH RESET statement:

```
   ods path reset;
```

A third approach is to save your customized definition as part of your SAS program and delete the corresponding template from your Sasuser.Templat template store.

Example 15.7 illustrates all the steps of displaying, editing, saving and using customized templates.

Introducing the Template Language for Graphics

Graph template definitions are written in a *graph template language*, which has been added to the TEMPLATE procedure in SAS 9.1. This language includes statements for specifying plot layouts (such as grids or overlays), plot types (such as scatter plots and histograms), and text elements (such as titles, footnotes, and insets). It also provides support for built-in computations (such as histogram binning) and evaluation of expressions. Options are available for specifying colors, marker symbols, and other attributes of plot features.

Graph template definitions begin with a DEFINE STATGRAPH statement in PROC TEMPLATE, and they end with an END statement. You can specify the DYNAMIC statement to define dynamic variables, the MVAR and NMVAR statements to define macro variables, and the NOTES statement to provide descriptive information about the graph.

The statements available in the graph template language can be classified as follows:

- **Control statements**, which specify conditional or iterative flow of control. By default, flow of control is sequential. In other words, each statement is used in the order in which it appears.
- **Layout statements**, which specify the arrangement of the components of the graph. Layout statements are arranged in blocks which begin with a LAYOUT statement and end with an ENDLAYOUT statement. The blocks can be nested. Within a layout block, you can specify plot, text, and other statement types to define one or more graph components. Statement options provide control for attributes of layouts and components.
- **Plot statements**, which specify a number of commonly used displays, including scatter plots, histograms, contour plots, surface plots, and box plots. Plot statements are always provided within a layout block. The plot statements include options to specify which data columns from the source objects are used in the graph. For example, in the SCATTERPLOT statement used to define a scatter plot, there are mandatory X= and Y= options that specify which data columns are used for the x- and y-variables in the plot, and there is a GROUP= option that specifies a data column as an optional classification variable.

- **Text statements**, which specify descriptions accompanying the graphs. An entry is any textual description, including titles, footnotes, and legends, and it can include symbols to identify graph elements.

As an illustration, the following statements display the template definition of the scatter plot available in PROC KDE (see Output 36.7.1 in Chapter 36, "The KDE Procedure").

```
proc template;
   define statgraph Stat.KDE.Graphics.ScatterPlot;
      dynamic _TITLE _DEPLABEL _DEPLABEL2;
      layout Gridded;
         layout overlay / padbottom = 5;
            entrytitle _TITLE;
         endlayout;
         scatterplot x=X y=Y /
            markersymbol = GraphDataDefault:markersymbol
            markercolor  = GraphDataDefault:contrastcolor
            markersize   = GraphDataDefault:markersize;
      EndLayout;
   end;
run;
```

The DEFINE STATGRAPH statement in PROC TEMPLATE creates the graph template definition. The DYNAMIC statement defines three dynamic variables. The variable _TITLE provides the title of the graph. The variables _DEPLABEL and _DEPLABEL2 contain the names of the x and y-variables, respectively. You can use these dynamic text variables in any text element of the graph definition.

The overall display is specified with the LAYOUT GRIDDED statement. The title of the graph is specified with the ENTRYTITLE statement inside a layout overlay block, which is nested within the main layout. The main plot is a scatter plot specified with the SCATTERPLOT statement. The options in the SCATTERPLOT statement, which are given after the slash, specify the symbol, color, and size for the markers using indirect references to style attributes of the form ***`style-element:attribute`***. The values of these attributes are specified in the definition of the style you are using, and so they are automatically set to different values if you specify a different style. For more information about style references see the "Styles for Graphics" section on page 344.

The second ENDLAYOUT statement ends the main layout block and the END statement ends the graph template definition.

Note: Graph template definitions are self-contained and do not support parenting (inheritance) as do table definitions. The EDIT statement is not supported.

For details concerning the syntax of the graph template language, refer to the "TEMPLATE Procedure: Creating ODS Statistical Graphics Output (Experimental)" at http://support.sas.com/documentation/onlinedoc/base/.

Styles for Graphics

This section provides an overview of the style elements for ODS Graphics. It also describes how to customize a style definition and how to specify a default style for all your output.

Introducing Style Elements for Graphics

An ODS style definition is composed of a set of *style elements*. A style element is a collection of *style attributes* that apply to a particular feature or aspect of the output. A value is specified for each attribute in a style definition.

Style definitions control the overall appearance of ODS tables and graphs. For ODS tables, style definitions specify features such as background color, table borders, and color scheme, and they specify the fonts, sizes, and color for the text and values in a table and its headers. For ODS graphs, style definitions specify the following features:

- background color
- graph dimensions (height and width). See Example 15.13 for an illustration.
- borders
- line styles for axes and grid lines
- fonts, sizes, and colors for titles, footnotes, axis labels, axis values, and data labels. See Example 15.11 for an illustration.
- marker symbols, colors, and sizes for data points and outliers
- line styles for needles
- line and curve styles for fitted models and predicted values. See Example 15.12 for an illustration.
- line and curve styles for confidence and prediction limits
- fill colors for histogram bars, confidence bands, and confidence ellipses
- colors for box plot features
- colors for surfaces
- color ramps for contour plots

In the templates supplied by SAS for ODS graphs, options for plot features are always specified with a style reference of the form ***style-element:attribute*** rather than a hard-coded value. For example, the symbol, color, and size of markers for basic scatter plots are specified in a template SCATTERPLOT statement as follows:

```
scatterplot x=X y=Y /
   markersymbol = GraphDataDefault:markersymbol
   markercolor  = GraphDataDefault:contrastcolor
   markersize   = GraphDataDefault:markersize;
```

This guarantees a common appearance for markers used in all basic scatter plots, which is controlled by the **`GraphDataDefault`** element of the style definition that you are using.

In general, the ODS graph features listed above are determined by style element attributes unless they are overridden by a statement or option in the graph template. For example, suppose that a classification variable is specified with the GROUP= option in a template SCATTERPLOT statement such as

```
scatterplot x=X y=Y / group=GroupVar;
```

Then the colors for markers corresponding to the classification levels are assigned the style element attributes **`GraphData1:contrastcolor`** through **`GraphData12:contrastcolor`**.

In order to create your own style definition or to modify a style definition for use with ODS Graphics, you need to understand the relationships between style elements and graph features. This information is provided in the section “ODS Statistical Graphics and ODS Styles: Usage and Reference (Experimental)” at http://support.sas.com/documentation/onlinedoc/base/.

Style definitions are created and modified with the TEMPLATE procedure. For more information, refer to the “TEMPLATE Procedure: Creating a Style Definition” in the *SAS Output Delivery System User’s Guide*.

Customizing Style Definitions

The default style definitions that SAS provides are stored in the “Styles” directory of Sashelp.Tmplmst.

You can display, edit, and save style definitions using the same methods available for modifying template definitions, as explained in the sections beginning on page 339. In particular, you can display style definitions using one of these methods:

- If you are using the Templates window in the SAS windowing environment, expand the **Sashelp.Tmplmst** node under **Templates**, and then select **Styles** to display the contents of this folder.
- Use the SOURCE statement in PROC TEMPLATE. For example, the following statements display the “Journal” style definition in the SAS log.

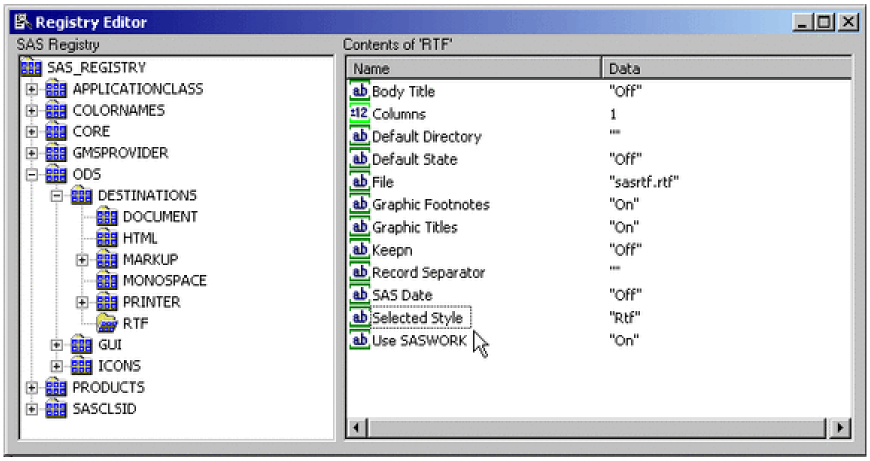
```
proc template;
   source Styles.Journal;
run;
```

Specifying a Default Style

The default style for each ODS destination is specified in the SAS Registry. For example, the default style for the HTML destination is “Default,” and for the RTF destination it is “Rtf.”

You can specify a default style for all your output in a particular ODS destination. This is useful if you want to use a different SAS-supplied style, if you have modified one of the SAS-supplied styles (see page 345), or if you have defined your own style. For example, you can specify the “Journal” style for all your RTF output.

The recommended approach for specifying a default style is as follows. Open the SAS Registry Editor by typing **regedit** in the command line. Expand the node **ODS** → **DESTINATIONS** and select a destination (for example, select **RTF**). Double-click the **Selected Style** item, as illustrated in Figure 15.15, and specify a style. This can be any SAS-supplied style or a user-defined style, as long as it can be found with the current ODS path (for example, specify **Journal**). You can specify a default style for the HTML, MARKUP, and PRINTER destinations in a similar way.



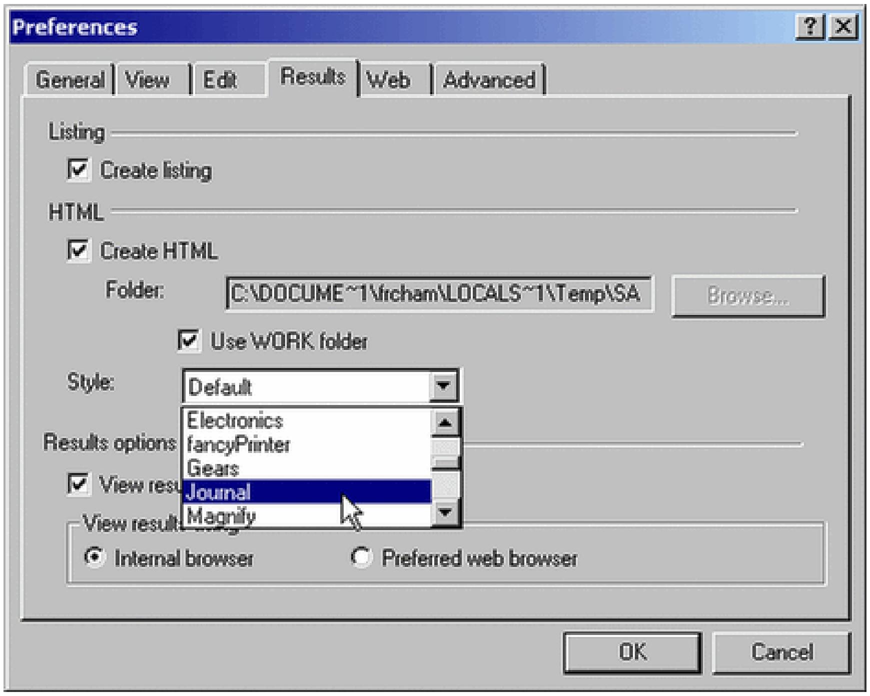
Name	Data
Body Title	"Off"
Columns	1
Default Directory	""
Default State	"Off"
File	"sasrtf.rtf"
Graphic Footnotes	"On"
Graphic Titles	"On"
Keepn	"Off"
Record Separator	""
SAS Date	"Off"
Selected Style	"Rtf"
Use SASWORK	"On"

Figure 15.15. SAS Registry Editor

Note: ODS searches sequentially through each element of the ODS PATH list for the first style definition that matches the name of the style specified in the SAS Registry. The first style definition found is used. If you are specifying a customized style as your default style, the following are useful suggestions:

- If you save your style in Sasuser.Templat, verify that the name of your default style matches the name of the style specified in the SAS Registry. For example suppose the “Rtf” style is specified for the RTF destination in the SAS Registry. You can name your style Rtf and save it in Sasuser.Templat. This blocks the “Rtf” style in Sashelp.Tmplmst.
- If you save your style in a user-defined template store, verify that this template store is the first in the current ODS PATH list. Include the ODS PATH statement in your SAS autoexec file so that it is executed at startup.

For the HTML destination, an alternative approach for specifying a default style is as follows. From the menu at the top of the main SAS window select **Tools** → **Options** → **Preferences. . . .** In the **Results** tab check the **Create HTML** box and select a style from the pull-down menu. This is illustrated in Figure 15.16.



☑ Create listing

☑ Create HTML

Folder: C:\DOCUME~1\frcham\LOCALS~1\Temp\SA

☑ Use WORK folder

Style: Default

☑ View resu

View result

● Internal browser ○ Preferred web browser

Figure 15.16. Selecting a Default Style for HTML Destination

Details

Procedures Supporting ODS Graphics

The following SAS procedures support ODS Graphics in SAS 9.1:

Base SAS

- CORR

SAS/ETS

- ARIMA
- AUTOREG
- ENTROPY
- EXPAND
- MODEL
- SYSLIN
- TIMESERIES
- UCM
- VARMAX
- X12

SAS High-Performance Forecasting

- HPF

SAS/STAT

- ANOVA
- CORRESP
- GAM
- GENMOD
- GLM
- KDE
- LIFETEST
- LOESS
- LOGISTIC
- MI
- MIXED
- PHREG
- PRINCOMP
- PRINQUAL
- REG
- ROBUSTREG

For details on the specific graphs available with a particular procedure, see the “ODS Graphics” section in the corresponding procedure chapter.

Operating Environments Supporting ODS Graphics

The following operating systems are supported:

- Windows (32- and 64- bit)
- OpenVMS Alpha
- z/OS (OS/390)
- UNIX (AIX, HP-UX, Tru64 UNIX, Solaris, Linux)

For information specific to an operating system, refer to the SAS Companion for that operating system.

Creating ODS Graphics in z/OS

Creating ODS Graphics with the z/OS (OS/390) operating system requires the following to be configured by your System Administrator:

- Java
- UNIX File System components

For more information, refer to the sections "Installing UNIX File System Components" and "Configuring SAS Software for Use with the Java Platform" of the *SAS System Configuration Guide*.

In addition, when you specify an ODS HTML destination you must specify the PATH= or GPATH= option with a valid UNIX directory.

ODS GRAPHICS Statement

The basic syntax for enabling ODS Graphics is

```
ods graphics on;
```

You specify this statement prior to your procedure statements, as illustrated in the "Using the ODS GRAPHICS Statement" section on page 321. Any procedure that supports ODS Graphics then produces graphics, either by default or when you specify procedure options for requesting particular graphs.

To disable ODS Graphics, specify

```
ods graphics off;
```

The following is a summary of the ODS GRAPHICS statement syntax. You can find the complete syntax in the section ODS Graphics Statement in the "Dictionary of ODS Language Statements" (*SAS Output Delivery System User's Guide*).

Syntax

ODS GRAPHICS < **OFF | ON** < / *options* > > **;**

enables ODS to create graphics automatically. The default is ON.

Options

ANTIALIAS | NOANTIALIAS
ANTIALIAS = ON | OFF

controls the use of antialiasing to smooth the components of a graph.

OFF

suppresses the use of antialiasing for components other than text.

ON

specifies that antialiasing is to be used to smooth jagged edges of all of the components in a graph.

Text displayed in a graph is always antialiased. If the number of observations in the ODS output object exceeds 250, then antialiasing is not used, even if you specify the option ANTIALIAS=ON. The default is ON.

IMAGEFMT = < *image-file-type* **| STATIC | STATICMAP** >

specifies the image file type (directly or indirectly) for displaying graphics in ODS output. The default image file type depends on the ODS destination; it is used when you specify IMAGEFMT=STATIC. You can also specify other supported image file types. This option only applies to ODS Graphics, and it has no effect on traditional high resolution graphics that rely on GOPTIONS values. The default is STATIC.

image-file-type

specifies the type of image you want to add to your graph. If the image file type is not valid for the active output destination, the default is used instead. Table 15.3 lists the image file types supported for the ODS destinations that are valid with ODS Graphics.

STATIC

specifies the best quality image file type for the active output destination.

STATICMAP

applies only with the HTML destination and specifies that an HTML image map is to be created for tool tip support. The image file type used is the same as with STATIC. For an illustration see Example 15.2. If the number of observations in the data set exceeds 500, the image map is not generated.

Table 15.3. Supported Destinations and Image File Types

Destination	**Values for IMAGEFMT= Option**
HTML	GIF (default), JPEG, PNG
LATEX	PS (default), EPSI, GIF, JPEG, PNG
PCL	Not applicable
PDF	Not applicable
PS	Not applicable
RTF	Not applicable

Note: For PCL, PDF, PS, and RTF, the IMAGEFMT= option is not applicable because the graph is contained in the output file. See Table 15.2.

IMAGENAME = <*file-name*>

specifies the base image file name. The default is the name of the output object. You can determine the name of the output object by using the ODS TRACE statement. The base image name should not include an extension. ODS automatically adds the increment value and the appropriate extension (which is specific to the output destination that has been selected).

RESET

resets the index counter appended to image file names.

Note: The index counter is initialized to zero at the beginning of your SAS session or if you specify the RESET option in the ODS GRAPHICS statement. Graphics image files with the same name are overwritten.

Label Collision Avoidance

Label collision avoidance is supported in ODS Graphics by an algorithm which positions point labels so that label overlap is minimized. However, if the number of data labels is greater than 100, the labels are displayed at fixed positions and collision avoidance is not attempted.

Label collision avoidance is illustrated in Figure 15.17, which is a scatter plot taken from Example 24.3 in Chapter 24, "The CORRESP Procedure."

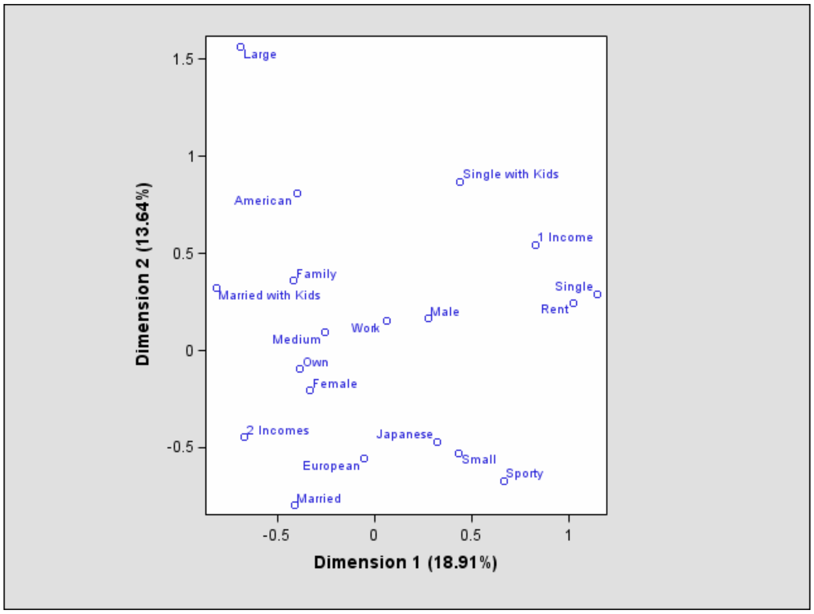


Figure 15.17. Label Collision Avoidance

Examples

This section provides a series of examples which illustrate various tasks that can be performed with ODS Graphics. The examples are presented in increasing order of task complexity and should be read sequentially.

Example 15.1. Selecting and Excluding Graphs

This example illustrates how to select and exclude ODS graphs from your output.

The "Getting Started" example on page 321 uses the REG procedure to produce a panel, shown in Figure 15.1, which consists of eight different diagnostics plots. The panel is produced by default. To display the plots individually, specify the PLOTS(UNPACK) option in the PROC REG statement as follows:

```
ods trace on;

ods html;
ods graphics on;

proc reg data = Class plots(unpack);
   model Weight = Height;
run;
quit;

ods graphics off;
ods html close;

ods trace off;
```

The ODS TRACE ON statement requests a record of the output objects created by ODS, which is displayed in the SAS log as shown in Output 15.1.1.

Output 15.1.1. Partial ODS Trace Record in SAS Log

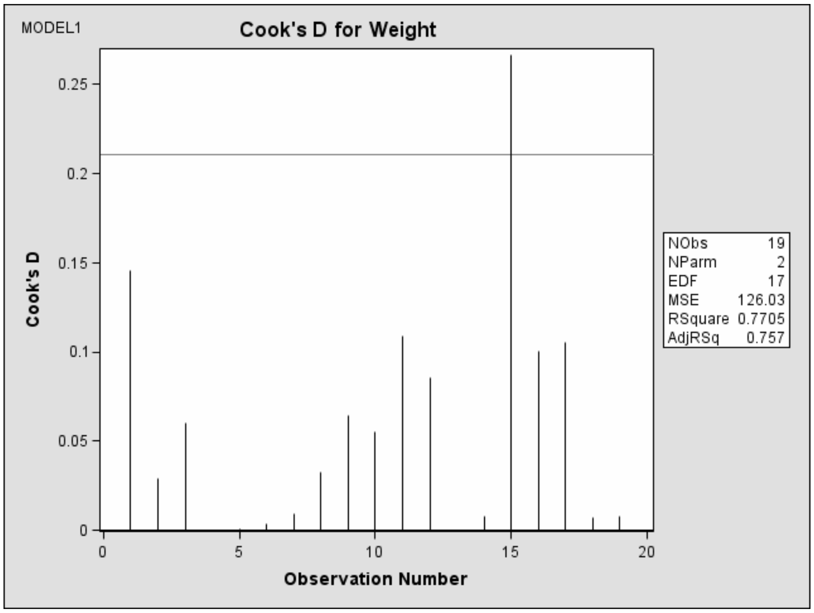
```
Output Added:
-------------
Name:       NObs
Label:      Number of Observations
Template:   Stat.Reg.NObs
Path:       Reg.MODEL1.Fit.Weight.NObs
-------------
   .
   .
   .
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL1.Fit.Weight.ParameterEstimates
-------------
WARNING: Statistical graphics displays created with ODS are experimental in
         this release.

Output Added:
-------------
Name:       ResidualHistogram
Label:      Residual Histogram
Template:   Stat.REG.Graphics.ResidualHistogram
Path:       Reg.MODEL1.ObswiseStats.Weight.DiagnosticPlots.ResidualHistogram
-------------
   .
   .
   .

Output Added:
-------------
Name:       CooksD
Label:      Cook’s D
Template:   Stat.REG.Graphics.CooksD
Path:       Reg.MODEL1.ObswiseStats.Weight.DiagnosticPlots.CooksD
-------------
   .
   .
   .

Output Added:
-------------
Name:       Fit
Label:      Fit Plot
Template:   Stat.REG.Graphics.Fit
Path:       Reg.MODEL1.ObswiseStats.Weight.DiagnosticPlots.Fit
-------------
```

You can use the ODS SELECT statement to restrict your output to a particular subset of ODS tables or graphs. The following statements restrict the output to the Cook’s D plot, which is shown in Output 15.1.2.

```
ods html;
ods graphics on;

ods select CooksD;

proc reg data = Class plots(unpack);
   model Weight = Height;
```

```
run;
quit;

ods graphics off;
ods html close;
```

Output 15.1.2. Cook's D Plot



Conversely, you can use the ODS EXCLUDE statement to display all the output with the exception of a particular subset of tables or graphs. For example, to exclude the fit plot from the output you specify

```
ods exclude Fit;
```

See the "Selecting and Excluding Graphs" section on page 330 for further information.

A sample program named odsgr01.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.2. Creating Graphs with Tool Tips in HTML

This example demonstrates how to request graphics in HTML with tool tip displays, which appear when you move a mouse over certain features of the graph. When you specify the HTML destination and the IMAGEFMT=STATICMAP option in the ODS GRAPHICS statement, then the HTML file output file is generated with an image map of coordinates for tool tips. The individual graphs are saved as GIF files.

Example 46.2 and Example 46.8 of Chapter 46, “The MIXED Procedure” analyze a data set with repeated growth measurements for twenty-seven children.

```
data pr;
   input Person Gender $ y1 y2 y3 y4 @@;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
datalines;
 1  F  21.0  20.0  21.5  23.0      2  F  21.0  21.5  24.0  25.5
 3  F  20.5  24.0  24.5  26.0      4  F  23.5  24.5  25.0  26.5
 5  F  21.5  23.0  22.5  23.5      6  F  20.0  21.0  21.0  22.5
 7  F  21.5  22.5  23.0  25.0      8  F  23.0  23.0  23.5  24.0
 9  F  20.0  21.0  22.0  21.5     10  F  16.5  19.0  19.0  19.5
11  F  24.5  25.0  28.0  28.0     12  M  26.0  25.0  29.0  31.0
13  M  21.5  22.5  23.0  26.5     14  M  23.0  22.5  24.0  27.5
15  M  25.5  27.5  26.5  27.0     16  M  20.0  23.5  22.5  26.0
17  M  24.5  25.5  27.0  28.5     18  M  22.0  22.0  24.5  26.5
19  M  24.0  21.5  24.5  25.5     20  M  23.0  20.5  31.0  26.0
21  M  27.5  28.0  31.0  31.5     22  M  23.0  23.0  23.5  25.0
23  M  21.5  23.5  24.0  28.0     24  M  17.0  24.5  26.0  29.5
25  M  22.5  25.5  25.5  26.0     26  M  23.0  24.5  26.0  30.0
27  M  22.0  21.5  23.5  25.0
;
```

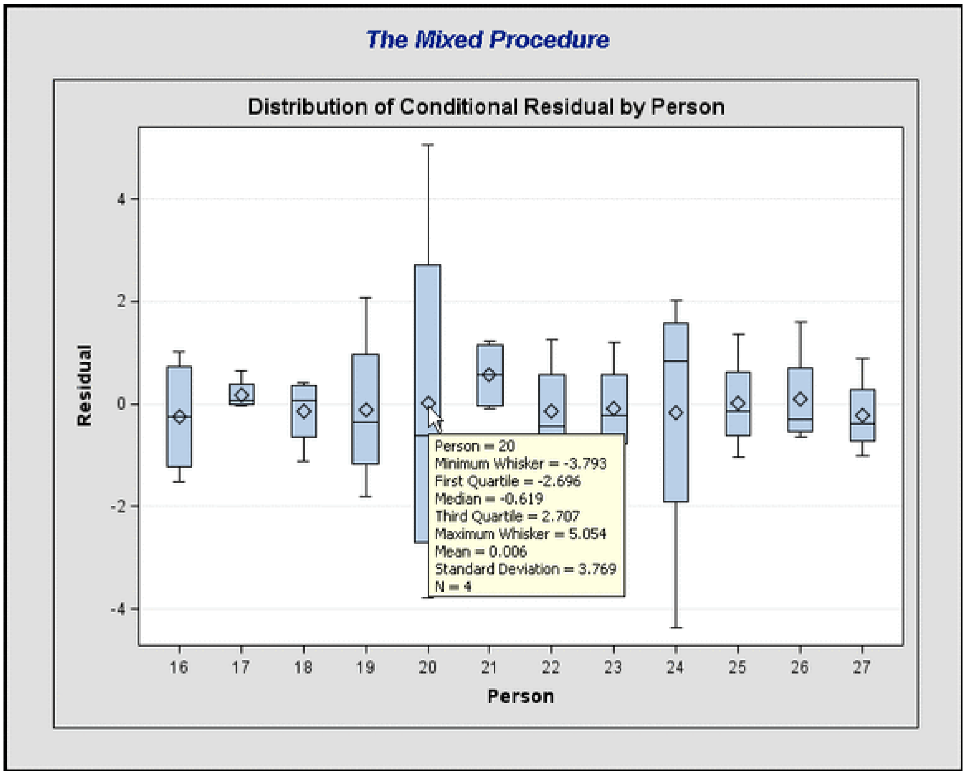
The following statements fit a mixed model with random intercepts and slopes for each child. The BOXPLOT option in the PROC MIXED statement requests box plots of observed values and residuals for each classification main effect in the model (Gender and Person).

```
ods html;
ods graphics on / imagefmt = staticmap;

proc mixed data=pr method=ml boxplot(npanel=15);
   class Person Gender;
   model y = Gender Age Gender*Age;
   random intercept Age / type=un subject=Person;
run;

ods graphics off;
ods html close;
```

The NPANEL=15 suboption limits the number of box plots per graph to at most fifteen. For example, the conditional residuals of the Person effect are displayed in two graphs, consisting of 15 and 12 box plots, respectively. Output 15.2.1 displays the second of these two graphs that are included in the HTML output.

Moving the mouse over a box plot displays a tool tip with summary statistics for the corresponding person.

Output 15.2.1. Box Plot with Tool Tips



Person = 20
Minimum Whisker = -3.793
First Quartile = -2.696
Median = -0.619
Third Quartile = 2.707
Maximum Whisker = 5.054
Mean = 0.006
Standard Deviation = 3.769
N = 4

Note: Graphics with tool tips are only supported for the HTML destination.

A sample program named odsgr03.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.3. Creating Graphs for a Presentation

The RTF destination provides the easiest way to create ODS graphs for inclusion in a document or presentation. You can specify the ODS RTF statement to create a file that is easily imported into a word processor (such as Microsoft Word or WordPerfect) or a presentation (such as Microsoft PowerPoint).

The following statements simulate 100 observations from the model $y = \log(x) + w$, where $x = 1, \ldots, 100$ and w has a normal distribution with mean 0 and variance 1.

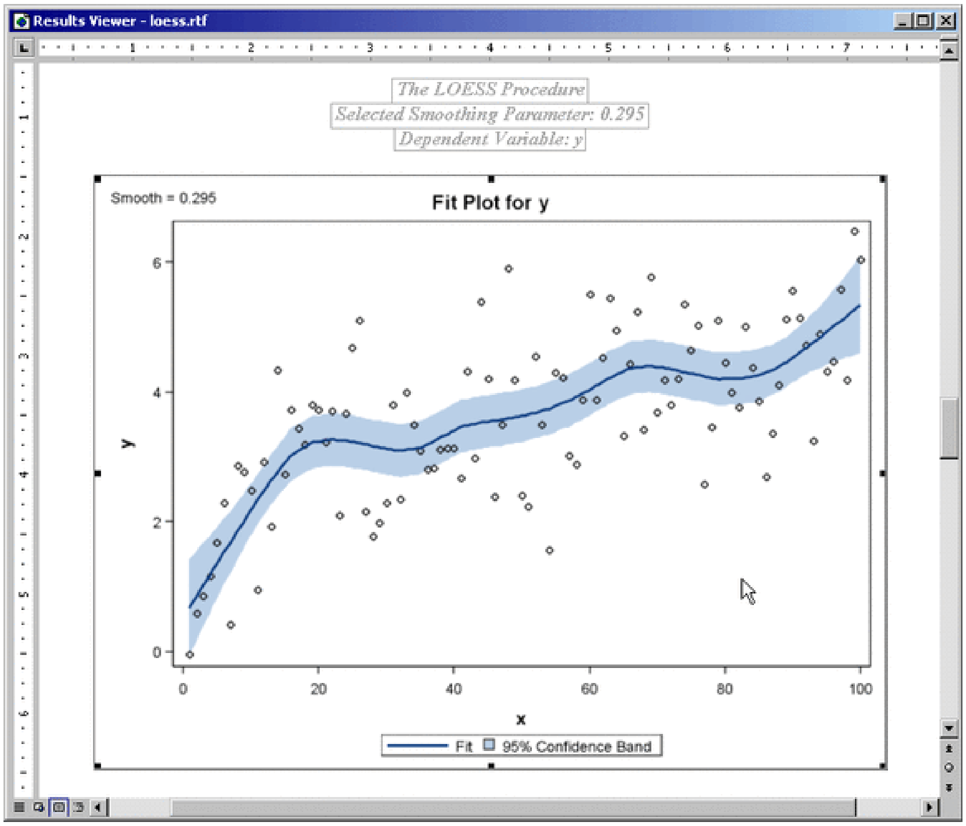
```
data one;
   do x = 1 to 100;
      y = log(x) + rannor(12345);
      output;
   end;
run;
```

The following statements request a loess fit and save the output in the file loess.rtf.

```
ods rtf file = "loess.rtf";
ods graphics on;

proc loess data = one;
   model y = x / clm residual;
run;

ods graphics off;
ods rtf close;
```

The output file includes various tables and the following plots: a plot of selection criterion versus smoothing parameter, a fit plot with 95% confidence bands, a plot of residual by regressors, and a diagnostics panel. The fit plot is shown in Output 15.3.1.

Output 15.3.1. Fit Plot



The LOESS Procedure

Selected Smoothing Parameter: 0.295

Dependent Variable: y

If you are running SAS in the Windows operating system, it is easy to include your graphs in a Microsoft PowerPoint presentation when you generate RTF output. You can open the RTF file in Microsoft Word and simply copy and paste the graphs into Microsoft PowerPoint. In general, RTF output is convenient for exchange of graphical results between Windows applications through the clipboard.

Alternatively, if you request ODS Graphics using the HTML destination, then your individual graphs are created as GIF files by default. You can insert the GIF files into a Microsoft PowerPoint presentation. See "Naming Graphics Image Files" and "Saving Graphics Image Files" for information on how the image files are named and saved.

A sample program named odsgr04.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.4. Creating Graphs in PostScript Files

This example illustrates how to create individual graphs in PostScript files, which is particularly useful when you want to include them in a LaTeX document.

The "Getting Started" section of Chapter 62, "The ROBUSTREG Procedure," creates the following data set to illustrate the use of the ROBUSTREG procedure for robust regression.

```
data stack;
   input  x1 x2 x3 y @@;
   datalines;
80  27  89  42    80  27  88  37    75  25  90  37
62  24  87  28    62  22  87  18    62  23  87  18
62  24  93  19    62  24  93  20    58  23  87  15
58  18  80  14    58  18  89  14    58  17  88  13
58  18  82  11    58  19  93  12    50  18  89   8
50  18  86   7    50  19  72   8    50  19  79   8
50  20  80   9    56  20  82  15    70  20  91  15
;
```

The following statements specify a LATEX destination[*] with the "Journal" style, and request a histogram of standardized robust residuals computed with the ROBUSTREG procedure.

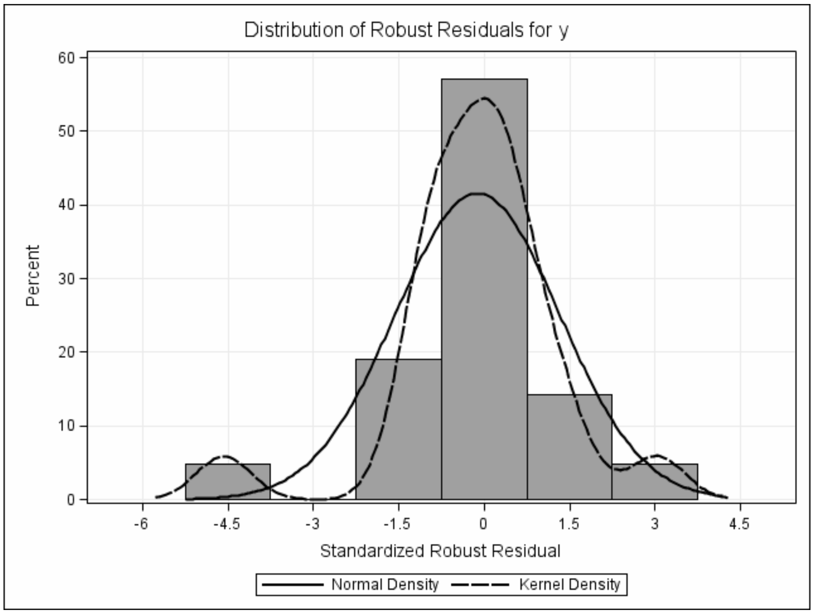
```
ods latex style = Journal;
ods graphics on;

proc robustreg plot=reshistogram data=stack;
   model y = x1 x2 x3;
run;

ods graphics off;
ods latex close;
```

The "Journal" style displays gray-scale graphs that are suitable for a journal. When you specify the ODS LATEX destination, ODS creates a PostScript file for each individual graph in addition to a LaTeX source file that includes the tabular output and references to the PostScript files. By default these files are saved in the SAS current folder. If you run this example at the beginning of your SAS session, the histogram

[*] The LATEX destination in ODS is experimental in SAS 9.1.

shown in Output 15.4.1 is saved by default in a file named ResidualHistogram0.ps. See page 335 for details about how graphics image files are named.

Output 15.4.1. Histogram Using Journal Style



If you are writing a paper, you can include the graphs in your own LaTeX source file by referencing the names of the individual PostScript graphics files. In this situation, you may not find necessary to use the LaTeX source file created by SAS.

If you specify PATH= and GPATH= options in the ODS LATEX statement, your tabular output is saved as a LaTeX source file in the directory specified with the PATH= option, and your graphs are saved as PostScript files in the directory specified with the GPATH= option. This is illustrated by the following statements:

```
ods latex path  = "C:\temp"
          gpath = "C:\temp\ps" (url="ps/")
          style = Journal;
ods graphics on;

   ...SAS statements...

ods graphics off;
ods latex close;
```

The URL= suboption is specified in the GPATH= option to create relative paths for graphs referenced in the LaTeX source file created by SAS. See the "HTML Destination" section on page 336 for further information.

A sample program named odsgr05.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.5. Creating Graphs in Multiple Destinations

This example illustrates how to send your output to more than one destination with a single execution of your SAS statements.

For instance, to create both HTML and RTF output, you can specify the ODS HTML and the ODS RTF statements before your procedure statements.

```
ods html;
ods rtf;

   ...SAS statements...

ods _all_ close;
```

The ODS _ALL_ CLOSE statement closes all open destinations.

You can also specify multiple instances of the same destination. For example, using the data in the "Using the ODS GRAPHICS Statement and Procedure Options" section on page 324, the following statements save the contour plot to the file contour.pdf and the surface plot to the file surface.pdf.

```
ods pdf file = "contour.pdf";
ods pdf select Contour;

ods pdf(id=srf) file = "surface.pdf";
ods pdf(id=srf) select SurfacePlot;

ods graphics on;

proc kde data = bivnormal;
   bivar x y / plots = contour surface;
run;

ods graphics off;
ods _all_ close;
```

The ID= option assigns the name srf to the second instance of the PDF destination. Without the ID= option, the second ODS PDF statement would close the destination that was opened by the previous ODS PDF statement, and it would open a new instance of the PDF destination. In that case, the file contour.pdf would contain no output. For more information, refer to the Example 1 of the ODS PDF statement in the "Dictionary of ODS Language Statements" (*SAS Output Delivery System User's Guide*).

Example 15.6. Displaying Graphs Using the DOCUMENT Procedure

This example illustrates the use of the DOCUMENT destination and the DOCUMENT procedure to display your ODS graphs. In particular, this is useful when you want to display your output (both tables and graphs) in one or more ODS destinations, or when you want to use different styles without rerunning your SAS program.

In general, when you send your output to the DOCUMENT destination you can use the DOCUMENT procedure to rearrange, duplicate, or remove output from the results of a procedure or a database query. You can also generate output for one or more ODS destinations. For more information, refer to the ODS DOCUMENT statement in the “Dictionary of ODS Language Statements” and “The DOCUMENT Procedure” (*SAS Output Delivery System User’s Guide*).

The following statements request a Q-Q plot using PROC ROBUSTREG with the stack data from Example 15.4. The ODS DOCUMENT statement stores the data for the tables and the residual Q-Q plot from this analysis in an ODS document named QQDoc. Neither the tables nor the plot are displayed.

```
ods listing close;
ods document name = QQDoc(write);
ods graphics on;

proc robustreg plot=resqqplot data=stack;
   model y = x1 x2 x3;
run;
quit;

ods graphics off;
ods document close;
ods listing;
```

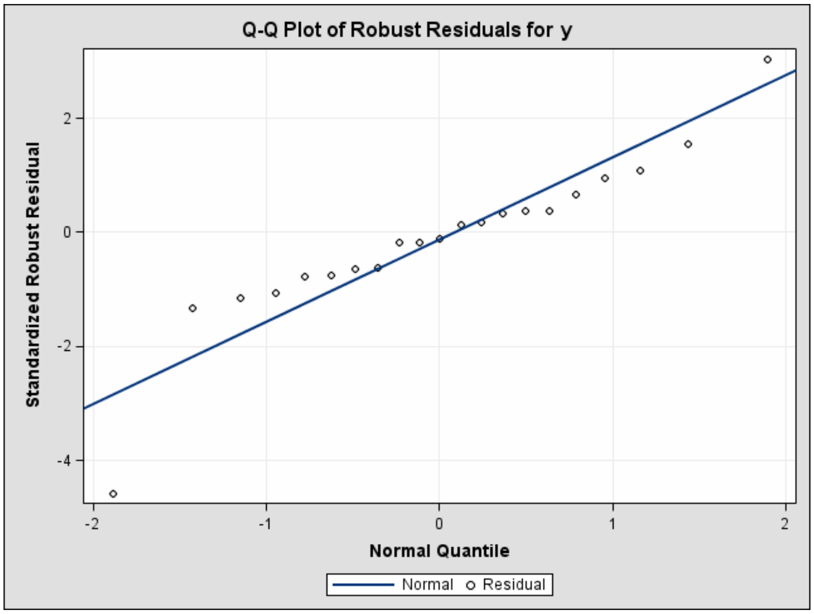
In order to display the Q-Q plot using PROC DOCUMENT, you first need to determine its name. You can do this by specifying the ODS TRACE ON statement prior to the procedure statements (see page 330 for more information). Alternatively, you can type **odsdocuments** (or **odsd** for short) in the command line to open the Documents window, which you can then use to manage your ODS documents.

The following statements specify an HTML destination and display the residual Q-Q plot using the REPLAY statement in PROC DOCUMENT.

```
ods html;
ods select ResidualQQPlot;

proc document name = QQDoc;
   replay;
run;
quit;

ods html close;
```

By default, the REPLAY statement attempts to replay every output object stored in the document, but only the Q-Q plot is displayed as specified by the ODS SELECT statement. The plot is displayed in Output 15.6.1.

Output 15.6.1. Q-Q Plot Displayed by PROC DOCUMENT



As an alternative to running PROC DOCUMENT with an ODS SELECT statement, you can run PROC DOCUMENT specifying a *document path* for the Q-Q plot in the REPLAY statement. This approach is preferable when the document contains a large volume of output, because PROC DOCUMENT does not attempt to process every piece of output stored in the document.

You can determine the document path for the Q-Q plot by specifying the LIST statement with the LEVELS=ALL option in PROC DOCUMENT.

```
proc document name = QQDoc;
   list / levels = all;
run;
quit;
```

This lists the entries of the QQDoc document, as shown in Output 15.6.2.

Output 15.6.2. Contents of QQDoc

```
Listing of: \Work.Qqdoc\
Order by: Insertion
Number of levels: All

  Obs    Path                                                              Type
--------------------------------------------------------------------------------
       1 \Robustreg#1                                                      Dir
       2 \Robustreg#1\ModelInfo#1                                          Table
       3 \Robustreg#1\NObs#1                                               Table
       4 \Robustreg#1\SummaryStatistics#1                                  Table
       5 \Robustreg#1\ParameterEstimates#1                                 Table
       6 \Robustreg#1\DiagSummary#1                                        Table
       7 \Robustreg#1\ResidualQQPlot#1                                     Graph
       8 \Robustreg#1\GoodFit#1                                            Table
```

The document path of the “ResidualQQPlot” entry in QQDoc, as shown in Output 15.6.2, is

```
\Robustreg#1\ResidualQQPlot#1
```

You can specify this path to display the residual Q-Q plot with PROC DOCUMENT as follows.

```
ods html;

proc document name = QQDoc;
   replay \Robustreg#1\ResidualQQPlot#1;
run;
quit;

ods html close;
```

You can also determine the document path from the Results window or the Documents window. Right-click on the object icon and select **Properties**.

A sample program named odsgr06.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.7. Customizing Graph Titles and Axes Labels

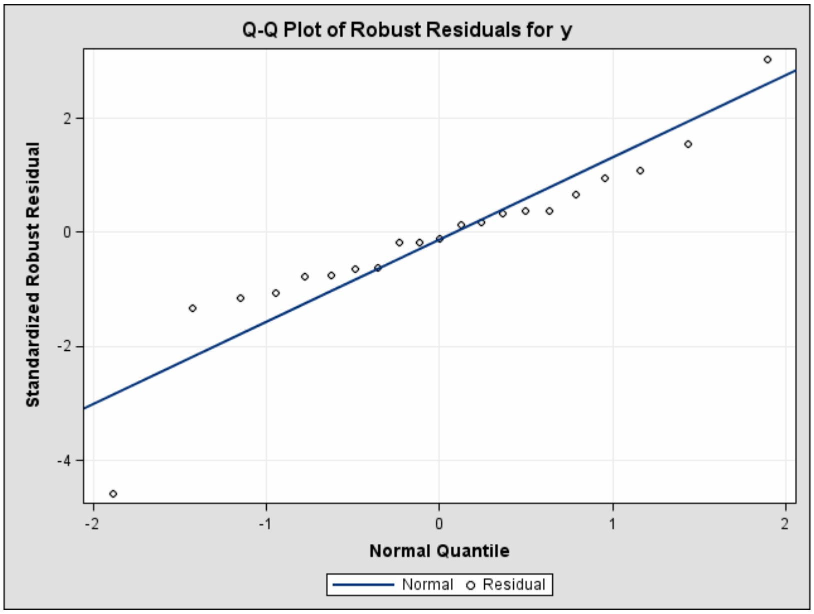
This example shows how to use PROC TEMPLATE to customize the appearance and content of an ODS graph. It illustrates the discussion in the section “Customizing Graphics with Templates” on page 338 in the context of changing the default title and y-axis label for a Q-Q plot created with the ROBUSTREG procedure.

The following statements request a Q-Q plot for robust residuals using PROC ROBUSTREG with the stack data from Example 15.4.

```
ods trace on;
ods html;
ods graphics on;

ods select ResidualQQPlot;

proc robustreg plot=resqqplot data=stack;
   model y = x1 x2 x3;
run;

ods graphics off;
ods html close;
ods trace off;
```

The Q-Q plot is shown in Output 15.7.1.

Output 15.7.1. Default Q-Q Plot from PROC ROBUSTREG



The ODS TRACE ON statement requests a record of all the ODS output objects created by PROC ROBUSTREG. A partial listing of the trace record, which is displayed in the SAS log, is shown in Output 15.7.2.

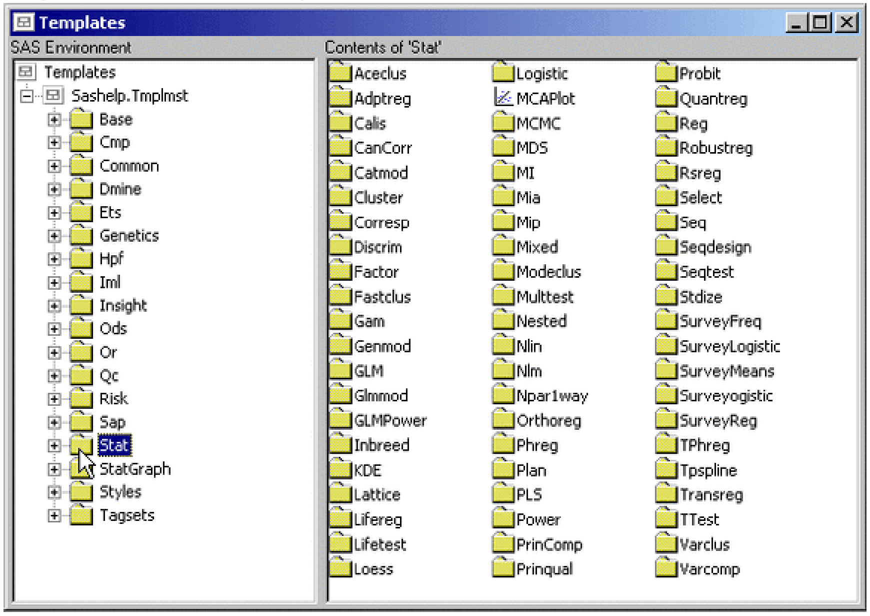
Output 15.7.2. Trace Record for Q-Q Plot

```
Output Added:
-------------
Name:       ResidualQQPlot
Label:      ResidualQQPlot
Template:   Stat.Robustreg.Graphics.ResidualQQPlot
Path:       Robustreg.Graphics.ResidualQQPlot
-------------
```

As shown in Output 15.7.2, ODS Graphics creates the Q-Q plot using an ODS output data object named “ResidualQQPlot” and a graph template named “Stat.Robustreg.Graphics.ResidualQQPlot,” which is the default template provided by SAS. Default templates supplied by SAS are saved in the Sashelp.Tmplmst template store (see page 338).

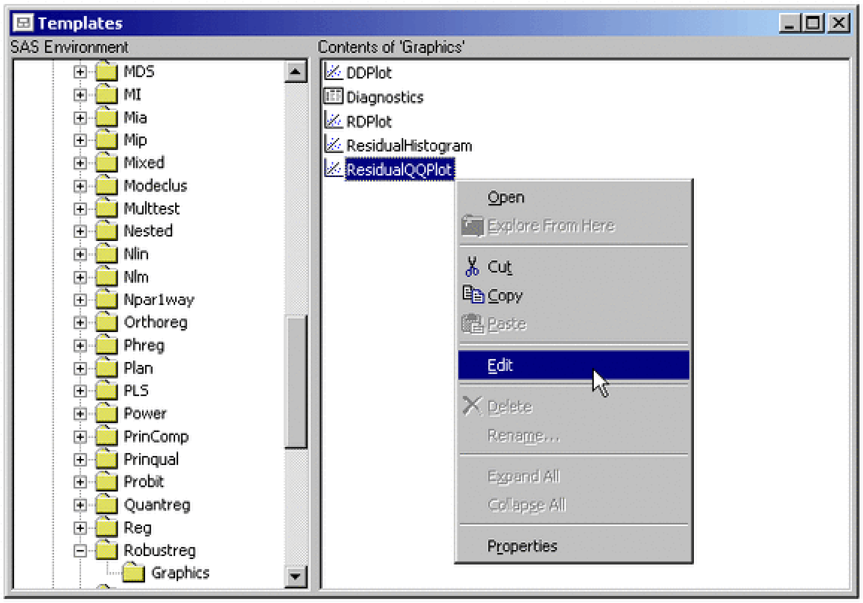
To display the default template definition, open the Templates window by typing **odstemplates** (or **odst** for short) in the command line. Expand **Sashelp.Tmplmst** and click on the **Stat** folder, as shown in Output 15.7.3.

Output 15.7.3. The Templates Window



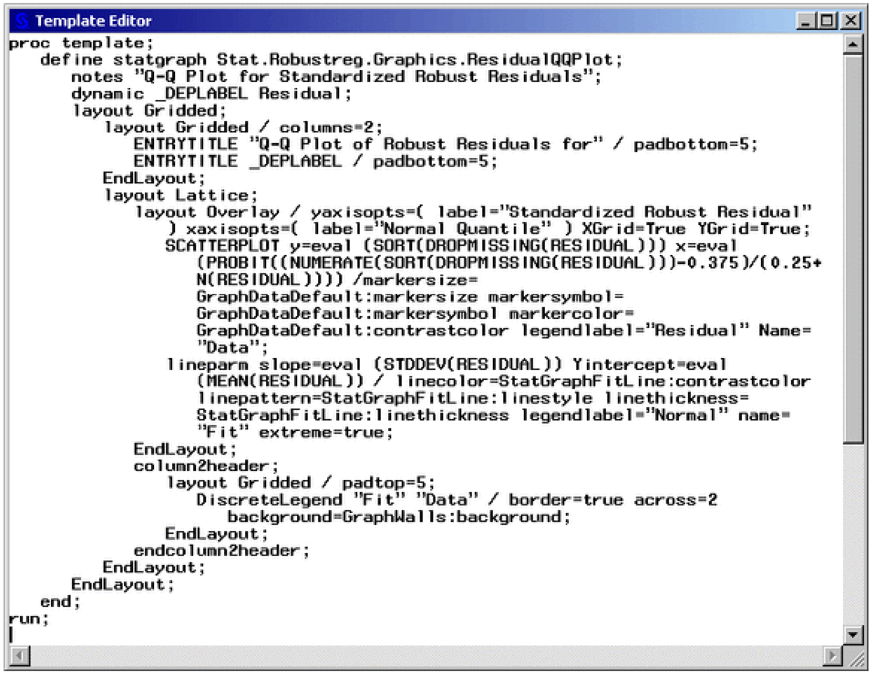
Next, open the **Robustreg** folder and then open the **Graphics** folder. Then right-click on the “ResidualQQPlot” template icon and select **Edit**, as shown in Output 15.7.4.

Output 15.7.4. Editing Templates in the Template Window



Selecting **Edit** opens a Template Editor window, as shown in Output 15.7.5. You can use this window to edit the template.

Output 15.7.5. Default Template Definition for Q-Q Plot



```
proc template;
   define statgraph Stat.Robustreg.Graphics.ResidualQQPlot;
      notes "Q-Q Plot for Standardized Robust Residuals";
      dynamic _DEPLABEL Residual;
      layout Gridded;
         layout Gridded / columns=2;
            ENTRYTITLE "Q-Q Plot of Robust Residuals for" / padbottom=5;
            ENTRYTITLE _DEPLABEL / padbottom=5;
         EndLayout;
         layout Lattice;
            layout Overlay / yaxisopts=( label="Standardized Robust Residual"
               ) xaxisopts=( label="Normal Quantile" ) XGrid=True YGrid=True;
               SCATTERPLOT y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval
                  (PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL)))-0.375)/(0.25+
                  N(RESIDUAL)))) /markersize=
                  GraphDataDefault:markersize markersymbol=
                  GraphDataDefault:markersymbol markercolor=
                  GraphDataDefault:contrastcolor legendlabel="Residual" Name=
                  "Data";
               lineparm slope=eval (STDDEV(RESIDUAL)) Yintercept=eval
                  (MEAN(RESIDUAL)) / linecolor=StatGraphFitLine:contrastcolor
                  linepattern=StatGraphFitLine:linestyle linethickness=
                  StatGraphFitLine:linethickness legendlabel="Normal" name=
                  "Fit" extreme=true;
            EndLayout;
            column2header;
               layout Gridded / padtop=5;
                  DiscreteLegend "Fit" "Data" / border=true across=2
                     background=GraphWalls:background;
               EndLayout;
            endcolumn2header;
         EndLayout;
      EndLayout;
   end;
run;
```

The template definition in Output 15.7.5 is discussed below and in subsequent examples. It is listed in a more convenient format by the following statements:

```
proc template;
define statgraph Stat.Robustreg.Graphics.ResidualQQPlot;
   notes "Q-Q Plot for Standardized Robust Residuals";
   dynamic _DEPLABEL Residual;
   layout Gridded;
      layout Gridded / columns = 2;
         ENTRYTITLE "Q-Q Plot of Robust Residuals for" / padbottom = 5;
         ENTRYTITLE _DEPLABEL / padbottom = 5;
      EndLayout;
      layout Lattice;
         layout Overlay /
               yaxisopts = (label = "Standardized Robust Residual")
               xaxisopts = (label = "Normal Quantile")
               XGrid = True YGrid = True;
            SCATTERPLOT
               y = eval(SORT(DROPMISSING(RESIDUAL)))
               x = eval(PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL)))
                        -0.375)/(0.25+N(RESIDUAL)))) /
                  markersize   = GraphDataDefault:markersize
                  markersymbol = GraphDataDefault:markersymbol
                  markercolor  = GraphDataDefault:contrastcolor
                  legendlabel  = "Residual"
                  name         = "Data";
            lineparm
               slope      = eval(STDDEV(RESIDUAL))
               Yintercept = eval(MEAN(RESIDUAL)) /
                  linecolor     = StatGraphFitLine:contrastcolor
                  linepattern   = StatGraphFitLine:linestyle
                  linethickness = StatGraphFitLine:linethickness
                  legendlabel   = "Normal"
                  name          = "Fit"
                  extreme       = true;
         EndLayout;
         column2header;
            layout Gridded / padtop = 5;
               DiscreteLegend "Fit" "Data" /
                  border     = true
                  across     = 2
                  background = GraphWalls:background;
            EndLayout;
         endcolumn2header;
      EndLayout;
   EndLayout;
end;
run;
```

As an alternative to using the Template Editor window, you can submit the following statements, which display the “ResidualQQPlot” template definition in the SAS log.

```
proc template;
   source Stat.Robustreg.Graphics.ResidualQQPlot;
run;
```

The SOURCE statement specifies the fully qualified template name. You can copy and paste the template source into the Program Editor, modify it, and submit it using PROC TEMPLATE. See the "Editing Templates" section on page 340 for more information.

In the template, the default title of the Q-Q plot is specified by the two ENTRYTITLE statements. Note that _DEPLABEL is a dynamic variable that provides the name of the dependent variable in the regression analysis (the name happens to be y in Output 15.7.1). The default label for the y-axis is specified by the LABEL= suboption of the YAXISOPTS= option for the LAYOUT OVERLAY statement.

Suppose you want to change the default title to My Favorite Title, and you want the y-axis label to display the name of the dependent variable. First, replace the two ENTRYTITLE statements with the single statement

```
   ENTRYTITLE "My Favorite Title" / padbottom = 5;
```

The PADBOTTOM= option specifies the amount of empty space (in pixel units) at the bottom of the layout component. In this case it creates an empty space of 5 pixels between the title and the adjacent layout component, which defines the plot itself.

Next, replace the LABEL= suboption with the following:

```
   label = _DEPLABEL
```

Note that you can reuse dynamic text variables such as _DEPLABEL in any text element.

You can then submit the modified template definition as you would any SAS program, for example, by selecting **Submit** from the **Run** menu.

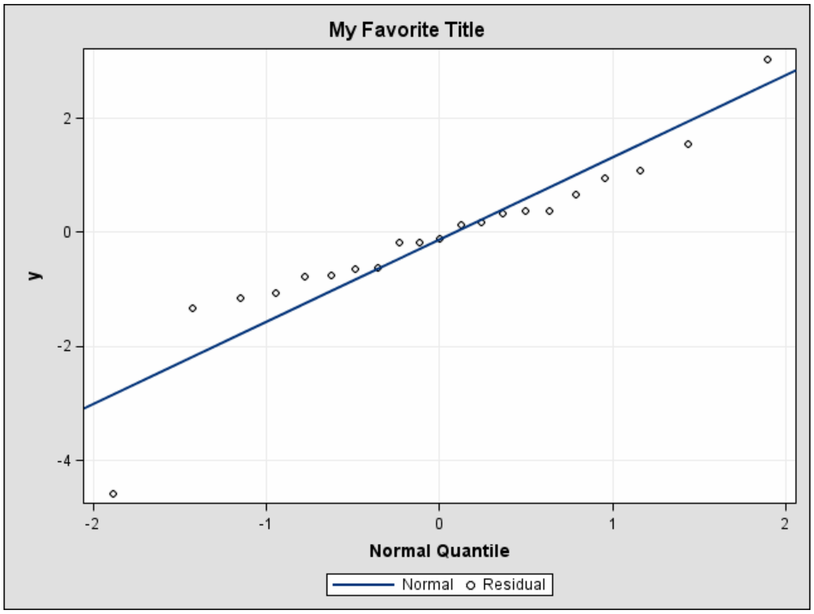
After submitting the PROC TEMPLATE statements you should see the following message in the SAS log:

```
 NOTE: STATGRAPH 'Stat.Robustreg.Graphics.ResidualQQPlot' has been
   saved to: SASUSER.TEMPLAT
```

Note: Graph definitions are self-contained and do not support parenting as do table definitions. For more information about graph definitions and the graph template language, see the "Introducing the Template Language for Graphics" section on page 342.

Finally, resubmit the PROC ROBUSTREG statements on page 363 to display the Q-Q plot created with your modified template, as shown in Output 15.7.6.

Output 15.7.6. Q-Q Plot with Modified Title and Y-Axis Label



If you have not changed the default ODS path, the modified template "ResidualQQPlot" is used automatically because Sasuser.Templat occurs before Sashelp.Tmplmst in the ODS search path. See the "Using Customized Templates" section on page 341 for additional information.

Note that you do not need to rerun the PROC ROBUSTREG analysis after you modify a graph template. After you modify your template, you can submit the PROC DOCUMENT statements in Example 15.6 to replay the Q-Q plot with the modified template.

See the "Reverting to Default Templates" section on page 341 for information on how to revert to the default template.

A sample program named odsgr07.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.8. Modifying Colors, Line Styles, and Markers

This example is a continuation of Example 15.7. Here the objective is to customize colors, line attributes, and marker symbol attributes by modifying the graph template.

In the "ResidualQQPlot" template definition shown in Output 15.7.5, the SCATTERPLOT statement specifies a scatter plot of normal quantiles versus ordered standardized residuals. The default marker symbol in the scatter plot is specified by the MARKERSYMBOL= option of the SCATTERPLOT statement:

```
   markersymbol = GraphDataDefault:markersymbol
```

The default value is a reference to the style attribute **markersymbol** of the style element **GraphDataDefault**. See the "Introducing Style Elements for Graphics" section on page 344 for more information. The actual value of the marker symbol depends on the style that you are using. In this case, since the "Default" style is used, the value of the marker symbol is Circle.

You can specify a filled circle as the marker symbol by modifying the value of the MARKERSYMBOL= option as follows.

```
markersymbol = CircleFilled
```

Note that the value of the option can be any valid marker symbol or a reference to a style attribute of the form ***style-element:attribute***. It is recommended that you use style attributes since these are chosen to provide consistency and appropriate emphasis based on display principles for statistical graphics. If you specify values directly in a template, you are overriding the style and run the risk of creating a graph that is inconsistent with the style definition.

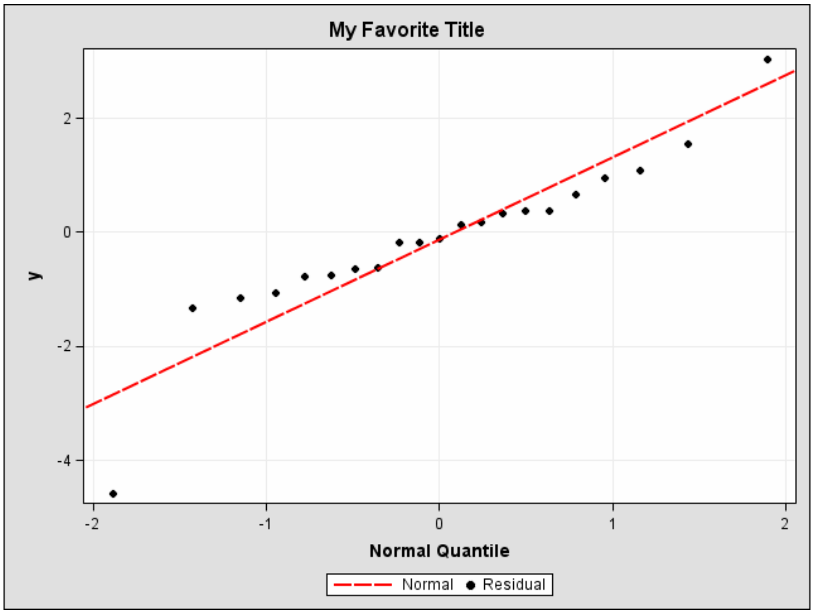
For more information about the syntax of the graphics template language and style elements for graphics, refer to the sections "TEMPLATE Procedure: Creating ODS Statistical Graphics Output (Experimental)" and "ODS Statistical Graphics and ODS Styles: Usage and Reference (Experimental)" at http://support.sas.com/documentation/onlinedoc/base/.

Similarly, you can change the line color and pattern with the LINECOLOR= and LINEPATTERN= options in the LINEPARM statement. The LINEPARM statement displays a straight line specified by slope and intercept parameters. The following statements change the default color of the Q-Q plot line to red, and the line pattern to dashed.

```
linecolor   = red
linepattern = dash
```

To display these modifications, shown in Output 15.8.1, submit the modified template definition and then resubmit the PROC ROBUSTREG statements on page 363. Alternatively, you can replay the plot using PROC DOCUMENT, as in Example 15.6.

Output 15.8.1. Q-Q Plot with Modified Marker Symbols and Line



A sample program named odsgr08.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.9. Swapping the Axes in a Graph

Sometimes a Q-Q plot is displayed with the normal quantiles plotted along the y-axis and the ordered variable values plotted along the x-axis. This example, which is a continuation of Example 15.7 and Example 15.8, illustrates how to interchange the axes with a simple modification of the graph template.

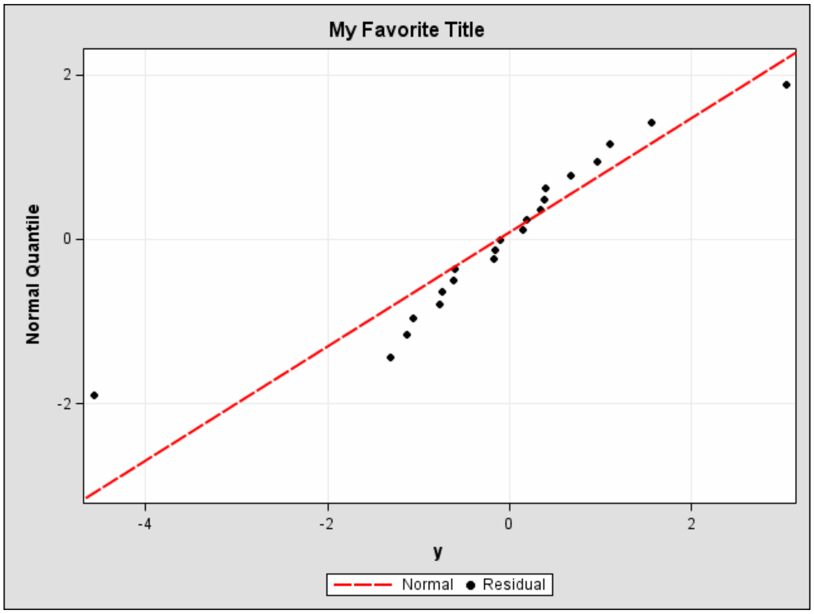
Begin by swapping the YAXISOPTS= and XAXISOPTS= options, and by swapping the X= and Y= options in the SCATTERPLOT statement.

Next, modify the LINEPARM statement. In Output 15.8.1, the slope of the line in the Q-Q plot is $\hat{\sigma}$, and y-intercept is $\hat{\mu}$. When you swap the axes, the values of the slope and y-intercept become $1/\hat{\sigma}$ and $-\hat{\mu}/\hat{\sigma}$, respectively. The modified template definition (including the changes from Example 15.7 and Example 15.8) is as follows:

```
proc template;
define statgraph Stat.Robustreg.Graphics.ResidualQQPlot;
   notes "Q-Q Plot for Standardized Robust Residuals";
   dynamic _DEPLABEL Residual;
   layout Gridded;
      layout Gridded / columns = 2;
         ENTRYTITLE "My Favorite Title" / padbottom=5;
      EndLayout;
      layout Lattice;
         layout Overlay /
               xaxisopts = (label = _DEPLABEL)
               yaxisopts = (label = "Normal Quantile")
               XGrid = True YGrid = True;
            SCATTERPLOT
               x = eval(SORT(DROPMISSING(RESIDUAL)))
               y = eval(PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL)))
                        -0.375)/(0.25+N(RESIDUAL)))) /
                  markersize   = GraphDataDefault:markersize
                  markersymbol = CircleFilled
                  markercolor  = GraphDataDefault:contrastcolor
                  legendlabel  = "Residual"
                  name         = "Data";
            lineparm
               slope      = eval(1/STDDEV(RESIDUAL))
               Yintercept = eval(-MEAN(RESIDUAL)/STDDEV(RESIDUAL)) /
                  linecolor     = red
                  linepattern   = dash
                  linethickness = StatGraphFitLine:linethickness
                  legendlabel   = "Normal"
                  name          = "Fit"
                  extreme       = true;
         EndLayout;
         column2header;
            layout Gridded / padtop = 5;
               DiscreteLegend "Fit" "Data" /
                  border     = true
                  across     = 2
                  background = GraphWalls:background;
            EndLayout;
         endcolumn2header;
      EndLayout;
   EndLayout;
end;
run;
```

The resulting Q-Q plot, after submitting the preceding statements and the PROC ROBUSTREG statements on page 363, is shown in Output 15.9.1.

Output 15.9.1. Q-Q Plot with Swapped Axes



A sample program named odsgr09.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.10. Modifying Tick Marks and Suppressing Grid Lines

This example, which is a continuation of Example 15.7, Example 15.8, and Example 15.9, illustrates how to modify the tick marks for an axis and suppress grid lines.

You can use the TICKS= suboption in the XAXISOPTS= or YAXISOPTS= options to specify the tick marks for an axis. For example, you can specify the following to request tick marks ranging from −3 to 3 in the y-axis for the Q-Q plots in Output 15.9.1:

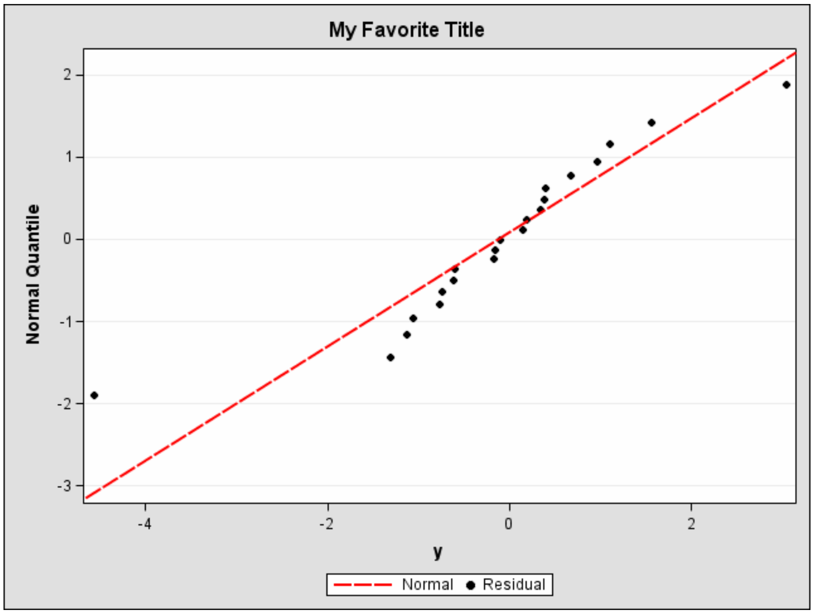
```
yaxisopts = (label = "Normal Quantile"
             ticks = (-3 -2 -1 0 1 2))
```

By default, the Q-Q plot in Output 15.9.1 displays grid lines since XGRID=TRUE and YGRID=TRUE are specified in the LAYOUT OVERLAY statement in the “ResidualQQPlot” template definition. You can suppress the grid lines by specifying

```
XGrid = False
```

The result of these changes, after submitting the modified template definition and the corresponding PROC ROBUSTREG statements on page 363, is displayed in Output 15.10.1.

Output 15.10.1. Q-Q Plot with Modified Y-Axis Tick Marks and Grids



A sample program named odsgr10.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.11. Modifying Graph Fonts in Styles

You can modify an ODS style to customize the general appearance of ODS Graphics, just as you can modify a style to customize the general appearance of ODS tables. The goal of this example is to customize the fonts used in ODS graphs. It is a continuation of Example 15.10.

The following statements define a style named NewStyle that replaces the graph fonts in the "Default" style with italic Times New Roman fonts.

```
proc template;
   define style Styles.NewStyle;
   parent = Styles.Default;
   replace GraphFonts
      "Fonts used in graph styles" /
      'GraphDataFont'     = ("Times New Roman",8pt,Italic)
      'GraphValueFont'    = ("Times New Roman",10pt,Italic)
      'GraphLabelFont'    = ("Times New Roman",12pt,Italic)
      'GraphFootnoteFont' = ("Times New Roman",12pt,Italic)
      'GraphTitleFont'    = ("Times New Roman",14pt,Italic Bold);
end;
run;
```

In general, the following graph fonts are specified in the ODS styles provided by SAS:

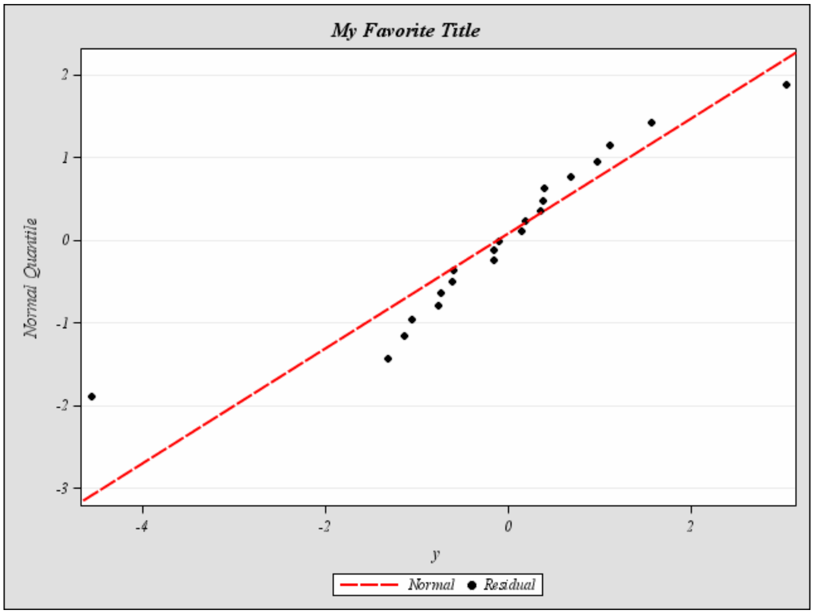
- **'GraphDataFont'** is the smallest font. It is used for text that needs to be small (labels for points in scatter plots, labels for contours, and so on)
- **'GraphValueFont'** is the next largest font. It is used for axis value (tick marks) labels and legend entry labels.
- **'GraphLabelFont'** is the next largest font. It is used for axis labels and legend titles.
- **'GraphFootnoteFont'** is the next largest font. It is used for all footnotes.
- **'GraphTitleFont'** is the largest font. It is used for all titles.

For more information about the DEFINE, PARENT, and REPLACE statements, refer to the "TEMPLATE Procedure: Creating a Style Definition" in the *SAS Output Delivery System User's Guide*.

The Q-Q plots in the preceding examples, beginning with Example 15.6, were created with the "Default" style; see, for instance, Output 15.10.1. In contrast, the Q-Q plot displayed in Output 15.11.1 was produced by specifying the NewStyle style in the following statements.

```
ods html style = NewStyle;
ods graphics on;

ods select ResidualQQPlot;

proc robustreg plot=resqqplot data=stack;
   model y = x1 x2 x3;
run;

ods graphics off;
ods html close;
```

Output 15.11.1. Q-Q Plot Using NewStyle



Although this example illustrates the use of a style with output from a particular procedure, note that a style is applied to *all* of your output (graphs and tables) in the destination for which you specify the style. See the "Specifying a Default Style" section on page 346 for information about specifying a default style for all your output.

A sample program named odsgr11.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.12. Modifying Other Graph Elements in Styles

This example, which is a continuation of Example 15.11, illustrates how to modify additional style elements for graphics, such as the thickness of a line.

The attributes of fitted lines in ODS Graphics are controlled by the style element **StatGraphFitLine**, which is defined in the "Default" style. For example, the line thickness of the normal distribution reference line in Output 15.11.1 is specified in the graph template by

```
linethickness = StatGraphFitLine:linethickness
```

To specify a line thickness of 4 pixels for the line, add the following statements to the definition of the NewStyle style in Example 15.11.

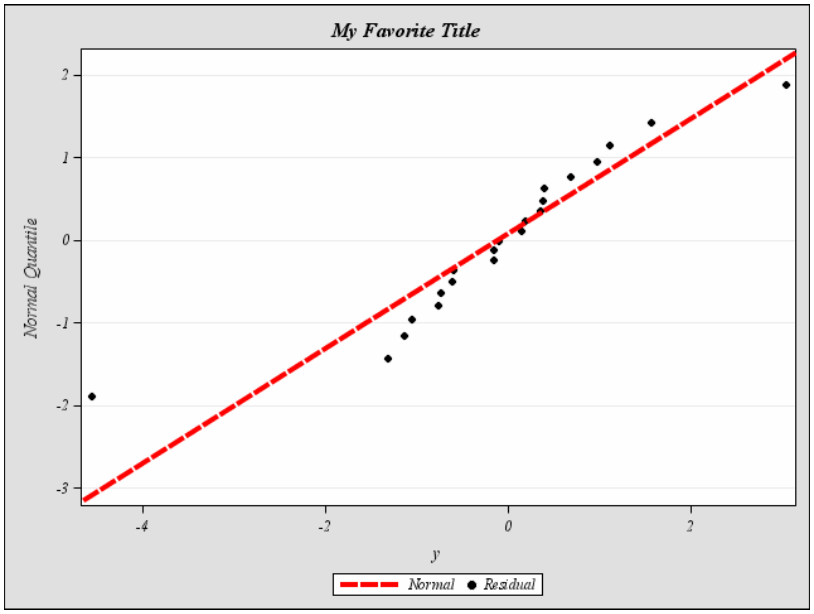
```
replace StatGraphFitLine /
   linethickness = 4px;
```

The complete revised NewStyle style is now defined by the following statements:

```
proc template;
   define style Styles.NewStyle;
   parent = Styles.Default;
   replace GraphFonts
      "Fonts used in graph styles" /
      'GraphDataFont'     = ("Times New Roman",8pt,Italic)
      'GraphValueFont'    = ("Times New Roman",10pt,Italic)
      'GraphLabelFont'    = ("Times New Roman",12pt,Italic)
      'GraphFootnoteFont' = ("Times New Roman",12pt,Italic)
      'GraphTitleFont'    = ("Times New Roman",14pt,Italic Bold);
   replace StatGraphFitLine /
      linethickness = 4px;
end;
run;
```

Output 15.12.1 shows the Q-Q plot created by the ROBUSTREG statements on page 375 with the new version of NewStyle.

Output 15.12.1. Q-Q Plot Using NewStyle with Thicker Line



You can use this approach to modify other attributes of the line, such as **`transparency`**, **`linestyle`**, **`contrastcolor`**, and **`foreground`**.

Note: Values specified directly in a graph template override style attributes. If you have customized a template, changes in a style may not have any effect. For more information, refer to the "ODS Statistical Graphics and ODS Styles: Usage and Reference (Experimental)" at http://support.sas.com/documentation/onlinedoc/base/.

A sample program named odsgr12.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.13. Modifying Graph Sizes Using Styles

This example demonstrates how to modify the size of your ODS graphs using a style definition.

You can specify the size of a graph in a graph template definition or in a style definition:

- To modify the size of a *particular* graph, specify the dimensions with the HEIGHT= and WIDTH= options in the outermost layout of the graph template definition.
- To modify the size of *all* your ODS graphs, specify the dimensions with the OUTPUTHEIGHT= and OUTPUTWIDTH= options in the style definition.

Dimensions specified in a graph template override those specified in a style.

Continuing the discussion in Example 15.12, you can add the following style element to the definition of NewStyle to change the size of all your graphs:

```
style Graph from Graph /
   outputwidth  = 400px
   outputheight = 300px;
```

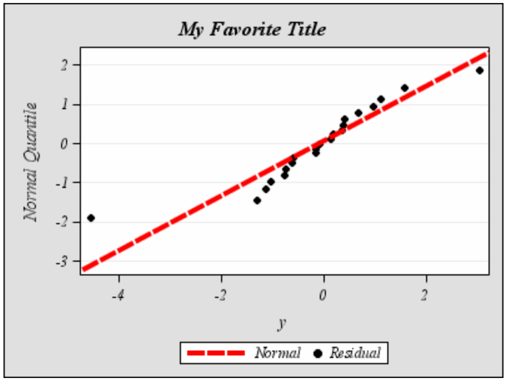
With all the changes introduced so far, NewStyle is defined as follows:

```
proc template;
   define style Styles.NewStyle;
   parent = Styles.Default;
   replace GraphFonts
      "Fonts used in graph styles" /
      'GraphDataFont'     = ("Times New Roman",8pt,Italic)
      'GraphValueFont'    = ("Times New Roman",10pt,Italic)
      'GraphLabelFont'    = ("Times New Roman",12pt,Italic)
      'GraphFootnoteFont' = ("Times New Roman",12pt,Italic)
      'GraphTitleFont'    = ("Times New Roman",14pt,Italic Bold);
   replace StatGraphFitLine /
      linethickness = 4px;
   style Graph from Graph /
      outputwidth  = 400px
      outputheight = 300px;
end;
run;
```

The dimensions of the graph must be specified in pixels. The actual size of the graph in inches depends on your printer or display device. For example, if the resolution of your printer is 100 dpi (100 dots per inch) and you want a graph that is 4 inches wide, you should set the width to 400 pixels.

You can create a smaller version of Output 15.12.1, shown in Output 15.13.1, by specifying the preceding PROC TEMPLATE statements followed by the ROBUSTREG statements on page 375.

Output 15.13.1. Q-Q Plot Using NewStyle with Smaller Dimensions



An alternative method for including smaller graphs in a document is to start with a style provided by SAS and define a modified style that *increases* the size of the graph fonts while preserving the default width and height attributes. Then you can include the graph in a document (for example in Microsoft Word) and manually rescale the graph to a smaller size while maintaining the fonts in a size that is still readable.*

The following style increases the size of the fonts but retains all the other style elements as assigned in the “Default” style:

```
proc template;
   define style Styles.BigFontStyle;
   parent = Styles.Default;
   replace GraphFonts
      "Fonts used in graph styles" /
      'GraphDataFont'     = ("Arial",12pt)
      'GraphValueFont'    = ("Arial",15pt)
      'GraphLabelFont'    = ("Arial",18pt)
      'GraphFootnoteFont' = ("Arial",18pt)
      'GraphTitleFont'    = ("Arial",21pt);
end;
run;
```

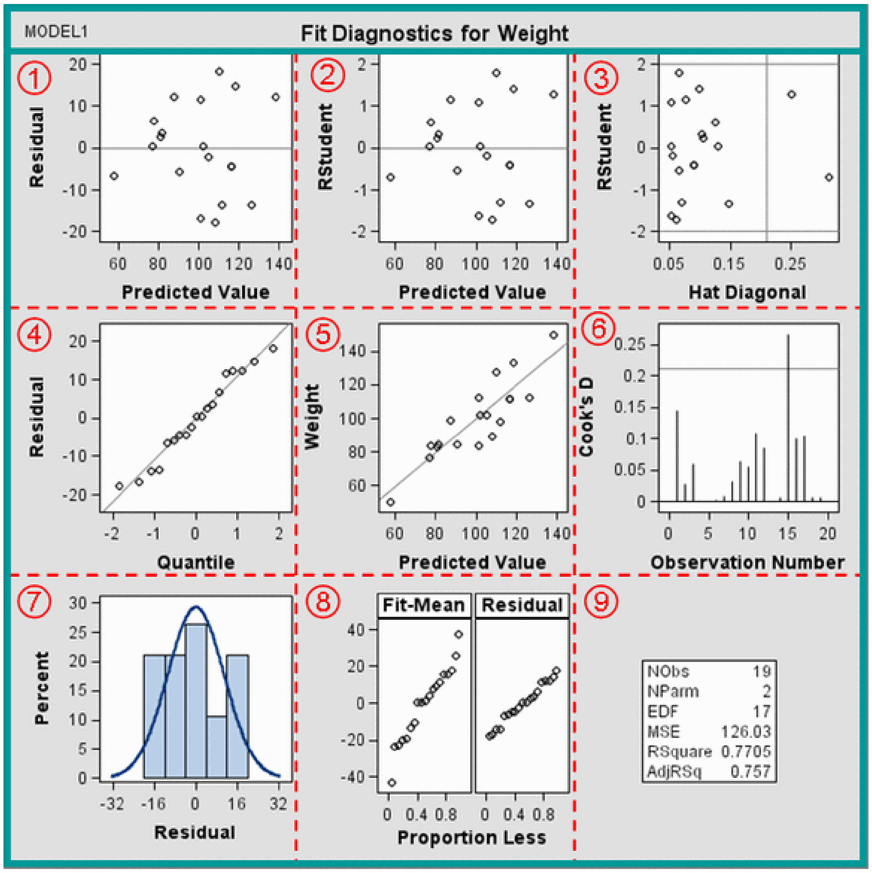
A sample program named odsgr13.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Example 15.14. Modifying Panels

This example illustrates how to modify the regression fit diagnostics panel in Figure 15.1 so that it displays a subset of component plots. The original panel consists of eight plots and a summary statistics box. These components are labeled 1 to 9 in the annotated version of Figure 15.1, which is shown in Output 15.14.1.

*In a markup language, such as HTML or $\LaTeX$, you can use a resize command.

Output 15.14.1. Diagnostics Panel Annotated to Indicate Layout Structure



```
NObs          19
NParm          2
EDF           17
MSE       126.03
RSquare   0.7705
AdjRSq     0.757
```

In the discussion that follows, the panel is modified so that it includes only the following components:

1. residual by predicted plot
4. residual Q-Q plot
6. Cook's D plot
7. residual histogram
9. summary statistics box

The panel to be produced is shown in Output 15.14.2. It displays components 1, 4, 6, and 7 in a 2 × 2 lattice, and it displays four of the summary statistics in component 9 in a box at the bottom.

The template that defines the original panel is "Stat.Reg.Graphics.DiagnosticPanel." The following listing is abbreviated to show the main structure of the template definition (see page 339 for details on how to display the complete template definition).

```
proc template;
define statgraph Stat.Reg.Graphics.DiagnosticsPanel;

   /* Dynamic variables */
   dynamic _TITLE _MODELLABEL _DEPLABEL _NOBS _NPARM _EDF _MSE
      _RSquare _AdjRSq;

   /* 3x3 LATTICE layout */
   layout lattice / columns = 3 rows = 3 ... ;

      sidebar / align=top;
         /* Statements for model label and graph title */
      endsidebar;

      /* 1. Residual By Predicted */
      layout overlay / ... ;
         lineparm slope = 0 yintercept = 0;
         scatterplot y = RESIDUAL x = PREDICTEDVALUE;
      endlayout;

      ...

      /* LAYOUT statements for components 2-8 */

      ...

      /* 9. Summary Statistics Box */
      layout overlay;
         layout gridded / ... ;
            entry "NObs";
            entry _NOBS / format=best6.;
               .
               .
               .
            entry "AdjRSq";
            entry _ADJRSQ / format=best6.;
         endlayout;
      endlayout;

   endlayout;   /* End of 3x3 LATTICE layout */
end;
run;
```

The overall display is defined by the LAYOUT LATTICE statement, which specifies a lattice of components, indicated by the solid grid annotated in Output 15.14.1. The COLUMNS=3 and ROWS=3 options in the LAYOUT LATTICE statement specify a 3 × 3 lattice, indicated by the dashed grid.

The model label and the graph title (top rectangle in Output 15.14.1) are specified inside the LATTICE layout with a SIDEBAR statement. The ALIGN=TOP option positions the sidebar at the top.

Each of the nine components of the lattice is defined by a LAYOUT statement. These statements define the components from left to right and top to bottom. Components 1 through 7 are defined with LAYOUT OVERLAY statements. Component 8 (RF plot) is defined with a LAYOUT LATTICE statement. The last LAYOUT OVERLAY statement defines a box with summary statistics for the fitted model.

The following abbreviated listing shows the basic structure of the template definition for a simplified panel that displays components 1, 4, 6, and 7 in a 2 × 2 lattice.* For the complete template definition, refer to the sample program odsgr14.sas in the SAS Sample Library for SAS/STAT software.

```
proc template;
define statgraph Stat.Reg.Graphics.DiagnosticsPanel;
   dynamic _TITLE _MODELLABEL _DEPLABEL _NOBS _NPARM _EDF _MSE
      _RSquare _AdjRSq;

   /* 2x2 LATTICE layout */
   /* Change COLUMNS= and ROWS= options */
   layout lattice / columns = 2 rows = 2 ... ;

      sidebar / align=top;
         /* Statements for model label and graph title */
      endsidebar;

      /* 1. Residual By Predicted */
      layout overlay / ... ;
         lineparm slope = 0 yintercept = 0;
         scatterplot y = RESIDUAL x = PREDICTEDVALUE;
      endlayout;

      /* 4. Q-Q Plot */
      layout overlay / ... ;
         lineparm slope      = eval(STDDEV(RESIDUAL))
                  yintercept = eval(...);
         scatterplot y = eval(...) x = eval(...);
      endlayout;

      /* Statements for components 6 and 7 (not listed) */

      /* Summary Statistics Box in a SIDEBAR */
      sidebar / align=bottom;
         layout gridded;
            layout lattice / rows=1 columns=4 ... ;
               .
               .
               .
            endlayout;
         endlayout;
      endsidebar;

   endlayout;   /* End of 2x2 LATTICE layout */
end;
run;
```

This template is a straightforward modification of the original template. The COLUMNS=2 and ROWS=2 options in the LAYOUT LATTICE statement request a 2 × 2 lattice. The LAYOUT statements for components 2, 3, 5, and 8 are deleted.

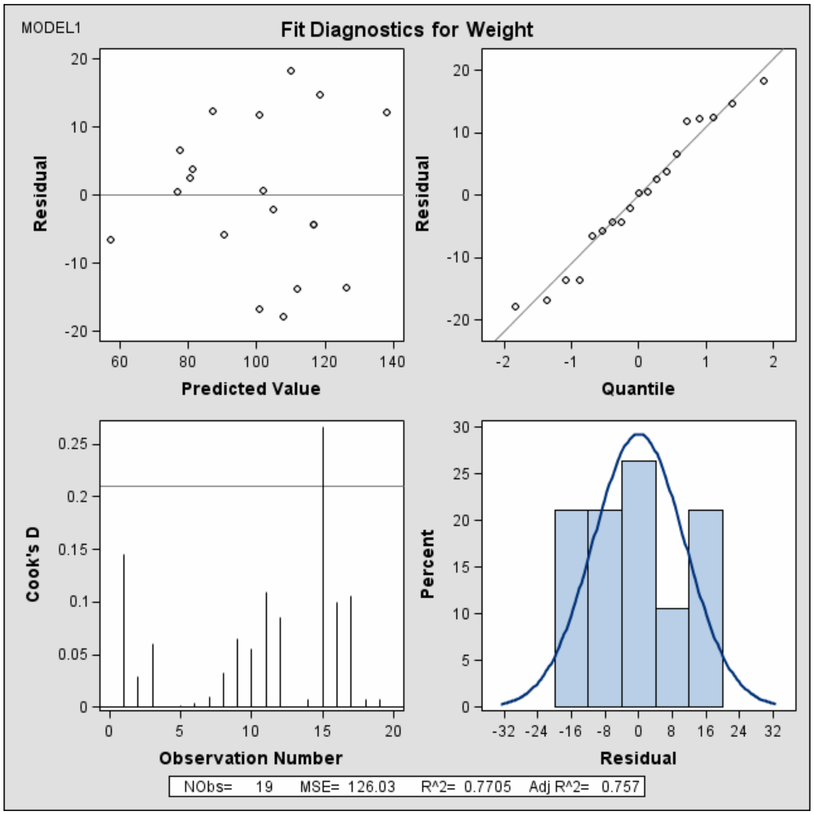
*See page 340 for details on how to edit the template definition.

A subset of the summary statistics are displayed at the bottom of the graph using a SIDEBAR statement with the ALIGN=BOTTOM option.

After submitting the preceding statements, which create the modified template and save it in Sasuser.Templat, you can run the following PROC REG statements to obtain the simplified panel, which is shown in Output 15.14.2.

```
ods html;
ods graphics on;

ods select DiagnosticsPanel;

proc reg data = Class;
   model Weight = Height;
run;
quit;

ods graphics off;
ods html close;
```

Output 15.14.2. Simplified Diagnostics Panel



NObs= 19 MSE= 126.03 R^2= 0.7705 Adj R^2= 0.757

A sample program named odsgr14.sas is available for this example in the SAS Sample Library for SAS/STAT software.

Chapter 16
The ACECLUS Procedure

Chapter Contents

Chapter 16
The ACECLUS Procedure

Overview

The ACECLUS (Approximate Covariance Estimation for CLUStering) procedure obtains approximate estimates of the pooled within-cluster covariance matrix when the clusters are assumed to be multivariate normal with equal covariance matrices. Neither cluster membership nor the number of clusters need be known. PROC ACECLUS is useful for preprocessing data to be subsequently clustered by the CLUSTER or the FASTCLUS procedure.

Many clustering methods perform well with spherical clusters but poorly with elongated elliptical clusters (Everitt 1980, 77–97). If the elliptical clusters have roughly the same orientation and eccentricity, you can apply a linear transformation to the data to yield a spherical within-cluster covariance matrix, that is, a covariance matrix proportional to the identity. Equivalently, the distance between observations can be measured in the metric of the inverse of the pooled within-cluster covariance matrix. The remedy is difficult to apply, however, because you need to know what the clusters are in order to compute the sample within-cluster covariance matrix. One approach is to estimate iteratively both cluster membership and within-cluster covariance (Wolfe 1970; Hartigan 1975). Another approach is provided by Art, Gnanadesikan, and Kettenring (1982). They have devised an ingenious method for estimating the within-cluster covariance matrix without knowledge of the clusters. The method can be applied before any of the usual clustering techniques, including hierarchical clustering methods.

First, Art, Gnanadesikan, and Kettenring (1982) obtain a decomposition of the total-sample sum-of-squares-and-cross-products (SSCP) matrix into within-cluster and between-cluster SSCP matrices computed from pairwise differences between observations, rather than differences between observations and means. Then, they show how the within-cluster SSCP matrix based on pairwise differences can be approximated without knowing the number or the membership of the clusters. The approximate within-cluster SSCP matrix can be used to compute distances for cluster analysis, or it can be used in a canonical analysis similar to canonical discriminant analysis. For more information, see Chapter 21, “The CANDISC Procedure.”

Art, Gnanadesikan, and Kettenring demonstrate by Monte Carlo calculations that their method can produce better clusters than the Euclidean metric even when the approximation to the within-cluster SSCP matrix is poor or the within-cluster covariances are moderately heterogeneous. The algorithm used by the ACECLUS procedure differs slightly from the algorithm used by Art, Gnanadesikan, and Kettenring. In the following sections, the PROC ACECLUS algorithm is described first; then, differences between PROC ACECLUS and the method used by Art, Gnanadesikan, and Kettenring are summarized.

Background

It is well known from the literature on nonparametric statistics that variances and, hence, covariances can be computed from pairwise differences instead of deviations from means. (For example, Puri and Sen (1971, pp. 51–52) show that the variance is a U statistic of degree 2.) Let $\mathbf{X} = (x_{ij})$ be the data matrix with n observations (rows) and v variables (columns), and let $\bar{x}_j$ be the mean of the jth variable. The sample covariance matrix $\mathbf{S} = (s_{jk})$ is usually defined as

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

The matrix $\mathbf{S}$ can also be computed as

$$s_{jk} = \frac{1}{n(n-1)} \sum_{i=2}^{n} \sum_{h=1}^{i-1} (x_{ij} - x_{hj})(x_{ik} - x_{hk})$$

Let $\mathbf{W} = (w_{jk})$ be the pooled within-cluster covariance matrix, q be the number of clusters, n_c be the number of observations in the cth cluster, and

$$d''_{ic} = \begin{cases} 1 & \text{if observation } i \text{ is in cluster } c \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{W}$ is normally defined as

$$w_{jk} = \frac{1}{n-q} \sum_{c=1}^{q} \sum_{i=1}^{n} d''_{ic} (x_{ij} - \bar{x}_{cj})(x_{ik} - \bar{x}_{ck})$$

where $\bar{x}_{cj}$ is the mean of the jth variable in cluster c. Let

$$d'_{ih} = \begin{cases} \frac{1}{n_c} & \text{if observations } i \text{ and } h \text{ are in cluster } c \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{W}$ can also be computed as

$$w_{jk} = \frac{1}{n-q} \sum_{i=2}^{n} \sum_{h=1}^{i-1} d'_{ih} (x_{ij} - x_{hj})(x_{ik} - x_{hk})$$

If the clusters are not known, d'_{ih} cannot be determined. However, an approximation to $\mathbf{W}$ can be obtained by using instead

$$d_{ih} = \begin{cases} 1 & \text{if } \sum_{j=1}^{v} \sum_{k=1}^{v} m_{jk} (x_{ij} - x_{hj})(x_{ik} - x_{hk}) \le u^2 \\ 0 & \text{otherwise} \end{cases}$$

where u is an appropriately chosen value and $\mathbf{M} = (m_{jk})$ is an appropriate metric. Let $\mathbf{A} = (a_{jk})$ be defined as

$$a_{jk} = \frac{\sum_{i=2}^{n} \sum_{h=1}^{i-1} d_{ih}(x_{ij} - x_{hj})(x_{ik} - x_{hk})}{2 \sum_{i=2}^{n} \sum_{h=1}^{i-1} d_{ih}}$$

If all of the following conditions hold, $\mathbf{A}$ equals $\mathbf{W}$:

- all within-cluster distances in the metric $\mathbf{M}$ are less than or equal to u
- all between-cluster distances in the metric $\mathbf{M}$ are greater than u
- all clusters have the same number of members n_c

If the clusters are of unequal size, $\mathbf{A}$ gives more weight to large clusters than $\mathbf{W}$ does, but this discrepancy should be of little importance if the population within-cluster covariance matrices are equal. There may be large differences between $\mathbf{A}$ and $\mathbf{W}$ if the cutoff u does not discriminate between pairs in the same cluster and pairs in different clusters. Lack of discrimination may occur for one of the following reasons:

- The clusters are not well separated.
- The metric $\mathbf{M}$ or the cutoff u is not chosen appropriately.

In the former case, little can be done to remedy the problem. The remaining question concerns how to choose $\mathbf{M}$ and u. Consider $\mathbf{M}$ first. The best choice for $\mathbf{M}$ is $\mathbf{W}^{-1}$, but $\mathbf{W}$ is not known. The solution is to use an iterative algorithm:

1. Obtain an initial estimate of $\mathbf{A}$, such as the identity or the total-sample covariance matrix. (See the INITIAL= option in the PROC ACECLUS statement for more information.)
2. Let $\mathbf{M}$ equal $\mathbf{A}^{-1}$.
3. Recompute $\mathbf{A}$ using the preceding formula.
4. Repeat steps 2 and 3 until the estimate stabilizes.

Convergence is assessed by comparing values of $\mathbf{A}$ on successive iterations. Let $\mathbf{A}_i$ be the value of $\mathbf{A}$ on the ith iteration and $\mathbf{A}_0$ be the initial estimate of $\mathbf{A}$. Let $\mathbf{Z}$ be a user-specified $v \times v$ matrix. (See the METRIC= option in the PROC ACECLUS statement for more information.) The convergence measure is

$$e_i = \frac{1}{v} \parallel \mathbf{Z}'(\mathbf{A}_i - \mathbf{A}_{i-1})\mathbf{Z} \parallel$$

where $\parallel \cdots \parallel$ indicates the Euclidean norm, that is, the square root of the sum of the squares of the elements of the matrix. In PROC ACECLUS, $\mathbf{Z}$ can be the identity or an inverse factor of $\mathbf{S}$ or diag($\mathbf{S}$). Iteration stops when e_i falls below a user-specified value. (See the CONVERGE= option or the MAXITER= option in the PROC ACECLUS statement for more information.)

The remaining question of how to choose u has no simple answer. In practice, you must try several different values. PROC ACECLUS provides four different ways of specifying u:

- You can specify a constant value for u. This method is useful if the initial estimate of $\mathbf{A}$ is quite good. (See the ABSOLUTE option and the THRESHOLD= option in the PROC ACECLUS statement for more information.)
- You can specify a threshold value $t > 0$ that is multiplied by the root mean square distance between observations in the current metric on each iteration to give u. Thus, the value of u changes from iteration to iteration. This method is appropriate if the initial estimate of $\mathbf{A}$ is poor. (See the THRESHOLD= option in the PROC ACECLUS statement for more information)
- You can specify a value p, $0 < p < 1$, to be transformed into a distance u such that approximately a proportion p of the pairwise Mahalanobis distances between observations in a random sample from a multivariate normal distribution will be less than u in repeated sampling. The transformation can be computed only if the number of observations exceeds the number of variables, preferably by at least 10 percent. This method also requires a good initial estimate of $\mathbf{A}$. (See the PROPORTION= option and the ABSOLUTE option in the PROC ACECLUS statement for more information.)
- You can specify a value p, $0 < p < 1$, to be transformed into a value t that is then multiplied by $1/\sqrt{2v}$ times the root mean square distance between observations in the current metric on each iteration to yield u. The value of u changes from iteration to iteration. This method can be used with a poor initial estimate of $\mathbf{A}$. (See the PROPORTION= option in the PROC ACECLUS statement for more information.)

In most cases, the analysis should begin with the last method using values of p between 0.5 and 0.01 and using the full covariance matrix as the initial estimate of $\mathbf{A}$.

Proportions p are transformed to distances t using the formula

$$t^2 = 2v \left\{ \left[F^{-1}_{v,n-v}(p) \right]^{\frac{n-v}{n-1}} \right\}$$

where $F^{-1}_{v,n-v}$ is the quantile (inverse cumulative distribution) function of an F random variable with v and $n-v$ degrees of freedom. The squared Mahalanobis distance between a single pair of observations sampled from a multivariate normal distribution is distributed as $2v$ times an F random variable with v and $n-v$ degrees of freedom. The distances between two pairs of observations are correlated if the pairs have an observation in common. The quantile function is raised to the power given in the preceding formula to compensate approximately for the correlations among distances between pairs of observations that share a member. Monte Carlo studies indicate that the approximation is acceptable if the number of observations exceeds the number of variables by at least 10 percent.

If $\mathbf{A}$ becomes singular, step 2 in the iterative algorithm cannot be performed because $\mathbf{A}$ cannot be inverted. In this case, let $\mathbf{Z}$ be the matrix as defined in discussing the convergence measure, and let $\mathbf{Z}'\mathbf{A}\mathbf{Z} = \mathbf{R}'\mathbf{\Lambda}\mathbf{R}$ where $\mathbf{R}'\mathbf{R} = \mathbf{R}\mathbf{R}' = \mathbf{I}$ and $\mathbf{\Lambda} = (\lambda_{jk})$ is diagonal. Let $\mathbf{\Lambda}^* = (\lambda^*_{jk})$ be a diagonal matrix where $\lambda^*_{jj} = \max(\lambda_{jj}, g\ \text{trace}(\mathbf{\Lambda}))$, and $0 < g < 1$ is a user-specified singularity criterion (see the SINGULAR= option in the PROC ACECLUS statement for more information). Then $\mathbf{M}$ is computed as $\mathbf{Z}\mathbf{R}'(\mathbf{\Lambda}^*)^{-1}\mathbf{R}\mathbf{Z}'$.

The ACECLUS procedure differs from the method used by Art, Gnanadesikan, and Kettenring (1982) in several respects.

- The Art, Gnanadesikan, and Kettenring method uses the identity matrix as the initial estimate, whereas the ACECLUS procedure enables you to specify any symmetric matrix as the initial estimate and defaults to the total-sample covariance matrix. The default initial estimate in PROC ACECLUS is chosen to yield invariance under nonsingular linear transformations of the data but may sometimes obscure clusters that become apparent if the identity matrix is used.
- The Art, Gnanadesikan, and Kettenring method carries out all computations with SSCP matrices, whereas the ACECLUS procedure uses estimated covariance matrices because covariances are easier to interpret than crossproducts.
- The Art, Gnanadesikan, and Kettenring method uses the m pairs with the smallest distances to form the new estimate at each iteration, where m is specified by the user, whereas the ACECLUS procedure uses all pairs closer than a given cutoff value. Kettenring (1984) says that the m-closest-pairs method seems to give the user more direct control. PROC ACECLUS uses a distance cutoff because it yields a slight decrease in computer time and because in some cases, such as widely separated spherical clusters, the results are less sensitive to the choice of distance cutoff than to the choice of m. Much research remains to be done on this issue.
- The Art, Gnanadesikan, and Kettenring method uses a different convergence measure. Let $\mathbf{A}_i$ be computed on each iteration using the m-closest-pairs method, and let $\mathbf{B}_i = \mathbf{A}^{-1}_{i-1}\mathbf{A}_i - \mathbf{I}$ where $\mathbf{I}$ is the identity matrix. The convergence measure is equivalent to trace$(\mathbf{B}^2_i)$.

Analyses of Fisher's (1936) iris data, consisting of measurements of petal and sepal length and width for fifty specimens from each of three iris species, are summarized in Table 16.1. The number of misclassified observations out of 150 is given for four clustering methods:

- k-means as implemented in PROC FASTCLUS with MAXC=3, MAXITER=99, and CONV=0
- Ward's minimum variance method as implemented in PROC CLUSTER
- average linkage on Euclidean distances as implemented in PROC CLUSTER

- the centroid method as implemented in PROC CLUSTER

Each hierarchical analysis is followed by the TREE procedure with NCL=3 to determine cluster assignments at the three-cluster level. Clusters with twenty or fewer observations are discarded by using the DOCK=20 option. The observations in a discarded cluster are considered unclassified.

Each method is applied to

- the raw data
- the data standardized to unit variance by the STANDARD procedure
- two standardized principal components accounting for 95 percent of the standardized variance and having an identity total-sample covariance matrix, computed by the PRINCOMP procedure with the STD option
- four standardized principal components having an identity total-sample covariance matrix, computed by PROC PRINCOMP with the STD option
- the data transformed by PROC ACECLUS using seven different settings of the PROPORTION= (P=) option
- four canonical variables having an identity pooled within-species covariance matrix, computed using the CANDISC procedure

Theoretically, the best results should be obtained by using the canonical variables from PROC CANDISC. PROC ACECLUS yields results comparable to PROC CANDISC for values of the PROPORTION= option ranging from 0.005 to 0.02. At PROPORTION=0.04, average linkage and the centroid method show some deterioration, but k-means and Ward's method continue to produce excellent classifications. At larger values of the PROPORTION= option, all methods perform poorly, although no worse than with four standardized principal components.

Table 16.1. Number of Misclassified and Unclassified Observations Using Fisher's (1936) Iris Data

	Clustering Method			
Data	k-means	Ward's	Average Linkage	Centroid
raw data	16^*	16^*	$25 + 12^{**}$	14^*
standardized data	25	26	33+4	33+4
two standardized principal components	29	31	30+9	27+32
four standardized principal components	39	27	32+7	45+11
transformed by ACECLUS P=0.32	39	10+9	7+25	
transformed by ACECLUS P=0.16	39	18+9	7+19	7+26
transformed by ACECLUS P=0.08	19	9	3+13	5+16
transformed by ACECLUS P=0.04	4	5	1+19	3+12
transformed by ACECLUS P=0.02	4	3	3	3
transformed by ACECLUS P=0.01	4	4	3	4
transformed by ACECLUS P=0.005	4	4	4	4
canonical variables	3	5	4	4+1

[*] A single number represents misclassified observations with no unclassified observations.

[**] Where two numbers are separated by a plus sign, the first is the number of misclassified observations; the second is the number of unclassified observations.

This example demonstrates the following:

- PROC ACECLUS can produce results as good as those from the optimal transformation.

- PROC ACECLUS can be useful even when the within-cluster covariance matrices are moderately heterogeneous.
- The choice of the distance cutoff as specified by the PROPORTION= or the THRESHOLD= option is important, and several values should be tried.
- Commonly used transformations such as standardization and principal components can produce poor classifications.

Although experience with the Art, Gnanadesikan, and Kettenring and PROC ACECLUS methods is limited, the results so far suggest that these methods help considerably more often than they hinder the subsequent cluster analysis, especially with normal-mixture techniques such as k-means and Ward's minimum variance method.

Getting Started

The following example demonstrates how you can use the ACECLUS procedure to obtain approximate estimates of the pooled within-cluster covariance matrix and to compute canonical variables for subsequent analysis. You use PROC ACECLUS to preprocess data before you cluster it using the FASTCLUS or CLUSTER procedure.

Suppose you want to determine whether national figures for birth rates, death rates, and infant death rates can be used to determine certain types or categories of countries. You want to perform a cluster analysis to determine whether the observations can be formed into groups suggested by the data. Previous studies indicate that the clusters computed from this type of data can be elongated and elliptical. Thus, you need to perform a linear transformation on the raw data before the cluster analysis.

The following data[*] from Rouncefield (1995) are the birth rates, death rates, and infant death rates for 97 countries. The following statements create the SAS data set Poverty:

```
data poverty;
   input Birth Death InfantDeath Country $15. @@;
   datalines;
24.7  5.7  30.8 Albania         12.5 11.9  14.4 Bulgaria
13.4 11.7  11.3 Czechoslovakia  12   12.4   7.6 Former_E._Germa
11.6 13.4  14.8 Hungary         14.3 10.2    16 Poland
13.6 10.7  26.9 Romania           14    9  20.2 Yugoslavia
17.7   10    23 USSR            15.2  9.5  13.1 Byelorussia
13.4 11.6    13 Ukrainian_SSR   20.7  8.4  25.7 Argentina
46.6   18   111 Bolivia         28.6  7.9    63 Brazil
23.4  5.8  17.1 Chile           27.4  6.1    40 Columbia
32.9  7.4    63 Ecuador         28.3  7.3    56 Guyana
34.8  6.6    42 Paraguay        32.9  8.3 109.9 Peru
  18  9.6  21.9 Uruguay         27.5  4.4  23.3 Venezuela
  29 23.2    43 Mexico            12 10.6   7.9 Belgium
13.2 10.1   5.8 Finland         12.4 11.9   7.5 Denmark
13.6  9.4   7.4 France          11.4 11.2   7.4 Germany
```

[*] These data have been compiled from the United Nations Demographic Yearbook 1990 (United Nations publications, Sales No. E/F.91.XII.1, copyright 1991, United Nations, New York) and are reproduced with the permission of the United Nations.

```
10.1  9.2    11 Greece          15.1  9.1   7.5 Ireland
 9.7  9.1   8.8 Italy           13.2  8.6   7.1 Netherlands
14.3 10.7   7.8 Norway          11.9  9.5  13.1 Portugal
10.7  8.2   8.1 Spain           14.5 11.1   5.6 Sweden
12.5  9.5   7.1 Switzerland     13.6 11.5   8.4 U.K.
14.9  7.4     8 Austria          9.9  6.7   4.5 Japan
14.5  7.3   7.2 Canada          16.7  8.1   9.1 U.S.A.
40.4 18.7 181.6 Afghanistan     28.4  3.8    16 Bahrain
42.5 11.5 108.1 Iran            42.6  7.8    69 Iraq
22.3  6.3   9.7 Israel          38.9  6.4    44 Jordan
26.8  2.2  15.6 Kuwait          31.7  8.7    48 Lebanon
45.6  7.8    40 Oman            42.1  7.6    71 Saudi_Arabia
29.2  8.4    76 Turkey          22.8  3.8    26 United_Arab_Emr
42.2 15.5   119 Bangladesh      41.4 16.6   130 Cambodia
21.2  6.7    32 China           11.7  4.9   6.1 Hong_Kong
30.5 10.2    91 India           28.6  9.4    75 Indonesia
23.5 18.1    25 Korea           31.6  5.6    24 Malaysia
36.1  8.8    68 Mongolia        39.6 14.8   128 Nepal
30.3  8.1 107.7 Pakistan        33.2  7.7    45 Philippines
17.8  5.2   7.5 Singapore       21.3  6.2  19.4 Sri_Lanka
22.3  7.7    28 Thailand        31.8  9.5    64 Vietnam
35.5  8.3    74 Algeria         47.2 20.2   137 Angola
48.5 11.6    67 Botswana        46.1 14.6    73 Congo
38.8  9.5  49.4 Egypt           48.6 20.7   137 Ethiopia
39.4 16.8   103 Gabon           47.4 21.4   143 Gambia
44.4 13.1    90 Ghana             47 11.3    72 Kenya
 44  9.4    82 Libya          48.3   25   130 Malawi
35.5  9.8    82 Morocco           45 18.5   141 Mozambique
  44 12.1   135 Namibia         48.5 15.6   105 Nigeria
48.2 23.4   154 Sierra_Leone    50.1 20.2   132 Somalia
32.1  9.9    72 South_Africa    44.6 15.8   108 Sudan
46.8 12.5   118 Swaziland       31.1  7.3    52 Tunisia
52.2 15.6   103 Uganda          50.5   14   106 Tanzania
45.6 14.2    83 Zaire           51.1 13.7    80 Zambia
41.7 10.3    66 Zimbabwe
;
```

The data set Poverty contains the character variable Country and the numeric variables Birth, Death, and InfantDeath, which represent the birth rate per thousand, death rate per thousand, and infant death rate per thousand. The $15. in the INPUT statement specifies that the variable Country is a character variable with a length of 15. The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values have been read.

It is often useful when beginning a cluster analysis to look at the data graphically. The following statements use the GPLOT procedure to make a scatter plot of the variables Birth and Death.

```
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot data=poverty;
   plot Birth*Death/
      frame cframe=white legend=legend1 vaxis=axis1 haxis=axis2;
run;
quit;
```

The plot, displayed in Figure 16.1, indicates the difficulty of dividing the points into clusters. Plots of the other variable pairs (not shown) display similar characteristics. The clusters that comprise these data may be poorly separated and elongated. Data with poorly separated or elongated clusters must be transformed.

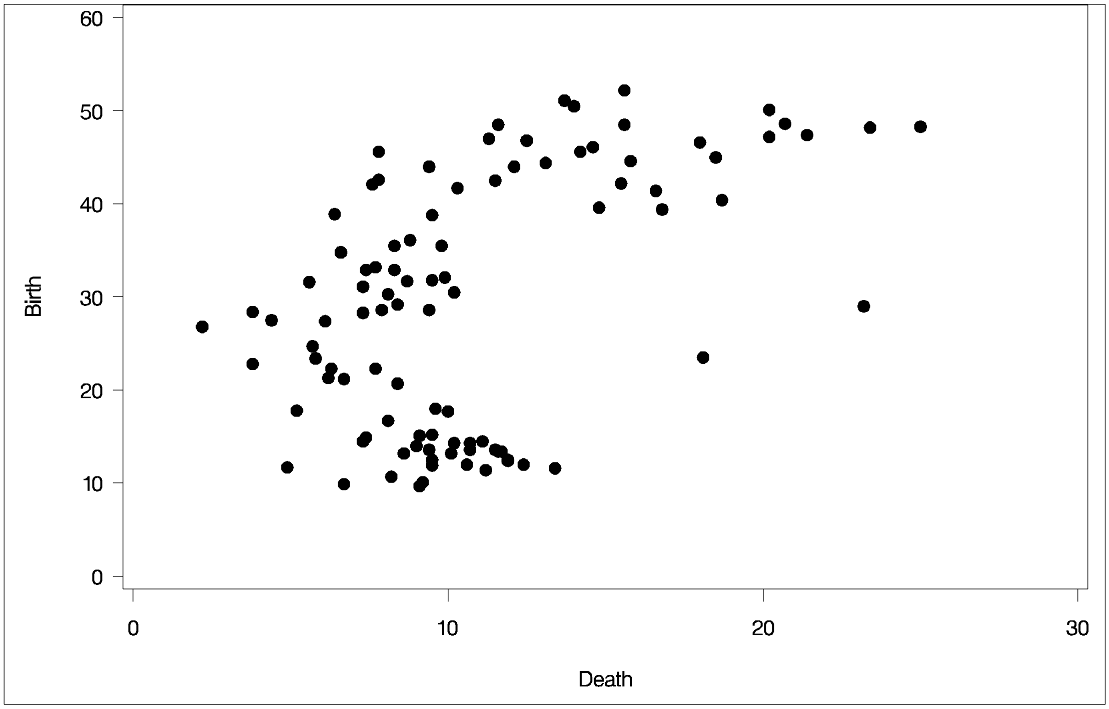


Figure 16.1. Scatter Plot of Original Poverty Data: Birth Rate versus Death Rate

If you know the within-cluster covariances, you can transform the data to make the clusters spherical. However, since you do not know what the clusters are, you cannot calculate exactly the within-cluster covariance matrix. The ACECLUS procedure estimates the within-cluster covariance matrix to transform the data, even when you have no knowledge of cluster membership or the number of clusters.

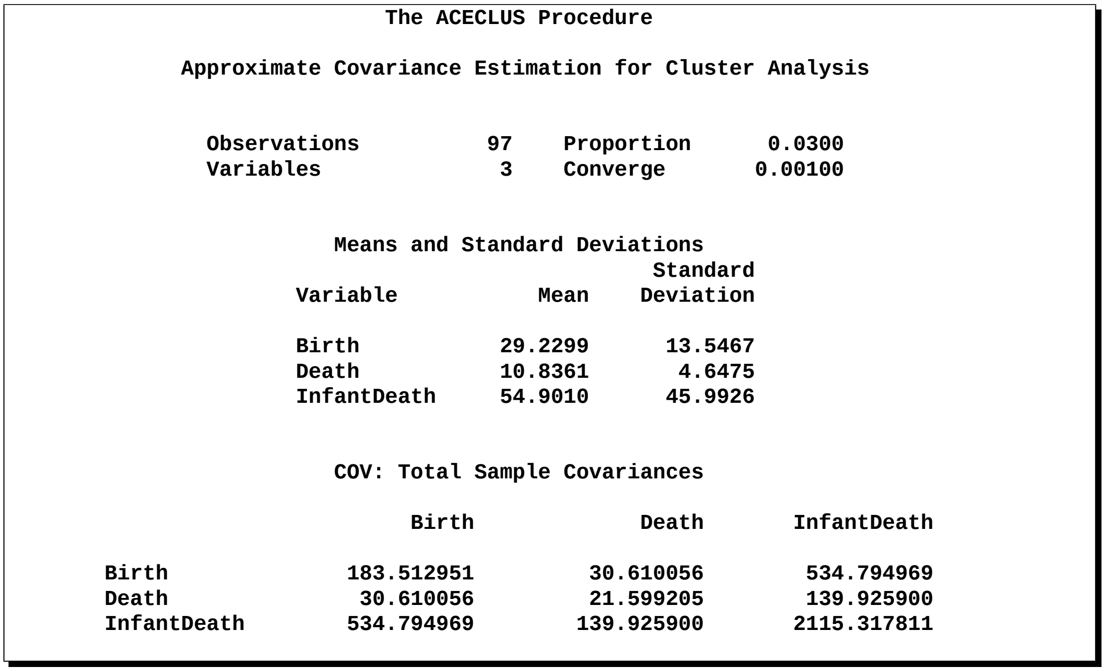
The following statements perform the ACECLUS procedure transformation using the SAS data set Poverty.

```
proc aceclus data=poverty out=ace proportion=.03;
   var Birth Death InfantDeath;
run;
```

The OUT= option creates an output data set called Ace to contain the canonical variable scores. The PROPORTION= option specifies that approximately three percent of the pairs are included in the estimation of the within-cluster covariance matrix. The VAR statement specifies that the variables Birth, Death, and InfantDeath are used in computing the canonical variables.

The results of this analysis are displayed in the following figures.

Figure 16.2 displays the number of observations, the number of variables, and the settings for the PROPORTION and CONVERGE options. The PROPORTION option is set at 0.03, as specified in the previous statements. The CONVERGE parameter is set at its default value of 0.001.

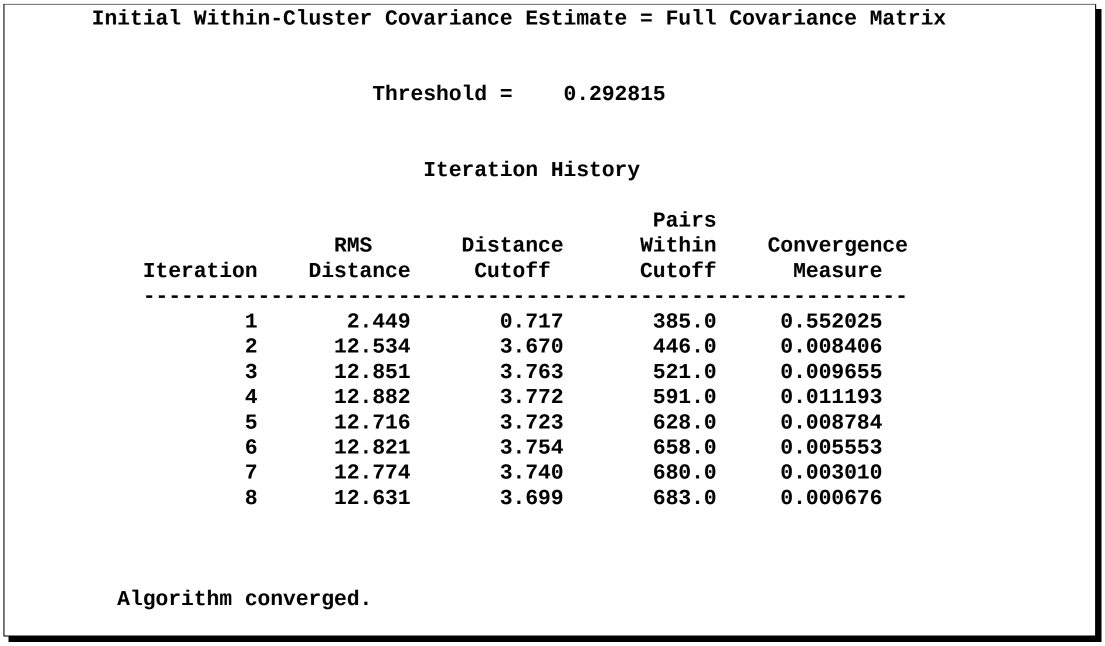


```
                         The ACECLUS Procedure

          Approximate Covariance Estimation for Cluster Analysis


            Observations          97    Proportion      0.0300
            Variables              3    Converge       0.00100


                     Means and Standard Deviations
                                                Standard
                   Variable           Mean     Deviation

                   Birth           29.2299       13.5467
                   Death           10.8361        4.6475
                   InfantDeath     54.9010       45.9926


                     COV: Total Sample Covariances

                         Birth             Death       InfantDeath

    Birth           183.512951         30.610056        534.794969
    Death            30.610056         21.599205        139.925900
    InfantDeath     534.794969        139.925900       2115.317811
```

Figure 16.2. Means, Standard Deviations, and Covariance Matrix from the ACECLUS Procedure

Figure 16.2 next displays the means, standard deviations, and sample covariance matrix of the analytical variables.

The type of matrix used for the initial within-cluster covariance estimate is displayed in Figure 16.3. In this example, that initial estimate is the full covariance matrix. The threshold value that corresponds to the PROPORTION=0.03 setting is given as 0.292815.



```
     Initial Within-Cluster Covariance Estimate = Full Covariance Matrix


                          Threshold =    0.292815


                              Iteration History

                                                           Pairs
                          RMS         Distance          Within     Convergence
         Iteration     Distance        Cutoff          Cutoff        Measure
         ------------------------------------------------------------
                 1        2.449         0.717           385.0       0.552025
                 2       12.534         3.670           446.0       0.008406
                 3       12.851         3.763           521.0       0.009655
                 4       12.882         3.772           591.0       0.011193
                 5       12.716         3.723           628.0       0.008784
                 6       12.821         3.754           658.0       0.005553
                 7       12.774         3.740           680.0       0.003010
                 8       12.631         3.699           683.0       0.000676



       Algorithm converged.
```

Figure 16.3. Table of Iteration History from the ACECLUS Procedure

Figure 16.3 displays the iteration history. For each iteration, PROC ACECLUS displays the following measures:

- root mean square distance between all pairs of observations
- distance cutoff for including pairs of observations in the estimate of within-cluster covariances (equal to RMS*Threshold)
- number of pairs within the cutoff
- convergence measure

Figure 16.4 displays the approximate within-cluster covariance matrix and the table of eigenvalues from the canonical analysis. The first column of the eigenvalues table contains numbers for the eigenvectors. The next column of the table lists the eigenvalues of Inv(ACE)*(COV-ACE).

```
              ACE: Approximate Covariance Estimate Within Clusters

                              Birth             Death       InfantDeath

       Birth             5.94644949       -0.63235725        6.28151537
       Death            -0.63235725        2.33464129        1.59005857
       InfantDeath       6.28151537        1.59005857       35.10327233


                         Eigenvalues of Inv(ACE)*(COV-ACE)

                      Eigenvalue    Difference    Proportion    Cumulative

                 1       63.5500       54.7313        0.8277        0.8277
                 2        8.8187        4.4038        0.1149        0.9425
                 3        4.4149                      0.0575        1.0000
```

Figure 16.4. Approximate Within–Cluster Covariance Estimates

The next three columns of the eigenvalue table (Figure 16.4) display measures of the relative size and importance of the eigenvalues. The first column lists the difference between each eigenvalue and its successor. The last two columns display the individual and cumulative proportions that each eigenvalue contributes to the total sum of eigenvalues.

The raw and standardized canonical coefficients are displayed in Figure 16.5. The coefficients are standardized by multiplying the raw coefficients with the standard deviation of the associated variable. The ACECLUS procedure uses these standardized canonical coefficients to create the transformed canonical variables, which are the linear transformations of the original input variables, Birth, Death, and InfantDeath.

```
                    Eigenvectors (Raw Canonical Coefficients)

                                 Can1          Can2          Can3

             Birth           0.125610      0.457037      0.003875
             Death           0.108402      0.163792      0.663538
             InfantDeath     0.134704      -.133620      -.046266


                       Standardized Canonical Coefficients

                                 Can1          Can2          Can3

             Birth            1.70160       6.19134       0.05249
             Death            0.50380       0.76122       3.08379
             InfantDeath      6.19540      -6.14553      -2.12790
```

Figure 16.5. Raw and Standardized Canonical Coefficients from the ACECLUS Procedure

The following statements invoke the CLUSTER procedure, using the SAS data set Ace created in the previous ACECLUS procedure.

```
proc cluster data=ace outtree=tree noprint method=ward;
   var can1 can2 can3 ;
   copy Birth--Country;
run;
```

The OUTTREE= option creates the output SAS data set Tree that is used in subsequent statements to draw a tree diagram. The NOPRINT option suppresses the display of the output. The METHOD= option specifies Ward's minimum-variance clustering method.

The VAR statement specifies that the canonical variables computed in the ACECLUS procedure are used in the cluster analysis. The COPY statement specifies that all the variables from the SAS data set Poverty (Birth—Country) are added to the output data set Tree.

The following statements use the TREE procedure to create an output SAS data set called New. The NCLUSTERS= option specifies the number of clusters desired in the SAS data set New. The NOPRINT option suppresses the display of the output.

```
proc tree data=tree out=new nclusters=3 noprint;
   copy Birth Death InfantDeath can1 can2 ;
   id Country;
run;
```

The COPY statement copies the canonical variables CAN1 and CAN2 (computed in the preceding ACECLUS procedure) and the original analytical variables Birth, Death, and InfantDeath into the output SAS data set New.

The following statements invoke the GPLOT procedure, using the SAS data set created by PROC TREE:

```
legend1 frame cframe=white cborder=black position=center
   value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot data=new;
   plot Birth*Death=cluster/
      frame cframe=white legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

The first plot statement requests a scatter plot of the two variables Birth and Death, using the variable CLUSTER as the identification variable.

The second PLOT statement requests a plot of the two canonical variables, using the value of the variable CLUSTER as the identification variable.

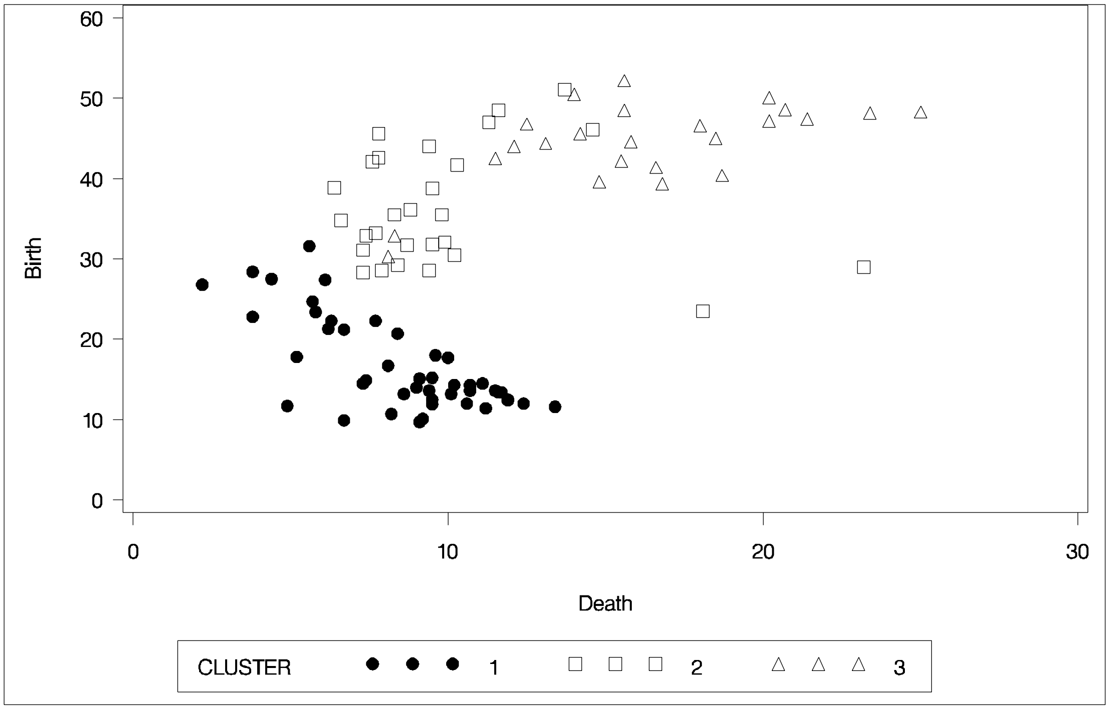


Figure 16.6. Scatter Plot of Poverty Data, Identified by Cluster

Figure 16.6 and Figure 16.7 display the separation of the clusters when three clusters are calculated.

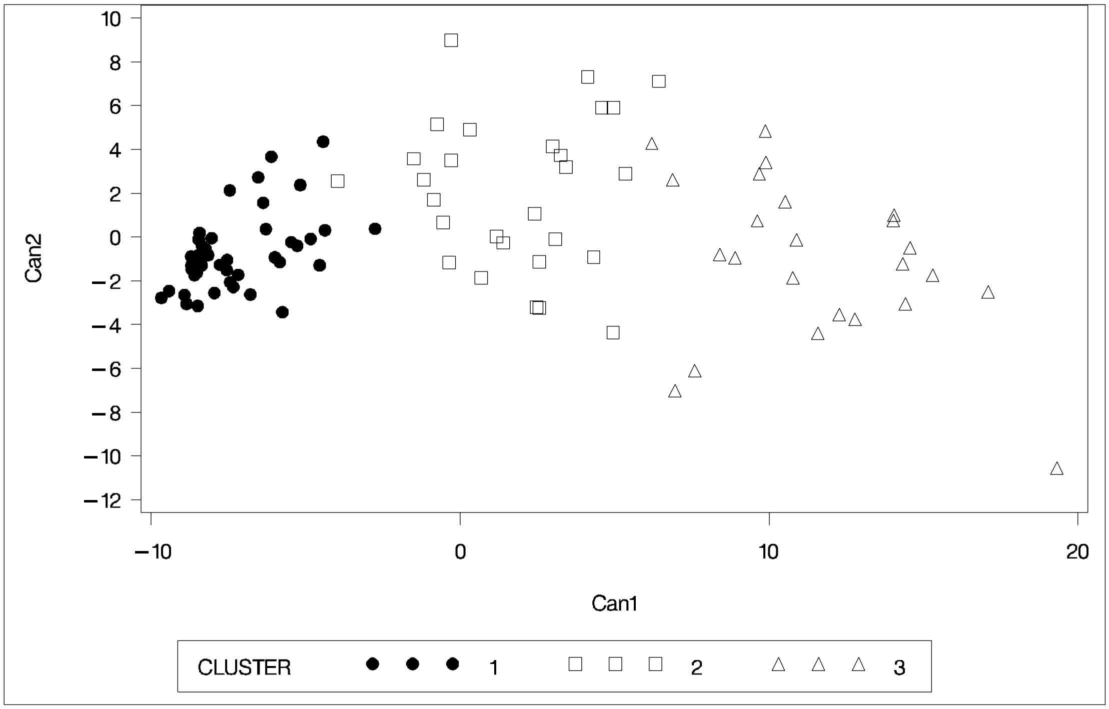


Figure 16.7. Scatter Plot of Canonical Variables

Syntax

The following statements are available in the ACECLUS procedure.

PROC ACECLUS *PROPORTION=p* **|** *THRESHOLD=t* < *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
VAR *variables* **;**
WEIGHT *variable* **;**

Usually you need only the VAR statement in addition to the required PROC ACECLUS statement. The optional BY, FREQ, VAR, and WEIGHT statements are described in alphabetical order after the PROC ACECLUS statement.

PROC ACECLUS Statement

PROC ACECLUS *PROPORTION=p* | *THRESHOLD=t* < *options* > **;**

The PROC ACECLUS statement starts the ACECLUS procedure. The options available with the PROC ACECLUS statement are summarized in Table 16.2 and discussed in the following sections. Note that, if you specify the METHOD=COUNT option, you must specify either the PROPORTION= or the MPAIRS= option. Otherwise, you must specify either the PROPORTION= or THRESHOLD= option.

Table 16.2. Summary of PROC ACECLUS Statement Options

Task	Options	Description
Specify clustering options		
	METHOD=	specify the clustering method
	MPAIRS=	specify number of pairs for estimating within-cluster covariance (when you specify the option METHOD=COUNT)
	PROPORTION=	specify proportion of pairs for estimating within-cluster covariance
	THRESHOLD=	specify the threshold for including pairs in the estimation of the within-cluster covariance
Specify input and output data sets		
	DATA=	specify input data set name
	OUT=	specify output data set name
	OUTSTAT=	specify output data set name containing various statistics
Specify iteration options		
	ABSOLUTE	use absolute instead of relative threshold
	CONVERGE=	specify convergence criterion
	INITIAL=	specify initial estimate of within-cluster covariance matrix
	MAXITER=	specify maximum number of iterations
	METRIC=	specify metric in which computations are performed
	SINGULAR=	specify singularity criterion
Specify canonical analysis options		
	N=	specify number of canonical variables
	PREFIX=	specify prefix for naming canonical variables
Control displayed output		
	NOPRINT	suppress the display of the output
	PP	produce PP-plot of distances between pairs from last iteration
	QQ	produce QQ-plot of power transformation of distances between pairs from last iteration
	SHORT	omit all output except for iteration history and eigenvalue table

The following list provides details on the options. The list is in alphabetical order.

ABSOLUTE

causes the THRESHOLD= value or the threshold computed from the PROPORTION= option to be treated absolutely rather than relative to the root mean square distance between observations. Use the ABSOLUTE option only when you are confident that the initial estimate of the within-cluster covariance matrix is close to the final estimate, such as when the INITIAL= option specifies a data set created by a previous execution of PROC ACECLUS using the OUTSTAT= option.

CONVERGE=*c*

specifies the convergence criterion. By default, CONVERGE= 0.001. Iteration stops when the convergence measure falls below the value specified by the CONVERGE= option or when the iteration limit as specified by the MAXITER= option is exceeded, whichever happens first.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. By default, PROC ACECLUS uses the most recently created SAS data set.

INITIAL=*name*

specifies the matrix for the initial estimate of the within-cluster covariance matrix. Valid values for *name* are as follows:

DIAGONAL | D
uses the diagonal matrix of sample variances as the initial estimate of the within-cluster covariance matrix.

FULL | F
uses the total-sample covariance matrix as the initial estimate of the within-cluster covariance matrix.

IDENTITY | I
uses the identity matrix as the initial estimate of the within-cluster covariance matrix.

INPUT=*SAS-data-set*
specifies a SAS data set from which to obtain the initial estimate of the within-cluster covariance matrix. The data set can be TYPE=CORR, COV, UCORR, UCOV, SSCP, or ACE, or it can be an ordinary SAS data set. (See Appendix 1, “Special SAS Data Sets,” for descriptions of CORR, COV, UCORR, UCOV, and SSCP data sets. See the section “Output Data Sets” on page 409 for a description of ACE data sets.)

If you do not specify the INITIAL= option, the default is the matrix specified by the METRIC= option. If neither the INITIAL= nor the METRIC= option is specified, INITIAL=FULL is used if there are enough observations to obtain a nonsingular total-sample covariance matrix; otherwise, INITIAL=DIAGONAL is used.

MAXITER=*n*
specifies the maximum number of iterations. By default, MAXITER=10.

METHOD= COUNT | C
METHOD= THRESHOLD | T
specifies the clustering method. The METHOD=THRESHOLD option requests a method (also the default) that uses all pairs closer than a given cutoff value to form the estimate at each iteration. The METHOD=COUNT option requests a method that uses a number of pairs, m, with the smallest distances to form the estimate at each iteration.

METRIC=*name*
specifies the metric in which the computations are performed, implies the default value for the INITIAL= option, and specifies the matrix $\mathbf{Z}$ used in the formula for the convergence measure e_i and for checking singularity of the $\mathbf{A}$ matrix. Valid values for *name* are as follows:

DIAGONAL | D	uses the diagonal matrix of sample variances diag$(\mathbf{S})$ and sets $\mathbf{Z} = \text{diag}(\mathbf{S})^{-\frac{1}{2}}$, where the superscript $-\frac{1}{2}$ indicates an inverse factor.
FULL | F	uses the total-sample covariance matrix $\mathbf{S}$ and sets $\mathbf{Z} = \mathbf{S}^{-\frac{1}{2}}$.
IDENTITY | I	uses the identity matrix $\mathbf{I}$ and sets $\mathbf{Z} = \mathbf{I}$.

If you do not specify the METRIC= option, METRIC=FULL is used if there are enough observations to obtain a nonsingular total-sample covariance matrix; otherwise, METRIC=DIAGONAL is used.

The option METRIC= is rather technical. It affects the computations in a variety of ways, but for well-conditioned data the effects are subtle. For most data sets, the METRIC= option is not needed.

MPAIRS=*m*
specifies the number of pairs to be included in the estimation of the within-cluster covariance matrix when METHOD=COUNT is requested. The values of m must be greater than 0 but less than or equal to $(\textit{totfq} \times (\textit{totfq} - 1))/2$, where *totfq* is the sum of nonmissing frequencies specified in the FREQ statement. If there is no FREQ statement, *totfq* equals the number of total nonmissing observations.

N=*n*
specifies the number of canonical variables to be computed. The default is the number of variables analyzed. N=0 suppresses the canonical analysis.

NOPRINT
suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, “Using the Output Delivery System.”

OUT=*SAS-data-set*

creates an output SAS data set that contains all the original data as well as the canonical variables having an estimated within-cluster covariance matrix equal to the identity matrix. If you want to create a permanent SAS data set, you must specify a two-level name. See Chapter 16, "SAS Data Files" in *SAS Language Reference: Concepts* for information on permanent SAS data sets.

OUTSTAT=*SAS-data-set*

specifies a TYPE=ACE output SAS data set that contains means, standard deviations, number of observations, covariances, estimated within-cluster covariances, eigenvalues, and canonical coefficients. If you want to create a permanent SAS data set, you must specify a two-level name. See Chapter 16, "SAS Data Files" in *SAS Language Reference: Concepts* for information on permanent SAS data sets.

PROPORTION=*p*
PERCENT=*p*
P=*p*

specifies the percentage of pairs to be included in the estimation of the within-cluster covariance matrix. The value of p must be greater than 0. If p is greater than or equal to 1, it is interpreted as a percentage and divided by 100; PROPORTION=0.02 and PROPORTION=2 are equivalent. When you specify METHOD=THRESHOLD, a threshold value is computed from the PROPORTION= option under the assumption that the observations are sampled from a multivariate normal distribution.

When you specify METHOD=COUNT, the number of pairs, m, is computed from PROPORTION=p as

$$m = \text{floor}\left(\frac{p}{2} \times \textit{totfq} \times (\textit{totfq} - 1)\right)$$

where *totfq* is the number of total non-missing observations.

PP

produces a PP probability plot of distances between pairs of observations computed in the last iteration.

PREFIX=*name*

specifies a prefix for naming the canonical variables. By default the names are CAN1, CAN2, . . . , CANn. If you specify PREFIX=ABC, the variables are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the name length defined by the VALIDVARNAME= system option. For more information on the VALIDVARNAME= system option, refer to *SAS Language Reference: Dictionary*.

QQ

produces a QQ probability plot of a power transformation of the distances between pairs of observations computed in the last iteration. **Caution:** The QQ plot may require an enormous amount of computer time.

SHORT

omits all items from the standard output except for the iteration history and the eigenvalue table.

SINGULAR=*g*
SING=*g*

specifies a singularity criterion $0 < g < 1$ for the total-sample covariance matrix $\mathbf{S}$ and the approximate within-cluster covariance estimate $\mathbf{A}$. The default is SINGULAR=1E$-$4.

THRESHOLD=*t*
T=*t*

specifies the threshold for including pairs of observations in the estimation of the within-cluster covariance matrix. A pair of observations is included if the Euclidean distance between them is less than or equal to t times the root mean square distance computed over all pairs of observations.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ACECLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ACECLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

If you specify the INITIAL=INPUT= option and the INITIAL=INPUT= data set does not contain any of the BY variables, the entire INITIAL=INPUT= data set provides the initial value for the matrix $\mathbf{A}$ for each BY group in the DATA= data set.

If the INITIAL=INPUT= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the INITIAL=INPUT= data set as in the DATA= data set, then PROC ACECLUS displays an error message and stops.

If all the BY variables appear in the INITIAL=INPUT= data set with the same type and length as in the DATA= data set, then each BY group in the INITIAL=INPUT= data set provides the initial value for $\mathbf{A}$ for the corresponding BY group in the DATA=

data set. All BY groups in the DATA= data set must also appear in the INITIAL= INPUT= data set. The BY groups in the INITIAL=INPUT= data set must be in the same order as in the DATA= data set. If you specify NOTSORTED in the BY statement, identical BY groups must occur in the same order in both data sets. If you do not specify NOTSORTED, some BY groups can appear in the INITIAL= INPUT= data set, but not in the DATA= data set; such BY groups are not used in the analysis.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in your data set represents the frequency of occurrence for the observation, include the name of that variable in the FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. If a value of the FREQ variable is not integral, it is truncated to the largest integer not exceeding the given value. Observations with FREQ values less than one are not included in the analysis. The total number of observations is considered equal to the sum of the FREQ variable.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not specified in other statements are analyzed.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to specify relative weights for each observation in the input data set, place the weights in a variable in the data set and specify that variable name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. The values of the WEIGHT variable can be non-integral and are not truncated. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

The WEIGHT and FREQ statements have a similar effect, except in calculating the divisor of the $\mathbf{A}$ matrix.

Details

Missing Values

Observations with missing values are omitted from the analysis and are given missing values for canonical variable scores in the OUT= data set.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. The N= option determines the number of new variables. The OUT= data set is not created if N=0. The names of the new variables are formed by concatenating the value given by the PREFIX= option (or the prefix CAN if the PREFIX= option is not specified) and the numbers 1, 2, 3, and so on. The OUT= data set can be used as input to PROC CLUSTER or PROC FASTCLUS. The cluster analysis should be performed on the canonical variables, not on the original variables.

OUTSTAT= Data Set

The OUTSTAT= data set is a TYPE=ACE data set containing the following variables.

- the BY variables, if any
- the two new character variables, _TYPE_ and _NAME_
- the variables analyzed, that is, those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

TYPE	
MEAN	mean of each variable
STD	standard deviation of each variable
N	number of observations on which the analysis is based. This value is the same for each variable.
SUMWGT	sum of the weights if a WEIGHT statement is used. This value is the same for each variable.
COV	covariances between each variable and the variable named by the _NAME_ variable. The number of observations with _TYPE_=COV is equal to the number of variables being analyzed.
ACE	estimated within-cluster covariances between each variable and the variable named by the _NAME_ variable. The number of observations with _TYPE_=ACE is equal to the number of variables being analyzed.

EIGENVAL eigenvalues of INV(ACE)*(COV−ACE). If the N= option requests fewer than the maximum number of canonical variables, only the specified number of eigenvalues are produced, with missing values filling out the observation.

RAWSCORE raw canonical coefficients.

To obtain the canonical variable scores, these coefficients should be multiplied by the raw data centered by means obtained from the observation with _TYPE_='MEAN'.

SCORE standardized canonical coefficients. The _NAME_ variable contains the name of the corresponding canonical variable as constructed from the PREFIX= option. The number of observations with _TYPE_=SCORE equals the number of canonical variables computed.

To obtain the canonical variable scores, these coefficients should be multiplied by the standardized data using means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD'.

The OUTSTAT= data set can be used

- to initialize another execution of PROC ACECLUS
- to compute canonical variable scores with the SCORE procedure
- as input to the FACTOR procedure, specifying METHOD=SCORE, to rotate the canonical variables

Computational Resources

Let

$$\begin{aligned} n &= \text{number of observations} \\ v &= \text{number of variables} \\ i &= \text{number of iterations} \end{aligned}$$

Memory

The memory in bytes required by PROC ACECLUS is approximately

$$8(2n(v+1)+21v+5v^2)$$

bytes. If you request the PP or QQ option, an additional $4n(n-1)$ bytes are needed.

Time

The time required by PROC ACECLUS is roughly proportional to

$$2nv^2 + 10v^3 + i\left(\frac{n^2v}{2} + nv^2 + 5v^3\right)$$

Displayed Output

Unless the SHORT option is specified, the ACECLUS procedure displays the following items:

- Means and Standard Deviations of the input variables
- the **S** matrix, labeled COV: Total Sample Covariances
- the name or value of the matrix used for the Initial Within-Cluster Covariance Estimate
- the Threshold value if the PROPORTION= option is specified

For each iteration, PROC ACECLUS displays

- the Iteration number
- RMS Distance, the root mean square distance between all pairs of observations
- the Distance Cutoff (u) for including pairs of observations in the estimate of the within-cluster covariances, which equals the RMS distance times the threshold
- the number of Pairs Within Cutoff
- the Convergence Measure (e_i) as specified by the METRIC= option

If the SHORT option is not specified, PROC ACECLUS also displays the **A** matrix, labeled ACE: Approximate Covariance Estimate Within Clusters.

The ACECLUS procedure displays a table of eigenvalues from the canonical analysis containing the following items:

- Eigenvalues of Inv(ACE)*(COV−ACE)
- the Difference between successive eigenvalues
- the Proportion of variance explained by each eigenvalue
- the Cumulative proportion of variance explained

If the SHORT option is not specified, PROC ACECLUS displays

- the Eigenvectors or raw canonical coefficients
- the standardized eigenvectors or standard canonical coefficients

ODS Table Names

PROC ACECLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 16.3. ODS Tables Produced in PROC ACECLUS

ODS Table Name	Description	Statement	Option
ConvergenceStatus	Convergence status	PROC	default
DataOptionInfo	Data and option information	PROC	default
Eigenvalues	Eigenvalues of Inv(ACE)*(COV-ACE)	PROC	default
Eigenvectors	Eigenvectors (raw canonical coefficients)	PROC	default
InitWithin	Initial within-cluster covariance estimate	PROC	INITIAL=INPUT
IterHistory	Iteration history	PROC	default
SimpleStatistics	Simple statistics	PROC	default
StdCanCoef	Standardized canonical coefficients	PROC	default
Threshold	Threshold value	PROC	PROPORTION=
TotSampleCov	Total sample covariances	PROC	default
Within	Approximate covariance estimate within clusters	PROC	default

Example

Example 16.1. Transformation and Cluster Analysis of Fisher Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analyses of the iris data.

In this example PROC ACECLUS is used to transform the data, and the clustering is performed by PROC FASTCLUS. Compare this with the example in Chapter 28, "The FASTCLUS Procedure." The results from the FREQ procedure display fewer misclassifications when PROC ACECLUS is used. The following statements produce Output 16.1.1 through Output 16.1.5.

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;
```

```
data iris;
   title ’Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
```

```
proc aceclus data=iris out=ace p=.02 outstat=score;
   var SepalLength SepalWidth PetalLength PetalWidth ;
run;
```

```
legend1 frame cframe=white  cborder=black
   position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot data=ace;
   plot can2*can1=Species /
      frame cframe=white legend=legend1 vaxis=axis1 haxis=axis2;
   format Species specname. ;
run;
quit;
```

```
proc fastclus data=ace maxc=3 maxiter=10 conv=0 out=clus;
   var can:;
run;
```

```
proc freq;
   tables cluster*Species;
run;
```

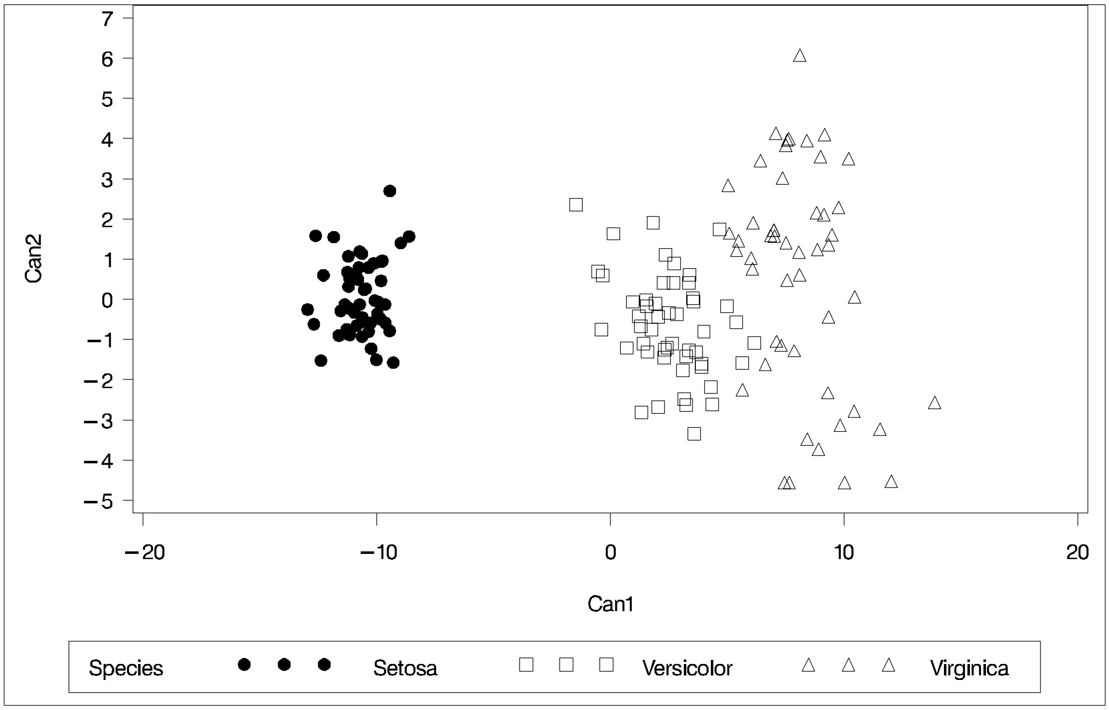
Output 16.1.1. Using PROC ACECLUS to Transform Fisher’s Iris Data

```
                           Fisher (1936) Iris Data

                            The ACECLUS Procedure

              Approximate Covariance Estimation for Cluster Analysis


                Observations         150    Proportion      0.0200
                Variables              4    Converge       0.00100


                      Means and Standard Deviations
                                   Standard
          Variable           Mean    Deviation    Label

          SepalLength     58.4333       8.2807    Sepal Length in mm.
          SepalWidth      30.5733       4.3587    Sepal Width in mm.
          PetalLength     37.5800      17.6530    Petal Length in mm.
          PetalWidth      11.9933       7.6224    Petal Width in mm.


                      COV: Total Sample Covariances

               SepalLength        SepalWidth       PetalLength        PetalWidth

SepalLength     68.5693512        -4.2434004       127.4315436        51.6270694
SepalWidth      -4.2434004        18.9979418       -32.9656376       -12.1639374
PetalLength    127.4315436       -32.9656376       311.6277852       129.5609396
PetalWidth      51.6270694       -12.1639374       129.5609396        58.1006264

      Initial Within-Cluster Covariance Estimate = Full Covariance Matrix


                           Threshold =    0.334211


                               Iteration History

                                                  Pairs
                          RMS       Distance     Within    Convergence
          Iteration    Distance      Cutoff      Cutoff      Measure
          ------------------------------------------------------------
                  1       2.828       0.945       408.0     0.465775
                  2      11.905       3.979       559.0     0.013487
                  3      13.152       4.396       940.0     0.029499
                  4      13.439       4.491      1506.0     0.046846
                  5      13.271       4.435      2036.0     0.046859
                  6      12.591       4.208      2285.0     0.025027
                  7      12.199       4.077      2366.0     0.009559
                  8      12.121       4.051      2402.0     0.003895
                  9      12.064       4.032      2417.0     0.002051
                 10      12.047       4.026      2429.0     0.000971



        Algorithm converged.
```

Output 16.1.2. Eigenvalues, Raw Canonical Coefficients, and Standardized Canonical Coefficients

```
              ACE: Approximate Covariance Estimate Within Clusters

                  SepalLength       SepalWidth      PetalLength       PetalWidth

SepalLength       11.73342939       5.47550432       4.95389049       2.02902429
SepalWidth         5.47550432       6.91992590       2.42177851       1.74125154
PetalLength        4.95389049       2.42177851       6.53746398       2.35302594
PetalWidth         2.02902429       1.74125154       2.35302594       2.05166735


                         Eigenvalues of Inv(ACE)*(COV-ACE)

                     Eigenvalue    Difference    Proportion    Cumulative

                1       63.7716       61.1593        0.9367        0.9367
                2        2.6123        1.5561        0.0384        0.9751
                3        1.0562        0.4167        0.0155        0.9906
                4        0.6395                     0.00939        1.0000


                    Eigenvectors (Raw Canonical Coefficients)

                                              Can1        Can2        Can3        Can4

 SepalLength   Sepal Length in mm.     -.012009    -.098074    -.059852    0.402352
 SepalWidth    Sepal Width in mm.      -.211068    -.000072    0.402391    -.225993
 PetalLength   Petal Length in mm.     0.324705    -.328583    0.110383    -.321069
 PetalWidth    Petal Width in mm.      0.266239    0.870434    -.085215    0.320286


                       Standardized Canonical Coefficients

                                              Can1        Can2        Can3        Can4

 SepalLength   Sepal Length in mm.     -0.09944    -0.81211    -0.49562     3.33174
 SepalWidth    Sepal Width in mm.      -0.91998    -0.00031     1.75389    -0.98503
 PetalLength   Petal Length in mm.      5.73200    -5.80047     1.94859    -5.66782
 PetalWidth    Petal Width in mm.       2.02937     6.63478    -0.64954     2.44134
```

Output 16.1.3. Plot of Transformed Iris Data: PROC PLOT



Output 16.1.4. Clustering of Transformed Iris Data: Partial Output from PROC FASTCLUS

```
                          The FASTCLUS Procedure
           Replace=FULL  Radius=0  Maxclusters=3 Maxiter=10  Converge=0

                             Cluster Summary

                                        Maximum Distance
                               RMS Std         from Seed     Radius     Nearest
 Cluster     Frequency       Deviation    to Observation    Exceeded    Cluster
 -----------------------------------------------------------------------------
    1               50        1.1016            5.2768                        3
    2               50        1.8880            6.8298                        3
    3               50        1.4138            5.3152                        2

                             Cluster Summary

                                       Distance Between
                          Cluster     Cluster Centroids
                          -----------------------------
                             1                  13.2845
                             2                   5.8580
                             3                   5.8580


                          Statistics for Variables

       Variable     Total STD    Within STD      R-Square     RSQ/(1-RSQ)
       ------------------------------------------------------------------
       Can1           8.04808       1.48537      0.966394       28.756658
       Can2           1.90061       1.85646      0.058725        0.062389
       Can3           1.43395       1.32518      0.157417        0.186826
       Can4           1.28044       1.27550      0.021025        0.021477
       OVER-ALL       4.24499       1.50298      0.876324        7.085666


                          Pseudo F Statistic =   520.80


               Approximate Expected Over-All R-Squared =   0.80391


                      Cubic Clustering Criterion =    5.179


                               Cluster Means

Cluster                Can1              Can2              Can3              Can4
-------------------------------------------------------------------------------
   1         -10.67516964        0.06706906        0.27068819        0.11164209
   2           8.12988211        0.52566663        0.51836499        0.14915404
   3           2.54528754       -0.59273569       -0.78905317       -0.26079612


                         Cluster Standard Deviations

Cluster                Can1              Can2              Can3              Can4
-------------------------------------------------------------------------------
   1          0.953761025       0.931943571       1.398456061       1.058217627
   2          1.799159552       2.743869556       1.270344142       1.370523175
   3          1.572366584       1.393565864       1.303411851       1.372050319
```

Output 16.1.5. Crosstabulation of Cluster by Species for Fisher's Iris Data: PROC FREQ

```
             The FREQ Procedure

        Table of CLUSTER by Species

CLUSTER(Cluster)     Species

Frequency|
Percent  |
Row Pct  |
Col Pct  |Setosa  |Versicol|Virginic|  Total
         |        |or      |a       |
---------+--------+--------+--------+
       1 |     50 |      0 |      0 |     50
         |  33.33 |   0.00 |   0.00 |  33.33
         | 100.00 |   0.00 |   0.00 |
         | 100.00 |   0.00 |   0.00 |
---------+--------+--------+--------+
       2 |      0 |      2 |     48 |     50
         |   0.00 |   1.33 |  32.00 |  33.33
         |   0.00 |   4.00 |  96.00 |
         |   0.00 |   4.00 |  96.00 |
---------+--------+--------+--------+
       3 |      0 |     48 |      2 |     50
         |   0.00 |  32.00 |   1.33 |  33.33
         |   0.00 |  96.00 |   4.00 |
         |   0.00 |  96.00 |   4.00 |
---------+--------+--------+--------+
Total          50       50       50      150
            33.33    33.33    33.33   100.00
```

References

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), "Data-based Metrics for Cluster Analysis," *Utilitas Mathematica*, 21A, 75–99.

Everitt, B.S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books Ltd.

Fisher, R.A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Hartigan, J.A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons, Inc.

Kettenring, R. (1984), personal communication.

Mezzich, J.E and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press, Inc.

Puri, M.L. and Sen, P.K. (1971), *Nonparametric Methods in Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Rouncefield, M. (1995), "The Statistics of Poverty and Inequality," *Journal of Statistics Education*, 3(2). [Online]: [http://www.stat.ncsu.edu/info/jse], accessed Dec. 19, 1997.

Wolfe, J.H. (1970),“Pattern Clustering by Multivariate Mixture Analysis,” *Multivariate Behavioral Research*, 5, 329–350.

Chapter 17
The ANOVA Procedure

Chapter Contents

Chapter 17
The ANOVA Procedure

Overview

The ANOVA procedure performs *analysis of variance* (ANOVA) for balanced data from a wide variety of experimental designs. In analysis of variance, a continuous response variable, known as a *dependent variable*, is measured under experimental conditions identified by classification variables, known as *independent variables*. The variation in the response is assumed to be due to effects in the classification, with random error accounting for the remaining variation.

The ANOVA procedure is one of several procedures available in SAS/STAT software for analysis of variance. The ANOVA procedure is designed to handle balanced data (that is, data with equal numbers of observations for every combination of the classification factors), whereas the GLM procedure can analyze both balanced and unbalanced data. Because PROC ANOVA takes into account the special structure of a balanced design, it is faster and uses less storage than PROC GLM for balanced data.

Use PROC ANOVA for the analysis of balanced data only, with the following exceptions: one-way analysis of variance, Latin square designs, certain partially balanced incomplete block designs, completely nested (hierarchical) designs, and designs with cell frequencies that are proportional to each other and are also proportional to the background population. These exceptions have designs in which the factors are all orthogonal to each other. For further discussion, refer to Searle (1971, p. 138). PROC ANOVA works for designs with block diagonal $\mathbf{X}'\mathbf{X}$ matrices where the elements of each block all have the same value. The procedure partially tests this requirement by checking for equal cell means. However, this test is imperfect: some designs that cannot be analyzed correctly may pass the test, and designs that can be analyzed correctly may not pass. If your design does not pass the test, PROC ANOVA produces a warning message to tell you that the design is unbalanced and that the ANOVA analyses may not be valid; if your design is not one of the special cases described here, then you should use PROC GLM instead. Complete validation of designs is not performed in PROC ANOVA since this would require the whole $\mathbf{X}'\mathbf{X}$ matrix; if you're unsure about the validity of PROC ANOVA for your design, you should use PROC GLM.

Caution: If you use PROC ANOVA for analysis of unbalanced data, you must assume responsibility for the validity of the results.

Experimental graphics are now available with the ANOVA procedure. For more information, see the "ODS Graphics" section on page 460.

Getting Started

The following examples demonstrate how you can use the ANOVA procedure to perform analyses of variance for a one-way layout and a randomized complete block design.

One-Way Layout with Means Comparisons

A one-way analysis of variance considers one treatment factor with two or more treatment levels. The goal of the analysis is to test for differences among the means of the levels and to quantify these differences. If there are two treatment levels, this analysis is equivalent to a t test comparing two group means.

The assumptions of analysis of variance (Steel and Torrie 1980) are

- treatment effects are additive
- experimental errors
  - are random
  - are independently distributed
  - follow a normal distribution
  - have mean zero and constant variance

The following example studies the effect of bacteria on the nitrogen content of red clover plants. The treatment factor is bacteria strain, and it has six levels. Five of the six levels consist of five different *Rhizobium trifolii* bacteria cultures combined with a composite of five *Rhizobium meliloti* strains. The sixth level is a composite of the five *Rhizobium trifolii* strains with the composite of the *Rhizobium meliloti*. Red clover plants are inoculated with the treatments, and nitrogen content is later measured in milligrams. The data are derived from an experiment by Erdman (1946) and are analyzed in Chapters 7 and 8 of Steel and Torrie (1980). The following DATA step creates the SAS data set Clover:

```
title1 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;
```

The variable Strain contains the treatment levels, and the variable Nitrogen contains the response. The following statements produce the analysis.

```
proc anova data = Clover;
   class strain;
   model Nitrogen = Strain;
run;
```

The classification variable is specified in the CLASS statement. Note that, unlike the GLM procedure, PROC ANOVA does not allow continuous variables on the right-hand side of the model. Figure 17.1 and Figure 17.2 display the output produced by these statements.

```
                      Nitrogen Content of Red Clover Plants

                               The ANOVA Procedure

                             Class Level Information

         Class          Levels    Values

         Strain              6    3DOK1 3DOK13 3DOK4 3DOK5 3DOK7 COMPOS


                    Number of Observations Read          30
                    Number of Observations Used          30
```

Figure 17.1. Class Level Information

The “Class Level Information” table shown in Figure 17.1 lists the variables that appear in the CLASS statement, their levels, and the number of observations in the data set.

Figure 17.2 displays the ANOVA table, followed by some simple statistics and tests of effects.

```
Dependent Variable: Nitrogen

                                         Sum of
 Source                      DF         Squares     Mean Square    F Value    Pr > F

 Model                        5      847.046667      169.409333      14.37    <.0001

 Error                       24      282.928000       11.788667

 Corrected Total             29     1129.974667


             R-Square     Coeff Var      Root MSE    Nitrogen Mean

             0.749616      17.26515      3.433463         19.88667


 Source                      DF        Anova SS     Mean Square    F Value    Pr > F

 Strain                       5     847.0466667     169.4093333      14.37    <.0001
```

Figure 17.2. ANOVA Table

The degrees of freedom (DF) column should be used to check the analysis results. The model degrees of freedom for a one-way analysis of variance are the number of levels minus 1; in this case, $6 - 1 = 5$. The Corrected Total degrees of freedom are always the total number of observations minus one; in this case $30 - 1 = 29$. The sum of Model and Error degrees of freedom equal the Corrected Total.

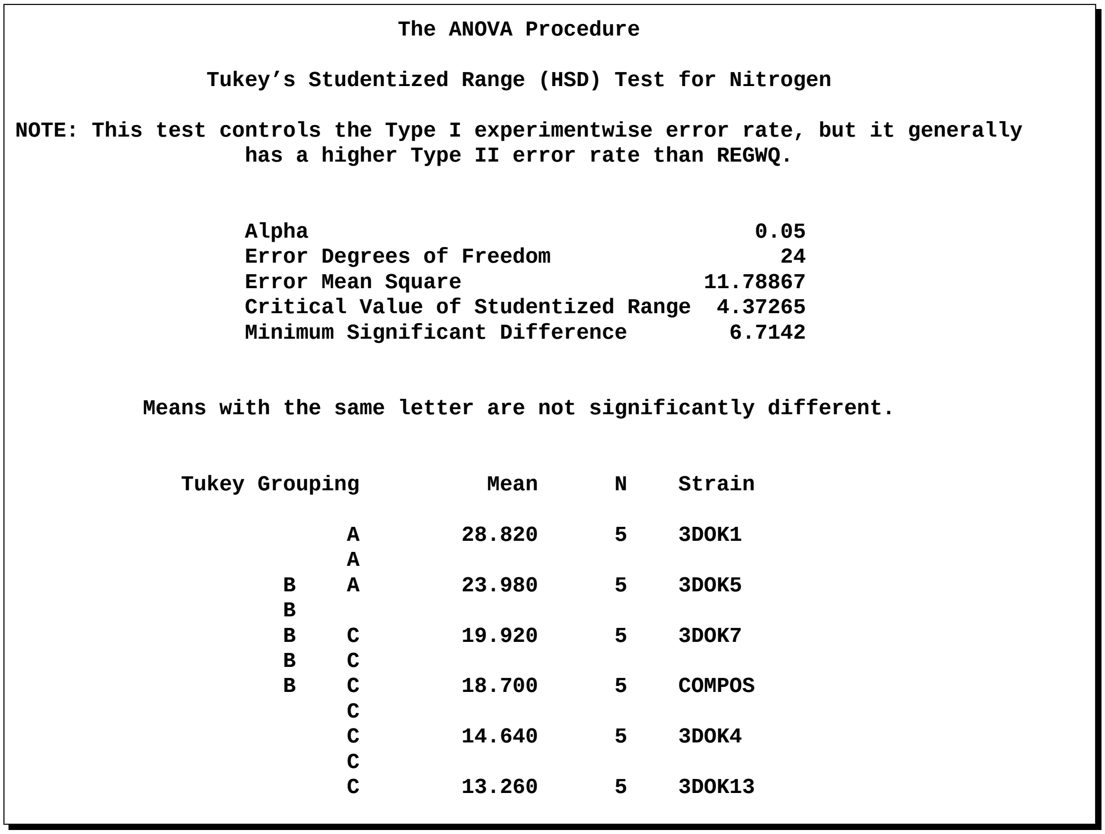
The overall F test is significant $(F = 14.37, p < 0.0001)$, indicating that the model as a whole accounts for a significant portion of the variability in the dependent variable. The F test for Strain is significant, indicating that some contrast between the means for the different strains is different from zero. Notice that the Model and Strain F tests are identical, since Strain is the only term in the model.

The F test for Strain $(F = 14.37, p < 0.0001)$ suggests that there are differences among the bacterial strains, but it does not reveal any information about the nature of the differences. Mean comparison methods can be used to gather further information. The interactivity of PROC ANOVA enables you to do this without re-running the entire analysis. After you specify a model with a MODEL statement and execute the ANOVA procedure with a RUN statement, you can execute a variety of statements (such as MEANS, MANOVA, TEST, and REPEATED) without PROC ANOVA recalculating the model sum of squares.

The following command requests means of the Strain levels with Tukey's studentized range procedure.

```
means strain / tukey;
```

Results of Tukey's procedure are shown in Figure 17.3.



```
                              The ANOVA Procedure

               Tukey’s Studentized Range (HSD) Test for Nitrogen

NOTE: This test controls the Type I experimentwise error rate, but it generally
                  has a higher Type II error rate than REGWQ.


                  Alpha                                   0.05
                  Error Degrees of Freedom                  24
                  Error Mean Square                   11.78867
                  Critical Value of Studentized Range  4.37265
                  Minimum Significant Difference        6.7142


          Means with the same letter are not significantly different.


             Tukey Grouping          Mean      N    Strain

                          A        28.820      5    3DOK1
                          A
                     B    A        23.980      5    3DOK5
                     B
                     B    C        19.920      5    3DOK7
                     B    C
                     B    C        18.700      5    COMPOS
                          C
                          C        14.640      5    3DOK4
                          C
                          C        13.260      5    3DOK13
```

Figure 17.3. Tukey’s Multiple Comparisons Procedure

The multiple comparisons results indicate, for example, that

- strain 3DOK1 fixes significantly more nitrogen than all but 3DOK5
- even though 3DOK5 is not significantly different from 3DOK1, it is also not significantly better than all the rest

Although the experiment has succeeded in separating the best strains from the worst, clearly distinguishing the very best strain requires more experimentation.

The experimental graphics features of PROC ANOVA enable you to visualize the distribution of nitrogen content for each treatment.

```
ods html;
ods graphics on;

proc anova data = Clover;
   class strain;
   model Nitrogen = Strain;
run;

ods graphics off;
ods html close;
```

When you specify the experimental ODS GRAPHICS statement and fit a one-way analysis of variance model, the ANOVA procedure output includes a box plot of the dependent variable values within each classification level of the independent variable. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the ANOVA procedure, see the section “ODS Graphics” on page 460.

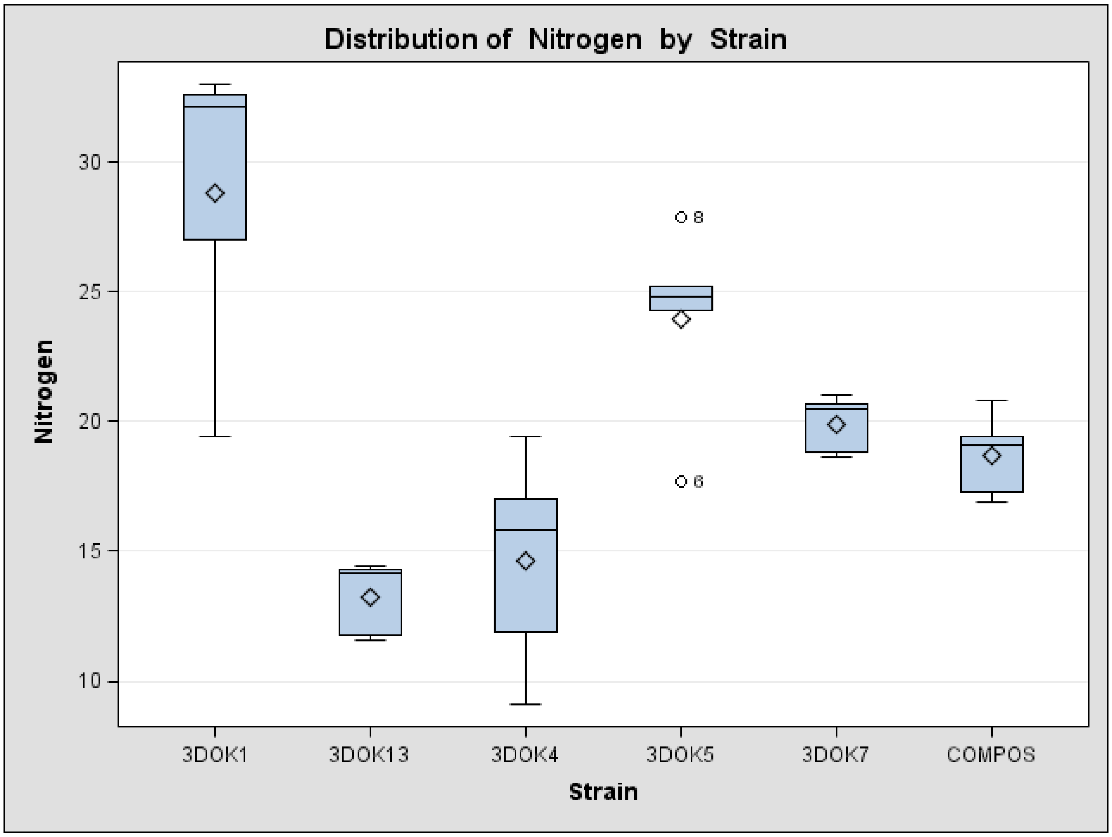


Figure 17.4. Box Plot of Nitrogen Content for each Treatment (Experimental)

Randomized Complete Block with One Factor

This example illustrates the use of PROC ANOVA in analyzing a randomized complete block design. Researchers are interested in whether three treatments have different effects on the yield and worth of a particular crop. They believe that the experimental units are not homogeneous. So, a blocking factor is introduced that allows the experimental units to be homogeneous within each block. The three treatments are then randomly assigned within each block.

The data from this study are input into the SAS data set RCB:

```
title1 'Randomized Complete Block';
data RCB;
   input Block Treatment $ Yield Worth @@;
   datalines;
```

```
1 A 32.6 112   1 B 36.4 130   1 C 29.5 106
2 A 42.7 139   2 B 47.1 143   2 C 32.9 112
3 A 35.3 124   3 B 40.1 134   3 C 33.6 116
;
```

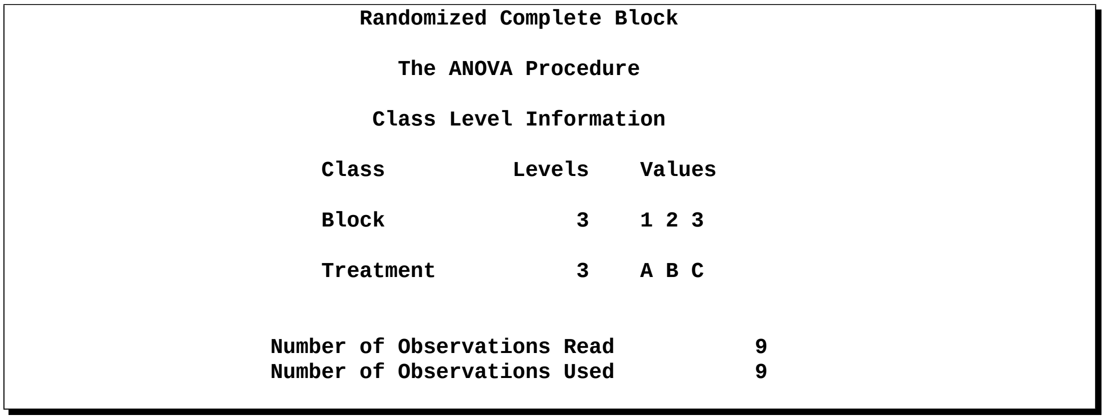
The variables Yield and Worth are continuous response variables, and the variables Block and Treatment are the classification variables. Because the data for the analysis are balanced, you can use PROC ANOVA to run the analysis.

The statements for the analysis are

```
proc anova data=RCB;
   class Block Treatment;
   model Yield Worth=Block Treatment;
run;
```

The Block and Treatment effects appear in the CLASS statement. The MODEL statement requests an analysis for each of the two dependent variables, Yield and Worth.

Figure 17.5 shows the “Class Level Information” table.



```
                       Randomized Complete Block

                          The ANOVA Procedure

                        Class Level Information

                   Class           Levels    Values

                   Block                3    1 2 3

                   Treatment            3    A B C


               Number of Observations Read           9
               Number of Observations Used           9
```

Figure 17.5. Class Level Information

The “Class Level Information” table lists the number of levels and their values for all effects specified in the CLASS statement. The number of observations in the data set are also displayed. Use this information to make sure that the data have been read correctly.

The overall ANOVA table for Yield in Figure 17.6 appears first in the output because it is the first response variable listed on the left side in the MODEL statement.

```
Dependent Variable: Yield

                                          Sum of
 Source                      DF          Squares     Mean Square    F Value    Pr > F

 Model                        4      225.2777778      56.3194444       8.94    0.0283

 Error                        4       25.1911111       6.2977778

 Corrected Total              8      250.4688889


              R-Square     Coeff Var      Root MSE    Yield Mean

              0.899424      6.840047      2.509537      36.68889
```

Figure 17.6. Overall ANOVA Table for Yield

The overall F statistic is significant $(F = 8.94, p = 0.02583)$, indicating that the model as a whole accounts for a significant portion of the variation in Yield and that you may proceed to tests of effects.

The degrees of freedom (DF) are used to ensure correctness of the data and model. The Corrected Total degrees of freedom are one less than the total number of observations in the data set; in this case, $9 - 1 = 8$. The Model degrees of freedom for a randomized complete block are $(b - 1) + (t - 1)$, where b =number of block levels and t =number of treatment levels. In this case, $(3 - 1) + (3 - 1) = 4$.

Several simple statistics follow the ANOVA table. The R-Square indicates that the model accounts for nearly 90% of the variation in the variable Yield. The coefficient of variation (C.V.) is listed along with the Root MSE and the mean of the dependent variable. The Root MSE is an estimate of the standard deviation of the dependent variable. The C.V. is a unitless measure of variability.

The tests of the effects shown in Figure 17.7 are displayed after the simple statistics.

```
Dependent Variable: Yield

 Source                      DF         Anova SS     Mean Square    F Value    Pr > F

 Block                        2       98.1755556      49.0877778       7.79    0.0417
 Treatment                    2      127.1022222      63.5511111      10.09    0.0274
```

Figure 17.7. Tests of Effects for Yield

For Yield, both the Block and Treatment effects are significant $(F = 7.79, p = 0.0417$ and $F = 10.09, p = 0.0274$, respectively) at the 95% level. From this you can conclude that blocking is useful for this variable and that some contrast between the treatment means is significantly different from zero.

Figure 17.8 shows the ANOVA table, simple statistics, and tests of effects for the variable Worth.

```
Dependent Variable: Worth

                                            Sum of
 Source                      DF            Squares     Mean Square    F Value    Pr > F

 Model                        4        1247.333333      311.833333       8.28    0.0323

 Error                        4         150.666667       37.666667

 Corrected Total              8        1398.000000


               R-Square     Coeff Var       Root MSE     Worth Mean

               0.892227      4.949450       6.137318       124.0000


 Source                      DF          Anova SS      Mean Square    F Value    Pr > F

 Block                        2       354.6666667      177.3333333       4.71    0.0889
 Treatment                    2       892.6666667      446.3333333      11.85    0.0209
```

Figure 17.8. ANOVA Table for Worth

The overall F test is significant $(F = 8.28, p = 0.0323)$ at the 95% level for the variable Worth. The Block effect is not significant at the 0.05 level but is significant at the 0.10 confidence level $(F = 4.71, p = 0.0889)$. Generally, the usefulness of blocking should be determined before the analysis. However, since there are two dependent variables of interest, and Block is significant for one of them (Yield), blocking appears to be generally useful. For Worth, as with Yield, the effect of Treatment is significant $(F = 11.85, p = 0.0209)$.

Issuing the following command produces the Treatment means.

```
      means Treatment;
   run;
```

Figure 17.9 displays the treatment means and their standard deviations for both dependent variables.

```
                                  The ANOVA Procedure

 Level of            ------------Yield-----------    ------------Worth-----------
 Treatment    N            Mean          Std Dev            Mean          Std Dev

 A            3      36.8666667       5.22908532      125.000000       13.5277493
 B            3      41.2000000       5.43415127      135.666667        6.6583281
 C            3      32.0000000       2.19317122      111.333333        5.0332230
```

Figure 17.9. Means of Yield and Worth

Syntax

The following statements are available in PROC ANOVA.

PROC ANOVA < *options* > **;**
CLASS *variables* < / *option* > **;**
MODEL *dependents=effects* < / *options* > **;**
ABSORB *variables* **;**
BY *variables* **;**
FREQ *variable* **;**
MANOVA < *test-options* >< / *detail-options* > **;**
MEANS *effects* < / *options* > **;**
REPEATED *factor-specification* < / *options* > **;**
TEST < **H=***effects* > **E=***effect* **;**

The PROC ANOVA, CLASS, and MODEL statements are required, and they must precede the first RUN statement. The CLASS statement must precede the MODEL statement. If you use the ABSORB, FREQ, or BY statement, it must precede the first RUN statement. The MANOVA, MEANS, REPEATED, and TEST statements must follow the MODEL statement, and they can be specified in any order. These four statements can also appear after the first RUN statement.

The following table summarizes the function of each statement (other than the PROC statement) in the ANOVA procedure:

Table 17.1. Statements in the ANOVA Procedure

Statement	Description
ABSORB	absorbs classification effects in a model
BY	specifies variables to define subgroups for the analysis
CLASS	declares classification variables
FREQ	specifies a frequency variable
MANOVA	performs a multivariate analysis of variance
MEANS	computes and compares means
MODEL	defines the model to be fit
REPEATED	performs multivariate and univariate repeated measures analysis of variance
TEST	constructs tests using the sums of squares for effects and the error term you specify

PROC ANOVA Statement

PROC ANOVA < *options* > **;**

The PROC ANOVA statement starts the ANOVA procedure.

You can specify the following options in the PROC ANOVA statement:

DATA=*SAS-data-set*

names the SAS data set used by the ANOVA procedure. By default, PROC ANOVA uses the most recently created SAS data set.

MANOVA

requests the multivariate mode of eliminating observations with missing values. If any of the dependent variables have missing values, the procedure eliminates that observation from the analysis. The MANOVA option is useful if you use PROC ANOVA in interactive mode and plan to perform a multivariate analysis.

MULTIPASS

requests that PROC ANOVA reread the input data set, when necessary, instead of writing the values of dependent variables to a utility file. This option decreases disk space usage at the expense of increased execution times and is useful only in rare situations where disk space is at an absolute premium.

NAMELEN=n

specifies the length of effect names to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you want to create only the output data set with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The following table shows how PROC ANOVA interprets values of the ORDER= option.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

OUTSTAT=*SAS-data-set*

names an output data set that contains sums of squares, degrees of freedom, F statistics, and probability levels for each effect in the model. If you use the CANONICAL option in the MANOVA statement and do not use an M= specification in the MANOVA statement, the data set also contains results of the canonical analysis. See the "Output Data Set" section on page 455 for more information.

ABSORB Statement

ABSORB *variables* **;**

Absorption is a computational technique that provides a large reduction in time and memory requirements for certain types of models. The *variables* are one or more variables in the input data set.

For a main effect variable that does not participate in interactions, you can absorb the effect by naming it in an ABSORB statement. This means that the effect can be adjusted out before the construction and solution of the rest of the model. This is particularly useful when the effect has a large number of levels.

Several variables can be specified, in which case each one is assumed to be nested in the preceding variable in the ABSORB statement.

Note: When you use the ABSORB statement, the data set (or each BY group, if a BY statement appears) must be sorted by the variables in the ABSORB statement. Including an absorbed variable in the CLASS list or in the MODEL statement may produce erroneous sums of squares. If the ABSORB statement is used, it must appear before the first RUN statement or it is ignored.

When you use an ABSORB statement and also use the INT option in the MODEL statement, the procedure ignores the option but produces the uncorrected total sum of squares (SS) instead of the corrected total SS.

See the "Absorption" section on page 1799 in Chapter 32, "The GLM Procedure," for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ANOVA to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ANOVA procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

Since sorting the data changes the order in which PROC ANOVA reads observations, the sorting order for the levels of the classification variables may be affected if you have also specified the ORDER=DATA option in the PROC ANOVA statement.

If the BY statement is used, it must appear before the first RUN statement or it is ignored. When you use a BY statement, the interactive features of PROC ANOVA are disabled.

When both a BY and an ABSORB statement are used, observations must be sorted first by the variables in the BY statement, and then by the variables in the ABSORB statement.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < */ option* > **;**

The CLASS statement names the classification variables to be used in the model. Typical class variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. The CLASS statement is required, and it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Concepts*.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

The analysis produced using a FREQ statement reflects the expanded number of observations. For example, means and total degrees of freedom reflect the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set would be identical. Each observation in the old data set would be replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

If the FREQ statement is used, it must appear before the first RUN statement or it is ignored.

MANOVA Statement

MANOVA < *test-options* >< / *detail-options* > **;**

If the MODEL statement includes more than one dependent variable, you can perform multivariate analysis of variance with the MANOVA statement. The *test-options* define which effects to test, while the *detail-options* specify how to execute the tests and what results to display.

When a MANOVA statement appears before the first RUN statement, PROC ANOVA enters a multivariate mode with respect to the handling of missing values; in addition to observations with missing independent variables, observations with *any* missing dependent variables are excluded from the analysis. If you want to use this mode of handling missing values but do not need any multivariate analyses, specify the MANOVA option in the PROC ANOVA statement.

Test Options

You can specify the following options in the MANOVA statement as *test-options* in order to define which multivariate tests to perform.

H=*effects* | **INTERCEPT** | **_ALL_**

specifies effects in the preceding model to use as hypothesis matrices. For each SSCP matrix $\mathbf{H}$ associated with an effect, the H= specification computes an analysis based on the characteristic roots of $\mathbf{E}^{-1}\mathbf{H}$, where $\mathbf{E}$ is the matrix associated with the error effect. The characteristic roots and vectors are displayed, along with the Hotelling-Lawley trace, Pillai's trace, Wilks' criterion, and Roy's maximum root criterion with approximate F statistics. By default, these statistics are tested with approximations based on the F distribution. To test them with exact (but computationally intensive) calculations, use the MSTAT=EXACT option.

Use the keyword INTERCEPT to produce tests for the intercept. To produce tests for all effects listed in the MODEL statement, use the keyword _ALL_ in place of a list of effects. For background and further details, see the "Multivariate Analysis of Variance" section on page 1823 in Chapter 32, "The GLM Procedure."

E=*effect*

specifies the error effect. If you omit the E= specification, the ANOVA procedure uses the error SSCP (residual) matrix from the analysis.

M=*equation,. . .,equation* | **(***row-of-matrix,. . .,row-of-matrix***)**

specifies a transformation matrix for the dependent variables listed in the MODEL statement. The equations in the M= specification are of the form

$$\begin{aligned} c_1 \times \textit{dependent-variable} \quad & \pm \quad c_2 \times \textit{dependent-variable} \\ \cdots \quad & \pm \quad c_n \times \textit{dependent-variable} \end{aligned}$$

where the c_i values are coefficients for the various *dependent-variables*. If the value of a given c_i is 1, it may be omitted; in other words $1 \times Y$ is the same as Y. Equations should involve two or more dependent variables. For sample syntax, see the "Examples" section on page 439.

Alternatively, you can input the transformation matrix directly by entering the elements of the matrix with commas separating the rows, and parentheses surrounding the matrix. When this alternate form of input is used, the number of elements in each row must equal the number of dependent variables. Although these combinations actually represent the columns of the $\mathbf{M}$ matrix, they are displayed by rows.

When you include an M= specification, the analysis requested in the MANOVA statement is carried out for the variables defined by the equations in the specification, not the original dependent variables. If you omit the M= option, the analysis is performed for the original dependent variables in the MODEL statement.

If an M= specification is included without either the MNAMES= or the PREFIX= option, the variables are labeled MVAR1, MVAR2, and so forth by default. For further information, see the section “Multivariate Analysis of Variance” on page 1823 in Chapter 32, “The GLM Procedure.”

MNAMES=*names*

provides names for the variables defined by the equations in the M= specification. Names in the list correspond to the M= equations or the rows of the $\mathbf{M}$ matrix (as it is entered).

PREFIX=*name*

is an alternative means of identifying the transformed variables defined by the M= specification. For example, if you specify PREFIX=DIFF, the transformed variables are labeled DIFF1, DIFF2, and so forth.

Detail Options

You can specify the following options in the MANOVA statement after a slash as *detail-options*:

CANONICAL

produces a canonical analysis of the $\mathbf{H}$ and $\mathbf{E}$ matrices (transformed by the $\mathbf{M}$ matrix, if specified) instead of the default display of characteristic roots and vectors.

MSTAT=FAPPROX
MSTAT=EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the “Multivariate Tests” section in Chapter 2, “Introduction to Regression Procedures.” Alternatively, you can specify MSTAT=EXACT to compute exact *p*-values for three of the four tests (Wilks’ Lambda, the Hotelling-Lawley Trace, and Roy’s Greatest Root) and an improved F-approximation for the fourth (Pillai’s Trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy’s Greatest Root, computations for the exact *p*-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information on the results of MSTAT=EXACT, see the “Multivariate Analysis of Variance” section on page 1823 in Chapter 32, “The GLM Procedure.”

ORTH

requests that the transformation matrix in the M= specification of the MANOVA statement be orthonormalized by rows before the analysis.

PRINTE

displays the error SSCP matrix $\mathbf{E}$. If the $\mathbf{E}$ matrix is the error SSCP (residual) matrix from the analysis, the partial correlations of the dependent variables given the independent variables are also produced.

For example, the statement

```
manova / printe;
```

displays the error SSCP matrix and the partial correlation matrix computed from the error SSCP matrix.

PRINTH
displays the hypothesis SSCP matrix **H** associated with each effect specified by the H= specification.

SUMMARY
produces analysis-of-variance tables for each dependent variable. When no **M** matrix is specified, a table is produced for each original dependent variable from the MODEL statement; with an **M** matrix other than the identity, a table is produced for each transformed variable defined by the **M** matrix.

Examples

The following statements give several examples of using a MANOVA statement.

```
proc anova;
   class A B;
   model Y1-Y5=A B(A);
   manova h=A e=B(A) / printh printe;
   manova h=B(A) / printe;
   manova h=A e=B(A) m=Y1-Y2,Y2-Y3,Y3-Y4,Y4-Y5
          prefix=diff;

   manova h=A e=B(A) m=(1 -1  0  0  0,
                        0  1 -1  0  0,
                        0  0  1 -1  0,
                        0  0  0  1 -1) prefix=diff;
run;
```

The first MANOVA statement specifies A as the hypothesis effect and B(A) as the error effect. As a result of the PRINTH option, the procedure displays the hypothesis SSCP matrix associated with the A effect; and, as a result of the PRINTE option, the procedure displays the error SSCP matrix associated with the B(A) effect.

The second MANOVA statement specifies B(A) as the hypothesis effect. Since no error effect is specified, PROC ANOVA uses the error SSCP matrix from the analysis as the **E** matrix. The PRINTE option displays this **E** matrix. Since the **E** matrix is the error SSCP matrix from the analysis, the partial correlation matrix computed from this matrix is also produced.

The third MANOVA statement requests the same analysis as the first MANOVA statement, but the analysis is carried out for variables transformed to be successive differences between the original dependent variables. The PREFIX=DIFF specification labels the transformed variables as DIFF1, DIFF2, DIFF3, and DIFF4.

Finally, the fourth MANOVA statement has the identical effect as the third, but it uses an alternative form of the M= specification. Instead of specifying a set of equations, the fourth MANOVA statement specifies rows of a matrix of coefficients for the five dependent variables.

As a second example of the use of the M= specification, consider the following:

```
proc anova;
   class group;
   model dose1-dose4=group / nouni;
   manova h = group
          m = -3*dose1 -   dose2 +   dose3 + 3*dose4,
                 dose1 -   dose2 -   dose3 +   dose4,
                -dose1 + 3*dose2 - 3*dose3 +   dose4
          mnames = Linear Quadratic Cubic
          / printe;
run;
```

The M= specification gives a transformation of the dependent variables dose1 through dose4 into orthogonal polynomial components, and the MNAMES= option labels the transformed variables as LINEAR, QUADRATIC, and CUBIC, respectively. Since the PRINTE option is specified and the default residual matrix is used as an error term, the partial correlation matrix of the orthogonal polynomial components is also produced.

For further information, see the “Multivariate Analysis of Variance” section on page 1823 in Chapter 32, “The GLM Procedure.”

MEANS Statement

MEANS *effects* < **/** *options* > **;**

PROC ANOVA can compute means of the dependent variables for any effect that appears on the right-hand side in the MODEL statement.

You can use any number of MEANS statements, provided that they appear after the MODEL statement. For example, suppose A and B each have two levels. Then, if you use the following statements

```
proc anova;
   class A B;
   model Y=A B A*B;
   means A B / tukey;
   means A*B;
run;
```

means, standard deviations, and Tukey’s multiple comparison tests are produced for each level of the main effects A and B, and just the means and standard deviations for each of the four combinations of levels for A*B. Since multiple comparisons options apply only to main effects, the single MEANS statement

```
means A B A*B / tukey;
```

produces the same results.

Options are provided to perform multiple comparison tests for only main effects in the model. PROC ANOVA does not perform multiple comparison tests for interaction terms in the model; for multiple comparisons of interaction terms, see the LSMEANS statement in Chapter 32, "The GLM Procedure."

The following table summarizes categories of options available in the MEANS statement.

Table 17.2. Options Available in the MEANS Statement

Task	**Available options**
Perform multiple comparison tests	BON
	DUNCAN
	DUNNETT
	DUNNETTL
	DUNNETTU
	GABRIEL
	GT2
	LSD
	REGWQ
	SCHEFFE
	SIDAK
	SMM
Perform multiple comparison tests	SNK
	T
	TUKEY
	WALLER
Specify additional details for multiple comparison tests	ALPHA=
	CLDIFF
	CLM
	E=
	KRATIO=
	LINES
	NOSORT
Test for homogeneity of variances	HOVTEST
Compensate for heterogeneous variances	WELCH

Descriptions of these options follow. For a further discussion of these options, see the section "Multiple Comparisons" on page 1806 in Chapter 32, "The GLM Procedure."

ALPHA=*p*

specifies the level of significance for comparisons among the means. By default, ALPHA=0.05. You can specify any value greater than 0 and less than 1.

BON

performs Bonferroni t tests of differences between means for all main effect means in the MEANS statement. See the CLDIFF and LINES options, which follow, for a discussion of how the procedure displays results.

CLDIFF

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, GT2, T, LSD, and TUKEY options as confidence intervals for all pairwise differences between means, and the results of the DUNNETT, DUNNETTU, and DUNNETTL options as confidence intervals for differences with the control. The CLDIFF option is the default for unequal cell sizes unless the DUNCAN, REGWQ, SNK, or WALLER option is specified.

CLM

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, T, and LSD options as intervals for the mean of each level of the variables specified in the MEANS statement. For all options except GABRIEL, the intervals are confidence intervals for the true means. For the GABRIEL option, they are *comparison intervals* for comparing means pairwise: in this case, if the intervals corresponding to two means overlap, the difference between them is insignificant according to Gabriel's method.

DUNCAN

performs Duncan's multiple range test on all main effect means given in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

DUNNETT < **(***formatted-control-values***)** >

performs Dunnett's two-tailed t test, testing if any treatments are significantly different from a single control for all main effects means in the MEANS statement.

To specify which level of the effect is the control, enclose the formatted value in quotes in parentheses after the keyword. If more than one effect is specified in the MEANS statement, you can use a list of control values within the parentheses. By default, the first level of the effect is used as the control. For example,

```
means a / dunnett('CONTROL');
```

where CONTROL is the formatted control value of A. As another example,

```
means a b c / dunnett('CNTLA' 'CNTLB' 'CNTLC');
```

where CNTLA, CNTLB, and CNTLC are the formatted control values for A, B, and C, respectively.

DUNNETTL < **(***formatted-control-value***)** >

performs Dunnett's one-tailed t test, testing if any treatment is significantly less than the control. Control level information is specified as described previously for the DUNNETT option.

DUNNETTU < **(***formatted-control-value***)** >

performs Dunnett's one-tailed t test, testing if any treatment is significantly greater than the control. Control level information is specified as described previously for the DUNNETT option.

E=*effect*

specifies the error mean square used in the multiple comparisons. By default, PROC ANOVA uses the residual Mean Square (MS). The effect specified with the E= option must be a term in the model; otherwise, the procedure uses the residual MS.

GABRIEL

performs Gabriel's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

GT2

see the SMM option.

HOVTEST
HOVTEST=BARTLETT
HOVTEST=BF
HOVTEST=LEVENE <(TYPE=ABS | SQUARE)>
HOVTEST=OBRIEN <(W=*number*)>

requests a homogeneity of variance test for the groups defined by the MEANS effect. You can optionally specify a particular test; if you do not specify a test, Levene's test (Levene 1960) with TYPE=SQUARE is computed. Note that this option is ignored unless your MODEL statement specifies a simple one-way model.

The HOVTEST=BARTLETT option specifies Bartlett's test (Bartlett 1937), a modification of the normal-theory likelihood ratio test.

The HOVTEST=BF option specifies Brown and Forsythe's variation of Levene's test (Brown and Forsythe 1974).

The HOVTEST=LEVENE option specifies Levene's test (Levene 1960), which is widely considered to be the standard homogeneity of variance test. You can use the TYPE= option in parentheses to specify whether to use the absolute residuals (TYPE=ABS) or the squared residuals (TYPE=SQUARE) in Levene's test. The default is TYPE=SQUARE.

The HOVTEST=OBRIEN option specifies O'Brien's test (O'Brien 1979), which is basically a modification of HOVTEST=LEVENE(TYPE=SQUARE). You can use the W= option in parentheses to tune the variable to match the suspected kurtosis of the underlying distribution. By default, W=0.5, as suggested by O'Brien (1979, 1981).

See the section "Homogeneity of Variance in One-Way Models" on page 1818 in Chapter 32, "The GLM Procedure," for more details on these methods. Example 32.10 on page 1892 in the same chapter illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances.

KRATIO=*value*

specifies the Type 1/Type 2 error seriousness ratio for the Waller-Duncan test. Reasonable values for KRATIO are 50, 100, and 500, which roughly correspond for the two-level case to ALPHA levels of 0.1, 0.05, and 0.01. By default, the procedure uses the default value of 100.

LINES

presents results of the BON, DUNCAN, GABRIEL, REGWQ, SCHEFFE, SIDAK, SMM, GT2, SNK, T, LSD, TUKEY, and WALLER options by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding means. The LINES option is appropriate for equal cell sizes, for which it is the default. The LINES option is also the default if the DUNCAN, REGWQ, SNK, or WALLER option is specified, or if there are only two cells of unequal size. If the cell sizes are unequal, the harmonic mean of the cell sizes is used, which may lead to somewhat liberal tests if the cell sizes are highly disparate. The LINES option cannot be used in combination with the DUNNETT, DUNNETTL, or DUNNETTU option. In addition, the procedure has a restriction that no more than 24 overlapping groups of means can exist. If a mean belongs to more than 24 groups, the procedure issues an error message. You can either reduce the number of levels of the variable or use a multiple comparison test that allows the CLDIFF option rather than the LINES option.

LSD

see the T option.

NOSORT

prevents the means from being sorted into descending order when the CLDIFF or CLM option is specified.

REGWQ

performs the Ryan-Einot-Gabriel-Welsch multiple range test on all main effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

SCHEFFE

performs Scheffé's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SIDAK

performs pairwise t tests on differences between means with levels adjusted according to Sidak's inequality for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SMM
GT2

performs pairwise comparisons based on the studentized maximum modulus and Sidak's uncorrelated-t inequality, yielding Hochberg's GT2 method when sample sizes are unequal, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SNK

performs the Student-Newman-Keuls multiple range test on all main effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

T
LSD

performs pairwise t tests, equivalent to Fisher's least-significant-difference test in the case of equal cell sizes, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

TUKEY

performs Tukey's studentized range test (HSD) on all main effect means in the MEANS statement. (When the group sizes are different, this is the Tukey-Kramer test.) See the CLDIFF and LINES options for discussions of how the procedure displays results.

WALLER

performs the Waller-Duncan k-ratio t test on all main effect means in the MEANS statement. See the KRATIO= option for information on controlling details of the test, and see the LINES option for a discussion of how the procedure displays results.

WELCH

requests Welch's (1951) variance-weighted one-way ANOVA. This alternative to the usual analysis of variance for a one-way model is robust to the assumption of equal within-group variances. This option is ignored unless your MODEL statement specifies a simple one-way model.

Note that using the WELCH option merely produces one additional table consisting of Welch's ANOVA. It does not affect all of the other tests displayed by the ANOVA procedure, which still require the assumption of equal variance for exact validity.

See the "Homogeneity of Variance in One-Way Models" section on page 1818 in Chapter 32, "The GLM Procedure," for more details on Welch's ANOVA. Example 32.10 on page 1892 in the same chapter illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances.

MODEL Statement

MODEL *dependents=effects* < */ options* > **;**

The MODEL statement names the dependent variables and independent effects. The syntax of effects is described in the section "Specification of Effects" on page 451. For any model effect involving classification variables (interactions as well as main effects), the number of levels can not exceed 32,767. If no independent effects are specified, only an intercept term is fit. This tests the hypothesis that the mean of the dependent variable is zero. All variables in effects that you specify in the MODEL statement must appear in the CLASS statement because PROC ANOVA does not allow for continuous effects.

You can specify the following options in the MODEL statement; they must be separated from the list of independent effects by a slash.

INTERCEPT
INT

displays the hypothesis tests associated with the intercept as an effect in the model. By default, the procedure includes the intercept in the model but does not display associated tests of hypotheses. Except for producing the uncorrected total SS instead of the corrected total SS, the INT option is ignored when you use an ABSORB statement.

NOUNI

suppresses the display of univariate statistics. You typically use the NOUNI option with a multivariate or repeated measures analysis of variance when you do not need the standard univariate output. The NOUNI option in a MODEL statement does not affect the univariate output produced by the REPEATED statement.

REPEATED Statement

REPEATED *factor-specification* < **/** *options* > **;**

When values of the dependent variables in the MODEL statement represent repeated measurements on the same experimental unit, the REPEATED statement enables you to test hypotheses about the measurement factors (often called *within-subject factors*), as well as the interactions of within-subject factors with independent variables in the MODEL statement (often called *between-subject factors*). The REPEATED statement provides multivariate and univariate tests as well as hypothesis tests for a variety of single-degree-of-freedom contrasts. There is no limit to the number of within-subject factors that can be specified. For more details, see the "Repeated Measures Analysis of Variance" section on page 1825 in Chapter 32, "The GLM Procedure."

The REPEATED statement is typically used for handling repeated measures designs with one repeated response variable. Usually, the variables on the left-hand side of the equation in the MODEL statement represent one repeated response variable. This does not mean that only one factor can be listed in the REPEATED statement. For example, one repeated response variable (hemoglobin count) might be measured 12 times (implying variables Y1 to Y12 on the left-hand side of the equal sign in the MODEL statement), with the associated within-subject factors treatment and time (implying two factors listed in the REPEATED statement). See the "Examples" section on page 449 for an example of how PROC ANOVA handles this case. Designs with two or more repeated response variables can, however, be handled with the IDENTITY transformation;see Example 32.9 on page 1886 in Chapter 32, "The GLM Procedure," for an example of analyzing a doubly-multivariate repeated measures design.

When a REPEATED statement appears, the ANOVA procedure enters a multivariate mode of handling missing values. If any values for variables corresponding to each combination of the within-subject factors are missing, the observation is excluded from the analysis.

The simplest form of the REPEATED statement requires only a *factor-name*. With two repeated factors, you must specify the *factor-name* and number of levels (*levels*) for each factor. Optionally, you can specify the actual values for the levels (*level-values*), a *transformation* that defines single-degree-of freedom contrasts, and *options* for additional analyses and output. When more than one within-subject factor is specified, *factor-names* (and associated level and transformation information) must be separated by a comma in the REPEATED statement. These terms are described in the following section, "Syntax Details."

Syntax Details

You can specify the following terms in the REPEATED statement.

factor-specification

The *factor-specification* for the REPEATED statement can include any number of individual factor specifications, separated by commas, of the following form:

factor-name levels < **(***level-values***)** > < *transformation* >

where

factor-name names a factor to be associated with the dependent variables. The name should not be the same as any variable name that already exists in the data set being analyzed and should conform to the usual conventions of SAS variable names.

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently.

levels specifies the number of levels associated with the factor being defined. When there is only one within-subject factor, the number of levels is equal to the number of dependent variables. In this case, *levels* is optional. When more than one within-subject factor is defined, however, *levels* is required, and the product of the number of levels of all the factors must equal the number of dependent variables in the MODEL statement.

(level-values) specifies values that correspond to levels of a repeated-measures factor. These values are used to label output; they are also used as spacings for constructing orthogonal polynomial contrasts if you specify a POLYNOMIAL transformation. The number of level values specified must correspond to the number of levels for that factor in the REPEATED statement. Enclose the *level-values* in parentheses.

The following *transformation* keywords define single-degree-of-freedom contrasts for factors specified in the REPEATED statement. Since the number of contrasts generated is always one less than the number of levels of the factor, you have some control over which contrast is omitted from the analysis by which transformation you select. The only exception is the IDENTITY transformation; this transformation is not composed of contrasts, and it has the same degrees of freedom as the factor has levels. By default, the procedure uses the CONTRAST transformation.

CONTRAST $<$ **(***ordinal-reference-level* **)** $>$ generates contrasts between levels of the factor and a reference level. By default, the procedure uses the last level; you can optionally specify a reference level in parentheses after the keyword CONTRAST. The reference level corresponds to the ordinal value of the level rather than the level value specified. For example, to generate contrasts between the first level of a factor and the other levels, use

```
contrast(1)
```

HELMERT generates contrasts between each level of the factor and the mean of subsequent levels.

IDENTITY generates an identity transformation corresponding to the associated factor. This transformation is *not* composed of contrasts; it has n degrees of freedom for an n-level factor, instead of $n - 1$. This can be used for doubly-multivariate repeated measures.

MEAN $<$ **(***ordinal-reference-level* **)** $>$ generates contrasts between levels of the factor and the mean of all other levels of the factor. Specifying a reference level eliminates the contrast between that level and the mean. Without a reference level, the contrast involving the last level is omitted. See the CONTRAST transformation for an example.

POLYNOMIAL generates orthogonal polynomial contrasts. Level values, if provided, are used as spacings in the construction of the polynomials; otherwise, equal spacing is assumed.

PROFILE generates contrasts between adjacent levels of the factor.

For examples of the transformation matrices generated by these contrast transformations, see the section "Repeated Measures Analysis of Variance" on page 1825 in Chapter 32, "The GLM Procedure."

You can specify the following options in the REPEATED statement after a slash:

CANONICAL

performs a canonical analysis of the $\mathbf{H}$ and $\mathbf{E}$ matrices corresponding to the transformed variables specified in the REPEATED statement.

MSTAT=FAPPROX
MSTAT=EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the "Multivariate

Tests" section in Chapter 2, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact *p*-values for three of the four tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root) and an improved F-approximation for the fourth (Pillai's Trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's Greatest Root, computations for the exact *p*-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information on the results of MSTAT=EXACT, see the "Multivariate Analysis of Variance" section on page 1823 in Chapter 32, "The GLM Procedure." .

NOM

displays only the results of the univariate analyses.

NOU

displays only the results of the multivariate analyses.

PRINTE

displays the $\mathbf{E}$ matrix for each combination of within-subject factors, as well as partial correlation matrices for both the original dependent variables and the variables defined by the transformations specified in the REPEATED statement. In addition, the PRINTE option provides sphericity tests for each set of transformed variables. If the requested transformations are not orthogonal, the PRINTE option also provides a sphericity test for a set of orthogonal contrasts.

PRINTH

displays the $\mathbf{H}$ (SSCP) matrix associated with each multivariate test.

PRINTM

displays the transformation matrices that define the contrasts in the analysis. PROC ANOVA always displays the $\mathbf{M}$ matrix so that the transformed variables are defined by the rows, not the columns, of the displayed $\mathbf{M}$ matrix. In other words, PROC ANOVA actually displays $\mathbf{M}'$.

PRINTRV

produces the characteristic roots and vectors for each multivariate test.

SUMMARY

produces analysis-of-variance tables for each contrast defined by the within-subjects factors. Along with tests for the effects of the independent variables specified in the MODEL statement, a term labeled MEAN tests the hypothesis that the overall mean of the contrast is zero.

Examples

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently. For example, assume that three treatments are administered at each of four times, for a total of twelve dependent variables on each experimental unit. If the variables are listed in the MODEL statement as Y1 through Y12, then the following REPEATED statement

```
   repeated trt 3, time 4;
```

implies the following structure:

	Dependent Variables											
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
Value of trt	1	1	1	1	2	2	2	2	3	3	3	3
Value of time	1	2	3	4	1	2	3	4	1	2	3	4

The REPEATED statement always produces a table like the preceding one. For more information on repeated measures analysis and on using the REPEATED statement, see the section "Repeated Measures Analysis of Variance" on page 1825 in Chapter 32, "The GLM Procedure."

TEST Statement

TEST < **H=** *effects* > **E=** *effect* **;**

Although an F value is computed for all SS in the analysis using the residual MS as an error term, you can request additional F tests using other effects as error terms. You need a TEST statement when a nonstandard error structure (as in a split plot) exists.

Caution: The ANOVA procedure does not check any of the assumptions underlying the F statistic. When you specify a TEST statement, you assume sole responsibility for the validity of the F statistic produced. To help validate a test, you may want to use the GLM procedure with the RANDOM statement and inspect the expected mean squares. In the GLM procedure, you can also use the TEST option in the RANDOM statement.

You can use as many TEST statements as you want, provided that they appear after the MODEL statement.

You can specify the following terms in the TEST statement.

H=*effects*
specifies which effects in the preceding model are to be used as hypothesis (numerator) effects.

E=*effect*
specifies one, and only one, effect to use as the error (denominator) term. The E= specification is required.

The following example uses two TEST statements and is appropriate for analyzing a split-plot design.

```
proc anova;
   class a b c;
   model y=a|b(a)|c;
   test h=a e=b(a);
   test h=c a*c e=b*c(a);
run;
```

Details

Specification of Effects

In SAS analysis-of-variance procedures, the variables that identify levels of the classifications are called *classification variables*, and they are declared in the CLASS statement. Classification variables are also called *categorical*, *qualitative*, *discrete*, or *nominal variables*. The values of a class variable are called *levels*. Class variables can be either numeric or character. This is in contrast to the *response* (or *dependent*) *variables*, which are continuous. Response variables must be numeric.

The analysis-of-variance model specifies *effects*, which are combinations of classification variables used to explain the variability of the dependent variables in the following manner:

- Main effects are specified by writing the variables by themselves in the CLASS statement: A B C. Main effects used as independent variables test the hypothesis that the mean of the dependent variable is the same for each level of the factor in question, ignoring the other independent variables in the model.
- Crossed effects (interactions) are specified by joining the class variables with asterisks in the MODEL statement: A*B A*C A*B*C. Interaction terms in a model test the hypothesis that the effect of a factor does not depend on the levels of the other factors in the interaction.
- Nested effects are specified by following a main effect or crossed effect with a class variable or list of class variables enclosed in parentheses in the MODEL statement. The main effect or crossed effect is nested within the effects listed in parentheses: B(A) C*D(A B). Nested effects test hypotheses similar to interactions, but the levels of the nested variables are not the same for every combination within which they are nested.

The general form of an effect can be illustrated using the class variables A, B, C, D, E, and F:

$$\mathsf{A} * \mathsf{B} * \mathsf{C}(\mathsf{D}\ \mathsf{E}\ \mathsf{F})$$

The crossed list should come first, followed by the nested list in parentheses. Note that no asterisks appear within the nested list or immediately before the left parenthesis.

Main Effects Models

For a three-factor main effects model with A, B, and C as the factors and Y as the dependent variable, the necessary statements are

```
proc anova;
   class A B C;
   model Y=A B C;
run;
```

Models with Crossed Factors

To specify interactions in a factorial model, join effects with asterisks as described previously. For example, these statements specify a complete factorial model, which includes all the interactions:

```
proc anova;
   class A B C;
   model Y=A B C A*B A*C B*C A*B*C;
run;
```

Bar Notation

You can shorten the specifications of a full factorial model by using bar notation. For example, the preceding statements can also be written

```
proc anova;
   class A B C;
   model Y=A|B|C;
run;
```

When the bar (|) is used, the expression on the right side of the equal sign is expanded from left to right using the equivalents of rules 2–4 given in Searle (1971, p. 390). The variables on the right- and left-hand sides of the bar become effects, and the cross of them becomes an effect. Multiple bars are permitted. For instance, A | B | C is evaluated as follows:

$$\begin{aligned} \mathsf{A} \mid \mathsf{B} \mid \mathsf{C} \quad &\rightarrow \quad \{\, \mathsf{A} \mid \mathsf{B} \,\} \mid \mathsf{C} \\ &\rightarrow \quad \{\, \mathsf{A} \;\; \mathsf{B} \;\; \mathsf{A{*}B} \,\} \mid \mathsf{C} \\ &\rightarrow \quad \mathsf{A} \;\; \mathsf{B} \;\; \mathsf{A{*}B} \;\; \mathsf{A{*}C} \;\; \mathsf{B{*}C} \;\; \mathsf{A{*}B{*}C} \end{aligned}$$

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect. For example, the specification A | B | C@2 results in only those effects that contain two or fewer variables; in this case, A B A*B C A*C and B*C.

The following table gives more examples of using the bar and at operators.

<table>
<tr><td>A | C(B)</td><td>is equivalent to</td><td>A C(B) A*C(B)</td></tr>
<tr><td>A(B) | C(B)</td><td>is equivalent to</td><td>A(B) C(B) A*C(B)</td></tr>
<tr><td>A(B) | B(D E)</td><td>is equivalent to</td><td>A(B) B(D E)</td></tr>
<tr><td>A | B(A) | C</td><td>is equivalent to</td><td>A B(A) C A*C B*C(A)</td></tr>
<tr><td>A | B(A) | C@2</td><td>is equivalent to</td><td>A B(A) C A*C</td></tr>
<tr><td>A | B | C | D@2</td><td>is equivalent to</td><td>A B A*B C A*C B*C D A*D B*D C*D</td></tr>
</table>

Consult the “Specification of Effects” section on page 1784 in Chapter 32, “The GLM Procedure,” for further details on bar notation.

Nested Models

Write the effect that is nested within another effect first, followed by the other effect in parentheses. For example, if A and B are main effects and C is nested within A and B (that is, the levels of C that are observed are not the same for each combination of A and B), the statements for PROC ANOVA are

```
proc anova;
   class A B C;
   model y=A B C(A B);
run;
```

The identity of a level is viewed within the context of the level of the containing effects. For example, if City is nested within State, then the identity of City is viewed within the context of State.

The distinguishing feature of a nested specification is that nested effects never appear as main effects. Another way of viewing nested effects is that they are effects that pool the main effect with the interaction of the nesting variable. See the “Automatic Pooling” section, which follows.

Models Involving Nested, Crossed, and Main Effects

Asterisks and parentheses can be combined in the MODEL statement for models involving nested and crossed effects:

```
proc anova;
   class A B C;
   model Y=A B(A) C(A) B*C(A);
run;
```

Automatic Pooling

In line with the general philosophy of the GLM procedure, there is no difference between the statements

```
model Y=A B(A);
```

and

```
model Y=A A*B;
```

The effect B becomes a nested effect by virtue of the fact that it does not occur as a main effect. If B is not written as a main effect in addition to participating in A*B, then the sum of squares that is associated with B is pooled into A*B.

This feature allows the automatic pooling of sums of squares. If an effect is omitted from the model, it is automatically pooled with all the higher-level effects containing the class variables in the omitted effect (or within-error). This feature is most useful in split-plot designs.

Using PROC ANOVA Interactively

PROC ANOVA can be used interactively. After you specify a model in a MODEL statement and run PROC ANOVA with a RUN statement, a variety of statements (such as MEANS, MANOVA, TEST, and REPEATED) can be executed without PROC ANOVA recalculating the model sum of squares.

The "Syntax" section (page 432) describes which statements can be used interactively. You can execute these interactive statements individually or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement cannot be repeated; the ANOVA procedure allows only one MODEL statement.

If you use PROC ANOVA interactively, you can end the procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you use PROC ANOVA interactively, additional RUN statements do not end the procedure but tell PROC ANOVA to execute additional statements.

When a WHERE statement is used with PROC ANOVA, it should appear before the first RUN statement. The WHERE statement enables you to select only certain observations for analysis without using a subsetting DATA step. For example, the statement `where group ne 5` omits observations with GROUP=5 from the analysis. Refer to *SAS Language Reference: Dictionary* for details on this statement.

When a BY statement is used with PROC ANOVA, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

Interactivity is also disabled when there are different patterns of missing values among the dependent variables. For details, see the section "Missing Values," which follows.

Missing Values

For an analysis involving one dependent variable, PROC ANOVA uses an observation if values are nonmissing for that dependent variable and for all the variables used in independent effects.

For an analysis involving multiple dependent variables without the MANOVA or REPEATED statement, or without the MANOVA option in the PROC ANOVA statement, a missing value in one dependent variable does not eliminate the observation from the analysis of other nonmissing dependent variables. For an analysis with the MANOVA or REPEATED statement, or with the MANOVA option in the PROC ANOVA statement, the ANOVA procedure requires values for all dependent variables to be nonmissing for an observation before the observation can be used in the analysis.

During processing, PROC ANOVA groups the dependent variables by their pattern of missing values across observations so that sums and cross products can be collected in the most efficient manner.

If your data have different patterns of missing values among the dependent variables, interactivity is disabled. This could occur when some of the variables in your data set have missing values and

- you do not use the MANOVA option in the PROC ANOVA statement
- you do not use a MANOVA or REPEATED statement before the first RUN statement

Output Data Set

The OUTSTAT= option in the PROC ANOVA statement produces an output data set that contains the following:

- the BY variables, if any
- _TYPE_, a new character variable. This variable has the value 'ANOVA' for observations corresponding to sums of squares; it has the value 'CANCORR', 'STRUCTUR', or 'SCORE' if a canonical analysis is performed through the MANOVA statement and no M= matrix is specified.
- _SOURCE_, a new character variable. For each observation in the data set, _SOURCE_ contains the name of the model effect from which the corresponding statistics are generated.

- _NAME_, a new character variable. The variable _NAME_ contains the name of one of the dependent variables in the model or, in the case of canonical statistics, the name of one of the canonical variables (CAN1, CAN2, and so on).
- four new numeric variables, SS, DF, F, and PROB, containing sums of squares, degrees of freedom, F values, and probabilities, respectively, for each model or contrast sum of squares generated in the analysis. For observations resulting from canonical analyses, these variables have missing values.
- if there is more than one dependent variable, then variables with the same names as the dependent variables represent
  - for _TYPE_=‘ANOVA’, the crossproducts of the hypothesis matrices
  - for _TYPE_=‘CANCORR’, canonical correlations for each variable
  - for _TYPE_=‘STRUCTUR’, coefficients of the total structure matrix
  - for _TYPE_=‘SCORE’, raw canonical score coefficients

The output data set can be used to perform special hypothesis tests (for example, with the IML procedure in SAS/IML software), to reformat output, to produce canonical variates (through the SCORE procedure), or to rotate structure matrices (through the FACTOR procedure).

Computational Method

Let $\mathbf{X}$ represent the $n \times p$ design matrix. The columns of $\mathbf{X}$ contain only 0s and 1s. Let $\mathbf{Y}$ represent the $n \times 1$ vector of dependent variables.

In the GLM procedure, $\mathbf{X'X}$, $\mathbf{X'Y}$, and $\mathbf{Y'Y}$ are formed in main storage. However, in the ANOVA procedure, only the diagonals of $\mathbf{X'X}$ are computed, along with $\mathbf{X'Y}$ and $\mathbf{Y'Y}$. Thus, PROC ANOVA saves a considerable amount of storage as well as time. The memory requirements for PROC ANOVA are asymptotically linear functions of n^2 and nr, where n is the number of dependent variables and r the number of independent parameters.

The elements of $\mathbf{X'Y}$ are cell totals, and the diagonal elements of $\mathbf{X'X}$ are cell frequencies. Since PROC ANOVA automatically pools omitted effects into the next higher-level effect containing the names of the omitted effect (or within-error), a slight modification to the rules given by Searle (1971, p. 389) is used.

1. PROC ANOVA computes the sum of squares for each effect as if it is a main effect. In other words, for each effect, PROC ANOVA squares each cell total and divides by its cell frequency. The procedure then adds these quantities together and subtracts the correction factor for the mean (total squared over N).
2. For each effect involving two class names, PROC ANOVA subtracts the SS for any main effect with a name that is contained in the two-factor effect.

3. For each effect involving three class names, PROC ANOVA subtracts the SS for all main effects and two-factor effects with names that are contained in the three-factor effect. If effects involving four or more class names are present, the procedure continues this process.

Displayed Output

PROC ANOVA first displays a table that includes the following:

- the name of each variable in the CLASS statement
- the number of different values or Levels of the Class variables
- the Values of the Class variables
- the Number of observations in the data set and the number of observations excluded from the analysis because of missing values, if any

PROC ANOVA then displays an analysis-of-variance table for each dependent variable in the MODEL statement. This table breaks down

- the Total Sum of Squares for the dependent variable into the portion attributed to the Model and the portion attributed to Error
- the Mean Square term, which is the Sum of Squares divided by the degrees of freedom (DF)

The analysis-of-variance table also lists the following:

- the Mean Square for Error (MSE), which is an estimate of σ^2, the variance of the true errors
- the F Value, which is the ratio produced by dividing the Mean Square for the Model by the Mean Square for Error. It tests how well the model as a whole (adjusted for the mean) accounts for the dependent variable's behavior. This F test is a test of the null hypothesis that all parameters except the intercept are zero.
- the significance probability associated with the F statistic, labeled “Pr > F”
- R-Square, R^2, which measures how much variation in the dependent variable can be accounted for by the model. The R^2 statistic, which can range from 0 to 1, is the ratio of the sum of squares for the model divided by the sum of squares for the corrected total. In general, the larger the R^2 value, the better the model fits the data.
- C.V., the coefficient of variation, which is often used to describe the amount of variation in the population. The C.V. is 100 times the standard deviation of the dependent variable divided by the Mean. The coefficient of variation is often a preferred measure because it is unitless.

- Root MSE, which estimates the standard deviation of the dependent variable. Root MSE is computed as the square root of Mean Square for Error, the mean square of the error term.
- the Mean of the dependent variable

For each effect (or source of variation) in the model, PROC ANOVA then displays the following:

- DF, degrees of freedom
- Anova SS, the sum of squares, and the associated Mean Square
- the F Value for testing the hypothesis that the group means for that effect are equal
- Pr > F, the significance probability value associated with the F Value

When you specify a TEST statement, PROC ANOVA displays the results of the requested tests. When you specify a MANOVA statement and the model includes more than one dependent variable, PROC ANOVA produces these additional statistics:

- the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$ for each $\mathbf{H}$ matrix
- the Hotelling-Lawley trace
- Pillai’s trace
- Wilks’ criterion
- Roy’s maximum root criterion

See Example 32.6 on page 1868 in Chapter 32, “The GLM Procedure,” for an example of the MANOVA results. These MANOVA tests are discussed in Chapter 2, “Introduction to Regression Procedures.”

ODS Table Names

PROC ANOVA assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 17.3. ODS Tables Produced in PROC ANOVA

ODS Table Name	Description	Statement / Option
AltErrTests	Anova tests with error other than MSE	TEST /E=
Bartlett	Bartlett’s homogeneity of variance test	MEANS / HOVTEST=BARTLETT
CLDiffs	Multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)

Table 17.3. (continued)

ODS Table Name	Description	Statement / Option
CLDiffsInfo	Information for multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLMeans	Multiple comparisons of means with confidence/comparison interval	MEANS / CLM with (BON or GABRIEL or SCHEFFE or SIDAK or SMM or T or LSD)
CLMeansInfo	Information for multiple comparisons of means with confidence/comparison interval	MEANS / CLM
CanAnalysis	Canonical analysis	(MANOVA or REPEATED) / CANONICAL
CanCoef	Canonical coefficients	(MANOVA or REPEATED) / CANONICAL
CanStructure	Canonical structure	(MANOVA or REPEATED) / CANONICAL
CharStruct	Characteristic roots and vectors	(MANOVA / not CANONICAL) or (REPEATED / PRINTRV)
ClassLevels	Classification variable levels	CLASS statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables with different patterns of missing values
Epsilons	Greenhouse-Geisser and Huynh-Feldt epsilons	REPEATED statement
ErrorSSCP	Error SSCP matrix	(MANOVA or REPEATED) / PRINTE
FitStatistics	R-Square, C.V., Root MSE, and dependent mean	default
HOVFTest	Homogeneity of variance ANOVA	MEANS / HOVTEST
HypothesisSSCP	Hypothesis SSCP matrix	(MANOVA or REPEATED) / PRINTE
MANOVATransform	Multivariate transformation matrix	MANOVA / M=
MCLines	Multiple comparisons LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not(CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesInfo	Information for multiple comparison LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesRange	Ranges for multiple range MC tests	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
Means	Group means	MEANS statement
ModelANOVA	ANOVA for model terms	default
MultStat	Multivariate tests	MANOVA statement

Table 17.3. (continued)

ODS Table Name	Description	Statement / Option
NObs	Number of observations	default
OverallANOVA	Over-all ANOVA	default
PartialCorr	Partial correlation matrix	(MANOVA or REPEATED) / PRINTE
RepTransform	Repeated transformation matrix	REPEATED (CONTRAST or HELMERT or MEAN or POLYNOMIAL or PROFILE)
RepeatedLevelInfo	Correspondence between dependents and repeated measures levels	REPEATED statement
Sphericity	Sphericity tests	REPEATED / PRINTE
Tests	Summary ANOVA for specified MANOVA H= effects	MANOVA / H= SUMMARY
Welch	Welch's ANOVA	MEANS / WELCH

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the ANOVA procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request these graphs you must specify the ODS GRAPHICS statement with an appropriate model, as discussed in the following. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

When the ODS GRAPHICS are in effect, then if you specify a one-way analysis of variance model, with just one independent classification variable, the ANOVA procedure will produce a grouped box plot of the response values versus the classification levels. For an example of the box plot, see the “One-Way Layout with Means Comparisons” section on page 424.

ODS Graph Names

PROC ANOVA assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 17.4.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 17.4. ODS Graphics Produced by PROC ANOVA

ODS Graph Name	Plot Description
BoxPlot	Box plot

Examples

Example 17.1. Randomized Complete Block With Factorial Treatment Structure

This example uses statements for the analysis of a randomized block with two treatment factors occuring in a factorial structure. The data, from Neter, Wasserman, and Kutner (1990, p. 941), are from an experiment examining the effects of codeine and acupuncture on post-operative dental pain in male subjects. Both treatment factors have two levels. The codeine levels are a codeine capsule or a sugar capsule. The acupuncture levels are two inactive acupuncture points or two active acupuncture points. There are four distinct treatment combinations due to the factorial treatment structure. The 32 subjects are assigned to eight blocks of four subjects each based on an assessment of pain tolerance.

The data for the analysis are balanced, so PROC ANOVA is used. The data are as follows:

```
title1 'Randomized Complete Block With Two Factors';
   data PainRelief;
      input PainLevel Codeine Acupuncture Relief @@;
      datalines;
   1 1 1 0.0  1 2 1 0.5  1 1 2 0.6  1 2 2 1.2
   2 1 1 0.3  2 2 1 0.6  2 1 2 0.7  2 2 2 1.3
   3 1 1 0.4  3 2 1 0.8  3 1 2 0.8  3 2 2 1.6
   4 1 1 0.4  4 2 1 0.7  4 1 2 0.9  4 2 2 1.5
   5 1 1 0.6  5 2 1 1.0  5 1 2 1.5  5 2 2 1.9
   6 1 1 0.9  6 2 1 1.4  6 1 2 1.6  6 2 2 2.3
   7 1 1 1.0  7 2 1 1.8  7 1 2 1.7  7 2 2 2.1
   8 1 1 1.2  8 2 1 1.7  8 1 2 1.6  8 2 2 2.4
   ;
```

The variable PainLevel is the blocking variable, and Codeine and Acupuncture represent the levels of the two treatment factors. The variable Relief is the pain relief score (the higher the score, the more relief the patient has).

The following code invokes PROC ANOVA. The blocking variable and treatment factors appear in the CLASS statement. The bar between the treatment factors Codeine and Acupuncture adds their main effects as well as their interaction Codeine*Acupuncture to the model.

```
proc anova data=PainRelief;
   class PainLevel Codeine Acupuncture;
   model Relief = PainLevel Codeine|Acupuncture;
```

The results from the analysis are shown in Output 17.1.1, Output 17.1.2, and Output 17.1.3.

Output 17.1.1. Class Level Information

```
                    Randomized Complete Block With Two Factors

                               The ANOVA Procedure

                             Class Level Information

                   Class             Levels    Values

                   PainLevel              8    1 2 3 4 5 6 7 8

                   Codeine                2    1 2

                   Acupuncture            2    1 2


                     Number of Observations Read          32
                     Number of Observations Used          32
```

Output 17.1.2. ANOVA Table

```
Dependent Variable: Relief

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                       10    11.33500000     1.13350000     78.37   <.0001

 Error                       21     0.30375000     0.01446429

 Corrected Total             31    11.63875000


               R-Square     Coeff Var      Root MSE    Relief Mean

               0.973902      10.40152      0.120268       1.156250
```

The Class Level Information and ANOVA table are shown in Output 17.1.1 and Output 17.1.2. The class level information summarizes the structure of the design. It is good to check these consistently in search of errors in the data step. The overall F test is significant, indicating that the model accounts for a significant amount of variation in the dependent variable.

Output 17.1.3. Tests of Effects

```
Dependent Variable: Relief

 Source                      DF       Anova SS    Mean Square   F Value   Pr > F

 PainLevel                    7     5.59875000     0.79982143     55.30   <.0001
 Codeine                      1     2.31125000     2.31125000    159.79   <.0001
 Acupuncture                  1     3.38000000     3.38000000    233.68   <.0001
 Codeine*Acupuncture          1     0.04500000     0.04500000      3.11   0.0923
```

Output 17.1.3 shows tests of the effects. The blocking effect is significant; hence, it is useful. The interaction between codeine and acupuncture is significant at the 90% level but not at the 95% level. The significance level of this test should be determined before the analysis. The main effects of both treatment factors are highly significant.

Example 17.2. Alternative Multiple Comparison Procedures

The following is a continuation of the first example in the the "One-Way Layout with Means Comparisons" section on page 424. You are studying the effect of bacteria on the nitrogen content of red clover plants, and the analysis of variance shows a highly significant effect. The following statements create the data set and compute the analysis of variance as well as Tukey's multiple comparisons test for pairwise differences between bacteria strains; the results are shown in Figure 17.1, Figure 17.2, and Figure 17.3

```
title1 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;
```

```
proc anova data=Clover;
   class Strain;
   model Nitrogen = Strain;
   means Strain / tukey;
run;
```

The interactivity of PROC ANOVA enables you to submit further MEANS statements without re-running the entire analysis. For example, the following command requests means of the Strain levels with Duncan's multiple range test and the Waller-Duncan k-ratio t test.

```
means Strain / duncan waller;
run;
```

Results of the Waller-Duncan k-ratio t test are shown in Output 17.2.1.

Output 17.2.1. Waller-Duncan K-ratio t Test

```
                     Nitrogen Content of Red Clover Plants

                              The ANOVA Procedure

                   Waller-Duncan K-ratio t Test for Nitrogen

NOTE: This test minimizes the Bayes risk under additive loss and certain other
                                  assumptions.


                    Kratio                              100
                    Error Degrees of Freedom             24
                    Error Mean Square              11.78867
                    F Value                           14.37
                    Critical Value of t             1.91873
                    Minimum Significant Difference   4.1665


          Means with the same letter are not significantly different.


            Waller Grouping          Mean      N    Strain

                          A        28.820      5    3DOK1

                          B        23.980      5    3DOK5
                          B
                     C    B        19.920      5    3DOK7
                     C
                     C    D        18.700      5    COMPOS
                          D
                     E    D        14.640      5    3DOK4
                     E
                     E             13.260      5    3DOK13



                     Nitrogen Content of Red Clover Plants

                              The ANOVA Procedure

                   Duncan's Multiple Range Test for Nitrogen

    NOTE: This test controls the Type I comparisonwise error rate, not the
                           experimentwise error rate.
```

The Waller-Duncan k-ratio t test is a multiple range test. Unlike Tukey's test, this test does not operate on the principle of controlling Type I error. Instead, it compares the Type I and Type II error rates based on Bayesian principles (Steel and Torrie 1980).

The Waller Grouping column in Output 17.2.1 shows which means are significantly different. From this test, you can conclude the following:

- The mean nitrogen content for strain 3DOK1 is higher than the means for all other strains.
- The mean nitrogen content for strain 3DOK5 is higher than the means for COMPOS, 3DOK4, and 3DOK13.
- The mean nitrogen content for strain 3DOK7 is higher than the means for 3DOK4 and 3DOK13.
- The mean nitrogen content for strain COMPOS is higher than the mean for 3DOK13.
- Differences between all other means are not significant based on this sample size.

Output 17.2.2 shows the results of Duncan's multiple range test. Duncan's test is a result-guided test that compares the treatment means while controlling the comparison-wise error rate. You should use this test for planned comparisons only (Steel and Torrie 1980). The results and conclusions for this example are the same as for the Waller-Duncan k-ratio t test. This is not always the case.

Output 17.2.2. Duncan’s Multiple Range Test

```
                   Waller-Duncan K-ratio t Test for Nitrogen

NOTE: This test minimizes the Bayes risk under additive loss and certain other
                                  assumptions.



                   Duncan’s Multiple Range Test for Nitrogen

    NOTE: This test controls the Type I comparisonwise error rate, not the
                           experimentwise error rate.


                       Alpha                        0.05
                       Error Degrees of Freedom       24
                       Error Mean Square        11.78867


     Number of Means          2          3          4          5          6
     Critical Range       4.482      4.707      4.852      4.954      5.031


          Means with the same letter are not significantly different.


            Duncan Grouping          Mean      N    Strain

                          A        28.820      5    3DOK1

                          B        23.980      5    3DOK5
                          B
                     C    B        19.920      5    3DOK7
                     C
                     C    D        18.700      5    COMPOS
                          D
                     E    D        14.640      5    3DOK4
                     E
                     E             13.260      5    3DOK13
```

Tukey and Least Significant Difference (LSD) tests are requested with the following MEANS statement. The CLDIFF option requests confidence intervals for both tests.

```
means strain/ lsd tukey cldiff ;
run;
```

The LSD tests for this example are shown in Output 17.2.3, and they give the same results as the previous two multiple comparison tests. Again, this is not always the case.

Output 17.2.3. T Tests (LSD)

```
                         The ANOVA Procedure

                      t Tests (LSD) for Nitrogen

 NOTE: This test controls the Type I comparisonwise error rate, not the
                      experimentwise error rate.


                 Alpha                              0.05
                 Error Degrees of Freedom             24
                 Error Mean Square              11.78867
                 Critical Value of t             2.06390
                 Least Significant Difference     4.4818


     Comparisons significant at the 0.05 level are indicated by ***.


                               Difference
             Strain             Between       95% Confidence
           Comparison            Means            Limits

         3DOK1  - 3DOK5          4.840       0.358    9.322  ***
         3DOK1  - 3DOK7          8.900       4.418   13.382  ***
         3DOK1  - COMPOS        10.120       5.638   14.602  ***
         3DOK1  - 3DOK4         14.180       9.698   18.662  ***
         3DOK1  - 3DOK13        15.560      11.078   20.042  ***
         3DOK5  - 3DOK1         -4.840      -9.322   -0.358  ***
         3DOK5  - 3DOK7          4.060      -0.422    8.542
         3DOK5  - COMPOS         5.280       0.798    9.762  ***
         3DOK5  - 3DOK4          9.340       4.858   13.822  ***
         3DOK5  - 3DOK13        10.720       6.238   15.202  ***
         3DOK7  - 3DOK1         -8.900     -13.382   -4.418  ***
         3DOK7  - 3DOK5         -4.060      -8.542    0.422
         3DOK7  - COMPOS         1.220      -3.262    5.702
         3DOK7  - 3DOK4          5.280       0.798    9.762  ***
         3DOK7  - 3DOK13         6.660       2.178   11.142  ***
         COMPOS - 3DOK1        -10.120     -14.602   -5.638  ***
         COMPOS - 3DOK5         -5.280      -9.762   -0.798  ***
         COMPOS - 3DOK7         -1.220      -5.702    3.262
         COMPOS - 3DOK4          4.060      -0.422    8.542
         COMPOS - 3DOK13         5.440       0.958    9.922  ***
         3DOK4  - 3DOK1        -14.180     -18.662   -9.698  ***
         3DOK4  - 3DOK5         -9.340     -13.822   -4.858  ***
         3DOK4  - 3DOK7         -5.280      -9.762   -0.798  ***
         3DOK4  - COMPOS        -4.060      -8.542    0.422
         3DOK4  - 3DOK13         1.380      -3.102    5.862
         3DOK13 - 3DOK1        -15.560     -20.042  -11.078  ***
         3DOK13 - 3DOK5        -10.720     -15.202   -6.238  ***
         3DOK13 - 3DOK7         -6.660     -11.142   -2.178  ***
         3DOK13 - COMPOS        -5.440      -9.922   -0.958  ***
         3DOK13 - 3DOK4         -1.380      -5.862    3.102



                         The ANOVA Procedure

              Tukey’s Studentized Range (HSD) Test for Nitrogen

      NOTE: This test controls the Type I experimentwise error rate.
```

If you only perform the LSD tests when the overall model F-test is significant, then this is called Fisher's protected LSD test. Note that the LSD tests should be used for planned comparisons.

The TUKEY tests shown in Output 17.2.4 find fewer significant differences than the other three tests. This is not unexpected, as the TUKEY test controls the Type I experimentwise error rate. For a complete discussion of multiple comparison methods, see the "Multiple Comparisons" section on page 1806 in Chapter 32, "The GLM Procedure."

Output 17.2.4. Tukey's Studentized Range Test

```
                          t Tests (LSD) for Nitrogen

     NOTE: This test controls the Type I comparisonwise error rate, not the
                          experimentwise error rate.



                Tukey's Studentized Range (HSD) Test for Nitrogen

          NOTE: This test controls the Type I experimentwise error rate.


                   Alpha                                   0.05
                   Error Degrees of Freedom                  24
                   Error Mean Square                   11.78867
                   Critical Value of Studentized Range  4.37265
                   Minimum Significant Difference        6.7142


         Comparisons significant at the 0.05 level are indicated by ***.


                                Difference
                  Strain          Between     Simultaneous 95%
                Comparison          Means    Confidence Limits

              3DOK1  - 3DOK5        4.840      -1.874   11.554
              3DOK1  - 3DOK7        8.900       2.186   15.614  ***
              3DOK1  - COMPOS      10.120       3.406   16.834  ***
              3DOK1  - 3DOK4       14.180       7.466   20.894  ***
              3DOK1  - 3DOK13      15.560       8.846   22.274  ***
              3DOK5  - 3DOK1       -4.840     -11.554    1.874
              3DOK5  - 3DOK7        4.060      -2.654   10.774
              3DOK5  - COMPOS       5.280      -1.434   11.994
              3DOK5  - 3DOK4        9.340       2.626   16.054  ***
              3DOK5  - 3DOK13      10.720       4.006   17.434  ***
              3DOK7  - 3DOK1       -8.900     -15.614   -2.186  ***
              3DOK7  - 3DOK5       -4.060     -10.774    2.654
              3DOK7  - COMPOS       1.220      -5.494    7.934
              3DOK7  - 3DOK4        5.280      -1.434   11.994
              3DOK7  - 3DOK13       6.660      -0.054   13.374
              COMPOS - 3DOK1      -10.120     -16.834   -3.406  ***
              COMPOS - 3DOK5       -5.280     -11.994    1.434
              COMPOS - 3DOK7       -1.220      -7.934    5.494
              COMPOS - 3DOK4        4.060      -2.654   10.774
              COMPOS - 3DOK13       5.440      -1.274   12.154
              3DOK4  - 3DOK1      -14.180     -20.894   -7.466  ***
              3DOK4  - 3DOK5       -9.340     -16.054   -2.626  ***
              3DOK4  - 3DOK7       -5.280     -11.994    1.434
              3DOK4  - COMPOS      -4.060     -10.774    2.654
              3DOK4  - 3DOK13       1.380      -5.334    8.094
              3DOK13 - 3DOK1      -15.560     -22.274   -8.846  ***
              3DOK13 - 3DOK5      -10.720     -17.434   -4.006  ***
              3DOK13 - 3DOK7       -6.660     -13.374    0.054
              3DOK13 - COMPOS      -5.440     -12.154    1.274
              3DOK13 - 3DOK4       -1.380      -8.094    5.334
```

Example 17.3. Split Plot

In some experiments, treatments can be applied only to groups of experimental observations rather than separately to each observation. When there are two nested groupings of the observations on the basis of treatment application, this is known as a *split plot design*. For example, in integrated circuit fabrication it is of interest to see how different manufacturing methods affect the characteristics of individual chips. However, much of the manufacturing process is applied to a relatively large wafer of material, from which many chips are made. Additionally, a chip's position within a wafer may also affect chip performance. These two groupings of chips—by wafer and by position-within-wafer—might form the *whole plots* and the *subplots*, respectively, of a split plot design for integrated circuits.

The following statements produce an analysis for a split-plot design. The CLASS statement includes the variables Block, A, and B, where B defines subplots within BLOCK*A whole plots. The MODEL statement includes the independent effects Block, A, Block*A, B, and A*B. The TEST statement asks for an F test of the A effect, using the Block*A effect as the error term. The following statements produce Output 17.3.1 and Output 17.3.2:

```
title1 'Split Plot Design';
data Split;
   input Block 1 A 2 B 3 Response;
   datalines;
142 40.0
141 39.5
112 37.9
111 35.4
121 36.7
122 38.2
132 36.4
131 34.8
221 42.7
222 41.6
212 40.3
211 41.6
241 44.5
242 47.6
231 43.6
232 42.8
;



proc anova data=Split;
   class Block A B;
   model Response = Block A Block*A B A*B;
   test h=A e=Block*A;
run;
```

Output 17.3.1. Class Level Information and ANOVA Table

```
                              Split Plot Design

                             The ANOVA Procedure

                           Class Level Information

                       Class         Levels    Values

                       Block              2    1 2

                       A                  4    1 2 3 4

                       B                  2    1 2


                   Number of Observations Read          16
                   Number of Observations Used          16


                              Split Plot Design

                             The ANOVA Procedure

Dependent Variable: Response

                                         Sum of
 Source                      DF         Squares    Mean Square   F Value   Pr > F

 Model                       11     182.0200000     16.5472727      7.85   0.0306

 Error                        4       8.4300000      2.1075000

 Corrected Total             15     190.4500000


              R-Square     Coeff Var      Root MSE    Response Mean

              0.955736      3.609007      1.451723         40.22500
```

First, notice that the overall F test for the model is significant.

Output 17.3.2. Tests of Effects

```
Dependent Variable: Response

 Source                      DF        Anova SS     Mean Square    F Value    Pr > F

 Block                        1     131.1025000     131.1025000      62.21    0.0014
 A                            3      40.1900000      13.3966667       6.36    0.0530
 Block*A                      3       6.9275000       2.3091667       1.10    0.4476
 B                            1       2.2500000       2.2500000       1.07    0.3599
 A*B                          3       1.5500000       0.5166667       0.25    0.8612


      Tests of Hypotheses Using the Anova MS for Block*A as an Error Term

 Source                      DF        Anova SS     Mean Square    F Value    Pr > F

 A                            3     40.19000000     13.39666667       5.80    0.0914
```

The effect of Block is significant. The effect of A is not significant: look at the F test produced by the TEST statement, not at the F test produced by default. Neither the B nor A*B effects are significant. The test for Block*A is irrelevant, as this is simply the main-plot error.

Example 17.4. Latin Square Split Plot

The data for this example is taken from Smith (1951). A Latin square design is used to evaluate six different sugar beet varieties arranged in a six-row (Rep) by six-column (Column) square. The data are collected over two harvests. The variable Harvest then becomes a split plot on the original Latin square design for whole plots. The following statements produce Output 17.4.1, Output 17.4.2, and Output 17.4.3:

```
title1 'Sugar Beet Varieties';
title3 'Latin Square Split-Plot Design';
data Beets;
   do Harvest=1 to 2;
      do Rep=1 to 6;
         do Column=1 to 6;
            input Variety Y @;
            output;
            end;
         end;
      end;
   datalines;
3 19.1 6 18.3 5 19.6 1 18.6 2 18.2 4 18.5
6 18.1 2 19.5 4 17.6 3 18.7 1 18.7 5 19.9
1 18.1 5 20.2 6 18.5 4 20.1 3 18.6 2 19.2
2 19.1 3 18.8 1 18.7 5 20.2 4 18.6 6 18.5
4 17.5 1 18.1 2 18.7 6 18.2 5 20.4 3 18.5
5 17.7 4 17.8 3 17.4 2 17.0 6 17.6 1 17.6
3 16.2 6 17.0 5 18.1 1 16.6 2 17.7 4 16.3
6 16.0 2 15.3 4 16.0 3 17.1 1 16.5 5 17.6
1 16.5 5 18.1 6 16.7 4 16.2 3 16.7 2 17.3
```

```
2 17.5 3 16.0 1 16.4 5 18.0 4 16.6 6 16.1
4 15.7 1 16.1 2 16.7 6 16.3 5 17.8 3 16.2
5 18.3 4 16.6 3 16.4 2 17.6 6 17.1 1 16.5
;



proc anova data=Beets;
   class Column Rep Variety Harvest;
   model Y=Rep Column Variety Rep*Column*Variety
           Harvest Harvest*Rep
           Harvest*Variety;
   test h=Rep Column Variety e=Rep*Column*Variety;
   test h=Harvest            e=Harvest*Rep;
run;
```

Output 17.4.1. Class Level Information

```
                              Sugar Beet Varieties

                          Latin Square Split-Plot Design

                              The ANOVA Procedure

                            Class Level Information

                      Class          Levels    Values

                      Column              6    1 2 3 4 5 6

                      Rep                 6    1 2 3 4 5 6

                      Variety             6    1 2 3 4 5 6

                      Harvest             2    1 2


                    Number of Observations Read          72
                    Number of Observations Used          72
```

Output 17.4.2. ANOVA Table

```
Dependent Variable: Y

                                            Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                       46       98.9147222      2.1503200      7.22   <.0001

 Error                       25        7.4484722      0.2979389

 Corrected Total             71      106.3631944


               R-Square     Coeff Var      Root MSE        Y Mean

               0.929971      3.085524      0.545838      17.69028


 Source                      DF         Anova SS    Mean Square   F Value   Pr > F

 Rep                          5       4.32069444     0.86413889      2.90   0.0337
 Column                       5       1.57402778     0.31480556      1.06   0.4075
 Variety                      5      20.61902778     4.12380556     13.84   <.0001
 Column*Rep*Variety          20       3.25444444     0.16272222      0.55   0.9144
 Harvest                      1      60.68347222    60.68347222    203.68   <.0001
 Rep*Harvest                  5       7.71736111     1.54347222      5.18   0.0021
 Variety*Harvest              5       0.74569444     0.14913889      0.50   0.7729
```

First, note from Output 17.4.2 that the overall model is significant.

Output 17.4.3. Tests of Effects

```
Dependent Variable: Y

 Tests of Hypotheses Using the Anova MS for Column*Rep*Variety as an Error Term

 Source                      DF         Anova SS    Mean Square   F Value   Pr > F

 Rep                          5       4.32069444     0.86413889      5.31   0.0029
 Column                       5       1.57402778     0.31480556      1.93   0.1333
 Variety                      5      20.61902778     4.12380556     25.34   <.0001


    Tests of Hypotheses Using the Anova MS for Rep*Harvest as an Error Term

 Source                      DF         Anova SS    Mean Square   F Value   Pr > F

 Harvest                      1      60.68347222    60.68347222     39.32   0.0015
```

Output 17.4.3 shows that the effects for Rep and Harvest are significant, while the Column effect is not. The average Ys for the six different Varietys are significantly different. For these four tests, look at the output produced by the two TEST statements, not at the usual ANOVA procedure output. The Variety*Harvest interaction is not significant. All other effects in the default output should either be tested using the results from the TEST statements or are irrelevant as they are only error terms for portions of the model.

Example 17.5. Strip-Split Plot

In this example, four different fertilizer treatments are laid out in vertical strips, which are then split into subplots with different levels of calcium. Soil type is stripped across the split-plot experiment, and the entire experiment is then replicated three times. The dependent variable is the yield of winter barley. The data come from the notes of G. Cox and A. Rotti.

The input data are the 96 values of Y, arranged so that the calcium value (Calcium) changes most rapidly, then the fertilizer value (Fertilizer), then the Soil value, and, finally, the Rep value. Values are shown for Calcium (0 and 1); Fertilizer (0, 1, 2, 3); Soil (1, 2, 3); and Rep (1, 2, 3, 4). The following example produces Output 17.5.1, Output 17.5.2, Output 17.5.3, and Output 17.5.4.

```
title1 'Strip-split Plot';
data Barley;
   do Rep=1 to 4;
      do Soil=1 to 3;                /* 1=d 2=h 3=p */
         do Fertilizer=0 to 3;
            do Calcium=0,1;
               input Yield @;
               output;
            end;
         end;
      end;
   end;
   datalines;
4.91 4.63 4.76 5.04 5.38 6.21 5.60 5.08
4.94 3.98 4.64 5.26 5.28 5.01 5.45 5.62
5.20 4.45 5.05 5.03 5.01 4.63 5.80 5.90
6.00 5.39 4.95 5.39 6.18 5.94 6.58 6.25
5.86 5.41 5.54 5.41 5.28 6.67 6.65 5.94
5.45 5.12 4.73 4.62 5.06 5.75 6.39 5.62
4.96 5.63 5.47 5.31 6.18 6.31 5.95 6.14
5.71 5.37 6.21 5.83 6.28 6.55 6.39 5.57
4.60 4.90 4.88 4.73 5.89 6.20 5.68 5.72
5.79 5.33 5.13 5.18 5.86 5.98 5.55 4.32
5.61 5.15 4.82 5.06 5.67 5.54 5.19 4.46
5.13 4.90 4.88 5.18 5.45 5.80 5.12 4.42
;
```

```
proc anova data=Barley;
   class Rep Soil Calcium Fertilizer;
   model Yield =
           Rep
           Fertilizer Fertilizer*Rep
           Calcium Calcium*Fertilizer Calcium*Rep(Fertilizer)
           Soil Soil*Rep
           Soil*Fertilizer Soil*Rep*Fertilizer
           Soil*Calcium Soil*Fertilizer*Calcium
           Soil*Calcium*Rep(Fertilizer);
   test h=Fertilizer                 e=Fertilizer*Rep;
   test h=Calcium calcium*fertilizer e=Calcium*Rep(Fertilizer);
   test h=Soil                       e=Soil*Rep;
   test h=Soil*Fertilizer            e=Soil*Rep*Fertilizer;
   test h=Soil*Calcium
          Soil*Fertilizer*Calcium    e=Soil*Calcium*Rep(Fertilizer);
   means Fertilizer Calcium Soil Calcium*Fertilizer;
run;
```

Output 17.5.1. Class Level Information

```
            Strip-split Plot

          The ANOVA Procedure

        Class Level Information

   Class           Levels    Values

   Rep                  4    1 2 3 4

   Soil                 3    1 2 3

   Calcium              2    0 1

   Fertilizer           4    0 1 2 3


Number of Observations Read          96
Number of Observations Used          96
```

Output 17.5.2. ANOVA Table

```
Dependent Variable: Yield

                                          Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                       95      31.89149583     0.33569996         .        .

 Error                        0       0.00000000         .

 Corrected Total             95      31.89149583


               R-Square     Coeff Var      Root MSE    Yield Mean

               1.000000             .             .      5.427292


 Source                      DF         Anova SS    Mean Square   F Value   Pr > F

 Rep                          3       6.27974583     2.09324861         .        .
 Fertilizer                   3       7.22127083     2.40709028         .        .
 Rep*Fertilizer               9       6.08211250     0.67579028         .        .
 Calcium                      1       0.27735000     0.27735000         .        .
 Calcium*Fertilizer           3       1.96395833     0.65465278         .        .
 Rep*Calcium(Fertili)        12       1.76705833     0.14725486         .        .
 Soil                         2       1.92658958     0.96329479         .        .
 Rep*Soil                     6       1.66761042     0.27793507         .        .
 Soil*Fertilizer              6       0.68828542     0.11471424         .        .
 Rep*Soil*Fertilizer         18       1.58698125     0.08816563         .        .
 Soil*Calcium                 2       0.04493125     0.02246562         .        .
 Soil*Calcium*Fertili         6       0.18936042     0.03156007         .        .
 Rep*Soil*Calc(Ferti)        24       2.19624167     0.09151007         .        .
```

As the model is completely specified by the MODEL statement, the entire top portion of output (Output 17.5.2) should be ignored. Look at the following output produced by the various TEST statements.

Output 17.5.3. Tests of Effects

```
Dependent Variable: Yield

   Tests of Hypotheses Using the Anova MS for Rep*Fertilizer as an Error Term

 Source                     DF       Anova SS    Mean Square   F Value   Pr > F

 Fertilizer                  3     7.22127083     2.40709028      3.56   0.0604


                   Tests of Hypotheses Using the Anova MS for
                     Rep*Calcium(Fertili) as an Error Term

 Source                     DF       Anova SS    Mean Square   F Value   Pr > F

 Calcium                     1     0.27735000     0.27735000      1.88   0.1950
 Calcium*Fertilizer          3     1.96395833     0.65465278      4.45   0.0255


      Tests of Hypotheses Using the Anova MS for Rep*Soil as an Error Term

 Source                     DF       Anova SS    Mean Square   F Value   Pr > F

 Soil                        2     1.92658958     0.96329479      3.47   0.0999


                   Tests of Hypotheses Using the Anova MS for
                      Rep*Soil*Fertilizer as an Error Term

 Source                     DF       Anova SS    Mean Square   F Value   Pr > F

 Soil*Fertilizer             6     0.68828542     0.11471424      1.30   0.3063


                   Tests of Hypotheses Using the Anova MS for
                     Rep*Soil*Calc(Ferti) as an Error Term

 Source                     DF       Anova SS    Mean Square   F Value   Pr > F

 Soil*Calcium                2     0.04493125     0.02246562      0.25   0.7843
 Soil*Calcium*Fertili        6     0.18936042     0.03156007      0.34   0.9059
```

The only significant effect is the Calcium*Fertilizer interaction.

Output 17.5.4. Results of MEANS statement

```
               Level of            ------------Yield-----------
               Fertilizer      N           Mean         Std Dev

               0              24     5.18416667      0.48266395
               1              24     5.12916667      0.38337082
               2              24     5.75458333      0.53293265
               3              24     5.64125000      0.63926801


                Level of           ------------Yield-----------
                Calcium       N            Mean         Std Dev

                0            48      5.48104167      0.54186141
                1            48      5.37354167      0.61565219


                Level of           ------------Yield-----------
                Soil          N            Mean         Std Dev

                1            32      5.54312500      0.55806369
                2            32      5.51093750      0.62176315
                3            32      5.22781250      0.51825224


         Level of     Level of              ------------Yield-----------
         Calcium      Fertilizer      N             Mean         Std Dev

         0            0              12       5.34666667      0.45029956
         0            1              12       5.08833333      0.44986530
         0            2              12       5.62666667      0.44707806
         0            3              12       5.86250000      0.52886027
         1            0              12       5.02166667      0.47615569
         1            1              12       5.17000000      0.31826233
         1            2              12       5.88250000      0.59856077
         1            3              12       5.42000000      0.68409197
```

The final portion of output shows the results of the MEANS statement. This portion shows means for various effects and combinations of effects, as requested. Because no multiple comparison procedures are requested, none are performed. You can examine the Calcium*Fertilizer means to understand the interaction better.

In this example, you could reduce memory requirements by omitting the Soil*Calcium*Rep(Fertilizer) effect from the model in the MODEL statement. This effect then becomes the ERROR effect, and you can omit the last TEST statement (in the code shown earlier). The test for the Soil*Calcium effect is then given in the Analysis of Variance table in the top portion of output. However, for all other tests, you should look at the results from the TEST statement. In large models, this method may lead to significant reductions in memory requirements.

References

Bartlett, M.S. (1937), “Properties of Sufficiency and Statistical Tests,” *Proceedings of the Royal Society of London, Series A* 160, 268–282.

Brown, M.B. and Forsythe, A.B. (1974), “Robust Tests for Equality of Variances,” *Journal of the American Statistical Association,* 69, 364–367.

Erdman, L.W. (1946), “Studies to Determine if Antibiosis Occurs among Rhizobia,” *Journal of the American Society of Agronomy,* 38, 251–258.

Fisher, R.A. (1942), *The Design of Experiments,* Third Edition, Edinburgh: Oliver & Boyd.

Freund, R.J., Littell, R.C., and Spector, P.C. (1986), *SAS System for Linear Models, 1986 Edition*, Cary, NC: SAS Institute Inc.

Graybill, F.A. (1961), *An Introduction to Linear Statistical Models,* Volume I, New York: McGraw-Hill Book Co.

Henderson, C.R. (1953), “Estimation of Variance and Covariance Components,” *Biometrics,* 9, 226–252.

Levene, H. (1960), “Robust Tests for the Equality of Variance,” in *Contributions to Probability and Statistics,* ed. I. Olkin, Palo Alto, CA: Stanford University Press, 278–292.

Neter, J., Wasserman, W., and Kutner, M.H. (1990), *Applied Linear Statistical Models: Regression, Analysis of Variance, and Experimental Designs*, Homewood, IL: Richard D. Irwin, Inc.

O’Brien, R.G. (1979), “A General ANOVA Method for Robust Tests of Additive Models for Variances,” *Journal of the American Statistical Association*, 74, 877–880.

O’Brien, R.G. (1981), “A Simple Test for Variance Effects in Experimental Designs,” *Psychological Bulletin,* 89(3), 570–574.

Remington, R.D. and Schork, M.A. (1970), *Statistics with Applications to the Biological and Health Sciences*, Englewood Cliffs, NJ: Prentice-Hall, Inc.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons, Inc.

Schlotzhauer, S.D. and Littell, R.C. (1987), *SAS System for Elementary Statistical Analysis*, Cary, NC: SAS Institute Inc.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Smith, W.G. (1951), Dissertation Notes on Canadian Sugar Factories, Ltd., Alberta, Canada: Taber.

Snedecor, G.W. and Cochran, W.G. (1967), *Statistical Methods,* Sixth Edition, Ames, IA: Iowa State University Press.

Steel, R.G.D. and Torrie, J.H. (1980), *Principles and Procedures of Statistics,* New York: McGraw-Hill Book Co.

Chapter 18
The BOXPLOT Procedure

Chapter Contents

Chapter 18
The BOXPLOT Procedure

Overview

The BOXPLOT procedure creates side-by-side box-and-whisker plots of measurements organized in groups. A box-and-whisker plot displays the mean, quartiles, and minimum and maximum observations for a group. Throughout this chapter, this type of plot, which can contain one or more box-and-whisker plots, is referred to as a *box plot*.

The PLOT statement of the BOXPLOT procedure produces a box plot. You can specify more than one PLOT statement to produce multiple box plots.

You can use options in the PLOT statement to

- control the style of the box-and-whisker plots
- specify one of several methods for calculating quantile statistics (percentiles)
- add block legends and symbol markers to reveal stratification in data
- display vertical and horizontal reference lines
- control axis values and labels
- overlay the box plot with plots of additional variables
- control the layout and appearance of the plot

The INSET and INSETGROUP statements produce boxes or tables (referred to as *insets*) of summary statistics or other data on a box plot. An INSET statement produces an inset of statistics pertaining to the entire box plot. An INSETGROUP statement produces an inset containing statistics calculated separately for each group. An INSET or INSETGROUP statement by itself does not produce a display; it must be used with a PLOT statement.

You can use options in an INSET or INSETGROUP statement to

- specify the position of the inset
- specify a header for the inset
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

Getting Started

Creating Box Plots from Raw Data

This section demonstrates how you can use the BOXPLOT procedure to produce box plots for your data.

Suppose that a petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set called Turbine that contains the power output measurements for 10 nonconsecutive days.

```
data Turbine;
   informat day date7.;
   format day date5.;
   label kwatts='Average Power Output';
   input day @;
   do i=1 to 10;
      input kwatts @;
      output;
      end;
   drop i;
   datalines;
05JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
05JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
06JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
06JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
07JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
07JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
08JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
08JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
11JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
11JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
12JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
12JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
13JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
13JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
14JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
14JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
15JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
15JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
18JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
18JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
;
run;
```

In the data set Turbine, each observation contains the date and the power output for a single heating. The first 20 observations contain the outputs for the first day, the second 20 observations contain the outputs for the second day, and so on. Because the variable day classifies the observations into groups, it is referred to as the *group* *variable*. The variable kwatts contains the output measurements and is referred to as the *analysis variable*.

You can create a box plot to examine the distribution of power output for each day. The following statements create the box plot shown in Figure 18.1.

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
run;
```

The input data set Turbine is specified with the DATA= option in the PROC BOXPLOT statement. The PLOT statement requests a box-and-whisker plot for each group of data. After the keyword PLOT, you specify the analysis variable (in this case, kwatts), followed by an asterisk and the group variable (day).

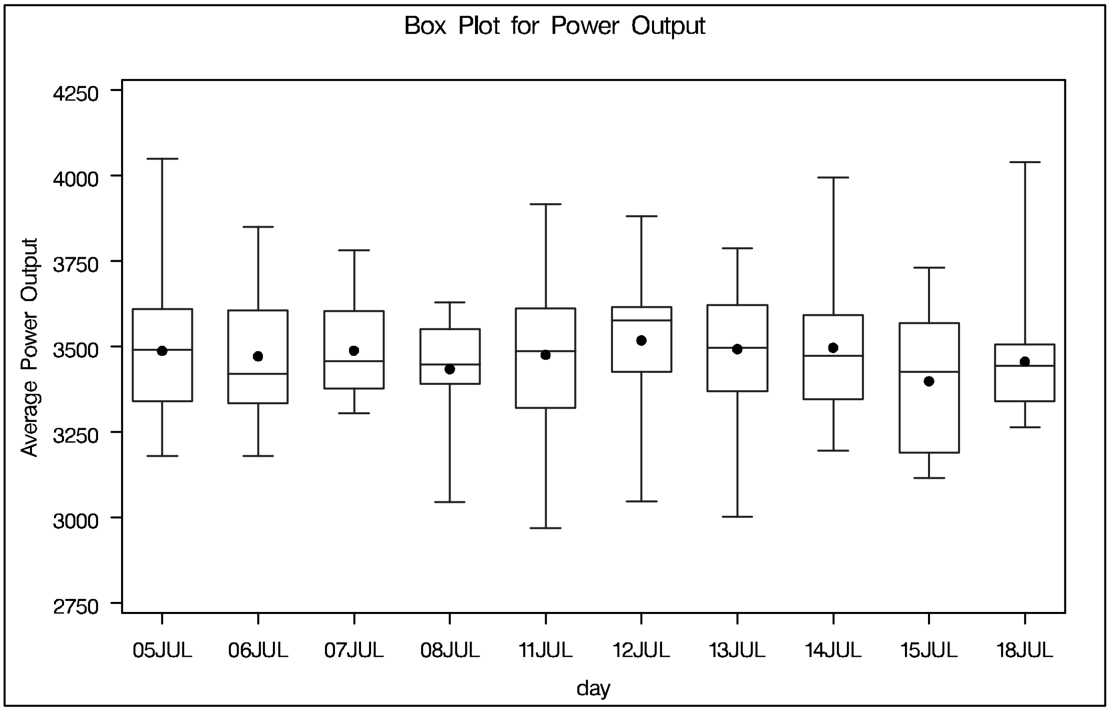


Figure 18.1. Box Plot for Power Output Data

The box plot displayed in Figure 18.1 represents summary statistics for the analysis variable kwatts; each of the 10 box-and-whisker plots describes the variable kwatts for a particular day. The plot elements and the statistics they represent are as follows:

- the length of the box represents the interquartile range (the distance between the 25th and the 75th percentiles)
- the dot in the box interior represents the mean
- the horizontal line in the box interior represents the median
- the vertical lines issuing from the box extend to the minimum and maximum values of the analysis variable

Displaying Summary Statistics on a Box Plot

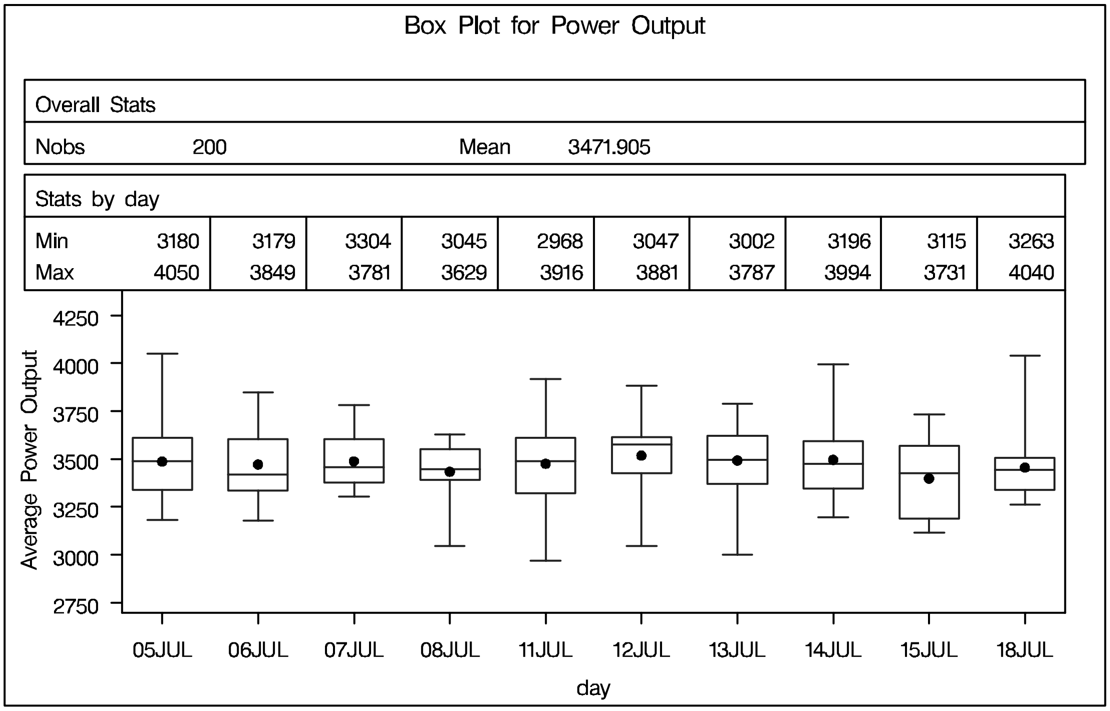
This section demonstrates how you can use the INSET and INSETGROUP statements to include tables of summary statistics on your box plots.

The following code produces a box plot of the Turbine data set from the previous section, augmented with insets containing summary statistics. The resulting plot is shown in Figure 18.2.

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
   inset nobs mean / header = 'Overall Stats'
      pos    = tm;
   insetgroup min max / header = 'Stats by day';
run;
```

The INSET statement produces an inset of overall summary statistics. The NOBS and MEAN keywords request the total number of observations and the overall mean across all days. The POS=TM option places the inset in the top margin of the plot.

The INSETGROUP statement produces an inset containing statistics calculated for each day separately. The MIN and MAX keywords request the minimum and maximum observations in each group.



Overall Stats

Nobs	200	Mean	3471.905

Stats by day

Min	3180	3179	3304	3045	2968	3047	3002	3196	3115	3263
Max	4050	3849	3781	3629	3916	3881	3787	3994	3731	4040

Figure 18.2. Box Plot with Insets

Syntax

The syntax for the BOXPLOT procedure is as follows:

PROC BOXPLOT < *options* > **;**
  PLOT *analysis-variable*group-variable* < (*block-variables*) >
    < =*symbol-variable* > < / *options* > **;**
  INSET *keywords* < / *options* > **;**
  INSETGROUP *keywords* < / *options* > **;**
  BY *variables***;**
  ID *variables***;**

Both the PROC BOXPLOT and PLOT statements are required. You can specify any number of PLOT statements within a single PROC BOXPLOT invocation.

PROC BOXPLOT Statement

PROC BOXPLOT < *options* > **;**

The PROC BOXPLOT statement starts the BOXPLOT procedure. The following options can appear in the PROC BOXPLOT statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*, that enhances all box plots requested in subsequent PLOT statements.

BOX=*SAS-data-set*

names an input data set containing group summary statistics and outlier values. Typically, this data set is created as an OUTBOX= data set in a previous run of PROC BOXPLOT. Each group summary statistic and outlier value is recorded in a separate observation in a BOX= data set, so there are multiple observations per group. You cannot use a BOX= data set together with a DATA= or HISTORY= data set. If you do not specify one of these input data sets, the procedure uses the most recently created SAS data set as a DATA= data set.

DATA=*SAS-data-set*

names an input data set containing raw data to be analyzed. You cannot use a DATA= data set together with a BOX= or a HISTORY= data set. If you do not specify one of these input data sets, the procedure uses the most recently created SAS data set as a DATA= data set.

GOUT=<*libref.*>*output catalog*

specifies the SAS catalog in which to save the graphics output that is produced by the BOXPLOT procedure. If you omit the libref, PROC BOXPLOT looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist.

HISTORY=*SAS-data-set*
HIST=*SAS-data-set*

names an input data set containing group summary statistics. Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC BOXPLOT, but it can also be created using a SAS summarization procedure such as PROC MEANS. The HISTORY= data set can contain only one observation for each value of the *group-variable*. You cannot use a HISTORY= data set with a DATA= or a BOX= data set. If you do not specify one of these three input data sets, PROC BOXPLOT uses the most recently created data set as a DATA= data set.

PLOT Statement

PLOT *(analysis-variables)*group-variable <(block-variables) >*
< =symbol-variable > < / options >;

You can specify multiple PLOT statements after the PROC BOXPLOT statement. The components of the PLOT statement are as follows.

analysis-variables

identify one or more variables to be analyzed. An analysis variable is required. If you specify more than one analysis variable, enclose the list in parentheses. For example, the following statements request distinct box plots for the variables weight, length, and width:

```
proc boxplot data=summary;
   plot (weight length width)*day;
run;
```

group-variable specifies the variable that identifies groups in the data. The group variable is required. In the preceding PLOT statement, day is the group variable.

block-variables specify optional variables that group the data into blocks of consecutive groups. These blocks are labeled in a legend, and each block variable provides one level of labels in the legend.

symbol-variable specifies an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the symbol variable. You can specify the symbol markers with SYMBOLn statements (refer to *SAS/GRAPH Software: Reference* for complete details).

options enhance the appearance of the box plot, request additional analyses, save results in data sets, and so on. Complete descriptions for each option follow.

Table 18.1 lists all options in the PLOT statement by function.

PLOT Statement Options

Table 18.1. PLOT Statement Options

Option	Description
Options for Controlling Box Appearance	
BOXCONNECT=	connects features of adjacent box-and-whisker plots with line segments
BOXSTYLE=	specifies style of box-and-whisker plots
BOXWIDTH=	specifies width of box-and-whisker plots
BOXWIDTHSCALE=	specifies that widths of box-and-whisker plots vary proportionately to group size
CBOXES=	specifies color for outlines of box-and-whisker plots
CBOXFILL=	specifies fill color for interior of box-and-whisker plots
IDCOLOR=	specifies outlier symbol color in schematic box-and-whisker plots
IDCTEXT=	specifies outlier label color in schematic box-and-whisker plots
IDFONT=	specifies outlier label font in schematic box-and-whisker plots
IDHEIGHT=	specifies outlier label height in schematic box-and-whisker plots
IDSYMBOL=	specifies outlier symbol in schematic box-and-whisker plots
LBOXES=	specifies line types for outlines of box-and-whisker plots
NOSERIFS	eliminates serifs from the whiskers of box-and-whisker plots
NOTCHES	specifies that box-and-whisker plots are to be notched
PCTLDEF=	specifies percentile definition used for box-and-whisker plots
Options for Plotting and Labeling Points	
ALLLABEL=	labels means of box-and-whisker plots
CLABEL=	specifies color for labels requested with ALLLABEL= option
CCONNECT=	specifies color for line segments that connect points on plot
LABELANGLE=	specifies angle for labels requested with ALLLABEL= option
SYMBOLLEGEND=	specifies LEGEND statement for levels of the symbol variable
SYMBOLORDER=	specifies order in which symbols are assigned for levels of the symbol variable
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= option
CVREF=	specifies color for lines requested by VREF= option
HREF=	requests reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set is to be applied uniformly to plots created for all BY groups

Table 18.1. (continued)

Option	Description
VREF=	requests reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	specifies position of VREFLABELS= labels
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for the block variable legend
BLOCKLABTYPE=	specifies text size of the block variable legend
BLOCKPOS=	specifies vertical position of the block variable legend
BLOCKREP	repeats identical consecutive labels in the block variable legend
CBLOCKLAB=	specifies colors for filling frames enclosing block variable labels
CBLOCKVAR=	specifies colors for filling background of the block variable legend
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill color for frame for plot area
CONTINUOUS	produces horizontal axis for continuous group variable values
CTEXT=	specifies color for tick mark values and axis labels
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
NOHLABEL	suppresses label for horizontal axis
NOTICKREP	specifies that only the first occurrence of repeated, adjacent character group values is to be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal axis labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick marks
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Input Data Set Options	
MISSBREAK	specifies that missing values between identical character group values signify the start of a new group
Output Data Set Options	

Table 18.1. (continued)

Option	**Description**
OUTBOX=	produces an output data set containing group summary statistics and outlier values
OUTHISTORY=	produces an output data set containing group summary statistics
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to box plot
BWSLEGEND	displays a legend identifying the function of group size specified with the BOXWIDTHSCALE= option
DESCRIPTION=	specifies string that appears in the description field of the PROC GREPLAY master menu for box plot
FONT=	specifies software font for labels and legends on plots
HTML=	specifies URLs to be associated with box-and-whisker plots
NAME=	specifies name that appears in the name field of the PROC GREPLAY master menu for box plot
NLEGEND	requests a legend displaying group sizes
OUTHIGHHTML=	specifies URLs to be associated with high outliers on box-and-whisker plots
OUTLOWHTML=	specifies URLs to be associated with low outliers on box-and-whisker plots
PAGENUM=	specifies the form of the label used in pagination
PAGENUMPOS=	specifies the position of the page number requested with the PAGENUM= option
Grid Options	
CGRID=	specifies color for grid requested with ENDGRID or GRID option
ENDGRID	adds grid after last box-and-whisker plot
GRID	adds grid to box plot
LENDGRID=	specifies line type for grid requested with the ENDGRID option
LGRID=	specifies line type for grid requested with the GRID option
WGRID=	specifies width of grid lines
Plot Layout Options	
INTERVAL=	specifies natural time interval between consecutive group positions when time, date, or datetime format is associated with a numeric group variable
INTSTART=	specifies first major tick mark value on horizontal axis when a date, time, or datetime format is associated with numeric group variable
MAXPANELS=	specifies maximum number of panels for plot
NOCHART	suppresses creation of the box plot
NOFRAME	suppresses frame for plot area
NPANELPOS=	specifies number of group positions per panel on each plot

Table 18.1. (continued)

Option	**Description**
REPEAT	repeats last group position on panel as first group position of next panel
TOTPANELS=	specifies number of panels to be used to display plot
Overlay Options	
CCOVERLAY=	specifies colors for line segments connecting points on overlays
COVERLAY=	specifies colors for points on overlays
LOVERLAY=	specifies line types for line segments connecting points on overlays
NOOVERLAYLEGEND	suppresses overlay legend
OVERLAY=	specifies variables to be plotted on overlays
OVERLAYHTML=	specifies URLs to be associated with overlay plot points
OVERLAYID=	specifies labels for overlay plot points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols used for overlays
OVERLAYSYMHT=	specifies heights for overlay symbols
WOVERLAY=	specifies widths for line segments connecting points on overlays
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme values are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR=	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
COVERLAYCLIP=	specifies color for clipped points on overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays

Following are explanations of the options that you can specify in the PLOT statement after a slash (/).

ALLLABEL=VALUE
ALLLABEL=(*variable*)

labels the point plotted for the mean of each box-and-whisker plot with its VALUE or with the value of a *variable* in the input data set.

ANNOTATE=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*.

BLOCKLABELPOS=ABOVE | LEFT

specifies the position of a block variable label in the block legend. The keyword ABOVE places the label immediately above the legend, and LEFT places the label to the left of the legend. Use the keyword LEFT with labels that are short enough to fit in the margin of the plot; otherwise, they are truncated. The default keyword is ABOVE.

BLOCKLABTYPE=SCALED | TRUNCATED
BLOCKLABTYPE=*height*

specifies how lengthy block variable values are to be treated when there is insufficient space to display them in the block legend. If you specify BLOCKLABTYPE=SCALED, the values are uniformly reduced in height so that they fit. If you specify BLOCKLABTYPE=TRUNCATED, lengthy values are truncated on the right until they fit. You can also specify a text height in vertical percent screen units for the values. By default, lengthy values are not displayed. For more information, see the section "Displaying Blocks of Data" on page 530.

BLOCKPOS=*n*

specifies the vertical position of the legend for the values of the block variables. Values of n and the corresponding positions are as follows. By default, BLOCKPOS=1.

n	Legend Position
1	top of plot, offset from axis frame
2	top of plot, immediately above axis frame
3	bottom of plot, immediately above horizontal axis
4	bottom of plot, below horizontal axis label

BLOCKREP

specifies that block variable values for all groups are to be displayed. By default, only the first block variable value in any block is displayed, and repeated block variable values are not displayed.

BOXCONNECT=MEAN | MEDIAN | MAX | MIN | Q1 | Q3
BOXCONNECT

specifies that the points in adjacent box-and-whisker plots representing group means, medians, maximum values, minimum values, first quartiles, or third quartiles are to be connected with line segments. If the BOXCONNECT option is specified without a keyword identifying the points to be connected, group means are connected. By default, no points are connected.

BOXSTYLE=*keyword*

specifies the style of the box-and-whisker plots displayed. If you specify BOXSTYLE=SKELETAL, the whiskers are drawn from the edges of the box to the extreme values of the group. This plot is sometimes referred to as a skeletal box-and-whisker plot. By default, the whiskers are drawn with serifs: you can specify the NOSERIFS option to draw the whiskers without serifs.

In the following descriptions, the terms *fence* and *far fence* refer to the distance from the first and third quartiles (25th and 75th percentiles, respectively), expressed in terms of the interquartile range (IQR). For example, the lower fence is located at $1.5 \times \mathrm{IQR}$ below the 25th percentile; the upper fence is located at $1.5 \times \mathrm{IQR}$ above the 75th percentile. Similarly, the lower far fence is located at $3 \times \mathrm{IQR}$ below the 25th percentile; the upper far fence is located at $3 \times \mathrm{IQR}$ above the 75th percentile.

If you specify BOXSTYLE=SCHEMATIC, a whisker is drawn from the upper edge of the box to the largest observed value within the upper fence and from the lower edge of the box to the smallest observed value within the lower fence. Serifs are added to the whiskers by default. Observations outside the fences are identified with a special symbol; you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. The default symbol is a square. This type of plot corresponds to the schematic box-and-whisker plot described in Chapter 2 of Tukey (1977). See Figure 18.5 and the discussion in the section "Styles of Box Plots" on page 522 for more information.

If you specify BOXSTYLE=SCHEMATICID, a schematic box-and-whisker plot is displayed in which an ID variable value is used to label the symbol marking each observation outside the upper and lower fences. A BOX= data set can contain a variable named _ID_ that is used as the ID variable. Otherwise, the first variable listed in the ID statement provides the labels.

If you specify BOXSTYLE=SCHEMATICIDFAR, a schematic box-and-whisker plot is displayed in which the value of the ID variable is used to label the symbol marking each observation outside the lower and upper far fences. Observations between the fences and the far fences are identified with a symbol but are not labeled with the ID variable.

Figure 18.3 illustrates the elements of a skeletal box-and-whisker plot.

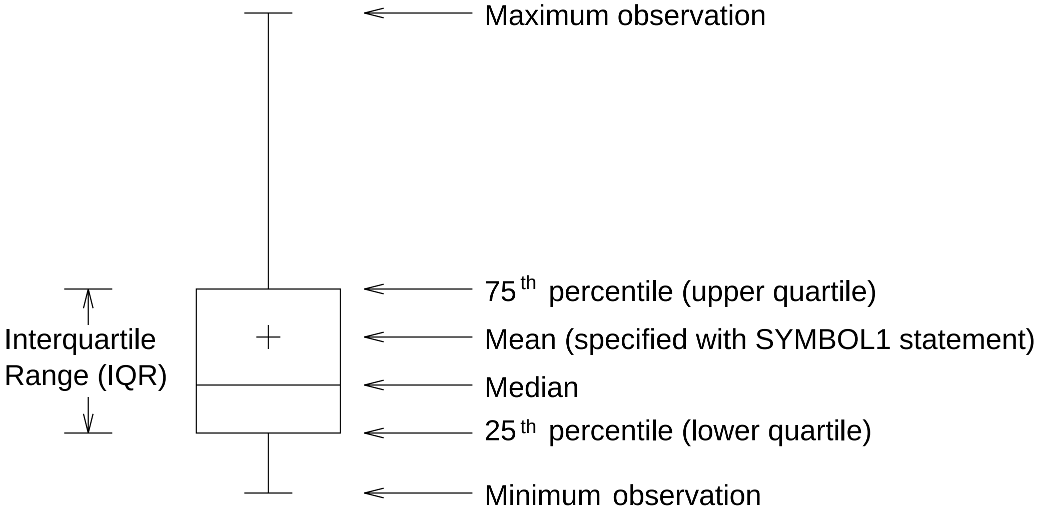

Maximum observation
Interquartile
Range (IQR)
75th percentile (upper quartile)
Mean (specified with SYMBOL1 statement)
Median
25th percentile (lower quartile)
Minimum observation

Figure 18.3. Skeletal Box-and-Whisker Plot

The skeletal style of the box-and-whisker plot shown in Figure 18.3 is the default.

BOXWIDTH=*value*

specifies the width (in horizontal percent screen units) of the box-and-whisker plots.

BOXWIDTHSCALE=*value*

specifies that the box-and-whisker plot width is to vary proportionately to a particular function of the group size n. The function is determined by the *value*.

If you specify a positive value, the widths are proportional to n^{value}. In particular, if you specify BOXWIDTHSCALE=1, the widths are proportional to the group size. If you specify BOXWIDTHSCALE=0.5, the widths are proportional to $\sqrt{n}$, as described by McGill, Tukey, and Larsen (1978). If you specify BOXWIDTHSCALE=0, the widths are proportional to $\log(n)$. See Example 18.4 on page 543 for an illustration of the BOXWIDTHSCALE= option.

You can specify the BWSLEGEND option to display a legend identifying the function of n used to determine the box-and-whisker plot widths.

By default, the box widths are constant.

BWSLEGEND

displays a legend identifying the function of group size n specified with the BOXWIDTHSCALE= option. No legend is displayed if all group sizes are equal. The BWSLEGEND option is not applicable unless you also specify the BOXWIDTHSCALE= option.

CAXIS=*color*
CAXES=*color*
CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default value is the first color in the device color list.

CBLOCKLAB=*color* **|** *(color-list)*

specifies fill colors for the frames that enclose the block variable labels in a block legend. By default, these areas are not filled. Colors in the CBLOCKLAB= list are matched with block variables in the order in which they appear in the PLOT statement.

CBLOCKVAR=*variable* **|** *(variable-list)*

specifies variables whose values are colors for filling the background of the legend associated with block variables. Each CBLOCKVAR= variable must be a character variable of no more than eight characters in the input data set, and its values must be valid SAS/GRAPH color names (refer to *SAS/GRAPH Software: Reference* for complete details). A list of CBLOCKVAR= variables must be enclosed in parentheses.

The procedure matches the CBLOCKVAR= variables with block variables in the order specified. That is, each block legend is filled with the color value of the CBLOCKVAR= variable of the first observation in each block. In general, values of the ith CBLOCKVAR= variable are used to fill the block of the legend corresponding to the ith block variable.

By default, fill colors are not used for the block variable legend. The CBLOCKVAR= option is available only when block variables are used in the PLOT statement.

CBOXES=*color*
CBOXES=*(variable)*

specifies the colors for the outlines of the box-and-whisker plots created with the PLOT statement. You can use one of the following approaches:

- You can specify CBOXES=*color* to provide a single outline color for all the box-and-whisker plots.
- You can specify CBOXES=*(variable)* to provide a distinct outline color for each box-and-whisker plot as the value of the variable. The variable must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH color names (refer to *SAS/GRAPH Software: Reference* for complete details). The outline color of the plot displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that, if there are multiple observations per group in the input data set, the values of the variable should be identical for all the observations in a given group.

The default color is the second color in the device color list.

CBOXFILL=*color*
CBOXFILL=*(variable)*

specifies the interior fill colors for the box-and-whisker plots. You can use one of the following approaches:

- You can specify CBOXFILL=*color* to provide a single color for all of the box-and-whisker plots.
- You can specify CBOXFILL=*(variable)* to provide a distinct color for each box-and-whisker plot as the value of the variable. The variable must be a character variable of length 8 or less in the input data set, and its values must be valid SAS/GRAPH color names (or the value EMPTY, which you can use to suppress color filling). Refer to *SAS/GRAPH Software: Reference* for complete details. The interior color of the box displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that if there are multiple observations per group in the input data set, the values of the variable should be identical for all the observations in a given group.

By default, the interiors are not filled.

CCLIP=*color*

specifies a color for the plotting symbol that is specified with the CLIPSYMBOL= option to mark clipped values. The default color is the color specified in the COLOR= option in the SYMBOL1 statement.

CCONNECT=*color*

specifies the color for line segments connecting points on the plot. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. This option is not applicable unless you also specify the BOXCONNECT= option.

CCOVERLAY=*(color-list)*

specifies the colors for line segments connecting points on overlay plots. Colors in the CCOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points of an overlay plot.

CFRAME=*color*

specifies the color for filling the rectangle enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option.

CGRID=*color*

specifies the color for the grid requested by the ENDGRID or GRID option. By default, the grid is the same color as the axes.

CHREF=*color*

specifies the color for the lines requested by the HREF= option. The default value is the first color in the device color list.

CLABEL=*color*

specifies the color for labels produced by the ALLLABEL= option. The default color is the CTEXT= color.

CLIPFACTOR=*factor*

requests clipping of extreme values on the box plot. The *factor* that you specify determines the extent to which these values are clipped, and it must be greater than 1.

For examples of the CLIPFACTOR= option, see Figure 18.14 on page 534 and Figure 18.15 on page 535. Related clipping options are CCLIP=, CLIPLEGEND=, CLIPLEGPOS=, CLIPSUBCHAR=, and CLIPSYMBOL=.

CLIPLEGEND='*label***'**

specifies the *label* for the legend that indicates the number of clipped boxes when the CLIPFACTOR= option is used. The *label* must be no more than 16 characters and must be enclosed in quotes. For an example, see Figure 18.15 on page 535.

CLIPLEGPOS=TOP | BOTTOM

specifies the position for the legend that indicates the number of clipped boxes when the CLIPFACTOR= option is used. The keywords TOP and BOTTOM position the legend at the top or bottom of the chart, respectively. Do not specify CLIPLEGPOS=TOP together with the PHASELEGEND option or the BLOCKPOS=1 or BLOCKPOS=2 options. By default, CLIPLEGPOS=BOTTOM.

CLIPSUBCHAR=’*character*’

specifies a substitution character (such as #) for the label provided with the CLIPLEGEND= option. The substitution character is replaced with the number of boxes that are clipped. For example, suppose that the following statements produce a chart in which three boxes are clipped:

```
proc boxplot data=pistons;
   plot diameter*hour /
      clipfactor  = 1.5
      cliplegend  = ’Boxes clipped=#’
      clipsubchar = ’#’ ;
run;
```

Then the clipping legend displayed on the chart will be

```
Boxes clipped=3
```

CLIPSYMBOL=*symbol*

specifies a plot symbol used to identify clipped points on the chart and in the legend when the CLIPFACTOR= option is used. You should use this option in conjunction with the CLIPFACTOR= option. The default *symbol* is CLIPSYMBOL=SQUARE.

CLIPSYMBOLHT=*value*

specifies the height for the symbol marker used to identify clipped points on the chart when the CLIPFACTOR= option is used. The default is the height specified with the H= option in the SYMBOL statement.

For general information about clipping options, refer to “Clipping Extreme Values” on page 532.

CONTINUOUS

specifies that numeric group variable values are to be treated as continuous values. By default, the values of a numeric group variable are considered discrete values unless the HAXIS= option is specified. For more information, see the discussion in the section “Continuous Group Variables” on page 524.

COVERLAY=*(color-list)*

specifies the colors used to plot overlay variables. Colors in the COVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

COVERLAYCLIP=*color*

specifies the color used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default color is the color specified in the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

specifies the color for the lines requested by the VREF= option. The default value is the first color in the device color list.

DESCRIPTION=*'string'*
DES=*'string'*

specifies a description of the box plot, not longer than 40 characters, that appears in the PROC GREPLAY master menu. The default string is the variable name.

ENDGRID

adds a grid to the rightmost portion of the plot, beginning with the first labeled major tick mark position that follows the last box-and-whisker plot. You can use the HAXIS= option to force space to be added to the horizontal axis.

FONT=*font*

specifies a software font for labels and legends. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Hardware characters are used by default. Refer to *SAS/GRAPH Software: Reference* for more information on the GOPTIONS statement.

GRID

adds a grid to the box plot. Grid lines are horizontal lines positioned at labeled major tick marks, and they cover the length and height of the plotting area.

HAXIS=*values*
HAXIS=AXIS*n*

specifies tick mark values for the horizontal (group) axis. If the group variable is numeric, the values must be numeric and equally spaced. Optionally, you can specify an axis name defined in a previous AXIS statement. Refer to *SAS/GRAPH Software: Reference* for more information on the AXIS statement.

Specifying the HAXIS= option with a numeric group variable causes the group variable values to be treated as continuous values. For more information, see the description of the CONTINUOUS option and the discussion in the section "Continuous Group Variables" on page 524. Numeric values can be given in an explicit or implicit list. If the group variable is character, values must be quoted strings of length 16 or less. If a date, time, or datetime format is associated with a numeric group variable, SAS datetime literals can be used. Examples of HAXIS= lists follow:

- haxis=0 2 4 6 8 10
- haxis=0 to 10 by 2
- haxis='LT12A' 'LT12B' 'LT12C' 'LT15A' 'LT15B' 'LT15C'
- haxis='20MAY88'D to '20AUG88'D by 7
- haxis='01JAN88'D to '31DEC88'D by 30

If the group variable is numeric, the HAXIS= list must span the group variable values; if it is a character variable, the HAXIS= list must include all of the group variable values. You can add group positions to the box plot by specifying HAXIS= values that are not group variable values.

If you specify a large number of HAXIS= values, some of these may be thinned to avoid collisions between tick mark labels. To avoid thinning, use one of the following methods.

- Shorten values of the group variable by eliminating redundant characters. For example, if your group variable has values LOT1, LOT2, LOT3, and so on, you can use the SUBSTR function in a DATA step to eliminate LOT from each value, and you can modify the horizontal axis label to indicate that the values refer to lots.
- Use the TURNHLABELS option to turn the labels vertically.
- Use the NPANELPOS= option to force fewer group positions per panel.

HEIGHT=*value*

specifies the height (in vertical screen percent units) of the text for axis labels and legends. This value takes precedence over the HTEXT= value specified in the GOPTIONS statement. This option is recommended for use with software fonts specified with the FONT= option or with the FTEXT= option in the GOPTIONS statement. Refer to *SAS/GRAPH Software: Reference* for complete information on the GOPTIONS statement.

HMINOR=*n*
HM=*n*

specifies the number of minor tick marks between each major tick mark on the horizontal axis. Minor tick marks are not labeled. The default is HMINOR=0.

HOFFSET=*value*

specifies the length (in percent screen units) of the offset at both ends of the horizontal axis. You can eliminate the offset by specifying HOFFSET=0.

HREF=*values*
HREF=*SAS-data-set*

draws reference lines perpendicular to the horizontal (group) axis on the box plot. You can use this option in the following ways:

- You can specify the values for the lines with an HREF= list. If the group variable is numeric, the values must be numeric. If the group variable is character, the values must be quoted strings of up to 16 characters. If the group variable is formatted, the values must be given as internal values. Examples of HREF= values follow:

```
href=5
href=5 10 15 20 25 30
href='Shift 1' 'Shift 2' 'Shift 3'
```

- You can specify reference line values as the values of a variable named _REF_ in an HREF= data set. The type and length of _REF_ must match those of the group variable specified in the PLOT statement. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in plots for different analysis variables specified in the PLOT statement, you must include a character variable named _VAR_, whose values are the analysis variable names. If you do not include the variable _VAR_, all of the lines are displayed in all of the plots.

Each observation in an HREF= data set corresponds to a reference line. If BY variables are used in the input data set, the same BY variable structure must be used in the reference line data set unless you specify the NOBYREF option.

Unless the CONTINUOUS or HAXIS= option is specified, numeric group variable values are treated as discrete values, and only HREF= values matching these discrete values are valid. Other values are ignored.

HREFLABELS='*label*$_1$' ... '*label*$_n$'
HREFLABEL='*label*$_1$' ... '*label*$_n$'
HREFLAB='*label*$_1$' ... '*label*$_n$'

specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of the HREFLABEL= label, as described in the following table. By default, n=2.

HREFLABPOS=	Label Position
1	along top of plot area
2	staggered from top to bottom of plot area
3	along bottom of plot area
4	staggered from bottom to top of plot area

HTML=*variable*

specifies uniform resource locators (URLs) as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with box-and-whisker plots when graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the group variable.

IDCOLOR=*color*

specifies the color of the symbol marker used to identify outliers in schematic box-and-whisker plots (that is, when you specify one of the keywords SCHEMATIC, SCHEMATICID, or SCHEMATICIDFAR with the BOXSTYLE= option). The default color is the color specified with the CBOXES= option; otherwise, the second color in the device color list is used.

IDCTEXT=*color*

specifies the color for the text used to label outliers when you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default value is the color specified with the CTEXT= option.

IDFONT=*font*

specifies the font for the text used to label outliers when you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default font is SIMPLEX.

IDHEIGHT=*value*

specifies the height for the text used to label outliers when you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default value is the height specified with the HTEXT= option in the GOPTIONS statement. Refer to *SAS/GRAPH Software: Reference* for complete information on the GOPTIONS statement.

IDSYMBOL=*symbol*

specifies the symbol marker used to identify outliers in schematic box plots. The default symbol is SQUARE.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive group positions when a time, date, or datetime format is associated with a numeric group variable. By default, the INTERVAL= option uses the number of group positions per panel (screen or page) that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in the following table.

Format	**Default Keyword**	**Format**	**Default Keyword**
DATE	DAY	MONYY	MONTH
DATETIME	DTDAY	TIME	SECOND
DDMMYY	DAY	TOD	SECOND
HHMM	HOUR	WEEKDATE	DAY
HOUR	HOUR	WORDDATE	DAY
MMDDYY	DAY	YYMMDD	DAY
MMSS	MINUTE	YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of group positions per panel. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the group variable without having to associate a different format with the group variable.

For example, suppose that your formatted group values span an overall time interval of 100 days and a DATETIME format is associated with the group variable. Since the default interval for the DATETIME format is DTDAY and since NPANELPOS=25 by default, the plot is displayed with four panels.

Now, suppose that your data span an overall time interval of 100 hours and a DATETIME format is associated with the group variable. The plot for these data is created in a single panel, but the data occupy only a small fraction of the plot since the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 25 hours, matching the scale of the data, and the plot is displayed with four panels.

You should use the INTERVAL= option only in conjunction with the CONTINUOUS or HAXIS= option, which produces a horizontal axis of continuous group variable values. For more information, see the descriptions of the CONTINUOUS and HAXIS= options, and the discussion in the section "Continuous Group Variables" on page 524.

INTSTART=*value*

specifies the starting value for a numeric horizontal axis when a date, time, or datetime format is associated with the group variable. If the value specified is greater than the first group variable value, this option has no effect.

LABELANGLE=*angle*

specifies the angle at which labels requested with the ALLLABEL= option are drawn. A positive angle rotates the labels counterclockwise; a negative angle rotates them clockwise. By default, labels are oriented horizontally.

LBOXES=*linetype*
LBOXES=*(variable)*

specifies the line types for the outlines of the box-and-whisker plots. You can use one of the following approaches:

- You can specify LBOXES=*linetype* to provide a single linetype for all of the box-and-whisker plots.
- You can specify LBOXES=*(variable)* to provide a distinct line type for each box-and-whisker plot. The variable must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH linetype values (numbers ranging from 1 to 46). The line type for the plot displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that if there are multiple observations per group in the input data set, the values of the variable should be identical for all of the observations in a given group.

The default value is 1, which produces solid lines. Refer to the description of the SYMBOL statement in *SAS/GRAPH Software: Reference* for more information on valid linetypes.

LENDGRID=*n*

specifies the line type for the grid requested with the ENDGRID option. The default value is *n=1*, which produces a solid line. If you use the LENDGRID= option, you do not need to specify the ENDGRID option. Refer to the description of the SYMBOL statement in *SAS/GRAPH Software: Reference* for more information on valid linetypes.

LGRID=*n*

specifies the line type for the grid requested with the GRID option. The default value is *n=1*, which produces a solid line. If you use the LGRID= option, you do not need to specify the GRID option. Refer to the description of the SYMBOL statement in *SAS/GRAPH Software: Reference* for more information on valid linetypes.

LHREF=*linetype*
LH=*linetype*

specifies the line type for reference lines requested with the HREF= option. The default value is 2, which produces a dashed line. Refer to the description of the SYMBOL statement in *SAS/GRAPH Software: Reference* for more information on valid linetypes.

LOVERLAY=*(linetypes)*

specifies line types for the line segments connecting points on overlay plots. Line types in the LOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

LVREF=*linetype*
LV=*linetype*

specifies the line type for reference lines requested by the VREF= option. The default value is 2, which produces a dashed line. Refer to the description of the SYMBOL statement in *SAS/GRAPH Software: Reference* for more information on valid linetypes.

MAXPANELS=*n*

specifies the maximum number of panels (pages or screens) for a plot. By default, $n = 20$.

MISSBREAK

determines how groups are formed when observations are read from a DATA= data set and a character group variable is provided. When you specify the MISSBREAK option, observations with missing values of the group variable are not processed. Furthermore, the next observation with a nonmissing value of the group variable is treated as the beginning observation of a new group even if this value is identical to the most recent nonmissing group value. In other words, by specifying the option MISSBREAK and by inserting an observation with a missing group variable value into a group of consecutive observations with the same group variable value, you can split the group into two distinct groups of observations.

By default, (that is, when you omit the MISSBREAK option), observations with missing values of the group variable are not processed, and all remaining observations with the same consecutive value of the group variable are treated as a single group.

NAME=*'string'*

specifies a name for the box plot, not more than eight characters, that appears in the PROC GREPLAY master menu.

NLEGEND

requests a legend displaying group sizes. If the size is the same for each group, that number is displayed. Otherwise, the minimum and maximum group sizes are displayed.

NOBYREF

specifies that the reference line information in an HREF= or VREF= data set is to be applied uniformly to box plots created for all the BY groups in the input data set. If you specify the NOBYREF option, you do not need to provide BY variables in the reference line data set. By default, you must provide BY variables.

NOCHART

suppresses the creation of the chart. You typically specify the NOCHART option when you are using the procedure to compute group summary statistics and save them in an output data set.

NOFRAME

suppresses the default frame drawn around the plot.

NOHLABEL

suppresses the label for the horizontal (group) axis. Use the NOHLABEL option when the meaning of the axis is evident from the tick mark labels, such as when a date format is associated with the group variable.

NOOVERLAYLEGEND

suppresses the legend for overlay plots that is displayed by default when the OVERLAY= option is specified.

NOSERIFS

eliminates serifs from the whiskers of box-and-whisker plots.

NOTCHES

specifies that box-and-whisker plots are to be notched. The endpoints of the notches are located at the median plus and minus $1.58(\mathrm{IQR}/\sqrt{n})$, where IQR is the interquartile range and n is the group size. The medians (central lines) of two box-and-whisker plots are significantly different at approximately the 0.05 level if the corresponding notches do not overlap. Refer to McGill, Tukey, and Larsen (1978) for more information. Figure 18.4 illustrates the NOTCHES option. Notice the folding effect at the bottom, which happens when the endpoint of a notch is beyond its corresponding quartile. This situation typically occurs when the group size is small.

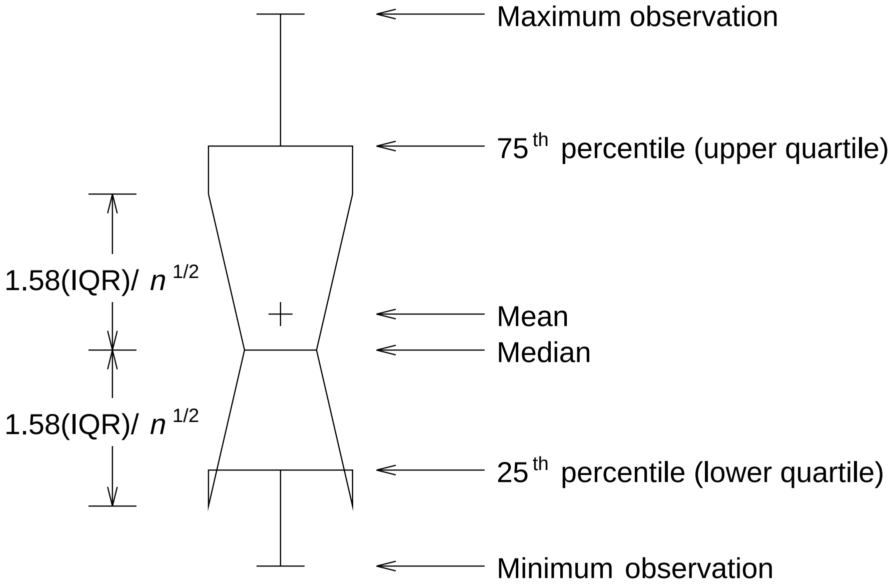


Figure 18.4. Box Plot: the NOTCHES Option

NOTICKREP

applies to character-valued group variables and specifies that only the first occurrence of repeated, adjacent group values is to be labeled on the horizontal axis.

NOVANGLE

requests vertical axis labels that are oriented vertically. By default, the labels are drawn at an angle of 90 degrees if a software font is used.

NPANELPOS=*n*
NPANEL=*n*

specifies the number of group positions per panel. A panel is defined as a screen or page. You typically specify the NPANELPOS= option to display more box-and-whisker plots on a panel than the default number, which is $n = 25$.

You can specify a positive or negative number for n. The absolute value of n must be at least 5. If n is positive, the number of positions is adjusted so that it is approximately equal to n and so that all panels display approximately the same number of group positions. If n is negative, no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions. In this case, the approximation is due only to axis scaling.

You can use the INTERVAL= option to change the effect of the NPANELPOS= option when a date or time format is associated with the group variable. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the group variable without having to associate a different format with the group variable.

OUTBOX=*SAS-data-set*

creates an output data set that contains group summary statistics and outlier values for a box plot. You can use an OUTBOX= data set as a BOX= input data set in a subsequent run of the procedure. See "OUTBOX= Data Set" for details.

OUTHIGHHTML=*variable*

specifies a variable whose values are URLs to be associated with outlier points above the upper fence on a schematic box plot when graphics output is directed into HTML.

OUTHISTORY=*SAS-data-set*

creates an output data set that contains the group summary statistics. You can use an OUTHISTORY= data set as a HISTORY= input data set in a subsequent run of the procedure. See "OUTHISTORY= Data Set" for details.

OUTLOWHTML=*variable*

specifies a variable whose values are URLs to be associated with outlier points below the lower fence on a schematic box plot when graphics output is directed into HTML.

OVERLAY=*(variable-list)*

specifies variables to be plotted as overlays on the box plot. One value for each overlay variable is plotted at each group position. If there are multiple observations with the same group variable value in the input data set, the overlay variable values from the first observation in each group are plotted. By default, the points in an overlay plot are connected with line segments.

OVERLAYCLIPSYM=*symbol*

specifies the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYCLIPSYMHT=*value*

specifies the height for the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYHTML=*(variable-list)*

specifies variables whose values are URLs to be associated with points on overlay plots when graphics output is directed into HTML. Variables in the OVERLAYHTML= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYID=*(variable-list)*

specifies variables whose formatted values are used to label points on overlays. Variables in the OVERLAYID= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYID= variable should be the same for each observation with a given value of the group variable.

OVERLAYLEGLAB=*'label'*

specifies the label displayed to the left of the overlay legend produced by the OVERLAY= option. The label can be up to 16 characters and must be enclosed in quotes. The default label is "Overlays:".

OVERLAYSYM=*(symbol-list)*

specifies symbols used to plot overlay variables. Symbols in the OVERLAYSYM= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYSYMHT=*(value-list)*

specifies the heights of symbols used to plot overlay variables. Symbol heights in the OVERLAYSYMHT= list are matched with variables in the corresponding positions in the OVERLAY= list.

PAGENUM=*'string'*

specifies the form of the label used for pagination. The string must be no longer than 16 characters, and it must include one or two occurrences of the substitution character '#'. The first '#' is replaced with the page number, and the optional second '#' is replaced with the total number of pages.

The PAGENUM= option is useful when you are working with a large number of groups, resulting in multiple pages of output. For example, suppose that each of the following PLOT statements produces multiple pages:

```
proc boxplot data=pistons;
   plot diameter*hour / pagenum='Page #';
   plot diameter*hour / pagenum='Page # of #';
   plot diameter*hour / pagenum='#/#';
run;
```

The third page produced by the first statement would be labeled *Page 3*. The third page produced by the second statement would be labeled *Page 3 of 5*. The third page produced by the third statement would be labeled *3/5*.

By default, no page number is displayed.

PAGENUMPOS=TL | TR | BL | BR | TL100 | TR100 | BL0 | BR0

specifies where to position the page number requested with the PAGENUM= option. The keywords TL, TR, BL, and BR correspond to the positions top left, top right, bottom left, and bottom right, respectively. You can use the TL100 and TR100 keywords to ensure that the page number appears at the very top of a page when a title is displayed. The BL0 and BR0 keywords ensure that the page number appears at the very bottom of a page when footnotes are displayed.

The default keyword is BR.

PCTLDEF=*index*

specifies one of five definitions used to calculate percentiles in the construction of box-and-whisker plots. The index can be 1, 2, 3, 4, or 5. The five corresponding percentile definitions are discussed in the section "Percentile Definitions" on page 523. The default index is 5.

REPEAT
REP

specifies that the horizontal axis of a plot that spans multiple pages is to be arranged so that the last group position on a page is repeated as the first group position on the next page. The REPEAT option facilitates cutting and pasting panels together. When a SAS DATETIME format is associated with the group variable, the REPEAT option is the default.

SKIPHLABELS=*n*
SKIPHLABEL=*n*

specifies the number n of consecutive tick mark labels, beginning with the second tick mark label, that are thinned (not displayed) on the horizontal (group) axis. For example, specifying SKIPHLABEL=1 causes every other label to be skipped. Specifying SKIPHLABEL=2 causes the second and third labels to be skipped, the fifth and sixth labels to be skipped, and so forth.

The default value of the SKIPHLABELS= option is the smallest value n for which tick mark labels do not collide. A specified n will be overridden to avoid collision. To reduce thinning, you can use the TURNHLABELS option.

SYMBOLLEGEND=LEGEND*n*
SYMBOLLEGEND=NONE

controls the legend for the levels of a symbol variable (see Example 18.1). You can specify SYMBOLLEGEND=LEGEND*n*, where *n* is the number of a LEGEND statement defined previously. You can specify SYMBOLLEGEND=NONE to suppress the default legend. Refer to *SAS/GRAPH Software: Reference* for more information on the LEGEND statement.

SYMBOLORDER=DATA | INTERNAL | FORMATTED
SYMORD=DATA | INTERNAL | FORMATTED

specifies the order in which symbols are assigned for levels of the symbol variable. The DATA keyword assigns symbols to values in the order in which values appear in the input data. The INTERNAL keyword assigns symbols based on sorted order of internal values of the symbol variable, and the FORMATTED keyword assigns them based on sorted formatted values. The default value is FORMATTED.

TOTPANELS=*n*

specifies the total number of panels (pages or screens) to be used to display the plot. This option overrides the NPANEL= option.

TURNHLABELS
TURNHLABEL

turns the major tick mark labels for the horizontal (group) axis so that they are arranged vertically. By default, labels are arranged horizontally. You should specify a software font (using the FONT= option) in conjunction with the TURNHLABELS option. Otherwise, the labels may be displayed with a mixture of hardware and software fonts.

Note that arranging the labels vertically may leave insufficient vertical space on the panel for a plot.

VAXIS=*value-list*
VAXIS=AXIS*n*

specifies major tick mark values for the vertical axis of a box plot. The values must be listed in increasing order, must be evenly spaced, and must span the range of values displayed on the plot. You can specify the values with an explicit list or with an implicit list, as shown in the following example:

```
proc boxplot;
   plot width*hour / vaxis=0 2 4 6 8;
   plot width*hour / vaxis=0 to 8 by 2;
run;
```

You can also specify a previously defined AXIS statement with the VAXIS= option.

VFORMAT=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of the box plot.

VMINOR=*n*
VM=*n*

specifies the number of minor tick marks between each major tick mark on the vertical axis. Minor tick marks are not labeled. By default, VMINOR=0.

VOFFSET=*value*

specifies the length in percent screen units of the offset at the ends of the vertical axis.

VREF=*value-list*
VREF=*SAS-data-set*

draws reference lines perpendicular to the vertical axis on the box plot. You can use this option in the following ways:

- Specify the values for the lines with a VREF= list. Examples of the VREF= option follow:

  ```
  vref=20
  vref=20 40 80
  ```

- Specify the values for the lines as the values of a numeric variable named _REF_ in a VREF= data set. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of length 16 or less. If you want distinct reference lines to be displayed in plots for different analysis variables specified in the PLOT statement, you must include a character variable named _VAR_, whose values are the names of the analysis variables. If you do not include the variable _VAR_, all of the lines are displayed in all of the plots.

  Each observation in the VREF= data set corresponds to a reference line. If BY variables are used in the input data set, the same BY variable structure must be used in the VREF= data set unless you specify the NOBYREF option.

VREFLABELS=*'label1' ... 'labeln'*

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of the VREFLABEL= label, as described in the following table. By default, *n=1*.

n	Label Position
1	left-justified in plot area
2	right-justified in plot area
3	left-justified in right margin

VZERO

forces the origin to be included in the vertical axis for a box plot.

WAXIS=*n*

specifies the width in pixels for the axis and frame lines. By default, *n=1*.

WGRID=*n*

specifies the width in pixels for grid lines requested with the ENDGRID and GRID options. By default, *n=1*.

WOVERLAY=*(value-list)*

specifies the widths in pixels for the line segments connecting points on overlay plots. Widths in the WOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, all overlay widths are 1.

INSET Statement

INSET *keywords < / options >;*

You can use any number of INSET statements in the BOXPLOT procedure. Each INSET statement produces one inset and must follow a PLOT statement. The inset appears in all panels produced by the last PLOT statement preceding it. The data requested using the *keywords* are displayed in the order in which they are specified. Summary statistics requested with an INSET statement are calculated using the observations in all groups.

keywords identify summary statistics or other data to be displayed in the inset. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format.

The keywords are listed in Table 18.2.

options control the appearance of the inset. Table 18.3 lists all the options in the INSET statement. Complete descriptions for each option follow.

Table 18.2. INSET Statement Keywords

DATA=	(label, value) pairs from *SAS-data-set*
MEAN	mean of all observations
MIN	minimum observed value
MAX	maximum observed value
NMIN	minimum group size
NMAX	maximum group size
NOBS	number of observations in box plot
STDDEV	pooled standard deviation

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable of length 24 or less whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset.

The *pooled standard deviation* requested with the STDDEV keyword is defined as

$$s_p = \sqrt{\frac{\sum_{i=1}^{N} s_i^2(n_i - 1)}{\sum_{i=1}^{N} (n_i - 1)}}$$

where N is the number of groups, n_i is the size of the ith group, and s_i^2 is the variance of the ith group.

Table 18.3. INSET Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CSHADOW=*color*	specifies color of drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER=*'quoted string'*	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Following are descriptions of the options that you can specify in the INSET statement after a slash (/).

CFILL=*color* **| BLANK**

specifies the color of the inset background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default, the background is empty. This means that items that overlap the inset (such as box-and-whisker plots or reference lines) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*

specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*

specifies the color of the frame around the inset. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*

specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CSHADOW=*color*
CS=*color*

specifies the color of the drop shadow. If you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*
CT=*color*

specifies the color of the text in the inset. By default, the inset text color is the same as the other text on the box plot.

DATA

specifies that data coordinates are to be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). See the entry for the POSITION= option.

FONT=*font*

specifies the font of the text. By default, the font is SIMPLEX if the inset is located in the interior of the plot, and the font is the same as the other text displayed on the plot if the inset is located in the exterior of the plot.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

specifies the height of the text.

NOFRAME

suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*

determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see "Positioning Insets" on page 526. By default, POSITION=NW, which positions the inset in the upper left (northwest) corner of the plot.

REFPOINT=BR | BL | TR | TL
RP=BR | BL | TR | TL

specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored.

INSETGROUP Statement

INSETGROUP *keywords* < */ options* >;

The INSETGROUP statement displays statistics associated with individual groups on the box plot produced by the last PLOT statement preceding it. No more than two INSETGROUP statements can be associated with a given PLOT statement: one above the box plot and one below it. The data requested using the *keywords* are displayed in the order in which they are specified.

keywords identify summary statistics to be displayed in the insets. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. The keywords are listed in Table 18.4.

options control the appearance of the insets. Table 18.5 lists all the options in the INSETGROUP statement. Complete descriptions for each option follow.

Table 18.4. INSETGROUP Statement Keywords

MEAN	group mean
MIN	minimum value in group
MAX	maximum value in group
N	number of observations in group
NHIGH	number of outliers above upper fence
NLOW	number of outliers below lower fence
NOUT	total number of outliers
Q1	first quartile
Q2	second quartile
Q3	third quartile
RANGE	range of group values
STDDEV	group standard deviation

Table 18.5 lists all options in the INSETGROUP statement.

Table 18.5. INSETGROUP Options

CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of header background
CFRAME=*color*	specifies color of frame
CHEADER=*color*	specifies color of header text
CTEXT=*color*	specifies color of inset text
FONT=*font*	specifies font of text
FORMAT=*format*	specifies format of values in inset
HEADER=*’quoted string’*	specifies header text
HEIGHT=*value*	specifies height of inset text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset

Following are descriptions of the options that you can specify in the INSETGROUP statement after a slash (/).

CFILL=*color*

specifies the color of the inset background (including the header background if you do not specify the CFILLH= option). If you do not specify the CFILL= option, then by default, the background is empty.

CFILLH=*color*

specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*

specifies the color of the frame around the inset. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*

specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CTEXT=*color*
CT=*color*

specifies the color of the inset text. By default, the inset text color is the same as the other text on the plot.

FONT=*font*

specifies the font of the inset text. By default, the font is SIMPLEX.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER= *'string'*

specifies the header text. The *string* cannot exceed 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

specifies the height of the text.

NOFRAME

suppresses the frame drawn around the text.

POSITION=*position*
POS=*position*

determines the position of the inset. Valid positions are TOP, TOPOFF, AXIS, and BOTTOM. By default, POSITION=TOP.

Position Keyword	Description
TOP	top of plot, immediately above axis frame
TOPOFF	top of plot, offset from axis frame
AXIS	bottom of plot, immediately above horizontal axis
BOTTOM	bottom of plot, below horizontal axis label

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC BOXPLOT to obtain separate box plots for each group defined by the levels of the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the BOXPLOT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

ID Statement

ID *variables* **;**

The ID statement specifies variables used to identify observations. The ID variables must be variables in the input data set.

If you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, the value of an ID variable is used to label each extreme observation. When you specify a BOX= data set, the label values come from the variable _ID_, if it is present in the data set. When you specify a DATA= or HISTORY= input data set, or a BOX= data set that does not contain the variable _ID_, the labels come from the first variable listed in the ID statement. If there is no ID statement, the outliers are not labeled.

Details

Summary Statistics Represented by Box Plots

Table 18.6 lists the summary statistics represented in each box-and-whisker plot.

Table 18.6. Summary Statistics Represented by Box Plots

Group Summary Statistic	Feature of Box-and-Whisker Plot
Maximum	Endpoint of upper whisker
Third quartile (75th percentile)	Upper edge of box
Median (50th percentile)	Line inside box
Mean	Symbol marker
First quartile (25th percentile)	Lower edge of box
Minimum	Endpoint of lower whisker

Note that you can request different box plot styles, as discussed in the section “Styles of Box Plots” on page 522, and as illustrated in Example 18.2 on page 538.

Output Data Sets

OUTBOX= Data Set

The OUTBOX= data set saves group summary statistics and outlier values. The following variables can be saved:

- the group variable
- the variable _VAR_, containing the analysis variable name
- the variable _TYPE_, identifying features of box-and-whisker plots
- the variable _VALUE_, containing values of box-and-whisker plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs associated with plot features

_ID_ is included in the OUTBOX= data set only if one of the keywords SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, or OUTLOWHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. The following table lists valid _TYPE_ variable values:

Table 18.7. Valid _TYPE_ Values in an OUTBOX= Data Set

TYPE Value	Description
N	group size
MIN	minimum group value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

Additionally, the following variables, if specified, are included:

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

OUTHISTORY= Data Set

The OUTHISTORY= data set saves group summary statistics. The following variables are saved:

- the group variable
- group minimum variables named by *analysis-variable* suffixed with *L*
- group first-quartile variables named by *analysis-variable* suffixed with *1*
- group mean variables named by *analysis-variable* suffixed with *X*
- group median variables named by *analysis-variable* suffixed with *M*
- group third-quartile variables named by *analysis-variable* suffixed with *3*
- group maximum variables named by *analysis-variable* suffixed with *H*
- group size variables named by *analysis-variable* suffixed with *N*

Subgroup summary variables are created for each *analysis-variable* specified in the PLOT statement. For example, consider the following statements:

```
proc boxplot data=steel;
   plot (width diameter)*lot / outhistory=summary;
run;
```

The data set SUMMARY contains variables named LOT, WIDTHL, WIDTH1, WIDTHM, WIDTHX, WIDTH3, WIDTHH, WIDTHN, DIAMTERL, DIAMTER1, DIAMTERM, DIAMTERX, DIAMTER3, DIAMTERH, and DIAMTERN.

Given an analysis variable name that contains the maximum of 32 characters, the procedure first shortens the name to its first 16 characters and its last 15 characters, and then it adds the suffix.

Additionally, the following variables, if specified, are included:

- BY variables
- *block-variables*
- *symbol-variable*
- ID variables

Note that an OUTHISTORY= data set does not contain outlier values, and therefore cannot be used, in general, to save a schematic box plot. You can use an OUTBOX= data set to save a schematic box plot summary.

Input Data Sets

DATA= Data Set

You can read data (analysis variable measurements) from a data set specified with the DATA= option in the PROC BOXPLOT statement. Each analysis variable specified in the PLOT statement must be a SAS variable in the data set. This variable provides measurements that are organized into groups indexed by the group variable. The group variable, specified in the PLOT statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each analysis variable and a value for the group variable. If the ith group contains n_i measurements, there should be n_i consecutive observations for which the value of the group variable is the index of the ith group. For example, if each group contains 20 items and there are 30 groups, the DATA= data set should contain 600 observations. Other variables that can be read from a DATA= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

BOX= Data Set

You can read group summary statistics and outlier information from a BOX= data set specified in the PROC BOXPLOT statement. This allows you to reuse OUTBOX= data sets that have been created in previous runs of the BOXPLOT procedure to reproduce schematic box plots.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the analysis variable name
- _TYPE_, identifying features of box-and-whisker plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one group's box-and-whisker plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. The following table lists valid _TYPE_ variable values:

Table 18.8. Valid _TYPE_ Values in a BOX= Data Set

TYPE Value	Description
N	group size
MIN	minimum group value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

The features identified by _TYPE_ values N, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each group.

Other variables that can be read from a BOX= data set include:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs to be associated with features on box plots
- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

When you specify one of the keywords SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

HISTORY= Data Set

You can read group summary statistics from a HISTORY= data set specified in the PROC BOXPLOT statement. This allows you to reuse OUTHISTORY= data sets that have been created in previous runs of the BOXPLOT procedure or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

Note that a HISTORY= data set does *not* contain outlier information. Therefore, in general you cannot reproduce a schematic box plot from summary statistics saved in an OUTHISTORY= data set. To save and reproduce schematic box plots, use OUTBOX= and BOX= data sets.

A HISTORY= data set must contain the following:

- the group variable
- a group minimum variable for each analysis variable
- a group first-quartile variable for each analysis variable
- a group median variable for each analysis variable
- a group mean variable for each analysis variable
- a group third-quartile variable for each analysis variable
- a group maximum variable for each analysis variable
- a group size variable for each analysis variable

The names of the group summary statistics variables must be the analysis variable name concatenated with the following special suffix characters:

Group Summary Statistic	Suffix Character
group minimum	L
group first-quartile	1
group median	M
group mean	X
group third-quartile	3
group maximum	H
group size	N

For example, consider the following statements:

```
proc boxplot history=summary;
   plot (weight yldstren) * batch;
run;
```

The data set SUMMARY must include the variables BATCH, WEIGHTL, WEIGHT1, WEIGHTM, WEIGHTX, WEIGHT3, WEIGHTH, WEIGHTN, YLDSRENL, YLDSREN1, YLDSRENM, YLDSRENX, YLDSREN3, YLDSRENH, and YLDSRENN.

Note that if you specify an analysis variable name that contains 32 characters, the names of the summary variables must be formed from the first 16 characters and the last 15 characters of the analysis variable name, suffixed with the appropriate character.

Other variables that can be read from a HISTORY= data set include

- *block-variables*
- *symbol-variable*
- BY variables
- ID variables

Styles of Box Plots

A box-and-whisker plot is displayed for the measurements in each group on the box plot. The skeletal style of the box-and-whisker plot shown in Figure 18.3 is the default. Figure 18.5 illustrates a typical schematic box plot and the locations of the fences (which are not displayed in actual output). See the description of the BOXSTYLE= option on page 493 for complete details.

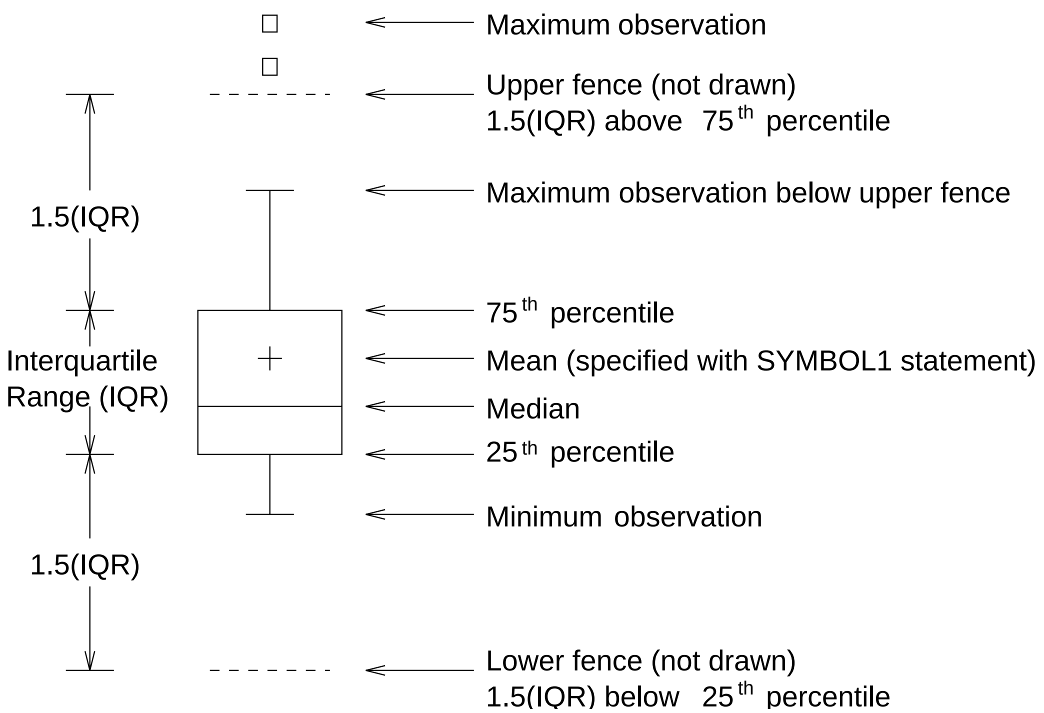


Figure 18.5. BOXSTYLE= SCHEMATIC

You can draw connecting lines between adjacent box-and-whisker plots using the BOXCONNECT=*keyword* option. For example, BOXCONNECT=MEAN connects the points representing the means of adjacent groups. Other available keywords are MIN, Q1, MEDIAN, Q3, and MAX. Specifying BOXCONNECT without a keyword is equivalent to specifying BOXCONNECT=MEAN. You can specify the color for the connecting lines with the CCONNECT= option.

Percentile Definitions

You can use the PCTLDEF= option to specify one of five definitions for computing quantile statistics (percentiles). Suppose that n equals the number of nonmissing values for a variable and that $x_1, x_2, \ldots, x_n$ represents the ordered values of the analysis variable. For the *t*th percentile, set $p = t/100$.

For the following definitions numbered 1, 2, 3, and 5, express np as

$$np = j + g$$

where j is the integer part of np, and g is the fractional part of np. For definition 4, let

$$(n+1)p = j + g$$

The *t*th percentile (call it y) can be defined as follows:

PCTLDEF=1 weighted average at x_{np}

$$y = (1-g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1

PCTLDEF=2 observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$ if $g \neq 1/2$. If $g = 1/2$, then $y = x_j$ if j is even, or $y = x_{j+1}$ if j is odd.

PCTLDEF=3 empirical distribution function

$$y = x_j \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

PCTLDEF=4 weighted average aimed at $x_{p(n+1)}$

$$y = (1-g)x_j + gx_{j+1}$$

where x_{n+1} is taken to be x_n

PCTLDEF=5 empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

Missing Values

An observation read from a DATA= data set is not analyzed if the value of the group variable is missing. For a particular analysis variable, an observation read from a DATA= data set is not analyzed if the value of the analysis variable is missing.

Missing values of analysis variables generally lead to unequal group sizes.

Continuous Group Variables

By default, the PLOT statement treats numerical group variable values as *discrete* values and spaces the boxes evenly on the plot. The following statements produce the plot shown in Figure 18.6:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
run;
```

The labels on the horizontal axis in Figure 18.6 do not represent 10 consecutive days, but the box-and-whisker plots are evenly spaced.

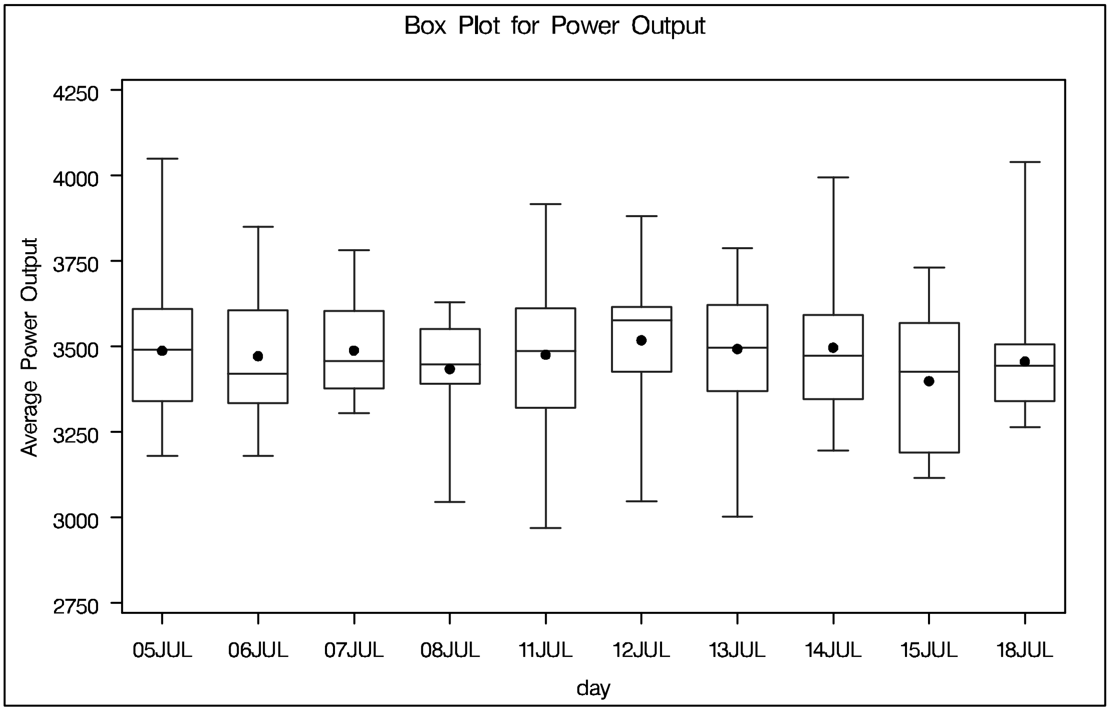


Figure 18.6. Box Plot with Discrete Group Variable

In order to treat the group variable as *continuous*, you can specify the CONTINUOUS or HAXIS= option. Either option produces a box plot with a horizontal axis scaled for continuous group variable values.

The following statements produce the plot shown in Figure 18.7. Note that the values on the horizontal axis represent consecutive days. (The TURNHLABEL option orients the horizontal axis labels vertically so there is room to display them all.) Box-and-whisker plots are not produced for days when no turbine data was collected.

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day / turnhlabel
                     continuous;
run;
```

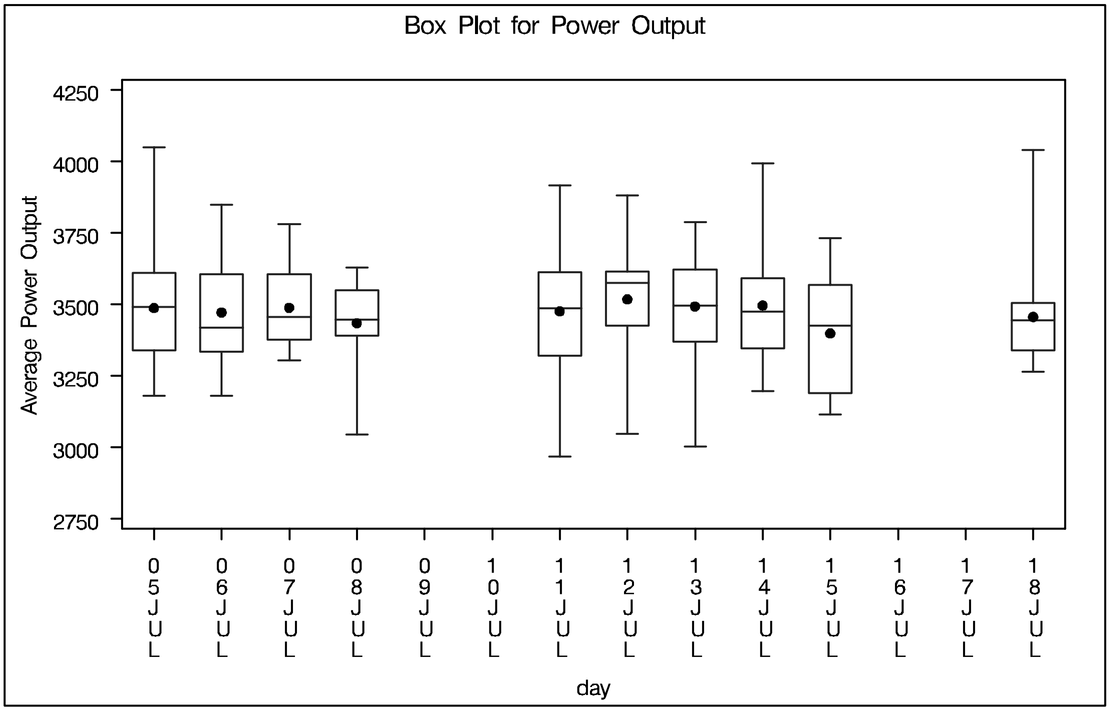


Figure 18.7. Box Plot with Continuous Group Variable

Positioning Insets

This section provides details on three different methods of positioning INSET boxes using the POSITION= option. With the POSITION= option, you can specify

- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points N, NE, E, SE, S, SW, W, and NW as keywords for the POSITION= option. The following statements create the display in Figure 18.8, which demonstrates all eight compass positions. The default is NW.

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
   inset nobs / height=3 cfill=blank header='NW' pos=nw;
   inset nobs / height=3 cfill=blank header='N ' pos=n ;
   inset nobs / height=3 cfill=blank header='NE' pos=ne;
   inset nobs / height=3 cfill=blank header='E ' pos=e ;
   inset nobs / height=3 cfill=blank header='SE' pos=se;
   inset nobs / height=3 cfill=blank header='S ' pos=s ;
   inset nobs / height=3 cfill=blank header='SW' pos=sw;
   inset nobs / height=3 cfill=blank header='W ' pos=w ;
run;
```

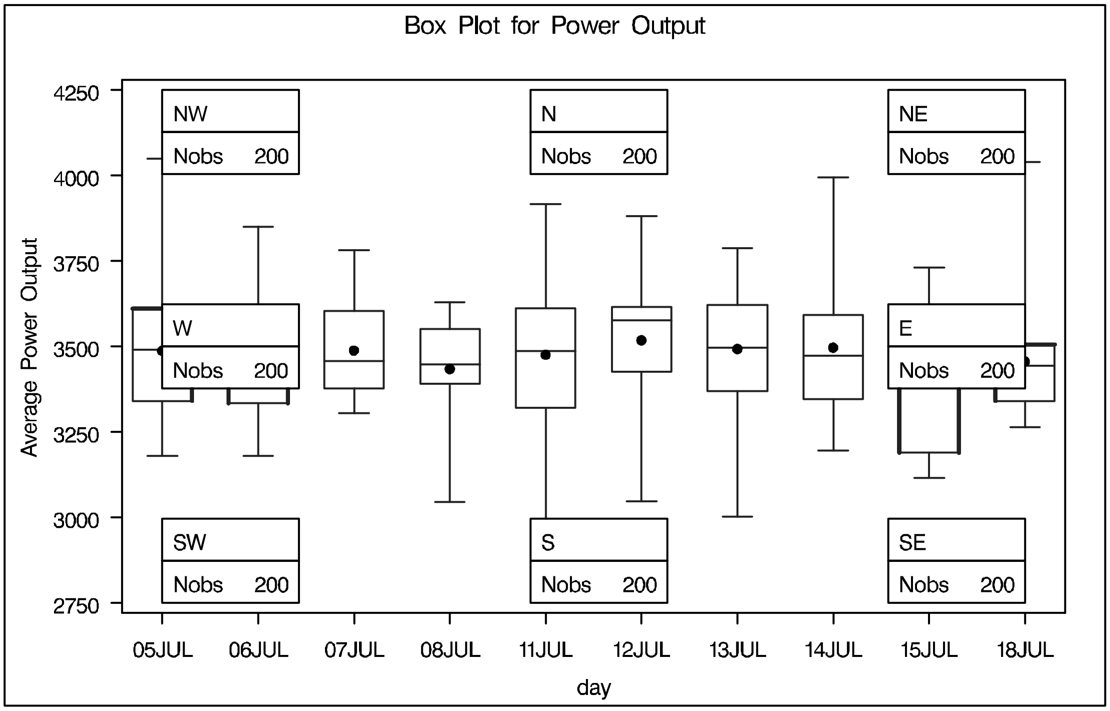


Figure 18.8. Insets Positioned Using Compass Points

Positioning the Inset in the Margins

Using the INSET statement you can also position an inset in one of the four margins surrounding the plot area using the margin keywords LM, RM, TM, or BM, as illustrated in Figure 18.9.

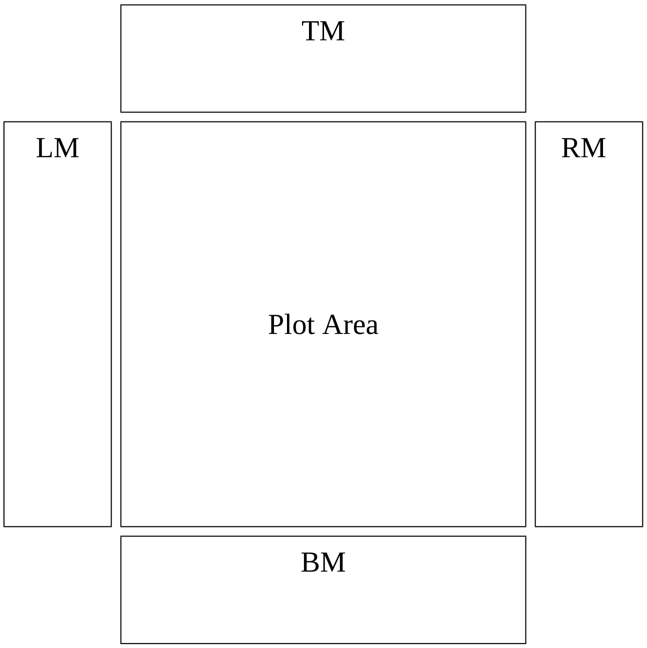


Figure 18.9. Positioning Insets in the Margins

For an example of an inset placed in the top margin, see Figure 18.2. Margin positions are recommended if a large number of statistics are listed in the INSET statement. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

You can also specify the position of the inset with coordinates: POSITION= (x, y). The coordinates can be given in axis percent units (the default) or in axis data units.

Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom left corner of the inset at 07JUL on the horizontal axis and 3950 on the vertical axis:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
   inset nobs /
      header   = 'Position=(07JUL,3950)'
      position = ('07JUL94'd, 3950) data;
run;
```

The box plot is displayed in Figure 18.10. By default, the specified coordinates determine the position of the bottom left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

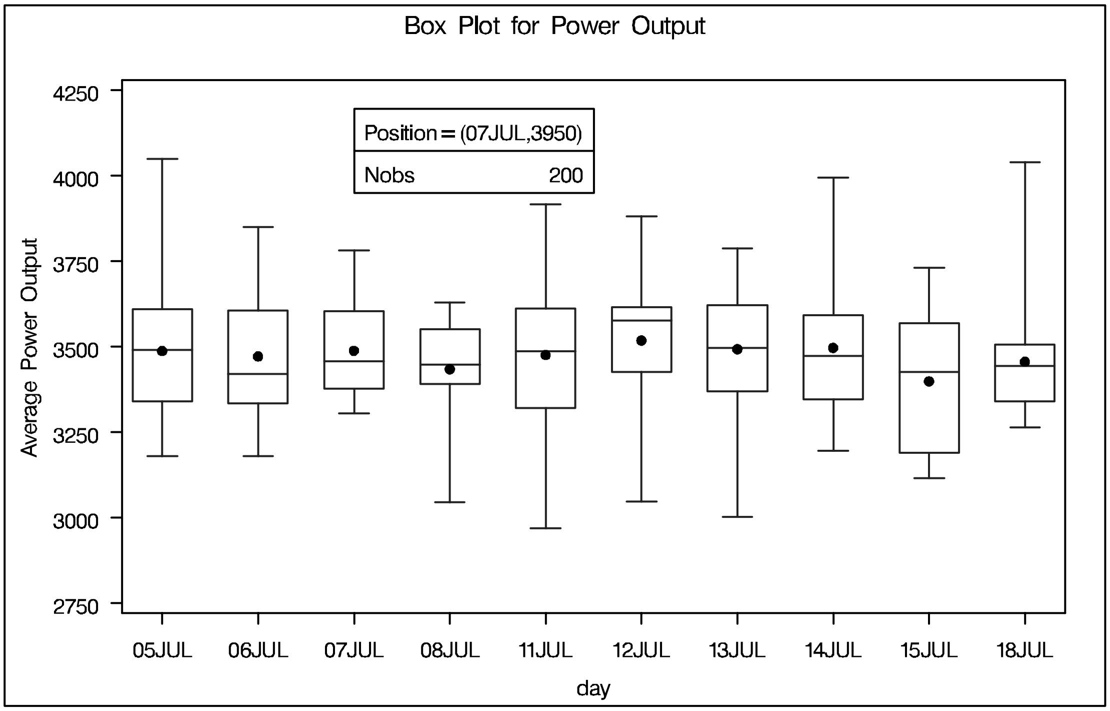


Figure 18.10. Inset Positioned Using Data Unit Coordinates

Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom left corner of the display are $(0, 0)$, while the upper right corner is $(100, 100)$. For example, the following statements create a box plot with two insets, both positioned using coordinates in axis percent units:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot kwatts*day;
   inset nmin / position = (5,25)
                header    = 'Position=(5,25)'
                height    = 3
                cfill     = blank
                refpoint = tl;
   inset nmax / position = (95,95)
                header    = 'Position=(95,95)'
                height    = 3
                cfill     = blank
                refpoint = tr;
run;
```

The display is shown in Figure 18.11. Notice that the REFPOINT= option is used to determine which corner of the inset is to be placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

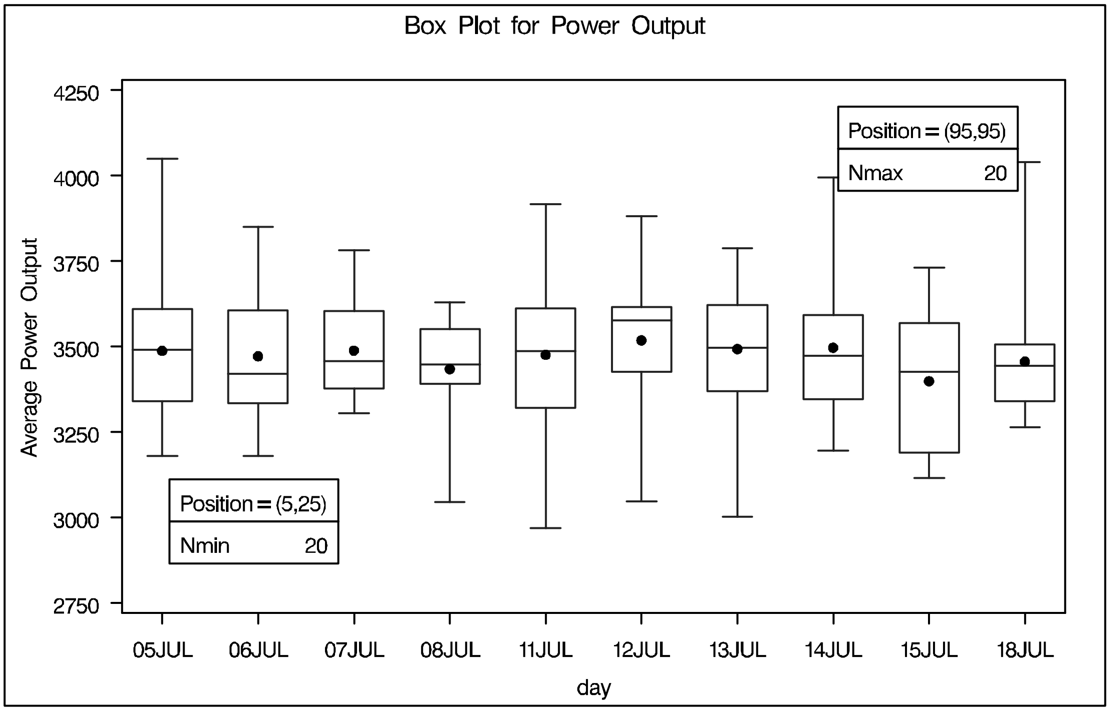


Figure 18.11. Inset Positioned Using Axis Percent Unit Coordinates

Displaying Blocks of Data

To display data organized in blocks of consecutive observations, specify one or more *block-variables* in parentheses after the *group-variable* in the PLOT statement. The block variables must be variables in the input data set. The procedure displays a legend identifying blocks of consecutive observations with identical values of the block variables. The legend displays one track of values for each block variable containing formatted values of the block variable.

The values of a block variable must be the same for all observations with the same value of the group variable. In other words, groups must be nested within blocks determined by block variables.

The following statements create a SAS data set containing diameter measurements for a part produced on three different machines:

```
data Parts;
   length machine $ 4;
   input sample machine $ @;
   do i= 1 to 4;
      input diam @;
      output;
   end;
   drop i;
 datalines;
 1  A386  4.32 4.55 4.16 4.44
 2  A386  4.49 4.30 4.52 4.61
 3  A386  4.44 4.32 4.25 4.50
 4  A386  4.55 4.15 4.42 4.49
 5  A386  4.21 4.30 4.29 4.63
 6  A386  4.56 4.61 4.29 4.56
 7  A386  4.63 4.30 4.41 4.58
 8  A386  4.38 4.65 4.43 4.44
 9  A386  4.12 4.49 4.30 4.36
10  A455  4.45 4.56 4.38 4.51
11  A455  4.62 4.67 4.70 4.58
12  A455  4.33 4.23 4.34 4.58
13  A455  4.29 4.38 4.28 4.41
14  A455  4.15 4.35 4.28 4.23
15  A455  4.21 4.30 4.32 4.38
16  C334  4.16 4.28 4.31 4.59
17  C334  4.14 4.18 4.08 4.21
18  C334  4.51 4.20 4.28 4.19
19  C334  4.10 4.33 4.37 4.47
20  C334  3.99 4.09 4.47 4.25
21  C334  4.24 4.54 4.43 4.38
22  C334  4.23 4.48 4.31 4.57
23  C334  4.27 4.40 4.32 4.56
24  C334  4.70 4.65 4.49 4.38
;
```

The following statements create a box plot for the data in the Parts data set grouped into blocks by the *block-variable* Machine. The plot is shown in Figure 18.12.

```
title 'Box Plot for Diameter Grouped By Machine';
proc boxplot data=Parts;
   plot diam*sample (machine);
   label sample  = 'Sample Number'
         machine = 'Machine'
         diam    = 'Diameter';
run;
```

The unique consecutive values of Machine (A386, A455, and C334) are displayed in a legend above the plot. Note the LABEL statement used to provide labels for the axes and for the block legend.

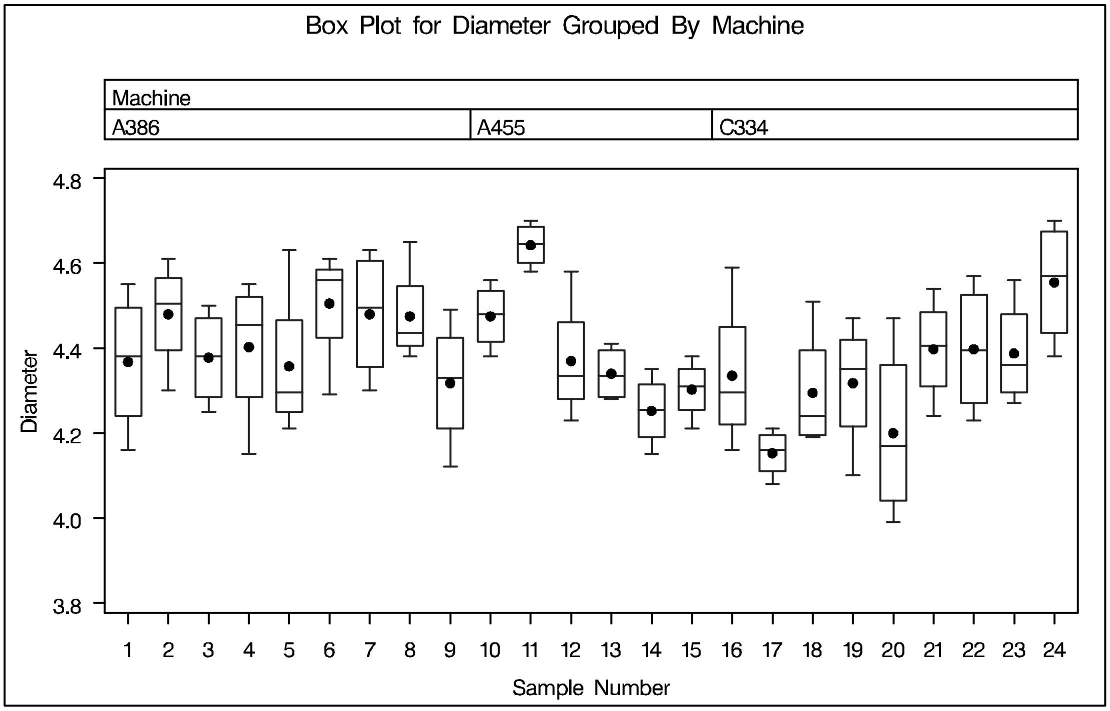


Figure 18.12. Box Plot Using a Block Variable

By default, the block legend is placed above the plot, as in Figure 18.12. You can control the position of the legend with the BLOCKPOS=*n* option; see the BLOCKPOS= option on page 493.

By default, block variable values that are too long to fit into the available space in a block legend are not displayed. You can specify the BLOCKLABTYPE= option to display lengthy labels. Specify BLOCKLABTYPE=SCALED to scale down the text size of the values so they all fit. Choose BLOCKLABTYPE=TRUNCATED to truncate lengthy values. You can also use BLOCKLABTYPE=*height* to specify a text height in vertical percent screen units for the values.

You can control the position of legend labels with the BLOCKLABELPOS=*keyword* option. The valid keywords are ABOVE (the default, as shown in Figure 18.12) and LEFT.

Clipping Extreme Values

By default a box plot's vertical axis is scaled to accommodate all the values in all groups. If the variation between groups is large with respect to the variation within groups, or if some groups contain extreme outlier values, the vertical axis scale can become so large that the box-and-whisker plots are compressed. In such cases, you can clip the extreme values so that a more readable plot is displayed, as illustrated in the following example.

A company produces copper tubing. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set NEWTUBES.

```
data newtubes;
   label diameter='Diameter in mm';
   do batch = 1 to 15;
      do  i = 1 to 5;
         input diameter @@;
         output;
      end;
   end;
datalines;
69.13  69.83  70.76  69.13  70.81
85.06  82.82  84.79  84.89  86.53
67.67  70.37  68.80  70.65  68.20
71.71  70.46  71.43  69.53  69.28
71.04  71.04  70.29  70.51  71.29
69.01  68.87  69.87  70.05  69.85
50.72  50.49  49.78  50.49  49.69
69.28  71.80  69.80  70.99  70.50
70.76  69.19  70.51  70.59  70.40
70.16  70.07  71.52  70.72  70.31
68.67  70.54  69.50  69.79  70.76
68.78  68.55  69.72  69.62  71.53
70.61  70.75  70.90  71.01  71.53
74.62  56.95  72.29  82.41  57.64
70.54  69.82  70.71  71.05  69.24
;
run;
```

The following statements create the box plot shown in Figure 18.13 for the tube diameter:

```
symbol value=plus;
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=newtubes;
   plot diameter*batch;
run;
```

Note that the diameters in batch 2 are significantly larger, and those in batch 7 significantly smaller, than those in most of the other batches. The default vertical axis scaling causes the box-and-whisker plots to be compressed.

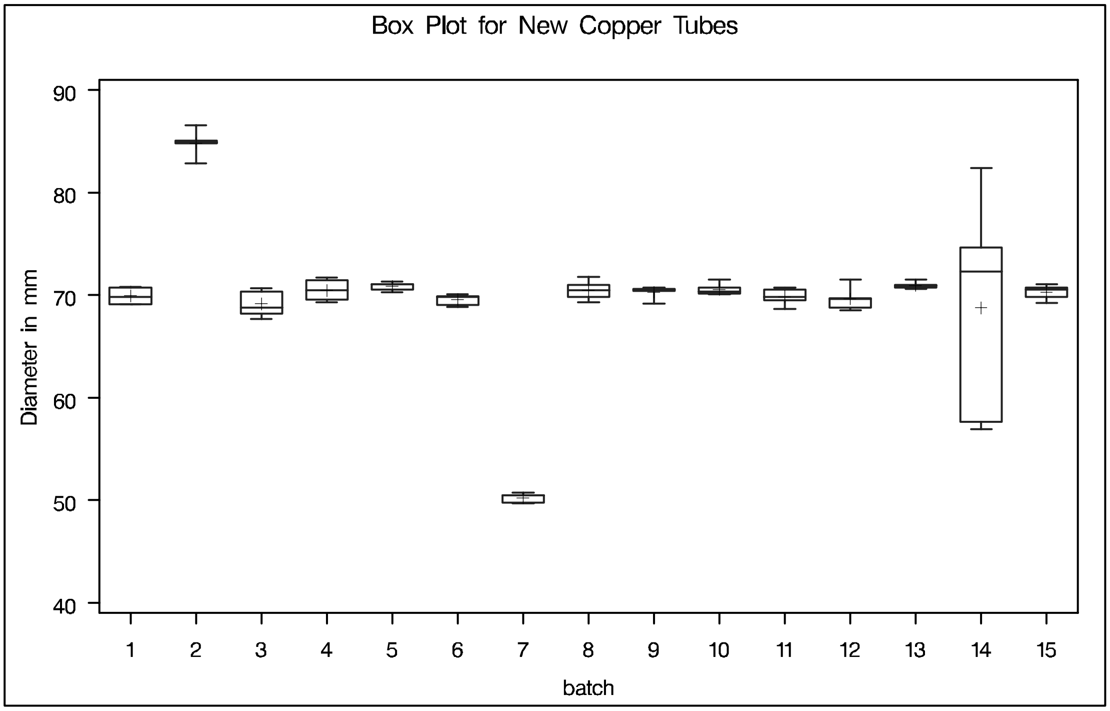


Figure 18.13. Compressed Box Plots

You can request clipping by specifying the CLIPFACTOR=*factor* option, where *factor* is a value greater than one. Clipping is applied as follows:

1. The mean of the first quartile values ($\overline{Q1}$) and the mean of the third quartile values ($\overline{Q3}$) are computed across all groups.
2. Any plotted statistic greater than y_{max} or less than y_{min} is ignored during vertical axis scaling, where

$$y_{\mathrm{max}} = \overline{Q1} + (\overline{Q3} - \overline{Q1}) \times \mathit{factor}$$

and

$$y_{\mathrm{min}} = \overline{Q3} - (\overline{Q3} - \overline{Q1}) \times \mathit{factor}$$

Notes:

- Clipping is applied only to the plotted statistics and not to the statistics saved in an output data set.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of boxes that were clipped.

The following statements create a box plot, shown in Figure 18.14, that use a clipping factor of 1.5:

```
symbol value=plus;
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=newtubes;
   plot diameter*batch /
      clipfactor = 1.5;
run;
```

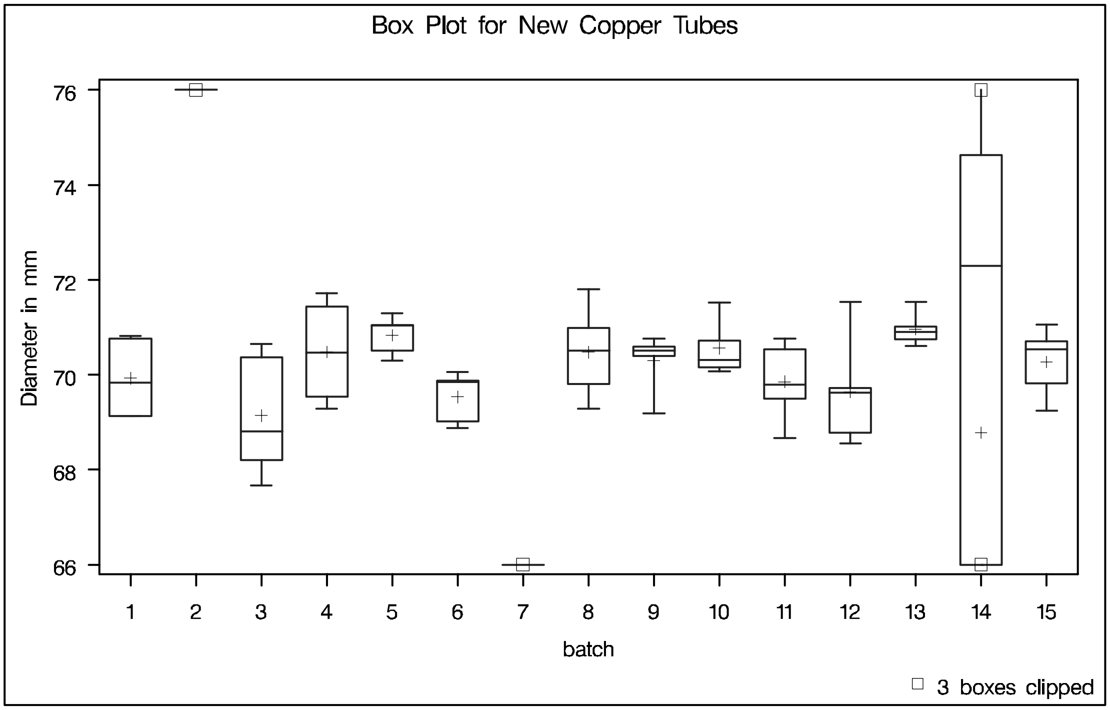


Figure 18.14. Box Plot with Clip Factor of 1.5

In Figure 18.14 the extreme values are clipped, making the remaining boxes more readable. The box-and-whisker plots for batches 2 and 7 are clipped completely, while batch 14 is clipped at both the top and bottom. Clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

Other clipping options are available, as illustrated by the following statements:

```
symbol value=plus;
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=newtubes;
   plot diameter*batch /
      clipfactor  = 1.5
      clipsymbol  = dot
      cliplegpos  = top
      cliplegend  = '# Clipped Boxes'
      clipsubchar = '#';
run;
```

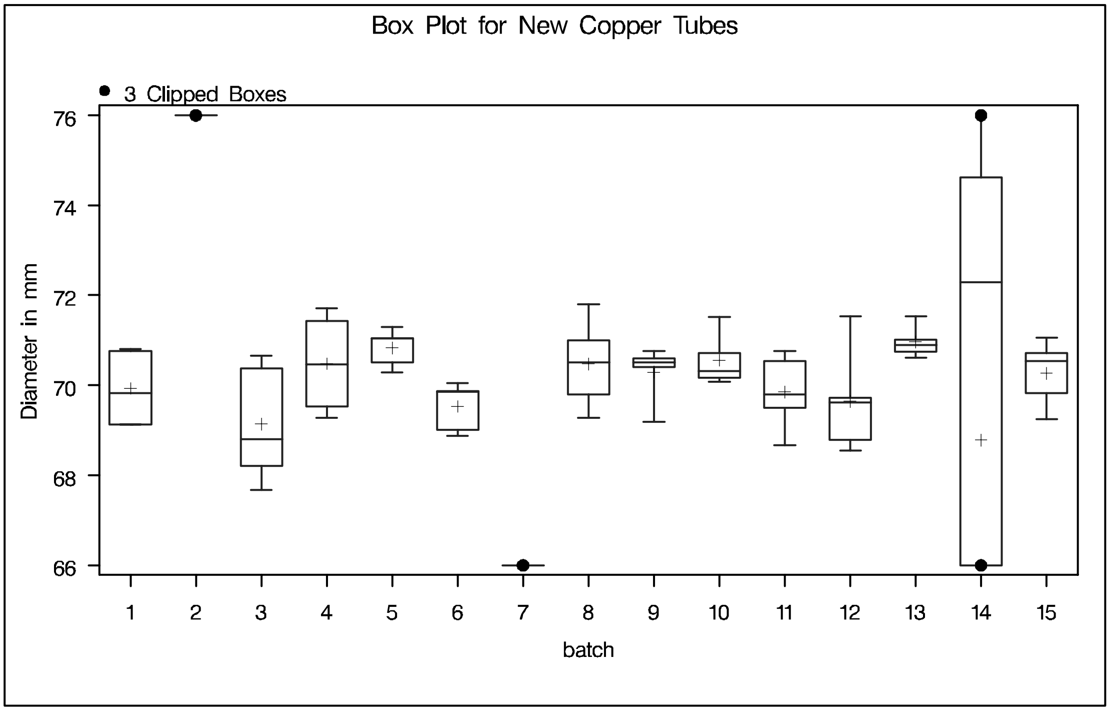


Figure 18.15. Box Plot Using Clipping Options

Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Boxes' and CLIPSUBCHAR='#' request the clipping legend *3 Clipped Boxes*. For more information about the clipping options, see the appropriate entries in "PLOT Statement Options."

Examples

This section provides advanced examples of the PLOT statement.

Example 18.1. Using Box Plots to Compare Groups

In the following example, a box plot is used to compare the delay times for airline flights during the Christmas holidays with the delay times prior to the holiday period. The following statements create a data set named Times with the delay times in minutes for 25 flights each day. When a flight is canceled, the delay is recorded as a missing value.

```
data Times;
   informat day date7. ;
   format   day date7. ;
   input day @ ;
   do flight=1 to 25;
      input delay @ ;
      output;
      end;
datalines;
16DEC88   4  12   2   2  18   5   6  21   0   0
          0  14   3   .   2   3   5   0   6  19
          7   4   9   5  10
17DEC88   1  10   3   3   0   1   5   0   .   .
          1   5   7   1   7   2   2  16   2   1
          3   1  31   5   0
18DEC88   7   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   3  12   8   6
          2   7   2   4   5
19DEC88  15   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  20   8   1  14   3
         10   0   1  11   7
20DEC88   2   1   0   4   4   6   2   2   1   4
          1  11   .   1   0   6   5   5   4   2
          2   6   6   4   0
21DEC88   2   6   6   2   7   7   5   2   5   0
          9   2   4   2   5   1   4   7   5   6
          5   0   4  36  28
22DEC88   3   7  22   1  11  11  39  46   7  33
         19  21   1   3  43  23   9   0  17  35
         50   0   2   1   0
23DEC88   6  11   8  35  36  19  21   .   .   4
          6  63  35   3  12  34   9   0  46   0
          0  36   3   0  14
24DEC88  13   2  10   4   5  22  21  44  66  13
          8   3   4  27   2  12  17  22  19  36
          9  72   2   4   4
25DEC88   4  33  35   0  11  11  10  28  34   3
         24   6  17   0   8   5   7  19   9   7
         21  17  17   2   6
26DEC88   3   8   8   2   7   7   8   2   5   9
```

```
           2   8   2  10  16   9   5  14  15   1
          12   2   2  14  18
;
run;
```

In the following statements, the MEANS procedure is used to count the number of canceled flights for each day. This information is then added to the data set Times.

```
proc means data=Times noprint;
   var delay;
   by day;
   output out=Cancel nmiss=ncancel;

data Times;
   merge Times Cancel;
   by day;
run;
```

The following statements create a data set named Weather that contains information about possible causes for delays. This data set is merged with the data set Times.

```
data Weather;
   informat day date7. ;
   format   day date7. ;
   length reason $ 16 ;
input day flight reason & ;
datalines;
16DEC88  8   Fog
17DEC88  18  Snow Storm
17DEC88  23  Sleet
21DEC88  24  Rain
21DEC88  25  Rain
22DEC88  7   Mechanical
22DEC88  15  Late Arrival
24DEC88  9   Late Arrival
24DEC88  22  Late Arrival
;
run;

data Times;
   merge Times Weather;
   by day flight;
run;
```

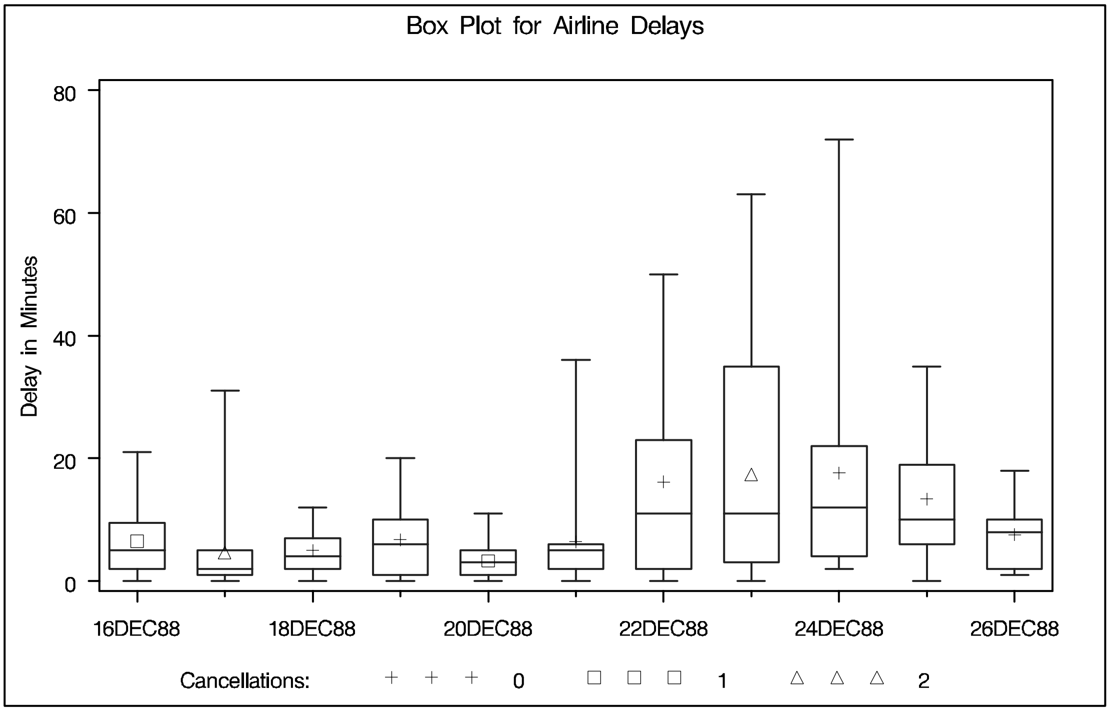
The following statements create a box plot for the complete set of data.

```
symbol1 v=plus     c=black;
symbol2 v=square   c=black;
symbol3 v=triangle c=black;
title 'Box Plot for Airline Delays';
```

```
proc boxplot data=Times;
   plot delay*day = ncancel /
       nohlabel
       symbollegend = legend1;
   legend1 label = ('Cancellations:');
   label delay = 'Delay in Minutes';
run;
```

The box plot is shown in Output 18.1.1. The level of the *symbol-variable* ncancel determines the symbol marker for each group mean, and the SYMBOLLEGEND= option controls the appearance of the legend for the symbols. The NOHLABEL option suppresses the label for the horizontal axis.

Output 18.1.1. Box Plot for Airline Data



The delay distributions from December 22 through December 25 are drastically different from the delay distributions during the pre-holiday period. Both the mean delay and the variability of the delays are much greater during the holiday period.

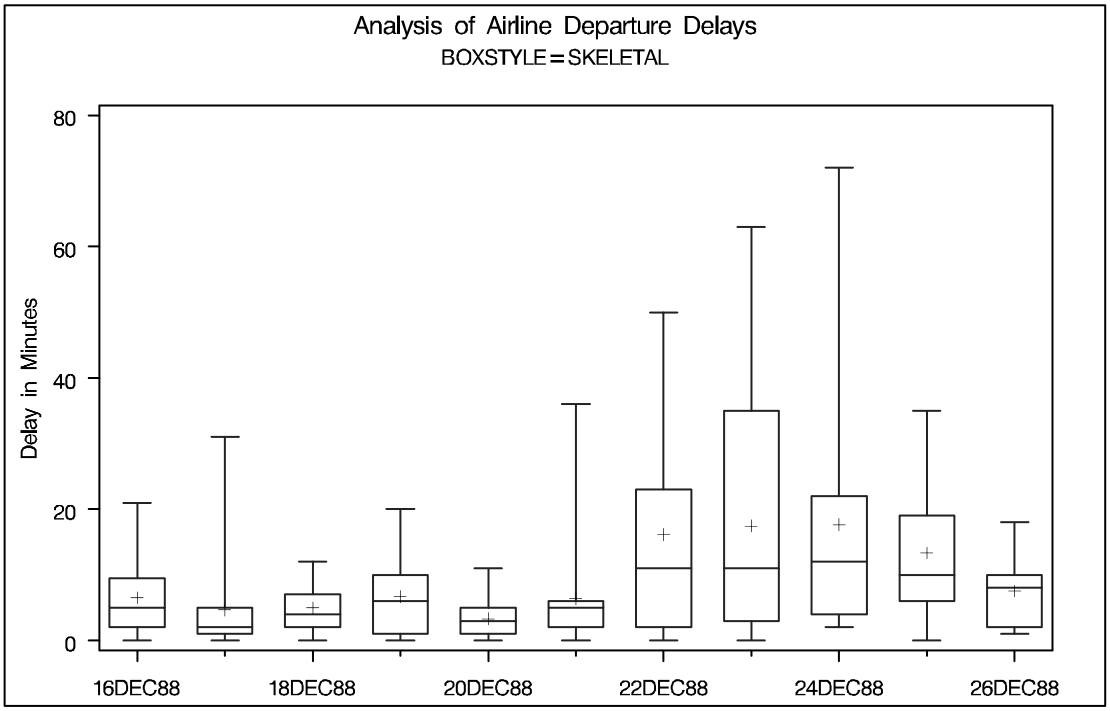
Example 18.2. Creating Various Styles of Box-and-Whisker Plots

The following example uses the flight delay data of the preceding example to illustrate how you can create box plots with various styles of box-and-whisker plots. The following statements create a plot, shown in Output 18.2.1, that displays skeletal box-and-whisker plots:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SKELETAL';
proc boxplot data=Times;
   plot delay*day /
      boxstyle = skeletal
      nohlabel;
   label delay = 'Delay in Minutes';
run;
```

In a skeletal box-and-whisker plot, the whiskers are drawn from the quartiles to the extreme values of the group. The skeletal box-and-whisker plot is the default style; consequently, you can also request this style by omitting the BOXSTYLE= option.

Output 18.2.1. BOXSTYLE=SKELETAL

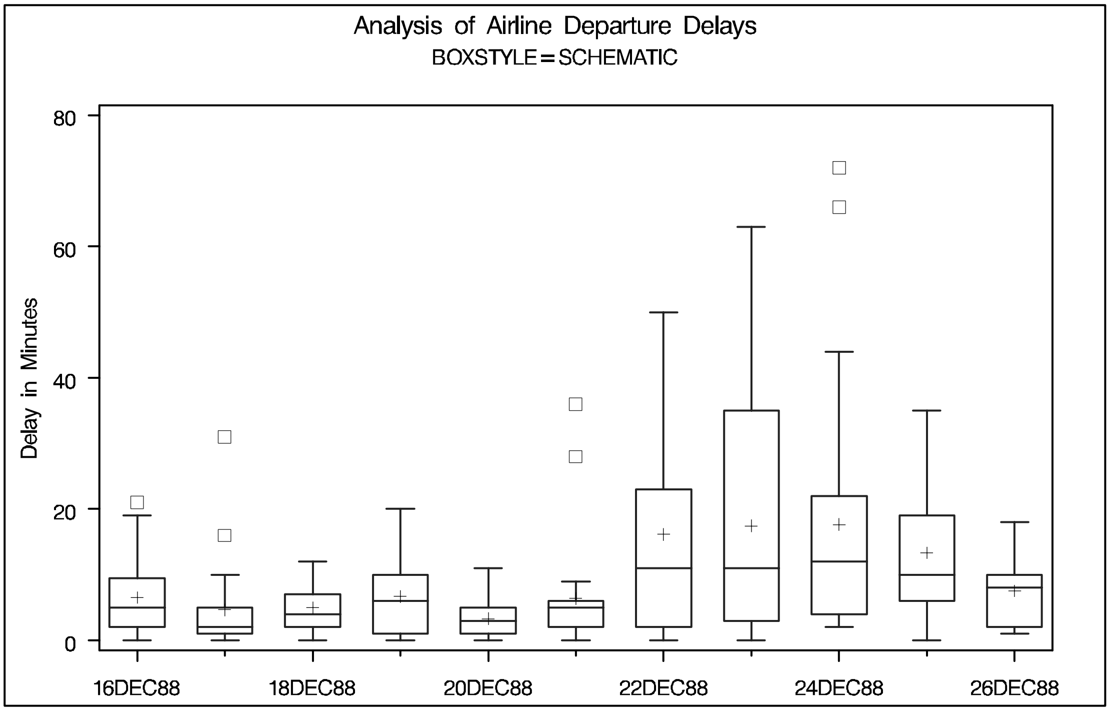


The following statements request a box plot with schematic box-and-whisker plots:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATIC';
proc boxplot data=Times;
   plot delay*day /
      boxstyle = schematic
      nohlabel;
   label delay = 'Delay in Minutes';
run;
```

The plot is shown in Output 18.2.2. When BOXSTYLE=SCHEMATIC is specified, the whiskers are drawn to the most extreme points in the group that lie within the *fences*. The *upper fence* is defined as the third quartile (represented by the upper edge of the box) plus 1.5 times the interquartile range (IQR). The *lower fence* is defined as the first quartile (represented by the lower edge of the box) minus 1.5 times the interquartile range. Observations outside the fences are identified with a special symbol. The default symbol is a square, and you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. Serifs are added to the whiskers by default. For further details, see the entry for the BOXSTYLE= option on page 493.

Output 18.2.2. BOXSTYLE=SCHEMATIC

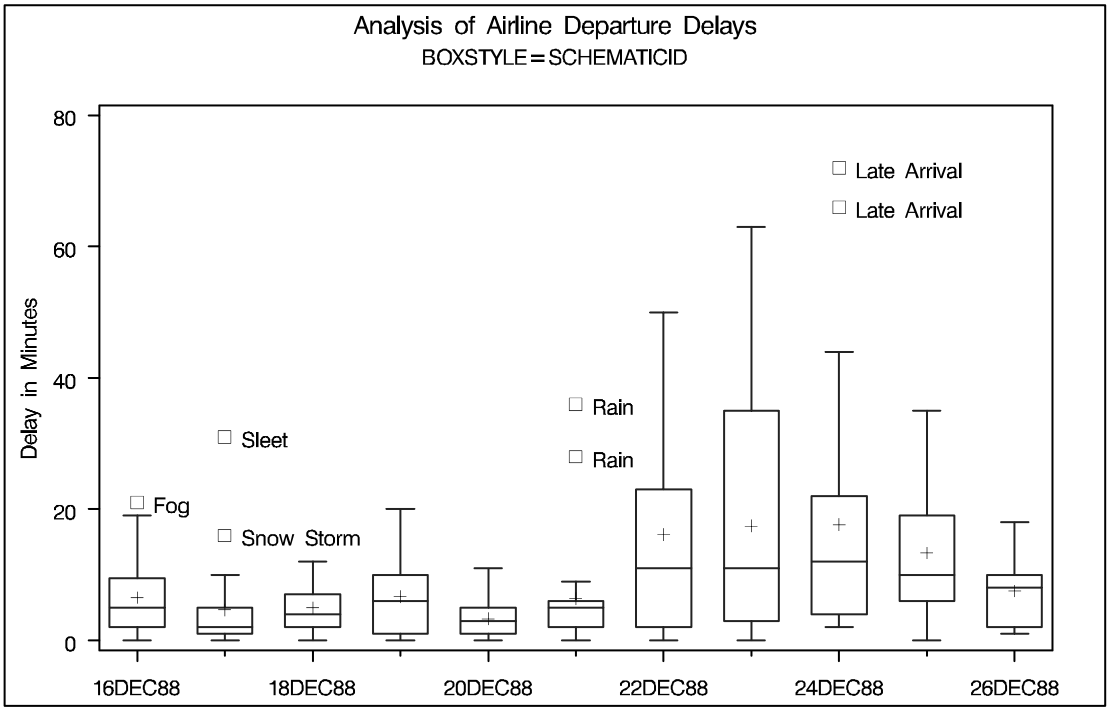


The following statements create a box plot with schematic box-and-whisker plots in which the observations outside the fences are labeled:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATICID';
proc boxplot data=Times;
   plot delay*day /
      boxstyle = schematicid
      nohlabel;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The plot is shown in Output 18.2.3. If you specify BOXSTYLE=SCHEMATICID, schematic box-and-whisker plots are displayed in which the value of the first ID variable (in this case, reason) is used to label each observation outside the fences.

Output 18.2.3. BOXSTYLE=SCHEMATICID

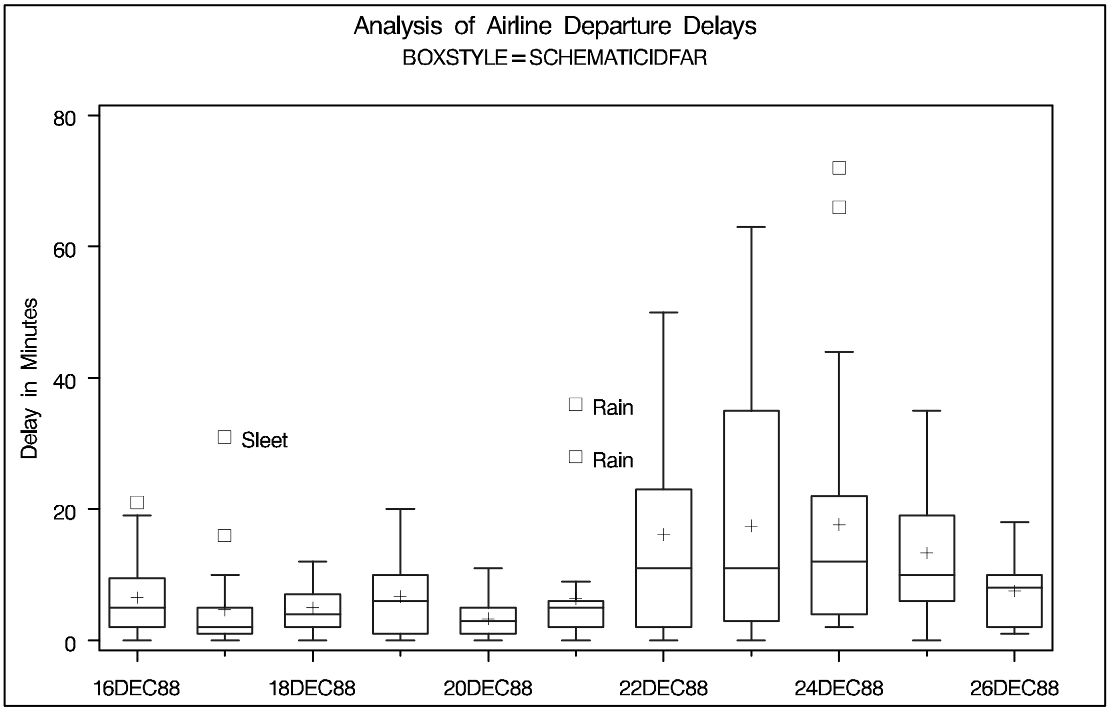


The following statements create a box plot with schematic box-and-whisker plots in which only the extreme observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATICIDFAR';
symbol v=plus color=black;
proc boxplot data=Times;
   plot delay*day /
      boxstyle = schematicidfar
      nohlabel;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The plot is shown in Output 18.2.4. If you specify BOXSTYLE=SCHEMATICIDFAR, schematic plots are displayed in which the value of the first ID variable is used to label each observation outside the *lower* and *upper far fences*. The lower and upper far fences are located 3×IQR below the 25th percentile and above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled.

Output 18.2.4. BOXSTYLE=SCHEMATICIDFAR



Other options for controlling the display of box-and-whisker plots include the BOXWIDTH=, BOXWIDTHSCALE=, CBOXES=, CBOXFILL=, and LBOXES= options.

Example 18.3. Creating Notched Box-and-Whisker Plots

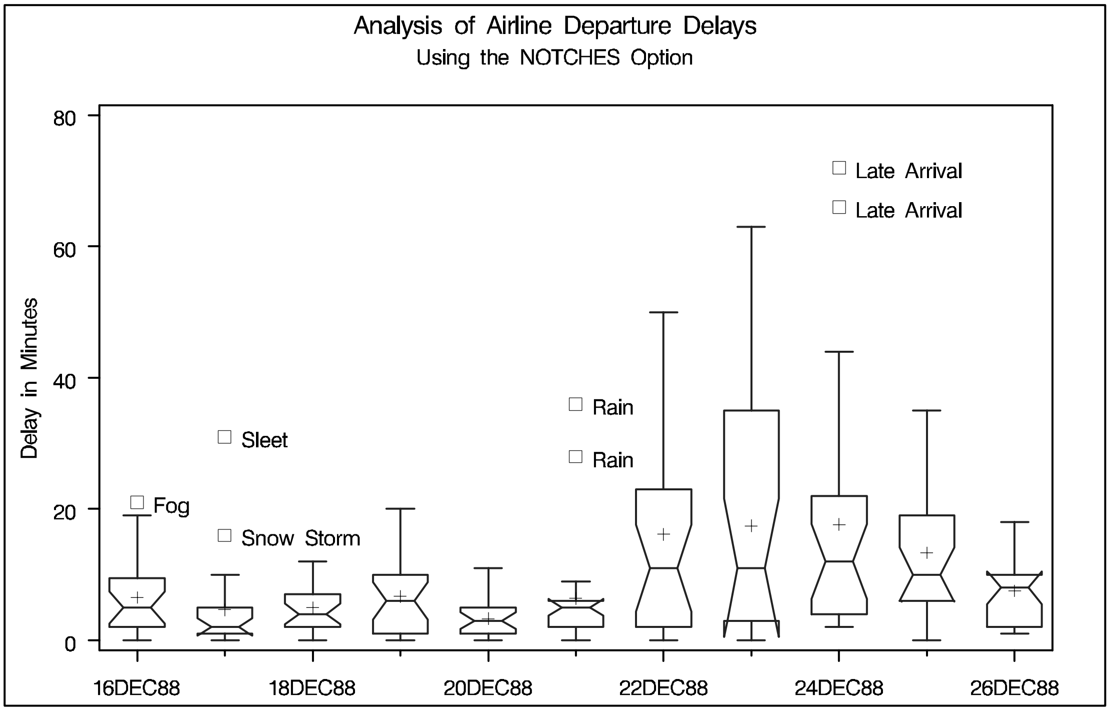
The following statements use the flight delay data of Example 18.1 to illustrate how to create box-and-whisker plots with notches:

```
symbol1 v=plus c=black;
title 'Analysis of Airline Departure Delays';
title2 'Using the NOTCHES Option';
proc boxplot data=Times;
   plot delay*day /
      boxstyle = schematicid
      nohlabel
      notches;
   id reason;
   label delay = 'Delay in Minutes';
run;
```

The notches, requested with the NOTCHES option, measure the significance of the difference between two medians. The medians of two box plots are significantly different at approximately the 0.05 level if the corresponding notches do not overlap.

For example, in Output 18.3.1, the median for December 20 is significantly different from the median for December 24.

Output 18.3.1. Notched Side-by-Side Box-and-Whisker Plots



Example 18.4. Creating Box-and-Whisker Plots with Varying Widths

The following example shows how to create a box plot with box-and-whisker plots whose widths vary proportionately with the group size. The following statements create a SAS data set named Times2 that contains flight departure delays (in minutes) recorded daily for eight consecutive days:

```
data Times2;
     label delay = 'Delay in Minutes';
     informat day date7. ;
     format   day date7. ;
     input day @ ;
     do flight=1 to 25;
        input delay @ ;
        output;
        end;
datalines;
01MAR90   12  4   2   2  15   8   0  11   0   0
           0  12  3   .   2   3   5   0   6  25
           7   4  9   5  10
02MAR90    1   .  3   .   0   1   5   0   .   .
```

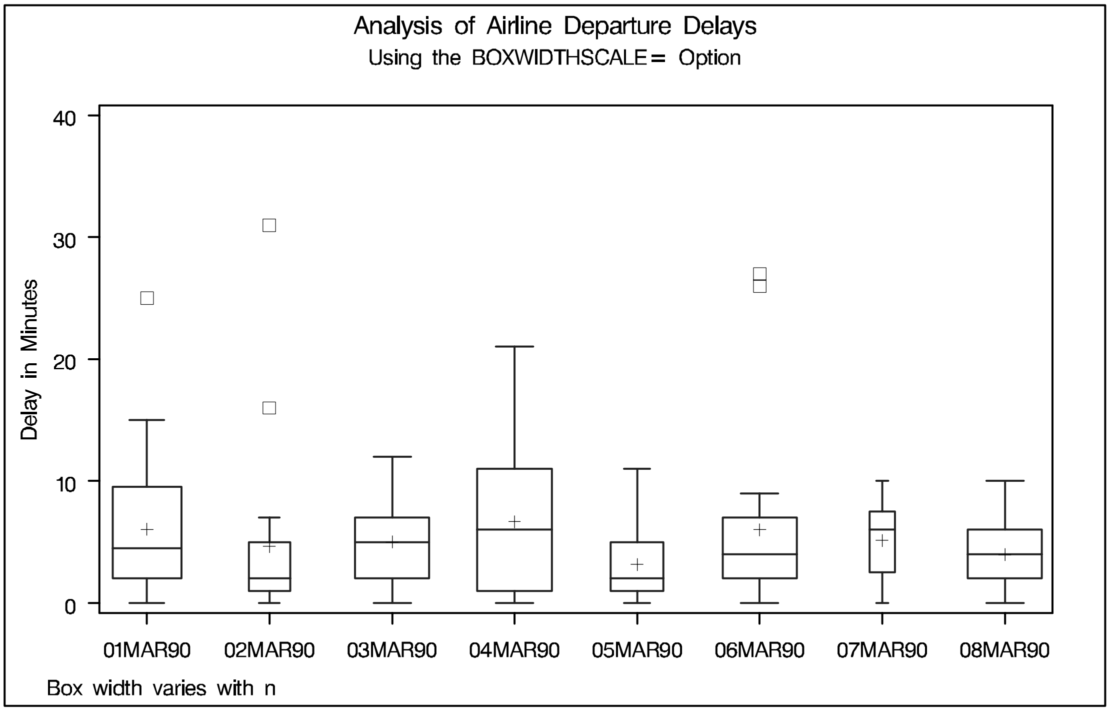
```
              1   5   7   .   7   2   2  16   2   1
              3   1  31   .   0
   03MAR90    6   8   4   2   3   2   7   6  11   3
              2   7   0   1  10   2   5  12   8   6
              2   7   2   4   5
   04MAR90   12   6   9   0  15   7   1   1   0   2
              5   6   5  14   7  21   8   1  14   3
             11   0   1  11   7
   05MAR90    2   1   0   4   .   6   2   2   1   4
              1  11   .   1   0   .   5   5   .   2
              3   6   6   4   0
   06MAR90    8   6   5   2   9   7   4   2   5   1
              2   2   4   2   5   1   3   9   7   8
              1   0   4  26  27
   07MAR90    9   6   6   2   7   8   .   .  10   8
              0   2   4   3   .   .   .   7   .   6
              4   0   .   .   .
   08MAR90    1   6   6   2   8   8   5   3   5   0
              8   2   4   2   5   1   6   4   5  10
              2   0   4   1   1
   run;
```

The following statements create the box plot shown in Output 18.4.1:

```
   title 'Analysis of Airline Departure Delays';
   title2 'Using the BOXWIDTHSCALE= Option';
   symbol1 v=plus c=black;
   proc boxplot data=Times2;
      plot delay*day /
         nohlabel
         boxstyle      = schematic
         boxwidthscale = 1
         bwslegend;
   run;
```

The BOXWIDTHSCALE=*value* option specifies that the width of box plots is to vary proportionately to a particular function of the group size n. The function is determined by the *value* and is identified on the plot with a legend if the BWSLEGEND option is specified. The BOXWIDTHSCALE= option is useful in situations where the group sizes vary widely.

Output 18.4.1. Box Plot with Box-and-Whisker Plots of Varying Widths



References

McGill, R., Tukey, J. W., and Larsen, W. A. (1978), “Variations of Box Plots,” *The American Statistician*, 32, 12–16.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley.

Chapter 19
The CALIS Procedure

Chapter Contents

Chapter 19
The CALIS Procedure

Overview

Structural equation modeling using covariance analysis is an important statistical tool in economics and behavioral sciences. Structural equations express relationships among several variables that can be either directly observed variables (manifest variables) or unobserved hypothetical variables (latent variables). For an introduction to latent variable models, refer to Loehlin (1987), Bollen (1989b), Everitt (1984), or Long (1983); and for manifest variables, refer to Fuller (1987).

In structural models, as opposed to functional models, all variables are taken to be random rather than having fixed levels. For maximum likelihood (default) and generalized least-squares estimation in PROC CALIS, the random variables are assumed to have an approximately multivariate normal distribution. Nonnormality, especially high kurtosis, can produce poor estimates and grossly incorrect standard errors and hypothesis tests, even in large samples. Consequently, the assumption of normality is much more important than in models with nonstochastic exogenous variables. You should remove outliers and consider transformations of nonnormal variables before using PROC CALIS with maximum likelihood (default) or generalized least-squares estimation. If the number of observations is sufficiently large, Browne's asymptotically distribution-free (ADF) estimation method can be used.

You can use the CALIS procedure to estimate parameters and test hypotheses for constrained and unconstrained problems in

- multiple and multivariate linear regression
- linear measurement-error models
- path analysis and causal modeling
- simultaneous equation models with reciprocal causation
- exploratory and confirmatory factor analysis of any order
- canonical correlation
- a wide variety of other (non)linear latent variable models

The parameters are estimated using the criteria of

- unweighted least squares (ULS)
- generalized least squares (GLS, with optional weight matrix input)
- maximum likelihood (ML, for multivariate normal data)
- weighted least squares (WLS, ADF, with optional weight matrix input)
- diagonally weighted least squares (DWLS, with optional weight matrix input)

The default weight matrix for generalized least-squares estimation is the sample covariance or correlation matrix. The default weight matrix for weighted least-squares estimation is an estimate of the asymptotic covariance matrix of the sample covariance or correlation matrix. In this case, weighted least-squares estimation is equivalent to Browne's (1982, 1984) asymptotic distribution-free estimation. The default weight matrix for diagonally weighted least-squares estimation is an estimate of the asymptotic variances of the input sample covariance or correlation matrix. You can also use an input data set to specify the weight matrix in GLS, WLS, and DWLS estimation.

You can specify the model in several ways:

- You can do a constrained (confirmatory) first-order factor analysis or component analysis using the FACTOR statement.
- You can specify simple path models using an easily formulated list-type RAM statement similar to that originally developed by J. McArdle (McArdle and McDonald 1984).
- If you have a set of structural equations to describe the model, you can use an equation-type LINEQS statement similar to that originally developed by P. Bentler (1985).
- You can analyze a broad family of matrix models using COSAN and MATRIX statements that are similar to the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980). It enables you to specify complex matrix models including nonlinear equation models and higher-order factor models.

You can specify linear and nonlinear equality and inequality constraints on the parameters with several different statements, depending on the type of input. Lagrange multiplier test indices are computed for simple constant and equality parameter constraints and for active boundary constraints. General equality and inequality constraints can be formulated using program statements. For more information, see the "SAS Program Statements" section on page 628.

PROC CALIS offers a variety of methods for the automatic generation of initial values for the optimization process:

- two-stage least-squares estimation
- instrumental variable factor analysis
- approximate factor analysis
- ordinary least-squares estimation
- McDonald's (McDonald and Hartmann 1992) method

In many common applications, these initial values prevent computational problems and save computer time.

Because numerical problems can occur in the (non)linearly constrained optimization process, the CALIS procedure offers several optimization algorithms:

- Levenberg-Marquardt algorithm (Moré, 1978)
- trust region algorithm (Gay 1983)
- Newton-Raphson algorithm with line search
- ridge-stabilized Newton-Raphson algorithm
- various quasi-Newton and dual quasi-Newton algorithms: Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, including a sequential quadratic programming algorithm for processing nonlinear equality and inequality constraints
- various conjugate gradient algorithms: automatic restart algorithm of Powell (1977), Fletcher-Reeves, Polak-Ribiere, and conjugate descent algorithm of Fletcher (1980)

The quasi-Newton and conjugate gradient algorithms can be modified by several line-search methods. All of the optimization techniques can impose simple boundary and general linear constraints on the parameters. Only the dual quasi-Newton algorithm is able to impose general nonlinear equality and inequality constraints.

The procedure creates an OUTRAM= output data set that completely describes the model (except for program statements) and also contains parameter estimates. This data set can be used as input for another execution of PROC CALIS. Small model changes can be made by editing this data set, so you can exploit the old parameter estimates as starting values in a subsequent analysis. An OUTEST= data set contains information on the optimal parameter estimates (parameter estimates, gradient, Hessian, projected Hessian and Hessian of Lagrange function for constrained optimization, the information matrix, and standard errors). The OUTEST= data set can be used as an INEST= data set to provide starting values and boundary and linear constraints for the parameters. An OUTSTAT= data set contains residuals and, for exploratory factor analysis, the rotated and unrotated factor loadings.

Automatic variable selection (using only those variables from the input data set that are used in the model specification) is performed in connection with the RAM and LINEQS input statements or when these models are recognized in an input model file. Also in these cases, the covariances of the exogenous manifest variables are recognized as given constants. With the PREDET option, you can display the predetermined pattern of constant and variable elements in the predicted model matrix before the minimization process starts. For more information, see the section "Automatic Variable Selection" on page 662 and the section "Exogenous Manifest Variables" on page 662.

PROC CALIS offers an analysis of linear dependencies in the information matrix (approximate Hessian matrix) that may be helpful in detecting unidentified models. You also can save the information matrix and the approximate covariance matrix of the parameter estimates (inverse of the information matrix), together with parameter estimates, gradient, and approximate standard errors, in an output data set for further analysis.

PROC CALIS does not provide the analysis of multiple samples with different sample size or a generalized algorithm for missing values in the data. However, the analysis of multiple samples with equal sample size can be performed by the analysis of a moment supermatrix containing the individual moment matrices as block diagonal submatrices.

Structural Equation Models

The Generalized COSAN Model

PROC CALIS can analyze matrix models of the form

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \cdots + \mathbf{F}_m\mathbf{P}_m\mathbf{F}_m'$$

where $\mathbf{C}$ is a symmetric correlation or covariance matrix, each matrix $\mathbf{F}_k$, $k = 1,\dots,m$, is the product of $n(k)$ matrices $\mathbf{F}_{k_1},\dots,\mathbf{F}_{k_{n(k)}}$, and each matrix $\mathbf{P}_k$ is symmetric, that is,

$$\mathbf{F}_k = \mathbf{F}_{k_1}\cdots\mathbf{F}_{k_{n(k)}} \quad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1,\dots,m$$

The matrices $\mathbf{F}_{k_j}$ and $\mathbf{P}_k$ in the model are parameterized by the matrices $\mathbf{G}_{k_j}$ and $\mathbf{Q}_k$

$$\mathbf{F}_{k_j} = \left\{ \begin{array}{c} \mathbf{G}_{k_j} \\ \mathbf{G}_{k_j}^{-1} \\ (\mathbf{I} - \mathbf{G}_{k_j})^{-1} \end{array} \right. \quad j = 1,\dots,n(k) \qquad \text{and} \qquad \mathbf{P}_k = \left\{ \begin{array}{c} \mathbf{Q}_k \\ \mathbf{Q}_k^{-1} \end{array} \right.$$

where you can specify the type of matrix desired.

The matrices $\mathbf{G}_{k_j}$ and $\mathbf{Q}_k$ can contain

- constant values
- parameters to be estimated
- values computed from parameters via programming statements

The parameters can be summarized in a parameter vector $\mathbf{X} = (x_1,\dots,x_t)$. For a given covariance or correlation matrix $\mathbf{C}$, PROC CALIS computes the unweighted least-squares (ULS), generalized least-squares (GLS), maximum likelihood (ML), weighted least-squares (WLS), or diagonally weighted least-squares (DWLS) estimates of the vector $\mathbf{X}$.

Some Special Cases of the Generalized COSAN Model

Original COSAN (Covariance Structure Analysis) Model (McDonald 1978, 1980)

Covariance Structure:

$$\mathbf{C} = \mathbf{F}_1\cdots\mathbf{F}_n\mathbf{P}\mathbf{F}_n'\cdots\mathbf{F}_1'$$

RAM (Reticular Action) Model (McArdle 1980; McArdle and McDonald 1984)

Structural Equation Model:

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

where $\mathbf{A}$ is a matrix of coefficients, and $\mathbf{v}$ and $\mathbf{u}$ are vectors of random variables. The variables in $\mathbf{v}$ and $\mathbf{u}$ can be manifest or latent variables. The endogenous variables corresponding to the components in $\mathbf{v}$ are expressed as a linear combination of the remaining variables and a residual component in $\mathbf{u}$ with covariance matrix $\mathbf{P}$.

Covariance Structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$ and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\} \qquad \text{and} \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

LINEQS (Linear Equations) Model (Bentler and Weeks 1980)

Structural Equation Model:

$$\boldsymbol{\eta} = \boldsymbol{\beta}\boldsymbol{\eta} + \boldsymbol{\gamma}\boldsymbol{\xi}$$

where $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are coefficient matrices, and $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are vectors of random variables. The components of $\boldsymbol{\eta}$ correspond to the endogenous variables; the components of $\boldsymbol{\xi}$ correspond to the exogenous variables and to error variables. The variables in $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ can be manifest or latent variables. The endogenous variables in $\boldsymbol{\eta}$ are expressed as a linear combination of the remaining endogenous variables, of the exogenous variables of $\boldsymbol{\xi}$, and of a residual component in $\boldsymbol{\xi}$. The coefficient matrix $\boldsymbol{\beta}$ describes the relationships among the endogenous variables of $\boldsymbol{\eta}$, and $I - \boldsymbol{\beta}$ should be nonsingular. The coefficient matrix $\boldsymbol{\gamma}$ describes the relationships between the endogenous variables of $\boldsymbol{\eta}$ and the exogenous and error variables of $\boldsymbol{\xi}$.

Covariance Structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}'((\mathbf{I} - \mathbf{B})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$, $\boldsymbol{\Phi} = \mathcal{E}\{\boldsymbol{\xi}\boldsymbol{\xi}'\}$, and

$$\mathbf{B} = \begin{pmatrix} \boldsymbol{\beta} & 0 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\gamma} \\ \mathbf{I} \end{pmatrix}$$

Keesling - Wiley - Jöreskog LISREL (Linear Structural Relationship) Model

Structural Equation Model and Measurement Models:

$$\boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta} \quad , \qquad \mathbf{y} = \boldsymbol{\Lambda}_y\boldsymbol{\eta} + \boldsymbol{\varepsilon} \quad , \qquad \mathbf{x} = \boldsymbol{\Lambda}_x\boldsymbol{\xi} + \boldsymbol{\delta}$$

where $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are vectors of latent variables (factors), and $\mathbf{x}$ and $\mathbf{y}$ are vectors of manifest variables. The components of $\boldsymbol{\eta}$ correspond to endogenous latent variables; the components of $\boldsymbol{\xi}$ correspond to exogenous latent variables. The endogenous and exogenous latent variables are connected by a system of linear equations (the structural model) with coefficient matrices $\mathbf{B}$ and $\boldsymbol{\Gamma}$ and an error vector $\boldsymbol{\zeta}$. It is assumed that matrix $\mathbf{I} - \mathbf{B}$ is nonsingular. The random vectors $\mathbf{y}$ and $\mathbf{x}$ correspond to manifest variables that are related to the latent variables $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ by two systems of linear equations (the measurement model) with coefficients $\boldsymbol{\Lambda}_y$ and $\boldsymbol{\Lambda}_x$ and with measurement errors $\boldsymbol{\varepsilon}$ and $\boldsymbol{\delta}$.

Covariance Structure:

$$\begin{aligned} \mathbf{C} &= \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}' \\ A &= \begin{pmatrix} 0 & 0 & \boldsymbol{\Lambda}_y & 0 \\ 0 & 0 & 0 & \boldsymbol{\Lambda}_x \\ 0 & 0 & \mathbf{B} & \boldsymbol{\Gamma} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} \boldsymbol{\Theta}_{\boldsymbol{\varepsilon}} & & & \\ & \boldsymbol{\Theta}_{\boldsymbol{\delta}} & & \\ & & \boldsymbol{\Psi} & \\ & & & \boldsymbol{\Phi} \end{pmatrix} \end{aligned}$$

with selection matrix $\mathbf{J}$, $\boldsymbol{\Phi} = \mathcal{E}\{\boldsymbol{\xi}\boldsymbol{\xi}'\}$, $\boldsymbol{\Psi} = \mathcal{E}\{\boldsymbol{\zeta}\boldsymbol{\zeta}'\}$, $\boldsymbol{\Theta}_\delta = \mathcal{E}\{\boldsymbol{\delta}\boldsymbol{\delta}'\}$, and $\boldsymbol{\Theta}_\varepsilon = \mathcal{E}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\}$.

Higher-Order Factor Analysis Models

First-order model:

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \mathbf{U}_1^2$$

Second-order model:

$$\mathbf{C} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

First-Order Autoregressive Longitudinal Factor Model

Example of McDonald (1980): k=3: Occasions of Measurement; n=3: Variables (Tests); m=2: Common Factors

$$\mathbf{C} = \mathbf{F}_1\mathbf{F}_2\mathbf{F}_3\mathbf{L}\mathbf{F}_3^{-1}\mathbf{F}_2^{-1}\mathbf{P}(\mathbf{F}_2^{-1})'(\mathbf{F}_3^{-1})'\mathbf{L}'\mathbf{F}_3'\mathbf{F}_2'\mathbf{F}_1' + \mathbf{U}^2$$

$$F_1 = \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix}$$

$$L = \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix} \quad , \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$S_2 = I_2 - D_2^2 \quad , \qquad S_3 = I_2 - D_3^2$$

For more information on this model, see Example 19.6 on page 739.

A Structural Equation Example

This example from Wheaton et al. (1977) illustrates the relationships among the RAM, LINEQS, and LISREL models. Different structural models for these data are in Jöreskog and Sörbom (1985) and in Bentler (1985, p. 28). The data set contains covariances among six (manifest) variables collected from 932 people in rural regions of Illinois:

Variable 1:	$V1, y_1$: Anomia 1967
Variable 2:	$V2, y_2$: Powerlessness 1967
Variable 3:	$V3, y_3$: Anomia 1971
Variable 4:	$V4, y_4$: Powerlessness 1971
Variable 5:	$V5, x_1$: Education (years of schooling)
Variable 6:	$V6, x_2$: Duncan's Socioeconomic Index (SEI)

It is assumed that anomia and powerlessness are indicators of an alienation factor and that education and SEI are indicators for a socioeconomic status (SES) factor. Hence, the analysis contains three latent variables:

Variable 7:	$F1, \eta_1$: Alienation 1967
Variable 8:	$F2, \eta_2$: Alienation 1971
Variable 9:	$F3, \xi_1$: Socioeconomic Status (SES)

The following path diagram shows the structural model used in Bentler (1985, p. 29) and slightly modified in Jöreskog and Sörbom (1985, p. 56). In this notation for the path diagram, regression coefficients between the variables are indicated as one-headed arrows. Variances and covariances among the variables are indicated as two-headed arrows. Indicating error variances and covariances as two-headed arrows with the same source and destination (McArdle 1988; McDonald 1985) is helpful in transforming the path diagram to RAM model list input for the CALIS procedure.

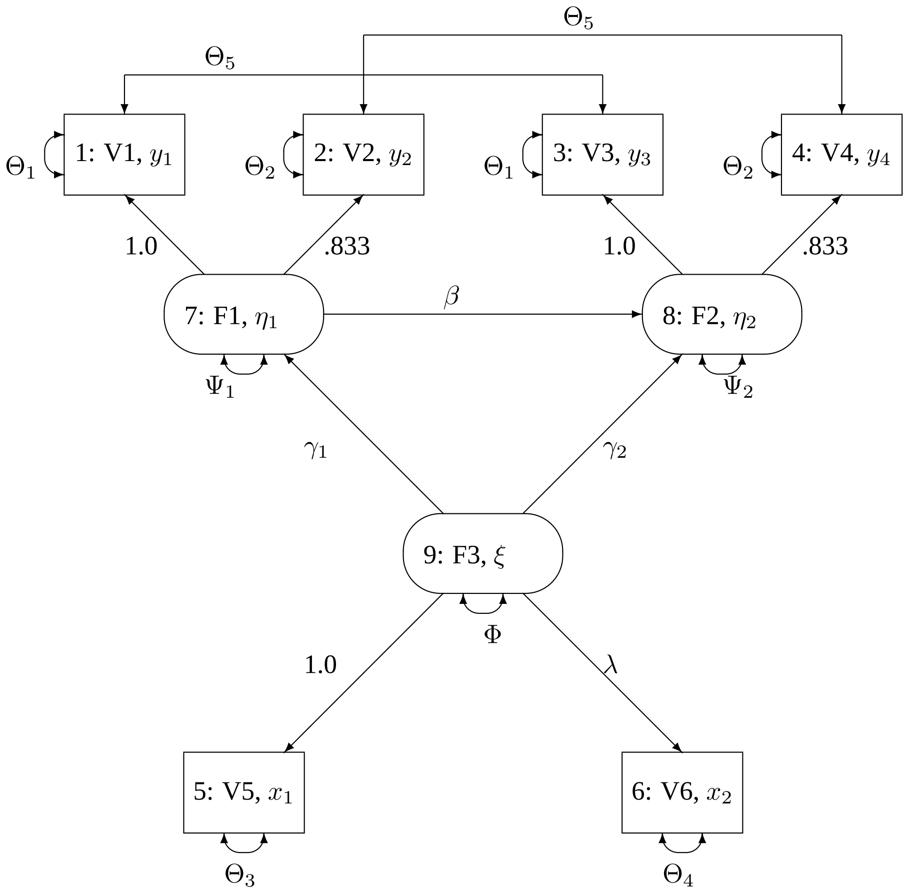


Figure 19.1. Path Diagram of Stability and Alienation Example

Variables in Figure 19.1 are as follows:

Variable 1:	$V1, y_1$: Anomia 1967
Variable 2:	$V2, y_2$: Powerlessness 1967
Variable 3:	$V3, y_3$: Anomia 1971
Variable 4:	$V4, y_4$: Powerlessness 1971
Variable 5:	$V5, x_1$: Education (years of schooling)
Variable 6:	$V6, x_2$: Duncan's Socioeconomic Index (SEI)
Variable 7:	$F1, \eta_1$: Alienation 1967
Variable 8:	$F2, \eta_2$: Alienation 1971
Variable 9:	$F3, \xi_1$: Socioeconomic Status (SES)

RAM Model

The vector $\mathbf{v}$ contains the six manifest variables $v_1 = V1, \ldots, v_6 = V6$ and the three latent variables $v_7 = F1, v_8 = F2, v_9 = F3$. The vector $\mathbf{u}$ contains the corresponding error variables $u_1 = E1, \ldots, u_6 = E6$ and $u_7 = D1, u_8 = D2, u_9 = D3$. The path diagram corresponds to the following set of structural equations of the RAM model:

$$
\begin{aligned}
v_1 &= 1.000 v_7 + u_1 \\
v_2 &= 0.833 v_7 + u_2 \\
v_3 &= 1.000 v_8 + u_3 \\
v_4 &= 0.833 v_8 + u_4 \\
v_5 &= 1.000 v_9 + u_5 \\
v_6 &= \lambda v_9 + u_6 \\
v_7 &= \gamma_1 v_9 + u_7 \\
v_8 &= \beta v_7 + \gamma_2 v_9 + u_8 \\
v_9 &= u_9
\end{aligned}
$$

This gives the matrices $\mathbf{A}$ and $\mathbf{P}$ in the RAM model:

$$
\mathbf{A} = \begin{pmatrix}
o & o & o & o & o & o & 1.000 & o & o \\
o & o & o & o & o & o & 0.833 & o & o \\
o & o & o & o & o & o & o & 1.000 & o \\
o & o & o & o & o & o & o & 0.833 & o \\
o & o & o & o & o & o & o & o & 1.000 \\
o & o & o & o & o & o & o & o & \lambda \\
o & o & o & o & o & o & o & o & \gamma_1 \\
o & o & o & o & o & o & \beta & o & \gamma_2 \\
o & o & o & o & o & o & o & o & o
\end{pmatrix}
$$

$$
\mathbf{P} = \begin{pmatrix}
\theta_1 & o & \theta_5 & o & o & o & o & o & o \\
o & \theta_2 & o & \theta_5 & o & o & o & o & o \\
\theta_5 & o & \theta_1 & o & o & o & o & o & o \\
o & \theta_5 & o & \theta_2 & o & o & o & o & o \\
o & o & o & o & \theta_3 & o & o & o & o \\
o & o & o & o & o & \theta_4 & o & o & o \\
o & o & o & o & o & o & \psi_1 & o & o \\
o & o & o & o & o & o & o & \psi_2 & o \\
o & o & o & o & o & o & o & o & \phi
\end{pmatrix}
$$

The RAM model input specification of this example for the CALIS procedure is in the "RAM Model Specification" section on page 563.

LINEQS Model

The vector $\boldsymbol{\eta}$ contains the six endogenous manifest variables $V1$, ... , $V6$ and the two endogenous latent variables $F1$ and $F2$. The vector $\boldsymbol{\xi}$ contains the exogenous error variables $E1$, ... , $E6$, $D1$, and $D2$ and the exogenous latent variable $F3$. The path diagram corresponds to the following set of structural equations of the LINEQS model:

$$\begin{aligned}
V1 &= 1.0F1 + E1 \\
V2 &= .833F1 + E2 \\
V3 &= 1.0F2 + E3 \\
V4 &= .833F2 + E4 \\
V5 &= 1.0F3 + E5 \\
V6 &= \lambda F3 + E6 \\
F1 &= \gamma_1 F3 + D1 \\
F2 &= \beta F1 + \gamma_2 F3 + D2
\end{aligned}$$

This gives the matrices $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\Phi}$ in the LINEQS model:

$$\boldsymbol{\beta} = \begin{pmatrix}
o & o & o & o & o & o & 1. & o \\
o & o & o & o & o & o & .833 & o \\
o & o & o & o & o & o & o & 1. \\
o & o & o & o & o & o & o & .833 \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & \beta & o
\end{pmatrix}, \quad
\boldsymbol{\gamma} = \begin{pmatrix}
1 & o & o & o & o & o & o & o & o \\
o & 1 & o & o & o & o & o & o & o \\
o & o & 1 & o & o & o & o & o & o \\
o & o & o & 1 & o & o & o & o & o \\
o & o & o & o & 1 & o & o & o & 1. \\
o & o & o & o & o & 1 & o & o & \lambda \\
o & o & o & o & o & o & 1 & o & \gamma_1 \\
o & o & o & o & o & o & o & 1 & \gamma_2
\end{pmatrix}$$

$$\boldsymbol{\Phi} = \begin{pmatrix}
\theta_1 & o & \theta_5 & o & o & o & o & o & o \\
o & \theta_2 & o & \theta_5 & o & o & o & o & o \\
\theta_5 & o & \theta_1 & o & o & o & o & o & o \\
o & \theta_5 & o & \theta_2 & o & o & o & o & o \\
o & o & o & o & \theta_3 & o & o & o & o \\
o & o & o & o & o & \theta_4 & o & o & o \\
o & o & o & o & o & o & \psi_1 & o & o \\
o & o & o & o & o & o & o & \psi_2 & o \\
o & o & o & o & o & o & o & o & \phi
\end{pmatrix}$$

The LINEQS model input specification of this example for the CALIS procedure is in the section “LINEQS Model Specification” on page 562.

LISREL Model

The vector $\mathbf{y}$ contains the four endogenous manifest variables $y_1 = V1, \ldots, y_4 = V4$, and the vector $\mathbf{x}$ contains the exogenous manifest variables $x_1 = V5$ and $x_2 = V6$. The vector $\boldsymbol{\varepsilon}$ contains the error variables $\varepsilon_1 = E1, \ldots, \varepsilon_4 = E4$ corresponding to $\mathbf{y}$, and the vector $\boldsymbol{\delta}$ contains the error variables $\delta_1 = E5$ and $\delta_2 = E6$ corresponding to $\mathbf{x}$. The vector $\boldsymbol{\eta}$ contains the endogenous latent variables (factors) $\eta_1 = F1$ and $\eta_2 = F2$, while the vector $\boldsymbol{\xi}$ contains the exogenous latent variable (factor) $\xi_1 = F3$. The vector $\boldsymbol{\zeta}$ contains the errors $\zeta_1 = D1$ and $\zeta_2 = D2$ in the equations (disturbance terms) corresponding to $\boldsymbol{\eta}$. The path diagram corresponds to the following set of structural equations of the LISREL model:

$$\begin{aligned}
y_1 &= 1.0\eta_1 + \epsilon_1 \\
y_2 &= .833\eta_1 + \epsilon_2 \\
y_3 &= 1.0\eta_2 + \epsilon_3 \\
y_4 &= .833\eta_2 + \epsilon_4 \\
x_1 &= 1.0\xi_1 + \delta_1 \\
x_2 &= \lambda\xi_1 + \delta_2 \\
\eta_1 &= \gamma_1\xi_1 + \zeta_1 \\
\eta_2 &= \beta\eta_1 + \gamma_2\xi_1 + \zeta_2
\end{aligned}$$

This gives the matrices $\boldsymbol{\Lambda}_y$, $\boldsymbol{\Lambda}_x$, $\mathbf{B}$, $\boldsymbol{\Gamma}$, and $\boldsymbol{\Phi}$ in the LISREL model:

$$\boldsymbol{\Lambda}_y = \begin{pmatrix} 1. & o \\ .833 & o \\ o & 1. \\ o & .833 \end{pmatrix}, \boldsymbol{\Lambda}_x = \begin{pmatrix} 1. \\ \lambda \end{pmatrix}, \mathbf{B} = \begin{pmatrix} o & o \\ \beta & o \end{pmatrix}, \boldsymbol{\Gamma} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}$$

$$\boldsymbol{\Theta}_\varepsilon^2 = \begin{pmatrix} \theta_1 & o & \theta_5 & o \\ o & \theta_2 & o & \theta_5 \\ \theta_5 & o & \theta_1 & o \\ o & \theta_5 & o & \theta_2 \end{pmatrix}, \boldsymbol{\Theta}_\delta^2 = \begin{pmatrix} \theta_3 & o \\ \theta_4 & o \end{pmatrix}, \boldsymbol{\Psi} = \begin{pmatrix} \psi_1 & o \\ o & \psi_2 \end{pmatrix}, \boldsymbol{\Phi} = (\phi)$$

The CALIS procedure does not provide a LISREL model input specification. However, any model that can be specified by the LISREL model can also be specified by using the COSAN, LINEQS, or RAM model specifications in PROC CALIS.

Getting Started

There are four sets of statements available in the CALIS procedure to specify a model. Since a LISREL analysis can be performed easily by using a RAM, COSAN, or LINEQS statement, there is no specific LISREL input form available in the CALIS procedure.

For COSAN-style input, you can specify the following statements:

COSAN *analysis model in matrix notation* **;**
MATRIX *definition of matrix elements* **;**
VARNAMES *names of additional variables* **;**
BOUNDS *boundary constraints* **;**
PARAMETERS *parameter names from program statements* **;**

For linear equations input, you can specify the following statements:

LINEQS *analysis model in equations notation* **;**
STD *variance pattern* **;**
COV *covariance pattern* **;**
BOUNDS *boundary constraints* **;**
PARAMETERS *parameter names from program statements* **;**

For RAM-style input, you can specify the following statements:

RAM *analysis model in list notation* **;**
VARNAMES *names of latent and error variables* **;**
BOUNDS *boundary constraints* **;**
PARAMETERS *parameter names from program statements* **;**

For (confirmatory) factor analysis input, you can specify the following statements:

FACTOR *options* **;**
MATRIX *definition of matrix elements* **;**
VARNAMES *names of latent and residual variables* **;**
BOUNDS *boundary constraints* **;**
PARAMETERS *parameter names from program statements* **;**

The model can also be obtained from an INRAM= data set, which is usually a version of an OUTRAM= data set produced by a previous PROC CALIS analysis (and possibly modified).

If no INRAM= data set is specified, you must use one of the four statements that defines the input form of the analysis model: COSAN, RAM, LINEQS, or FACTOR.

COSAN Model Specification

You specify the model for a generalized COSAN analysis with a COSAN statement and one or more MATRIX statements. The COSAN statement determines the name, dimension, and type (identity, diagonal, symmetric, upper, lower, general, inverse, and so forth) of each matrix in the model. You can specify the values of the constant elements in each matrix and give names and initial values to the elements that are to be estimated as parameters or functions of parameters using MATRIX statements. The resulting displayed output is in matrix form.

The following statements define the structural model of the alienation example as a COSAN model:

```
Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
Matrix A
          [ ,7] = 1. .833  5 * 0. Beta (.5) ,
          [ ,8] = 2 * 0.  1.  .833 ,
          [ ,9] = 4 * 0.  1.  Lamb Gam1-Gam2 (.5 2 * -.5);
Matrix P
          [1,1] = The1-The2 The1-The4 (6 * 3.) ,
          [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
          [3,1] = The5 (.2) ,
          [4,2] = The5 (.2) ;
```

The matrix model specified in the COSAN statement is the RAM model

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix **J** and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

The COSAN statement must contain only the matrices up to the central matrix **P** because of the symmetry of each matrix term in a COSAN model. Each matrix name is followed by one to three arguments in parentheses. The first argument is the number of columns. The second and third arguments are optional, and they specify the form of the matrix. The selection matrix **J** in the RAM model is specified by the 6×9 identity (IDE) (sub)matrix **J** because the first six variables in vector **v** correspond to the six manifest variables in the data set. The 9×9 parameter matrix **A** has a general (GEN) form and is used as $(\mathbf{I} - \mathbf{A})^{-1}$ in the analysis, as indicated by the identity-minus-inverse (IMI) argument. The central 9×9 matrix P is specified as a symmetric (SYM) matrix.

The MATRIX statement for matrix **A** specifies the values in columns 7, 8, and 9, which correspond to the three latent variables $F1$, $F2$, and $F3$, in accordance with the RAM model. The other columns of **A** are assumed to be zero. The initial values for the parameter elements in **A** are chosen as in the path diagram to be

$$\lambda = \beta = .5, \qquad \gamma_1 = \gamma_2 = -.5$$

In accordance with matrix $\mathbf{P}$ of the RAM model and the path model, the nine diagonal elements of matrix $\mathbf{P}$ are parameters with initial values

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = 3, \quad \psi_1 = \psi_2 = 4, \quad \phi = 6$$

There are also two off-diagonal elements in each triangle of $\mathbf{P}$ that are constrained to be equal, and they have an initial value of 0.2.

See the section "COSAN Model Statement" on page 591 for more information about the COSAN statement.

LINEQS Model Specification

You can also describe the model by a set of linear equations combined with variance and covariance specifications, using notation similar to that originally developed by P. Bentler for his EQS program. The displayed output can be in either equation form or matrix form.

The following statements define the structural model of the alienation example as a LINEQS model:

```
Lineqs
     V1 =        F1                   + E1,
     V2 =    .833 F1                  + E2,
     V3 =        F2                   + E3,
     V4 =    .833 F2                  + E4,
     V5 =        F3                   + E5,
     V6 = Lamb (.5) F3                + E6,
     F1 = Gam1(-.5) F3                + D1,
     F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;
Std
     E1-E6 = The1-The2 The1-The4 (6 * 3.),
     D1-D2 = Psi1-Psi2 (2 * 4.),
     F3    = Phi (6.) ;
Cov
     E1 E3 = The5 (.2),
     E4 E2 = The5 (.2);
```

The LINEQS statement shows the equations in the section "LINEQS Model" on page 558, except that in this case the coefficients to be estimated can be followed (optionally) by the initial value to use in the optimization process. If you do not specify initial values for the parameters in a LINEQS statement, PROC CALIS tries to assign these values automatically. The endogenous variables used on the left side can be manifest variables (with names that must be defined by the input data set) or latent variables (which must have names starting with F). The variables used on the right side can be manifest variables, latent variables (with names that must start with an F), or error variables (which must have names starting with an E or D). Commas separate the equations. The coefficients to be estimated are indicated by names. If no name is used, the coefficient is constant, either equal to a specified number or, if no number is used, equal to 1. The VAR statement in Bentler's notation is replaced here by the STD statement, because the VAR statement in PROC CALIS defines the subset of manifest variables in the data set to be analyzed. The variable names used in the STD or COV statement must be exogenous (that is, they should not occur on the left side of any equation). The STD and COV statements define the diagonal and off-diagonal elements in the $\mathbf{\Phi}$ matrix. The parameter specifications in the STD and COV statements are separated by commas. Using k variable names on the left of an equal sign in a COV statement means that the parameter list on the right side refers to all $k(k-1)/2$ distinct variable pairs in the $\mathbf{\Phi}$ matrix. Identical coefficient names indicate parameters constrained to be equal. You can also use prefix names to specify those parameters for which you do not need a precise name in any parameter constraint.

See the section "LINEQS Model Statement" on page 601 for more information about the precise syntax rules for a LINEQS statement.

RAM Model Specification

The RAM model allows a path diagram to be transcribed into a RAM statement in list form. The displayed output from the RAM statement is in matrix or list form.

The following statement defines the structural model of the alienation example as a RAM model:

```
Ram
     1   1  7  1.        ,
     1   2  7  .833      ,
     1   3  8  1.        ,
     1   4  8  .833      ,
     1   5  9  1.        ,
     1   6  9  .5    Lamb ,
     1   7  9  -.5   Gam1 ,
     1   8  7  .5    Beta ,
     1   8  9  -.5   Gam2 ,
     2   1  1  3.    The1 ,
     2   2  2  3.    The2 ,
     2   3  3  3.    The1 ,
     2   4  4  3.    The2 ,
     2   5  5  3.    The3 ,
     2   6  6  3.    The4 ,
     2   1  3  .2    The5 ,
     2   2  4  .2    The5 ,
     2   7  7  4.    Psi1 ,
     2   8  8  4.    Psi2 ,
     2   9  9  6.    Phi ;
```

You must assign numbers to the nodes in the path diagram. In the path diagram of Figure 19.1, the boxes corresponding to the six manifest variables $V1, \ldots, V6$ are assigned the number of the variable in the covariance matrix (1, . . . ,6); the circles corresponding to the three latent variables $F1$, $F2$, and $F3$ are given the numbers 7, 8, and 9. The path diagram contains 20 paths between the nine nodes; nine of the paths are one-headed arrows and eleven are two-headed arrows.

The RAM statement contains a list of items separated by commas. Each item corresponds to an arrow in the path diagram. The first entry in each item is the number of arrow heads (matrix number), the second entry shows where the arrow points to (row number), the third entry shows where the arrow comes from (column number), the fourth entry gives the (initial) value of the coefficient, and the fifth entry assigns a name if the path represents a parameter rather than a constant. If you specify the fifth entry as a parameter name, then the fourth list entry can be omitted, since PROC CALIS tries to assign an initial value to this parameter.

See the section "RAM Model Statement" on page 596 for more information about the RAM statement.

FACTOR Model Specification

You can specify the FACTOR statement to compute factor loadings **F** and unique variances **U** of an exploratory or confirmatory first-order factor (or component) analysis. By default, the factor correlation matrix **P** is an identity matrix.

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = diag$$

For a first-order confirmatory factor analysis, you can use MATRIX statements to define elements in the matrices **F**, **P**, and **U** of the more general model

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = diag$$

To perform a component analysis, specify the COMPONENT option to constrain the matrix **U** to a zero matrix; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}'$$

Note that the rank of $\mathbf{F}\mathbf{F}'$ is equal to the number m of components in **F**, and if m is smaller than the number of variables in the moment matrix **C**, the matrix of predicted model values is singular and maximum likelihood estimates for **F** cannot be computed. You should compute ULS estimates in this case.

The HEYWOOD option constrains the diagonal elements of **U** to be nonnegative; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{U} = diag$$

If the factor loadings are unconstrained, they can be orthogonally rotated by one of the following methods:

- principal axes rotation
- quartimax
- varimax
- equamax
- parsimax

The most common approach to factor analysis consists of two steps:

1. Obtain estimates for factor loadings and unique variances.
2. Apply an orthogonal or oblique rotation method.

PROC CALIS enables you to specify general linear and nonlinear equality and inequality constraints using the LINCON and NLINCON statements. You can specify the NLINCON statement to estimate orthogonal or oblique rotated factor loadings; refer to Browne and Du Toit (1992). Unlike PROC FACTOR, PROC CALIS currently does not provide standard errors for the rotated factor loadings.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates. If you use a MATRIX statement together with a FACTOR model specification, initial values are generally computed by McDonald's (McDonald and Hartmann 1992) method or are set by the START= option. See the section "FACTOR Model Statement" on page 606 and Example 19.3 on page 718 for more information about the FACTOR statement.

Constrained Estimation

- Simple equality constraints, $x_i = c_i$, $c_i = const$, and $x_i = x_j$, can be defined in each model by specifying constants or using the same name for parameters constrained to be equal.
- BOUNDS statement: You can specify boundary constraints, $l_i \leq x_i \leq u_i$, l_i, $u_i = const$, with the BOUNDS statement for the COSAN, LINEQS, and RAM models and in connection with an INRAM= data set. There may be serious convergence problems if negative values appear in the diagonal locations (variances) of the central model matrices during the minimization process. You can use the BOUNDS statement to constrain these parameters to have nonnegative values.
- LINCON statement: You can specify general linear equality and inequality constraints of the parameter estimates with the LINCON statement or by using an INEST= data set. The variables listed in the LINCON statements must be (a subset of) the model parameters. All optimization methods can be used with linear constraints.
- NLINCON statement: You can specify general nonlinear equality and inequality constraints of the parameter estimates with the NLINCON statement. The syntax of the NLINCON statement is almost the same as that for the BOUNDS statement with the exception that the BOUNDS statement can contain only names of the model parameters. However, the variables listed in the NLINCON statement can be defined by program statements. Only the quasi-Newton optimization method can be used when there are nonlinear constraints.
- Reparameterizing the Model: Complex linear equality and inequality constraints can be defined by means of program statements similar to those used in the DATA step. In this case, some of the parameters x_i are not elements of the matrices $\mathbf{G}_{kj}$ and $\mathbf{Q}_k$ but are instead defined in a PARAMETERS statement. Elements of the model matrices can then be computed by program statements

as functions of parameters in the PARAMETERS statement. This approach is similar to the classical COSAN program of R. McDonald, implemented by C. Fraser (McDonald 1978, 1980). One advantage of the CALIS procedure is that you need not supply code for the derivatives of the specified functions. The analytic derivatives of the user-written functions are computed automatically by PROC CALIS. The specified functions must be continuous and have continuous first-order partial derivatives. See the "SAS Program Statements" section on page 628 and the "Constrained Estimation Using Program Code" section on page 675 for more information about imposing linear and nonlinear restrictions on parameters by using program statements.

Although much effort has been made to implement reliable and numerically stable optimization methods, no practical algorithm exists that can always find the global optimum of a nonlinear function, especially when there are nonlinear constraints.

Syntax

```
PROC CALIS < options > ;
    COSAN matrix model ;
        MATRIX matrix elements ;
        VARNAMES variables ;
    LINEQS model equations ;
        STD variance pattern ;
        COV covariance pattern ;
    RAM model list ;
        VARNAMES variables ;
    FACTOR < options > ;
        MATRIX matrix elements ;
        VARNAMES variables ;
    BOUNDS boundary constraints ;
    BY variables ;
    FREQ variable ;
    LINCON linear constraints ;
    NLINCON nonlinear constraints ;
    NLOPTIONS optimization options ;
    PARAMETERS parameters ;
    PARTIAL variables ;
    STRUCTEQ variables ;
    VAR variables ;
    WEIGHT variable ;
    program statements
```

- If no INRAM= data set is specified, one of the four statements that defines the input form of the analysis model, COSAN, LINEQS, RAM, or FACTOR, must be used.
- The MATRIX statement can be used multiple times for the same or different matrices along with a COSAN or FACTOR statement. If the MATRIX

statement is used multiple times for the same matrix, later definitions override earlier ones.

- The STD and COV statements can be used only with the LINEQS model statement.
- You can formulate a generalized COSAN model using a COSAN statement. MATRIX statements can be used to define the elements of a matrix used in the COSAN statement. The input notation resembles the COSAN program of R. McDonald and C. Fraser (McDonald 1978, 1980).
- The RAM statement uses a simple list input that is especially suitable for describing J. McArdle's RAM analysis model (McArdle 1980, McArdle and McDonald 1984) for causal and path analysis problems.
- The LINEQS statement formulates the analysis model by means of a system of linear equations similar to P. Bentler's (1989) EQS program notation. The STD and COV statements can be used to define the variances and covariances corresponding to elements of matrix $\mathbf{\Phi}$ in the LINEQS model.
- A FACTOR statement can be used to compute a first-order exploratory or confirmatory factor (or component) analysis. The analysis of a simple exploratory factor analysis model performed by PROC CALIS is not as efficient as one performed by the FACTOR procedure. The CALIS procedure is designed for more general structural problems, and it needs significantly more computation time for a simple unrestricted factor or component analysis than does PROC FACTOR.
- You can add program statements to impose linear or nonlinear constraints on the parameters if you specify the model by means of a COSAN, LINEQS, or RAM statement. The PARAMETERS statement defines additional parameters that are needed as independent variables in your program code and that belong to the set of parameters to be estimated. Variable names used in the program code should differ from the preceding statement names. The code should respect the syntax rules of SAS statements usually used in the DATA step. See the "SAS Program Statements" section on page 628 for more information.
- The BOUNDS statement can be used to specify simple lower and upper boundary constraints for the parameters.
- You can specify general linear equality and inequality constraints with the LINCON statement (or via an INEST= data set). The NLINCON statement can be used to specify general nonlinear equality and inequality constraints by referring to nonlinear functions defined by program statements.
- The VAR, PARTIAL, WEIGHT, FREQ, and BY statements can be used in the same way as in other procedures, for example, the FACTOR or PRINCOMP procedure. You can select a subset of the input variables to analyze with the VAR statement. The PARTIAL statement defines a set of input variables that are chosen as partial variables for the analysis of a matrix of partial correlations or covariances. The BY statement specifies groups in which separate covariance structure analyses are performed.

PROC CALIS Statement

PROC CALIS < *options* > **;**

This statement invokes the procedure. The options available with the PROC CALIS statement are summarized in Table 19.1 and discussed in the following six sections.

Table 19.1. PROC CALIS Statement Options

Data Set Options	**Short Description**
DATA=	input data set
INEST=	input initial values, constraints
INRAM=	input model
INWGT=	input weight matrix
OUTEST=	covariance matrix of estimates
OUTJAC	Jacobian into OUTEST= data set
OUTRAM=	output model
OUTSTAT=	output statistic
OUTWGT=	output weight matrix

Data Processing	**Short Description**
AUGMENT	analyzes augmented moment matrix
COVARIANCE	analyzes covariance matrix
EDF=	defines nobs by number error df
NOBS=	defines number of observations nobs
NOINT	analyzes uncorrected moments
RDF=	defines nobs by number regression df
RIDGE	specifies ridge factor for moment matrix
UCORR	analyzes uncorrected CORR matrix
UCOV	analyzes uncorrected COV matrix
VARDEF=	specifies variance divisor

Estimation Methods	**Short Description**
METHOD=	estimation method
ASYCOV=	formula of asymptotic covariances
DFREDUCE=	reduces degrees of freedom
G4=	algorithm for STDERR
NODIAG	excludes diagonal elements from fit
WPENALTY=	penalty weight to fit correlations
WRIDGE=	ridge factor for weight matrix

Optimization Techniques	**Short Description**
TECHNIQUE=	minimization method
UPDATE=	update technique
LINESEARCH=	line-search method
FCONV=	function convergence criterion
GCONV=	gradient convergence criterion

Table 19.1. (continued)

Optimization Techniques	**Short Description**
INSTEP=	initial step length (RADIUS=, SALPHA=)
LSPRECISION=	line-search precision (SPRECISION=)
MAXFUNC=	max number function calls
MAXITER=	max number iterations

Displayed Output Options	**Short Description**
KURTOSIS	compute and display kurtosis
MODIFICATION	modification indices
NOMOD	no modification indices
NOPRINT	suppresses the displayed output
PALL	all displayed output (ALL)
PCORR	analyzed and estimated moment matrix
PCOVES	covariance matrix of estimates
PDETERM	determination coefficients
PESTIM	parameter estimates
PINITIAL	pattern and initial values
PJACPAT	displays structure of variable and constant elements of the Jacobian matrix
PLATCOV	latent variable covariances, scores
PREDET	displays predetermined moment matrix
PRIMAT	displays output in matrix form
PRINT	adds default displayed output
PRIVEC	displays output in vector form
PSHORT	reduces default output (SHORT)
PSUMMARY	displays only fit summary (SUMMARY)
PWEIGHT	weight matrix
RESIDUAL=	residual matrix and distribution
SIMPLE	univariate statistics
STDERR	standard errors
NOSTDERR	computes no standard errors
TOTEFF	displays total and indirect effects

Miscellaneous Options	**Short Description**
ALPHAECV=	probability Browne & Cudeck ECV
ALPHARMS=	probability Steiger & Lind RMSEA
BIASKUR	biased skewness and kurtosis
DEMPHAS=	emphasizes diagonal entries
FDCODE	uses numeric derivatives for code
HESSALG=	algorithm for Hessian
NOADJDF	no adjustment of df for active constraints
RANDOM=	randomly generated initial values
SINGULAR=	singularity criterion
ASINGULAR=	absolute singularity information matrix
COVSING=	singularity tolerance of information matrix

Table 19.1. (continued)

Miscellaneous Options	Short Description
MSINGULAR=	relative M singularity of information matrix
VSINGULAR=	relative V singularity of information matrix
SLMW=	probability limit for Wald test
START=	constant initial values

Data Set Options

DATA=*SAS-data-set*

specifies an input data set that can be an ordinary SAS data set or a specially structured TYPE=CORR, TYPE=COV, TYPE=UCORR, TYPE=UCOV, TYPE=SSCP, or TYPE=FACTOR SAS data set, as described in the section "Input Data Sets" on page 630. If the DATA= option is omitted, the most recently created SAS data set is used.

INEST | INVAR | ESTDATA=*SAS-data-set*

specifies an input data set that contains initial estimates for the parameters used in the optimization process and can also contain boundary and general linear constraints on the parameters. If the model did not change too much, you can specify an OUTEST= data set from a previous PROC CALIS analysis. The initial estimates are taken from the values of the PARMS observation.

INRAM=*SAS-data-set*

specifies an input data set that contains in RAM list form all information needed to specify an analysis model. The INRAM= data set is described in the section "Input Data Sets" on page 630. Typically, this input data set is an OUTRAM= data set (possibly modified) from a previous PROC CALIS analysis. If you use an INRAM= data set to specify the analysis model, you cannot use the model specification statements COSAN, MATRIX, RAM, LINEQS, STD, COV, FACTOR, or VARNAMES, but you can use the BOUNDS and PARAMETERS statements and program statements. If the INRAM= option is omitted, you must define the analysis model with a COSAN, RAM, LINEQS, or FACTOR statement.

INWGT=*SAS-data-set*

specifies an input data set that contains the weight matrix $\mathbf{W}$ used in generalized least-squares (GLS), weighted least-squares (WLS, ADF), or diagonally weighted least-squares (DWLS) estimation. If the weight matrix $\mathbf{W}$ defined by an INWGT= data set is not positive definite, it can be ridged using the WRIDGE= option. See the section "Estimation Criteria" on page 644 for more information. If no INWGT= data set is specified, default settings for the weight matrices are used in the estimation process. The INWGT= data set is described in the section "Input Data Sets" on page 630. Typically, this input data set is an OUTWGT= data set from a previous PROC CALIS analysis.

OUTEST | OUTVAR=*SAS-data-set*

creates an output data set containing the parameter estimates, their gradient, Hessian matrix, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the information matrix, the approximate covariance matrix of the parameter estimates ((generalized) inverse of information matrix), and approximate standard errors. If linear or nonlinear equality or active inequality constraints are present, the Lagrange multiplier estimates of the active constraints, the projected Hessian, and the Hessian of the Lagrange function are written to the data set. The OUTEST= data set also contains the Jacobian if the OUTJAC option is used.

The OUTEST= data set is described in the section "OUTEST= SAS-data-set" on page 634. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTJAC

writes the Jacobian matrix, if it has been computed, to the OUTEST= data set. This is useful when the information and Jacobian matrices need to be computed for other analyses.

OUTSTAT=*SAS-data-set*

creates an output data set containing the BY group variables, the analyzed covariance or correlation matrices, and the predicted and residual covariance or correlation matrices of the analysis. You can specify the correlation or covariance matrix in an OUTSTAT= data set as an input DATA= data set in a subsequent analysis by PROC CALIS. The OUTSTAT= data set is described in the section "OUTSTAT= SAS-data-set" on page 641. If the model contains latent variables, this data set also contains the predicted covariances between latent and manifest variables and the latent variables scores regression coefficients (see the PLATCOV option on page 586). If the FACTOR statement is used, the OUTSTAT= data set also contains the rotated and unrotated factor loadings, the unique variances, the matrix of factor correlations, the transformation matrix of the rotation, and the matrix of standardized factor loadings.

You can specify the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTRAM=*SAS-data-set*

creates an output data set containing the model information for the analysis, the parameter estimates, and their standard errors. An OUTRAM= data set can be used as an input INRAM= data set in a subsequent analysis by PROC CALIS. The OUTRAM= data set also contains a set of fit indices; it is described in more detail in the section "OUTRAM= SAS-data-set" on page 638. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTWGT=*SAS-data-set*

creates an output data set containing the weight matrix $\mathbf{W}$ used in the estimation process. You cannot create an OUTWGT= data set with an unweighted least-squares or maximum likelihood estimation. The fit function in GLS, WLS (ADF), and DWLS estimation contain the inverse of the (Cholesky factor of the) weight matrix $\mathbf{W}$ written in the OUTWGT= data set. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. An OUTWGT= data set can be used as an input INWGT= data set in a subsequent analysis by PROC CALIS. The OUTWGT= data set is described in the section "OUTWGT= SAS-data-set" on page 643. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

Data Processing Options

AUGMENT | AUG

analyzes the augmented correlation or covariance matrix. Using the AUG option is equivalent to specifying UCORR (NOINT but not COV) or UCOV (NOINT and COV) for a data set that is augmented by an intercept variable INTERCEPT that has constant values equal to 1. The variable INTERCEP can be used instead of the default INTERCEPT only if you specify the SAS option OPTIONS VALIDVARNAME=V6. The dimension of an augmented matrix is one higher than that of the corresponding correlation or covariance matrix. The AUGMENT option is effective only if the data set does not contain a variable called INTERCEPT and if you specify the UCOV, UCORR, or NOINT option.

Caution: The INTERCEPT variable is included in the moment matrix as the variable with number $n+1$. Using the RAM model statement assumes that the first n variable numbers correspond to the n manifest variables in the input data set. Therefore, specifying the AUGMENT option assumes that the numbers of the latent variables used in the RAM or path model have to start with number $n + 2$.

COVARIANCE | COV

analyzes the covariance matrix instead of the correlation matrix. By default, PROC CALIS (like the FACTOR procedure) analyzes a correlation matrix. If the DATA= input data set is a valid TYPE=CORR data set (containing a correlation matrix and standard deviations), using the COV option means that the covariance matrix is computed and analyzed.

DFE | EDF=*n*

makes the effective number of observations $n+i$, where i is 0 if the NOINT, UCORR, or UCOV option is specified without the AUGMENT option or where i is 1 otherwise. You can also use the NOBS= option to specify the number of observations.

DFR | RDF=*n*

makes the effective number of observations the actual number of observations minus the RDF= value. The degree of freedom for the intercept should not be included in the RDF= option. If you use PROC CALIS to compute a regression model, you can specify RDF= *number-of-regressor-variables* to get approximate standard errors equal to those computed by PROC REG.

NOBS= *nobs*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= and EDF= options override this default definition. You can use the RDF= option to modify the *nobs* specification. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, you can specify the number of observations either by using the NOBS=, EDF=, and RDF= options in the PROC CALIS statement or by including a _TYPE_='N' observation in the DATA= input data set.

NOINT

specifies that no intercept be used in computing covariances and correlations; that is, covariances or correlations are not corrected for the mean. You can specify this option (or UCOV or UCORR) to analyze mean structures in an uncorrected moment matrix, that is, to compute intercepts in systems of structured linear equations (see Example 19.2). The term NOINT is misleading in this case because an uncorrected covariance or correlation matrix is analyzed containing a constant (intercept) variable that is used in the analysis model. The degrees of freedom used in the variance divisor (specified by the VARDEF= option) and some of the assessment of the fit function (see the section "Assessment of Fit" on page 649) depend on whether an intercept variable is included in the model (the intercept is used in computing the corrected covariance or correlation matrix or is used as a variable in the uncorrected covariance or correlation matrix to estimate mean structures) or not included (an uncorrected covariance or correlation matrix is used that does not contain a constant variable).

RIDGE<=*r* >

defines a ridge factor r for the diagonal of the moment matrix $\mathbf{S}$ that is analyzed. The matrix $\mathbf{S}$ is transformed to

$$\mathbf{S} \longrightarrow \tilde{\mathbf{S}} = \mathbf{S} + r(diag(\mathbf{S}))$$

If you do not specify r in the RIDGE option, PROC CALIS tries to ridge the moment matrix $\mathbf{S}$ so that the smallest eigenvalue is about 10^{-3}.

Caution: The moment matrix in the OUTSTAT= output data set does not contain the ridged diagonal.

UCORR

analyzes the uncorrected correlation matrix instead of the correlation matrix corrected for the mean. Using the UCORR option is equivalent to specifying the NOINT option but not the COV option.

UCOV

analyzes the uncorrected covariance matrix instead of the covariance matrix corrected for the mean. Using the UCOV option is equivalent to specifying both the COV and NOINT options. You can specify this option to analyze mean structures in an uncorrected covariance matrix, that is, to compute intercepts in systems of linear structural equations (see Example 19.2).

VARDEF= DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of covariances and standard deviations. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table, where $i = 0$ if the NOINT option is used and $i = 1$ otherwise and where k is the number of partial variables specified in the PARTIAL statement. Using an intercept variable in a mean structure analysis, by specifying the AUGMENT option, includes the intercept variable in the analysis. In this case, $i = 1$. When a WEIGHT statement is used, w_j is the value of the WEIGHT variable in the jth observation, and the summation is performed only over observations with positive weight.

Value	**Description**	**Divisor**
DF	degrees of freedom	$N - k - i$
N	number of observations	N
WDF	sum of weights DF	$\sum_j^N w_j - k - i$
WEIGHT | WGT	sum of weights	$\sum_j^N w_j$

Estimation Methods

The default estimation method is maximum likelihood (METHOD=ML), assuming a multivariate normal distribution of the observed variables. The two-stage estimation methods METHOD=LSML, METHOD=LSGLS, METHOD=LSWLS, and METHOD=LSDWLS first compute unweighted least-squares estimates of the model parameters and their residuals. Afterward, these estimates are used as initial values for the optimization process to compute maximum likelihood, generalized least-squares, weighted least-squares, or diagonally weighted least-squares parameter estimates. You can do the same thing by using an OUTRAM= data set with least-squares estimates as an INRAM= data set for a further analysis to obtain the second set of parameter estimates. This strategy is also discussed in the section "Use of Optimization Techniques" on page 664. For more details, see the "Estimation Criteria" section on page 644.

METHOD | MET=*name*

specifies the method of parameter estimation. The default is METHOD=ML. Valid values for *name* are as follows:

ML | M | MAX — performs normal-theory maximum likelihood parameter estimation. The ML method requires a nonsingular covariance or correlation matrix.

GLS | G — performs generalized least-squares parameter estimation. If no INWGT= data set is specified, the GLS method uses the inverse sample covariance or correlation matrix as weight matrix $\mathbf{W}$. Therefore, METHOD=GLS requires a nonsingular covariance or correlation matrix.

WLS | W | ADF — performs weighted least-squares parameter estimation. If no INWGT= data set is specified, the WLS method uses the inverse matrix of estimated asymptotic covariances of the sample covariance or correlation matrix as the weight matrix $\mathbf{W}$. In this case, the WLS estimation method is equivalent to Browne's (1982, 1984) asymptotically distribution-free estimation. The WLS method requires a nonsingular weight matrix.

DWLS | D performs diagonally weighted least-squares parameter estimation. If no INWGT= data set is specified, the DWLS method uses the inverse diagonal matrix of asymptotic variances of the input sample covariance or correlation matrix as the weight matrix $\mathbf{W}$. The DWLS method requires a nonsingular diagonal weight matrix.

ULS | LS | U performs unweighted least-squares parameter estimation.

LSML | LSM | LSMAX performs unweighted least-squares followed by normal-theory maximum likelihood parameter estimation.

LSGLS | LSG performs unweighted least-squares followed by generalized least-squares parameter estimation.

LSWLS | LSW | LSADF performs unweighted least-squares followed by weighted least-squares parameter estimation.

LSDWLS | LSD performs unweighted least-squares followed by diagonally weighted least-squares parameter estimation.

NONE | NO uses no estimation method. This option is suitable for checking the validity of the input information and for displaying the model matrices and initial values.

ASYCOV | ASC=*name*

specifies the formula for asymptotic covariances used in the weight matrix $\mathbf{W}$ for WLS and DWLS estimation. The ASYCOV option is effective only if METHOD= WLS or METHOD=DWLS and no INWGT= input data set is specified. The following formulas are implemented:

BIASED: Browne's (1984) formula (3.4)
biased asymptotic covariance estimates; the resulting weight matrix is at least positive semidefinite. This is the default for analyzing a covariance matrix.

UNBIASED: Browne's (1984) formula (3.8)
asymptotic covariance estimates corrected for bias; the resulting weight matrix can be indefinite (that is, can have negative eigenvalues), especially for small N.

CORR: Browne and Shapiro's (1986) formula (3.2)
(identical to DeLeeuw's (1983) formulas (2,3,4)) the asymptotic variances of the diagonal elements are set to the reciprocal of the value r specified by the WPENALTY= option (default: $r = 100$). This formula is the default for analyzing a correlation matrix.

Caution: Using the WLS and DWLS methods with the ASYCOV=CORR option means that you are fitting a correlation (rather than a covariance) structure. Since the fixed diagonal of a correlation matrix for some models does not contribute to the model's degrees of freedom, you can specify the DFREDUCE=i option to reduce the degrees of freedom by the number of manifest variables used in the model. See the section "Counting the Degrees of Freedom" on page 676 for more information.

DFREDUCE | DFRED=*i*

reduces the degrees of freedom of the χ^2 test by i. In general, the number of degrees of freedom is the number of elements of the lower triangle of the predicted model matrix $\mathbf{C}$, $n(n+1)/2$, minus the number of parameters, t. If the NODIAG option is used, the number of degrees of freedom is additionally reduced by n. Because negative values of i are allowed, you can also increase the number of degrees of freedom by using this option. If the DFREDUCE= or NODIAG option is used in a correlation structure analysis, PROC CALIS does not additionally reduce the degrees of freedom by the number of constant elements in the diagonal of the predicted model matrix, which is otherwise done automatically. See the section "Counting the Degrees of Freedom" on page 676 for more information.

G4=*i*

specifies the algorithm to compute the approximate covariance matrix of parameter estimates used for computing the approximate standard errors and modification indices when the information matrix is singular. If the number of parameters t used in the model you analyze is smaller than the value of i, the time-expensive Moore-Penrose (G4) inverse of the singular information matrix is computed by eigenvalue decomposition. Otherwise, an inexpensive pseudo (G1) inverse is computed by sweeping. By default, $i = 60$. For more details, see the section "Estimation Criteria" on page 644.

NODIAG | NODI

omits the diagonal elements of the analyzed correlation or covariance matrix from the fit function. This option is useful only for special models with constant error variables. The NODIAG option does not allow fitting those parameters that contribute to the diagonal of the estimated moment matrix. The degrees of freedom are automatically reduced by n. A simple example for the usefulness of the NODIAG option is the fit of the first-order factor model, $\mathbf{S} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2$. In this case, you do not have to estimate the diagonal matrix of unique variances $\mathbf{U}^2$ that are fully determined by $diag(\mathbf{S} - \mathbf{F}\mathbf{F}')$.

WPENALTY | WPEN=*r*

specifies the penalty weight $r \geq 0$ for the WLS and DWLS fit of the diagonal elements of a correlation matrix (constant 1s). The criterion for weighted least-squares estimation of a correlation structure is

$$\mathbf{F}_{WLS} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \sum_{k=2}^{n} \sum_{l=1}^{k-1} w^{ij,kl} (s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_{i}^{n} (s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix $\mathbf{W}$. The second term is a penalty term to fit the diagonal elements of the correlation matrix. The default value is 100. The reciprocal of this value replaces the asymptotic variance corresponding to the diagonal elements of a correlation matrix in the weight matrix $\mathbf{W}$, and it is effective only with the ASYCOV=CORR option. The often used value $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. The default WPENALTY= value emphasizes the importance of the fit of the diagonal elements in the correlation matrix. You can decrease or increase the value of r if you want to decrease or increase the importance of the diagonal elements fit. This option is effective only with the WLS or DWLS estimation method and the analysis of a correlation matrix. See the section "Estimation Criteria" on page 644 for more details.

WRIDGE=*r*

defines a ridge factor r for the diagonal of the weight matrix $\mathbf{W}$ used in GLS, WLS, or DWLS estimation. The weight matrix $\mathbf{W}$ is transformed to

$$\mathbf{W} \longrightarrow \tilde{\mathbf{W}} = \mathbf{W} + r(diag(\mathbf{W}))$$

The WRIDGE= option is applied on the weight matrix

- before the WPENALTY= option is applied on it
- before the weight matrix is written to the OUTWGT= data set
- before the weight matrix is displayed

Optimization Techniques

Since there is no single nonlinear optimization algorithm available that is clearly superior (in terms of stability, speed, and memory) for all applications, different types of optimization techniques are provided in the CALIS procedure. Each technique can be modified in various ways. The default optimization technique for less than 40 parameters ($t < 40$) is TECHNIQUE=LEVMAR. For $40 \leq t < 400$, TECHNIQUE=QUANEW is the default method, and for $t \geq 400$, TECHNIQUE=CONGRA is the default method. For more details, see the section "Use of Optimization Techniques" on page 664. You can specify the following set of options in the PROC CALIS statement or in the NLOPTIONS statement.

TECHNIQUE | TECH=*name*
OMETHOD | OM=*name*

specifies the optimization technique. Valid values for *name* are as follows:

CONGRA | CG chooses one of four different conjugate-gradient optimization algorithms, which can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(t)$ memory compared to the $O(t^2)$ memory for the other three techniques, where t is the number of parameters. On the other hand, the conjugate-gradient techniques are significantly slower than other optimization techniques and should be used only when memory is insufficient for more efficient techniques. When you choose this option, UPDATE=PB by default. This is the default optimization technique if there are more than 400 parameters to estimate.

DBLDOG | DD performs a version of double dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is, in many aspects, very similar to the dual quasi-Newton method, but it does not use line search. The implementation is based on Dennis and Mei (1979) and Gay (1983).

LEVMAR | LM | MARQUARDT performs a highly stable but, for large problems, memory- and time-consuming Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. This is the default optimization technique if there are fewer than 40 parameters to estimate.

NEWRAP | NR | NEWTON performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging, and it can be modified with the LINESEARCH= option. In releases prior to Release 6.11, this option invokes the NRRIDG option.

NRRIDG | NRR | NR performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. This algorithm does not perform a line search. Since TECH=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of TECH=NRRIDG can be slower than that of TECH=NEWRAP, which works with Cholesky decomposition. However, usually TECH=NRRIDG needs less iterations than TECH=NEWRAP.

QUANEW | QN chooses one of four different quasi-Newton optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. If boundary constraints are used, these techniques sometimes converge slowly. When you choose this option, UPDATE=DBFGS by default. If nonlinear constraints are specified in the NLINCON statement, a modification of Powell's (1982a, 1982b) VMCWD algorithm is used, which is a sequential quadratic programming (SQP) method. This algorithm can be modified by specifying VERSION=1, which replaces the update of the Lagrange multiplier estimate vector μ to the original update of Powell (1978a, 1978b) that is used in the VF02AD algorithm. This can be helpful for applications with linearly dependent active constraints. The QUANEW technique is the default optimization technique if there are nonlinear constraints specified or if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG | TR performs a usually very stable but, for large problems, memory- and time-consuming trust region optimization technique. The algorithm is implemented similar to Gay (1983) and Moré and Sorensen (1983).

NONE | NO   does not perform any optimization. This option is similar to METHOD=NONE, but TECH=NONE also computes and displays residuals and goodness-of-fit statistics. If you specify METHOD=ML, METHOD=LSML, METHOD=GLS, METHOD=LSGLS, METHOD=WLS, or METHOD=LSWLS, this option allows computing and displaying (if the display options are specified) of the standard error estimates and modification indices corresponding to the input parameter estimates.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

For TECHNIQUE=CONGRA, the following updates can be used:

PB   performs the automatic restart update methodof Powell (1977) and Beale (1972). This is the default.

FR   performs the Fletcher-Reeves update (Fletcher 1980, p. 63).

PR   performs the Polak-Ribiere update (Fletcher 1980, p. 66).

CD   performs a conjugate-descent update of Fletcher (1987).

For TECHNIQUE=DBLDOG, the following updates (Fletcher 1987) can be used:

DBFGS   performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP   performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.

For TECHNIQUE=QUANEW, the following updates (Fletcher 1987) can be used:

BFGS   performs original BFGS update of the inverse Hessian matrix. This is the default for earlier releases.

DFP   performs the original DFP update of the inverse Hessian matrix.

DBFGS   performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP   performs the dual DFP update of the Cholesky factor of the Hessian matrix.

LINESEARCH | LIS | SMETHOD | SM=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1980) for an introduction to line-search techniques. The value of i can be $1, \dots, 8$; the default is $i = 2$.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function calls than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.

LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

FCONV | FTOL=*r*

specifies the relative function convergence criterion. The optimization process is terminated when the relative difference of the function values of two consecutive iterations is smaller than the specified value of r, that is

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where $FSIZE$ can be defined by the FSIZE= option in the NLOPTIONS statement. The default value is $r = 10^{-FDIGITS}$, where $FDIGITS$ either can be specified in the NLOPTIONS statement or is set by default to $-\log_{10}(\epsilon)$, where ϵ is the machine precision.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion (see the ABSGCONV= option on page 617 for the absolute gradient convergence criterion).

Termination of all techniques (except the CONGRA technique) requires the normalized predicted function reduction to be small,

$$\frac{[g(x^{(k)})]'[\mathbf{G}^{(k)}]^{-1}g(x^{(k)})}{\max(|f(x^{(k)})|, FSIZE)} \le r$$

where $FSIZE$ can be defined by the FSIZE= option in the NLOPTIONS statement. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{G}$ is not available),

$$\frac{\parallel g(x^{(k)}) \parallel_2^2 \quad \parallel s(x^{(k)}) \parallel_2}{\parallel g(x^{(k)}) - g(x^{(k-1)}) \parallel_2 \max(|f(x^{(k)})|, FSIZE)} \le r$$

is used. The default value is $r = 10^{-8}$.

Note that for releases prior to Release 6.11, the GCONV= option specified the absolute gradient convergence criterion.

INSTEP=*r*

For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithms TRUREG, DBLDOG, and LEVMAR or the default step length of the line-search algorithms can produce arithmetic overflows. If this occurs, specify decreasing values of $0 < r < 1$ such as INSTEP=1E$-$1, INSTEP=1E$-$2, INSTEP=1E$-$4, . . ., until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG, and LEVMAR), the INSTEP option specifies a positive factor for the initial radius of the trust region. The default initial trust-region radius is the length of the scaled gradient, and it corresponds to the default radius factor of $r = 1$.
- For line-search algorithms (NEWRAP, CONGRA, and QUANEW), INSTEP specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.

For releases prior to Release 6.11, specify the SALPHA= and RADIUS= options. For more details, see the section "Computational Problems" on page 678.

LSPRECISION | LSP=*r*
SPRECISION | SP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and more expensive line search may be necessary (Fletcher 1980, p.22). The second (default for NEWRAP, QUANEW, and CONGRA) and third line-search methods approach exact line search for small LSPRECISION= values. If you have numerical problems, you should decrease the LSPRECISION= value to obtain a more precise line search. The default LSPRECISION= values are displayed in the following table.

TECH=	UPDATE=	LSP default
QUANEW	DBFGS, BFGS	r = 0.4
QUANEW	DDFP, DFP	r = 0.06
CONGRA	all	r = 0.1
NEWRAP	no update	r = 0.9

For more details, refer to Fletcher (1980, pp. 25–29).

MAXFUNC | MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. The default values are displayed in the following table.

TECH=	MAXFUNC default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=125
DBLDOG, QUANEW	i=500
CONGRA	i=1000

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT=*i* <*n*>

specifies the maximum number i of iterations in the optimization process. The default values are displayed in the following table.

TECH=	MAXITER default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=50
DBLDOG, QUANEW	i=200
CONGRA	i=400

The default is used if you specify MAXITER=0 or if you omit the MAXITER option.

The optional second value n is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default is n=20. For example, specifying

```
maxiter= . 0
```

means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

RADIUS=*r*

is an alias for the INSTEP= option for Levenberg-Marquardt minimization.

SALPHA=*r*

is an alias for the INSTEP= option for line-search algorithms.

SPRECISION | SP=*r*

is an alias for the LSPRECISION= option.

Displayed Output Options

There are three kinds of options to control the displayed output:

- The PCORR, KURTOSIS, MODIFICATION, NOMOD, PCOVES, PDETERM, PESTIM, PINITIAL, PJACPAT, PLATCOV, PREDET, PWEIGHT, RESIDUAL, SIMPLE, STDERR, and TOTEFF options refer to specific parts of displayed output.
- The PALL, PRINT, PSHORT, PSUMMARY, and NOPRINT options refer to special subsets of the displayed output options mentioned in the first item. If the NOPRINT option is not specified, a default set of output is displayed. The PRINT and PALL options add other output options to the default output, and the PSHORT and PSUMMARY options reduce the default displayed output.
- The PRIMAT and PRIVEC options describe the form in which some of the output is displayed (the only nonredundant information displayed by PRIVEC is the gradient).

Output Options	**PALL**	**PRINT**	**default**	**PSHORT**	**PSUMMARY**
fit indices	*	*	*	*	*
linear dependencies	*	*	*	*	*
PREDET	*	(*)	(*)	(*)	
model matrices	*	*	*	*	
PESTIM	*	*	*	*	
iteration history	*	*	*	*	
PINITIAL	*	*	*		
SIMPLE	*	*	*		
STDERR	*	*	*		
RESIDUAL	*	*			
KURTOSIS	*	*			
PLATCOV	*	*			
TOTEFF	*	*			
PCORR	*				
MODIFICATION	*				
PWEIGHT	*				
PCOVES					
PDETERM					
PJACPAT					
PRIMAT					
PRIVEC					

KURTOSIS | KU

computes and displays univariate kurtosis and skewness, various coefficients of multivariate kurtosis, and the numbers of observations that contribute most to the normalized multivariate kurtosis. See the section "Measures of Multivariate Kurtosis" on page 658 for more information. Using the KURTOSIS option implies the SIMPLE display option. This information is computed only if the DATA= data set is a raw data set, and it is displayed by default if the PRINT option is specified. The multivariate LS kappa and the multivariate mean kappa are displayed only if you specify METHOD=WLS and the weight matrix is computed from an input raw data set. All measures of skewness and kurtosis are corrected for the mean. If an intercept variable is included in the analysis, the measures of multivariate kurtosis do not include the intercept variable in the corrected covariance matrix, as indicated by a displayed message. Using the BIASKUR option displays the biased values of univariate skewness and kurtosis.

MODIFICATION | MOD

computes and displays Lagrange multiplier test indices for constant parameter constraints, equality parameter constraints, and active boundary constraints, as well as univariate and multivariate Wald test indices. The modification indices are not computed in the case of unweighted or diagonally weighted least-squares estimation.

The Lagrange multiplier test (Bentler 1986; Lee 1985; Buse 1982) provides an estimate of the χ^2 reduction that results from dropping the constraint. For constant parameter constraints and active boundary constraints, the approximate change of the parameter value is displayed also. You can use this value to obtain an initial value if the parameter is allowed to vary in a modified model. For more information, see the section "Modification Indices" on page 673.

NOMOD

does not compute modification indices. The NOMOD option is useful in connection with the PALL option because it saves computing time.

NOPRINT | NOP

suppresses the displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

PALL | ALL

displays all optional output except the output generated by the PCOVES, PDETERM, PJACPAT, and PRIVEC options.

Caution: The PALL option includes the very expensive computation of the modification indices. If you do not really need modification indices, you can save computing time by specifying the NOMOD option in addition to the PALL option.

PCORR | CORR

displays the (corrected or uncorrected) covariance or correlation matrix that is analyzed and the predicted model covariance or correlation matrix.

PCOVES | PCE

displays the following:

- the information matrix (crossproduct Jacobian)
- the approximate covariance matrix of the parameter estimates (generalized inverse of the information matrix)
- the approximate correlation matrix of the parameter estimates

The covariance matrix of the parameter estimates is not computed for estimation methods ULS and DWLS. This displayed output is not included in the output generated by the PALL option.

PDETERM | PDE

displays three coefficients of determination: the determination of all equations (DETAE), the determination of the structural equations (DETSE), and the determination of the manifest variable equations (DETMV). These determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. The coefficients are displayed only when you specify a RAM or LINEQS model, but they are displayed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS.

You can use the STRUCTEQ statement to define which equations are structural equations. If you don't use the STRUCTEQ statement, PROC CALIS uses its own default definition to identify structural equations.

The term "structural equation" is not defined in a unique way. The LISREL program defines the structural equations by the user-defined BETA matrix. In PROC CALIS, the default definition of a structural equation is an equation that has a dependent left side variable that appears at least once on the right side of another equation, or an equation that has at least one right side variable that is the left side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

If the model contains structural equations, PROC CALIS also displays the "Stability Coefficient of Reciprocal Causation," that is, the largest eigenvalue of the $\mathbf{BB}'$ matrix, where $\mathbf{B}$ is the causal coefficient matrix of the structural equations. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). This displayed output is not included in the output generated by the PALL option.

PESTIM | PES

displays the parameter estimates. In some cases, this includes displaying the standard errors and t values.

PINITIAL | PIN

displays the input model matrices and the vector of initial values.

PJACPAT | PJP

displays the structure of variable and constant elements of the Jacobian matrix. This displayed output is not included in the output generated by the PALL option.

PLATCOV | PLC

displays the following:

- the estimates of the covariances among the latent variables
- the estimates of the covariances between latent and manifest variables
- the latent variable score regression coefficients

The estimated covariances between latent and manifest variables and the latent variable score regression coefficients are written to the OUTSTAT= data set. You can use the score coefficients with PROC SCORE to compute factor scores.

PREDET | PRE

displays the pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model. It is especially helpful in finding manifest variables that are not used or that are used as exogenous variables in a complex model specified in the COSAN statement. Those entries of the predicted moment matrix for which the model generates variable (rather than constant) elements are displayed as missing values. This output is displayed even without specifying the PREDET option if the model generates constant elements in the predicted model matrix different from those in the analysis moment matrix and if you specify at least the PSHORT amount of displayed output.

If the analyzed matrix is a correlation matrix (containing constant elements of 1s in the diagonal) and the model generates a predicted model matrix with q constant (rather than variable) elements in the diagonal, the degrees of freedom are automatically reduced by q. The output generated by the PREDET option displays those constant diagonal positions. If you specify the DFREDUCE= or NODIAG option, this automatic reduction of the degrees of freedom is suppressed. See the section "Counting the Degrees of Freedom" on page 676 for more information.

PRIMAT | PMAT

displays parameter estimates, approximate standard errors, and *t* values in matrix form if you specify the analysis model in the RAM or LINEQS statement. When a COSAN statement is used, this occurs by default.

PRINT | PRI

adds the options KURTOSIS, RESIDUAL, PLATCOV, and TOTEFF to the default output.

PRIVEC | PVEC

displays parameter estimates, approximate standard errors, the gradient, and *t* values in vector form. The values are displayed with more decimal places. This displayed output is not included in the output generated by the PALL option.

PSHORT | SHORT | PSH

excludes the output produced by the PINITIAL, SIMPLE, and STDERR options from the default output.

PSUMMARY | SUMMARY | PSUM

displays the fit assessment table and the ERROR, WARNING, and NOTE messages.

PWEIGHT | PW

displays the weight matrix $\mathbf{W}$ used in the estimation. The weight matrix is displayed after the WRIDGE= and the WPENALTY= options are applied to it.

RESIDUAL | RES $<$ = NORM | VARSTAND | ASYSTAND $>$

displays the absolute and normalized residual covariance matrix, the rank order of the largest residuals, and a bar chart of the residuals. This information is displayed by default when you specify the PRINT option.

Three types of normalized or standardized residual matrices can be chosen with the RESIDUAL= specification.

RESIDUAL= NORM	Normalized Residuals
RESIDUAL= VARSTAND	Variance Standardized Residuals
RESIDUAL= ASYSTAND	Asymptotically Standardized Residuals

For more details, see the section "Assessment of Fit" on page 649.

SIMPLE | S

displays means, standard deviations, skewness, and univariate kurtosis if available. This information is displayed when you specify the PRINT option. If you specify the UCOV, UCORR, or NOINT option, the standard deviations are not corrected for the mean. If the KURTOSIS option is specified, the SIMPLE option is set by default.

STDERR | SE

displays approximate standard errors if estimation methods other than unweighted least squares (ULS) or diagonally weighted least squares (DWLS) are used (and the NOSTDERR option is not specified). If you specify neither the STDERR nor the NOSTDERR option, the standard errors are computed for the OUTRAM= data set. This information is displayed by default when you specify the PRINT option.

NOSTDERR | NOSE

specifies that standard errors should not be computed. Standard errors are not computed for unweighted least-squares (ULS) or diagonally weighted least-squares (DWLS) estimation. In general, standard errors are computed even if the STDERR display option is not used (for file output).

TOTEFF | TE

computes and displays total effects and indirect effects.

Miscellaneous Options

ALPHAECV=α

specifies the significance level for a $1 - \alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Browne & Cudeck (1993) expected cross validation index (ECVI) . The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the ECVI.

ALPHARMS=α

specifies the significance level for a $1 - \alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Steiger & Lind (1980) root mean squared error of approximation (RMSEA) coefficient (refer to Browne and Du Toit 1992). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the RMSEA.

ASINGULAR | ASING=*r*

specifies an absolute singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

In the preceding criterion, $d_{j,j}$ is the diagonal pivot of the matrix, and *VSING* and *MSING* are the specified values of the VSINGULAR= and MSINGULAR= options. The default value for *ASING* is the square root of the smallest positive double precision value. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

BIASKUR

computes univariate skewness and kurtosis by formulas uncorrected for bias. See the section "Measures of Multivariate Kurtosis" on page 658 for more information.

COVSING=*r*

specifies a nonnegative threshold r, which determines whether the eigenvalues of the information matrix are considered to be zero. If the inverse of the information matrix is found to be singular (depending on the VSINGULAR=, MSINGULAR=, ASINGULAR=, or SINGULAR= option), a generalized inverse is computed using the eigenvalue decomposition of the singular matrix. Those eigenvalues smaller than r are considered to be zero. If a generalized inverse is computed and you do not specify the NOPRINT option, the distribution of eigenvalues is displayed.

DEMPHAS | DE=*r*

changes the initial values of all parameters that are located on the diagonals of the central model matrices by the relationship

$$diag_{new} = r(|diag_{old}| + 1)$$

The initial values of the diagonal elements of the central matrices should always be nonnegative to generate positive definite predicted model matrices in the first iteration. By using values of $r > 1$, for example, $r = 2$, $r = 10$, ..., you can increase these initial values to produce predicted model matrices with high positive eigenvalues in the first iteration. The DEMPHAS= option is effective independent of the way the initial values are set; that is, it changes the initial values set in the model specification as well as those set by an INRAM= data set and those automatically generated for RAM, LINEQS, or FACTOR model statements. It also affects the initial values set by the START= option, which uses, by default, DEMPHAS=100 if a covariance matrix is analyzed and DEMPHAS=10 for a correlation matrix.

FDCODE

replaces the analytic derivatives of the program statements by numeric derivatives (finite difference approximations). In general, this option is needed only when you have program statements that are too difficult for the built-in function compiler to differentiate analytically. For example, if the program code for the nonlinear constraints contains many arrays and many DO loops with array processing, the built-in function compiler can require too much time and memory to compute derivatives of the constraints with respect to the parameters. In this case, the Jacobian matrix of constraints is computed numerically by using finite difference approximations. The FDCODE option does not modify the kind of derivatives specified with the HESSALG= option.

HESSALG | HA = 1 | 2 | 3 | 4 | 5 | 6 | 11

specifies the algorithm used to compute the (approximate) Hessian matrix when TECHNIQUE=LEVMAR and NEWRAP, to compute approximate standard errors of the parameter estimates, and to compute Lagrange multipliers. There are different groups of algorithms available.

- analytic formulas: HA=*1,2,3,4,11*
- finite difference approximation: HA=*5,6*
- dense storage: HA=*1,2,3,4,5,6*
- sparse storage: HA=*11*

If the Jacobian is more than 25% dense, the dense analytic algorithm, HA$=1$, is used by default. The HA$=1$ algorithm is faster than the other dense algorithms, but it needs considerably more memory for large problems than HA$=2$,3,4. If the Jacobian is more than 75% sparse, the sparse analytic algorithm, HA$=11$, is used by default. The dense analytic algorithm HA$=4$ corresponds to the original COSAN algorithm; you are advised not to specify HA$=4$ due to its very slow performance. If there is not enough memory available for the dense analytic algorithm HA$=1$ and you must specify HA$=2$ or HA$=3$, it may be more efficient to use one of the quasi-Newton or conjugate-gradient optimization techniques since Levenberg-Marquardt and Newton-Raphson optimization techniques need to compute the Hessian matrix in each iteration. For approximate standard errors and modification indices, the Hessian matrix has to be computed at least once, regardless of the optimization technique.

The algorithms HA$=5$ and HA$=6$ compute approximate derivatives by using forward difference formulas. The HA$=5$ algorithm corresponds to the analytic HA$=1$: it is faster than HA$=6$, however it needs much more memory. The HA$=6$ algorithm corresponds to the analytic HA$=2$: it is slower than HA$=5$, however it needs much less memory.

Test computations of large sparse problems show that the sparse algorithm HA$=11$ can be up to ten times faster than HA$=1$ (and needs much less memory).

MSINGULAR | MSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and *ASING* and *VSING* are the specified values of the ASINGULAR= and VSINGULAR= options. If you do not specify the SINGULAR= option, the default value for *MSING* is 1E−12; otherwise, the default value is 1E−4 * SINGULAR. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

NOADJDF

turns off the automatic adjustment of degrees of freedom when there are active constraints in the analysis. When the adjustment is in effect, most fit statistics and the associated probability levels will be affected. This option should be used when the researcher believes that the active constraints observed in the current sample will have little chance to occur in repeated sampling.

RANDOM =*i*

specifies a positive integer as a seed value for the pseudo-random number generator to generate initial values for the parameter estimates for which no other initial value assignments in the model definitions are made. Except for the parameters in the diagonal locations of the central matrices in the model, the initial values are set to random numbers in the range $0 \leq r \leq 1$. The values for parameters in the diagonals of the central matrices are random numbers multiplied by 10 or 100. For more information, see the section "Initial Estimates" on page 661.

SINGULAR | SING =*r*

specifies the singularity criterion r, $0 < r < 1$, used, for example, for matrix inversion. The default value is the square root of the relative machine precision or, equivalently, the square root of the largest double precision value that, when added to 1, results in 1.

SLMW=*r*

specifies the probability limit used for computing the stepwise multivariate Wald test. The process stops when the univariate probability is smaller than r. The default value is $r = 0.05$.

START =*r*

In general, this option is needed only in connection with the COSAN model statement, and it specifies a constant r as an initial value for all the parameter estimates for which no other initial value assignments in the pattern definitions are made. Start values in the diagonal locations of the central matrices are set to $100|r|$ if a COV or UCOV matrix is analyzed and $10|r|$ if a CORR or UCORR matrix is analyzed. The default value is $r = .5$. Unspecified initial values in a FACTOR, RAM, or LINEQS model are usually computed by PROC CALIS. If none of the initialization methods are able to compute all starting values for a model specified by a FACTOR, RAM, or

LINEQS statement, then the start values of parameters that could not be computed are set to r, $10|r|$, or $100|r|$. If the DEMPHAS= option is used, the initial values of the diagonal elements of the central model matrices are multiplied by the value specified in the DEMPHAS= option. For more information, see the section "Initial Estimates" on page 661.

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and *ASING* and *MSING* are the specified values of the ASINGULAR= and MSINGULAR= options. If you do not specify the SINGULAR= option, the default value for *VSING* is 1E−8; otherwise, the default value is SINGULAR. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

COSAN Model Statement

COSAN *matrix_term* < + *matrix_term*. . . > **;**

where *matrix_term* represents
matrix_definition < ∗ *matrix_definition ...* >
and *matrix_definition* represents
matrix_name (column_number < *,general_form* < *,transformation* >> *)*

The COSAN statement constructs the symmetric matrix model for the covariance analysis mentioned earlier (see the section "The Generalized COSAN Model" on page 552):

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \cdots + \mathbf{F}_m\mathbf{P}_m\mathbf{F}_m',$$

$$\mathbf{F}_k = \mathbf{F}_{k_1} \cdots \mathbf{F}_{k_{n(k)}}, \qquad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1, \ldots, m$$

$$\mathbf{F}_{k_j} = \begin{cases} \mathbf{G}_{k_j} \\ \mathbf{G}_{k_j}^{-1} \\ (\mathbf{I} - \mathbf{G}_{k_j})^{-1} \end{cases} \quad j = 1, \ldots, n(k), \qquad \text{and} \quad \mathbf{P}_k = \begin{cases} \mathbf{Q}_k \\ \mathbf{Q}_k^{-1} \end{cases}$$

You can specify only one COSAN statement with each PROC CALIS statement. The COSAN statement contains m *matrix_terms* corresponding to the generalized COSAN formula. The *matrix_terms* are separated by plus signs (+) according to the addition of the terms within the model.

Each *matrix_term* of the COSAN statement contains the definitions of the first $n(k)+1$ matrices, $\mathbf{F}_{k_j}$ and $\mathbf{P}_k$, separated by asterisks (*) according to the multiplication of the matrices within the term. The matrices $\mathbf{F}'_k$ of the right-hand-side product are redundant and are not specified within the COSAN statement.

Each *matrix_definition* consists of the name of the matrix (*matrix_name*), followed in parentheses by the number of columns of the matrix (*column_number*) and, optionally, one or two matrix properties, separated by commas, describing the form of the matrix.

The number of rows of the first matrix in each term is defined by the input correlation or covariance matrix. You can reorder and reduce the variables in the input moment matrix using the VAR statement. The number of rows of the other matrices within the term is defined by the number of columns of the preceding matrix.

The first matrix property describes the general form of the matrix in the model. You can choose one of the following specifications of the first matrix property. The default first matrix property is GEN.

Code	**Description**
IDE	specifies an identity matrix; if the matrix is not square, this specification describes an identity submatrix followed by a rectangular zero submatrix.
ZID	specifies an identity matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by an identity submatrix.
DIA	specifies a diagonal matrix; if the matrix is not square, this specification describes a diagonal submatrix followed by a rectangular zero submatrix.
ZDI	specifies a diagonal matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by a diagonal submatrix.
LOW	specifies a lower triangular matrix; the matrix can be rectangular.
UPP	specifies an upper triangular matrix; the matrix can be rectangular.
SYM	specifies a symmetric matrix; the matrix cannot be rectangular.
GEN	specifies a general rectangular matrix (default).

The second matrix property describes the kind of inverse matrix transformation. If the second matrix property is omitted, no transformation is applied to the matrix.

Code	**Description**
INV	uses the inverse of the matrix.
IMI	uses the inverse of the difference between the identity and the matrix.

You cannot specify a nonsquare parameter matrix as an INV or IMI model matrix. Specifying a matrix of type DIA, ZDI, UPP, LOW, or GEN is not necessary if you do not use the *unspecified location* list in the corresponding MATRIX statements. After PROC CALIS processes the corresponding MATRIX statements, the matrix type DIA, ZDI, UPP, LOW, or GEN is recognized from the pattern of possibly nonzero elements. If you do not specify the first matrix property and you use the *unspecified location* list in a corresponding MATRIX statement, the matrix is recognized as a

GEN matrix. You can also generate an IDE or ZID matrix by specifying a DIA, ZDI, or IMI matrix and by using MATRIX statements that define the pattern structure. However, PROC CALIS would be unable to take advantage of the fast algorithms that are available for IDE and ZID matrices in this case.

For example, to specify a second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors and $m_2 = 2$ second-order factors and with $n = 9$ variables, you can use the following COSAN statement:

```
cosan F1(3) * F2(2) * P2(2,SYM)+F1(3) * U2(3,DIA) * I1(3,IDE)
      +U1(9,DIA) * I2(9,IDE)
```

MATRIX Statement

MATRIX *matrix-name* $<$ *location* $>$ = *list* $<$, *location* = *list* ... $>$ **;**

You can specify one or more MATRIX statements with a COSAN or FACTOR statement. A MATRIX statement specifies which elements of the matrix are constant and which are parameters. You can also assign values to the constant elements and initial values for the parameters. The input notation resembles that used in the COSAN program of R. McDonald and C. Fraser (personal communication), except that in PROC CALIS, parameters are distinguished from constants by giving parameters names instead of by using positive and negative integers.

A MATRIX statement cannot be used for an IDE or ZID matrix. For all other types of matrices, each element is assumed to be a constant of 0 unless a MATRIX statement specifies otherwise. Hence, there must be at least one MATRIX statement for each matrix mentioned in the COSAN statement except for IDE and ZID matrices. There can be more than one MATRIX statement for a given matrix. If the same matrix element is given different definitions, later definitions override earlier definitions.

At the start, all elements of each model matrix, except IDE or ZID matrices, are set equal to 0.

Description of *location*:

There are several ways to specify the starting *location* and continuation direction of a *list* with $n + 1$, $n \geq 0$, elements within the parameter matrix.

[*i* , *j*] The *list* elements correspond to the diagonally continued matrix elements [*i*,*j*] , [*i*+1,*j*+1] , ... , [*i*+*n*,*j*+*n*]. The number of elements is defined by the length of the list and eventually terminated by the matrix boundaries. If the list contains just one element (constant or variable), then it is assigned to the matrix element [*i*,*j*].

[*i* ,] The *list* elements correspond to the horizontally continued matrix elements [*i*,*j*], [*i*,*j*+1] , ... , [*i*,*j*+*n*], where the starting column *j* is the diagonal position

for a DIA, ZDI, or UPP matrix and is the first column for all other matrix types. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

[, *j*] The *list* elements correspond to the vertically continued matrix elements [*i*,*j*], [*i*+1,*j*] , ... , [*i*+*n*,*j*], where the starting row *i* is equal to the diagonal position for a DIA, ZDI, SYM, or LOW matrix and is the first row for each other matrix type. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

[,] unspecified location: The *list* is allocated to all valid matrix positions (except for a ZDI matrix) starting at the element [1,1] and continuing rowwise. The only valid matrix positions for a DIA or ZDI matrix are the diagonal elements; for an UPP or LOW matrix, the valid positions are the elements above or below the diagonal; and for a symmetric matrix, the valid positions are the elements in the lower triangle since the other triangle receives the symmetric allocation automatically. This *location* definition differs from the definitions with specified pattern locations in one important respect: if the number of elements in the *list* is smaller than the number of valid matrix elements, the list is repeated in the allocation process until all valid matrix elements are filled.

Omitting the left-hand-side term is equivalent to using [,] for an *unspecified location*.

Description of *list*:

The *list* contains numeric values or parameter names, or both, that are assigned to a list of matrix elements starting at a specified position and proceeding in a specified direction. A real number r in the list defines the corresponding matrix element as a constant element with this value. The notation $n * r$ generates n values of r in the list. A name in the list defines the corresponding matrix element as a parameter to be estimated. You can use numbered name lists (X1-X10) or the asterisk notation (5 *X means five occurrences of the parameter X). If a sublist of n_1 names inside a *list* is followed by a list of $n_2 \leq n_1$ real values inside parentheses, the last n_2 parameters in the name sublist are given the initial values mentioned inside the parenthesis. For example, the following *list*

```
0. 1. A2-A5 (1.4 1.9 2.5) 5.
```

specifies that the first two matrix elements (specified by the *location* to the left of the equal sign) are constants with values 0 and 1. The next element is parameter A2 with no specified initial value. The next three matrix elements are the variable parameters A3, A4, and A5 with initial values 1.4, 1.9, and 2.5, respectively. The next matrix element is specified by the seventh list element to be the constant 5.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. For example, if the prefix A (the parameter A1) is already used once in a *list*, the previous example would be identical to

```
0. 1. 4 * A: (1.4 1.9 2.5) 5.
```

To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

If you do not assign initial values to the parameters (listed in parentheses following a name sublist within the pattern list), PROC CALIS assigns initial values as follows:

- If the PROC CALIS statement contains a START=r option, each uninitialized parameter is given the initial value r. The uninitialized parameters in the diagonals of the central model matrices are given the initial value $10|r|$, $100|r|$, or $|r|$ multiplied by the value specified in the DEMPHAS= option.
- If the PROC CALIS statement contains a RANDOM=i option, each uninitialized parameter is given a random initial value $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- Otherwise, the initial value is set corresponding to START=0.5.

For example, to specify a confirmatory second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors, $m_2 = 2$ second-order factors, and $n = 9$ variables and the following matrix pattern,

$$\mathbf{F}_1 = \begin{pmatrix} X_1 & 0 & 0 \\ X_2 & 0 & 0 \\ X_3 & 0 & 0 \\ 0 & X_4 & 0 \\ 0 & X_5 & 0 \\ 0 & X_6 & 0 \\ 0 & 0 & X_7 \\ 0 & 0 & X_8 \\ 0 & 0 & X_9 \end{pmatrix} , \quad \mathbf{U}_1 = \begin{pmatrix} U_1 & & & & & & & & \\ & U_2 & & & & & & & \\ & & U_3 & & & & & & \\ & & & U_4 & & & & & \\ & & & & U_5 & & & & \\ & & & & & U_6 & & & \\ & & & & & & U_7 & & \\ & & & & & & & U_8 & \\ & & & & & & & & U_9 \end{pmatrix}$$

$$\mathbf{F}_2 = \begin{pmatrix} Y_1 & 0 \\ Y_1 & Y_2 \\ 0 & Y_2 \end{pmatrix} , \quad \mathbf{P}_2 = \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix} , \quad \mathbf{U}_2 = \begin{pmatrix} V_1 & & \\ & V_2 & \\ & & V_3 \end{pmatrix}$$

you can specify the following COSAN and MATRIX statements:

```
cosan f1(3) * f2(2) * p2(2,dia) + f1(3) * u2(3,dia) * i1(3,ide)
      + u1(9,dia) * i2(9,ide);
matrix f1
         [ ,1]= x1-x3,
         [ ,2]= 3 * 0 x4-x6,
         [ ,3]= 6 * 0 x7-x9;
matrix u1
         [1,1]=u1-u9;

matrix f2
         [ ,1]= 2 * y1,
         [ ,2]= 0. 2 * y2;
matrix u2 = 3 * v:;
matrix p2 = 2 * p;
run;
```

The matrix pattern includes several equality constraints. Two loadings in the first and second factor of $\mathbf{F}_2$ (parameter names Y1 and Y2) and the two factor correlations in the diagonal of matrix $\mathbf{P}_2$ (parameter name P) are constrained to be equal. There are many other ways to specify the same model. See Figure 19.2 for the path diagram of this model.

The MATRIX statement can also be used with the FACTOR model statement. See "Using the FACTOR and MATRIX Statements" on page 608 for the usage.

RAM Model Statement

RAM *list-entry* < , *list-entry* . . . > **;**

where *list-entry* represents
matrix-number row-number column-number <*value*><*parameter-name*>

The RAM statement defines the elements of the symmetric RAM matrix model

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

in the form of a list type input (McArdle and McDonald 1984).

The covariance structure is given by

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$ and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

You can specify only one RAM statement with each PROC CALIS statement. Using the RAM statement requires that the first n variable numbers in the path diagram and in the vector v correspond to the numbers of the n manifest variables of the given covariance or correlation matrix. If you are not sure what the order of the manifest variables in the DATA= data set is, use a VAR statement to specify the order of these observed variables. Using the AUGMENT option includes the INTERCEPT variable as a manifest variable with number $n + 1$ in the RAM model. In this case, latent variables have to start with $n + 2$. The box of each manifest variable in the path diagram is assigned the number of the variable in the covariance or correlation matrix.

The selection matrix **J** is always a rectangular identity (IDE) matrix, and it does not have to be specified in the RAM statement. A constant matrix element is defined in a RAM statement by a *list-entry* with four numbers. You define a parameter element by three or four numbers followed by a name for the parameter. Separate the list entries with a comma. Each *list-entry* in the RAM statement corresponds to a path in the diagram, as follows:

- The first number in each list entry (*matrix-number*) is the number of arrow heads of the path, which is the same as the number of the matrix in the RAM model (1 := **A** , 2 := **P**).
- The second number in each list entry (*row-number*) is the number of the node in the diagram to which the path points, which is the same as the row number of the matrix element.
- The third number in each list entry (*column-number*) is the number of the node in the diagram from which the path originates, which is the same as the column number of the matrix element.
- The fourth number (*value*) gives the (initial) value of the path coefficient. If you do not specify a fifth *list-entry*, this number specifies a constant coefficient; otherwise, this number specifies the initial value of this parameter. It is not necessary to specify the fourth item. If you specify neither the fourth nor the fifth item, the constant is set to 1 by default. If the fourth item (*value*) is not specified for a parameter, PROC CALIS tries to compute an initial value for this parameter.
- If the path coefficient is a parameter rather than a constant, then a fifth item in the list entry (*parameter-name*) is required to assign a name to the parameter. Using the same name for different paths constrains the corresponding coefficients to be equal.

If the initial value of a parameter is not specified in the list, the initial value is chosen in one of the following ways:

- If the PROC CALIS statement contains a RANDOM=i option, then the parameter obtains a randomly generated initial value r, such that $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values r multiplied by 10, 100, or the value specified in the DEMPHAS= option.

- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the START= option is used as an initial value.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify the confirmatory second-order factor analysis model (mentioned on page 595)

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

using the following RAM model statement.

```
ram
   1  1 10    x1,
   1  2 10    x2,
   1  3 10    x3,
   1  4 11    x4,
   1  5 11    x5,
   1  6 11    x6,
   1  7 12    x7,
   1  8 12    x8,
   1  9 12    x9,
   1 10 13    y1,
   1 11 13    y1,
   1 11 14    y2,
   1 12 14    y2,
   2  1  1    u:,
   2  2  2    u:,
   2  3  3    u:,
   2  4  4    u:,
   2  5  5    u:,
   2  6  6    u:,
   2  7  7    u:,
   2  8  8    u:,
   2  9  9    u:,
   2 10 10    v:,
   2 11 11    v:,
   2 12 12    v:,
   2 13 13    p ,
   2 14 14    p ;
run;
```

The confirmatory second-order factor analysis model corresponds to the path diagram displayed in Figure 19.2.

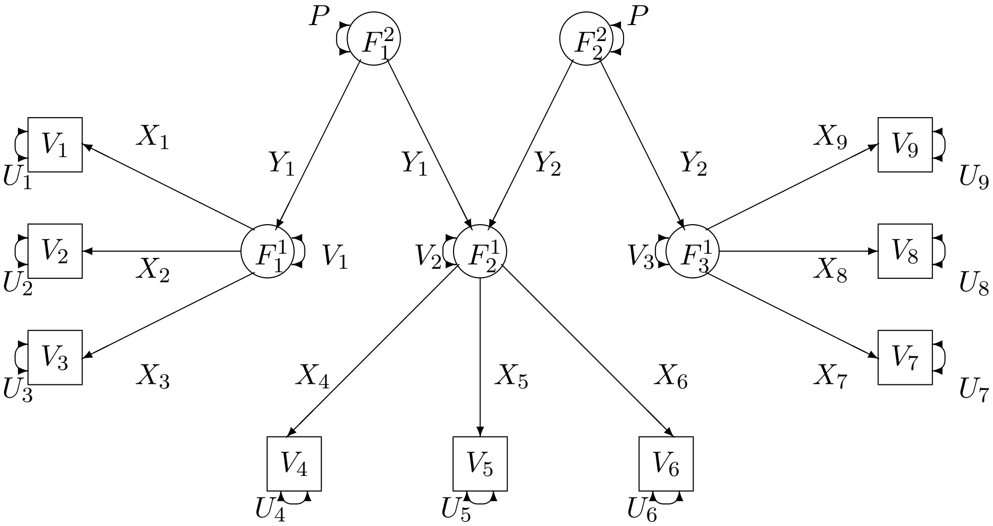
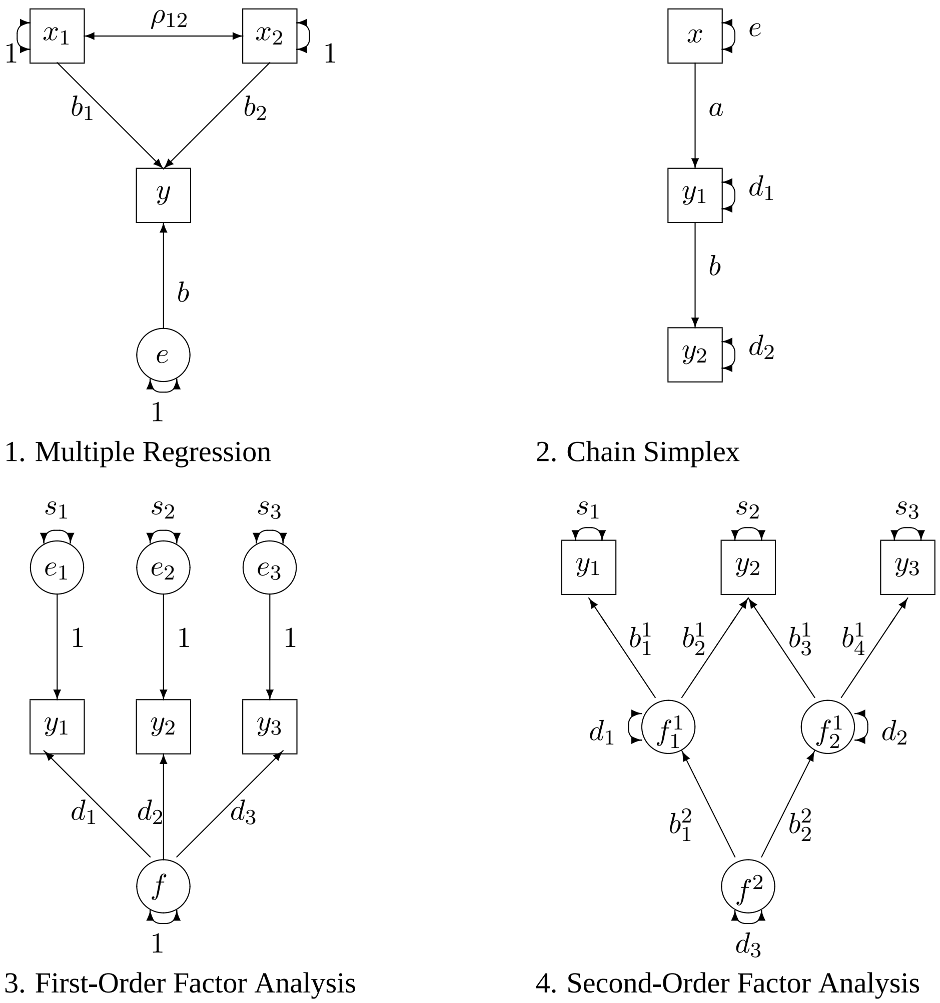


Figure 19.2. Path Diagram of Second-Order Factor Analysis Model

There is a very close relationship between the RAM model algebra and the specification of structural linear models by path diagrams. See Figure 19.3 for an example.



1. Multiple Regression

2. Chain Simplex

3. First-Order Factor Analysis

4. Second-Order Factor Analysis

Figure 19.3. Examples of RAM Nomography

Refer to McArdle (1980) for the interpretation of the models displayed in Figure 19.3.

LINEQS Model Statement

LINEQS *equation* < , *equation* . . . > **;**

where *equation* represents *dependent* = *term* < + *term* . . . >
and where *term* represents one of the following:

- *coefficient-name* < *(number)* > *variable-name*
- *prefix-name* < *(number)* > *variable-name*
- < *number* > *variable-name*

The LINEQS statement defines the LINEQS model

$$\begin{aligned}\boldsymbol{\eta} &= \boldsymbol{\beta\eta} + \boldsymbol{\gamma\xi} \\ \mathbf{C} &= \mathbf{J}(\mathbf{I} - \mathbf{B})^{-1}\boldsymbol{\Gamma\Phi\Gamma}'((\mathbf{I} - \mathbf{B})^{-1})'\mathbf{J}'\end{aligned}$$

You can specify only one LINEQS statement with each PROC CALIS statement. There are some differences from Bentler's notation in choosing the variable names. The length of each variable name is restricted to eight characters. The names of the manifest variables are defined in the DATA= input data set. The VAR statement can be used to select a subset of manifest variables in the DATA= input data set to analyze. You do not need to use a V prefix for manifest variables in the LINEQS statement nor do you need to use a numerical suffix in any variable name. The names of the latent variables must start with the prefix letter F (for Factor); the names of the residuals must start with the prefix letters E (for Error) or D (for Disturbance). The trailing part of the variable name can contain letters or digits. The prefix letter E is used for the errors of the manifest variables, and the prefix letter D is used for the disturbances of the latent variables. The names of the manifest variables in the DATA= input data set can start with F, E, or D, but these names should not coincide with the names of latent or error variables used in the model. The left-hand side (that is, endogenous *dependent* variable) of each equation should be either a manifest variable of the data set or a latent variable with prefix letter F. The left-hand-side variable should not appear on the right-hand side of the same equation; this means that matrix $\boldsymbol{\beta}$ should not have a nonzero diagonal element. Each equation should contain, at most, one E or D variable.

The equations must be separated by a comma. The order of the equations is arbitrary. The displayed output generally contains equations and terms in an order different from the input.

Coefficients to estimate are indicated in the equations by a name preceding the independent variable's name. The coefficient's name can be followed by a number inside parentheses indicating the initial value for this coefficient. A number preceding the independent variable's name indicates a constant coefficient. If neither a coefficient name nor a number precedes the independent variable's name, a constant coefficient of 1 is assumed.

If the initial value of a parameter is not specified in the equation, the initial value is chosen in one of the following ways:

- If you specify the RANDOM= option in the PROC CALIS statement, the variable obtains a randomly generated initial value r, such that $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the nonnegative random values r multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the START= option is used as an initial value.

In Bentler's notation, estimated coefficients are indicated by asterisks. Referring to a parameter in Bentler's notation requires the specification of two variable names that correspond to the row and column of the position of the parameter in the matrix. Specifying the estimated coefficients by parameter names makes it easier to impose additional constraints with code. You do not need any additional statements to express equality constraints. Simply specify the same name for parameters that should have equal values.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, you can specify confirmatory second-order factor analysis model (mentioned on page 595)

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

by using the LINEQS and STD statements:

```
lineqs
       V1 = X1 F1 + E1,
       V2 = X2 F1 + E2,
       V3 = X3 F1 + E3,
       V4 = X4 F2 + E4,
       V5 = X5 F2 + E5,
       V6 = X6 F2 + E6,
       V7 = X7 F3 + E7,
       V8 = X8 F3 + E8,
       V9 = X9 F3 + E9,
       F1 = Y1 F4 + D1,
       F2 = Y1 F4 + Y2 F5 + D2,
       F3 = Y2 F5 + D3;

std
       E1-E9 = 9 * U:,
       D1-D3 = 3 * V:,
       F4 F5 = 2 * P;
run;
```

STD Statement

STD *assignment* < , *assignment* . . . > **;**

where *assignment* represents *variables* = *pattern-definition*

The STD statement tells which variances are parameters to estimate and which are fixed. The STD statement can be used only with the LINEQS statement. You can specify only one STD statement with each LINEQS model statement. The STD statement defines the diagonal elements of the central model matrix $\mathbf{\Phi}$. These elements correspond to the variances of the exogenous variables and to the error variances of the endogenous variables. Elements that are not defined are assumed to be 0.

Each *assignment* consists of a variable list (*variables*) on the left-hand side and a pattern list (*pattern-definition*) on the right-hand side of an equal sign. The *assignments* in the STD statement must be separated by commas. The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the variance of the variable on the left-hand side in the same list position. A name on the right-hand side means that the corresponding variance is a parameter to estimate. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding variance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

The right-hand side can also contain prefixes. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names. For example, if the prefix A is not used in any previous statement, this STD statement

```
std E1-E6=6 * A: (6 * 3.) ;
```

defines the six error variances as free parameters $A1,\ldots,A6$, all with starting values of 3.

COV Statement

COV *assignment* < , *assignment* . . . > **;**

where *assignment* represents *variables* < ∗ *variables2* > = *pattern-definition*

The COV statement tells which covariances are parameters to estimate and which are fixed. The COV statement can be used only with the LINEQS statement. The COV statement differs from the STD statement only in the meaning of the left-hand-side *variables* list. You can specify only one COV statement with each LINEQS statement. The COV statement defines the off-diagonal elements of the central model matrix $\mathbf{\Phi}$. These elements correspond to the covariances of the exogenous variables and to the error covariances of the endogenous variables. Elements that are not defined are assumed to be 0. The *assignment*s in the COV statement must be separated by commas.

The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement, that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the covariance of a pair of variables in the list on the left-hand side. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding covariance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

You can use one of two alternatives to refer to parts of $\mathbf{\Phi}$. The first alternative uses only one variable list and refers to all distinct pairs of variables within the list. The second alternative uses two variable lists separated by an asterisk and refers to all pairs of variables among the two lists.

Within-List Covariances

Using *k* variable names in the *variables* list on the left-hand side of an equal sign in a COV statement means that the parameter list (*pattern-definition*) on the right-hand side refers to all $k(k-1)/2$ distinct variable pairs in the below-diagonal part of the $\mathbf{\Phi}$ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1-E4 = PHI1-PHI6 ;
```

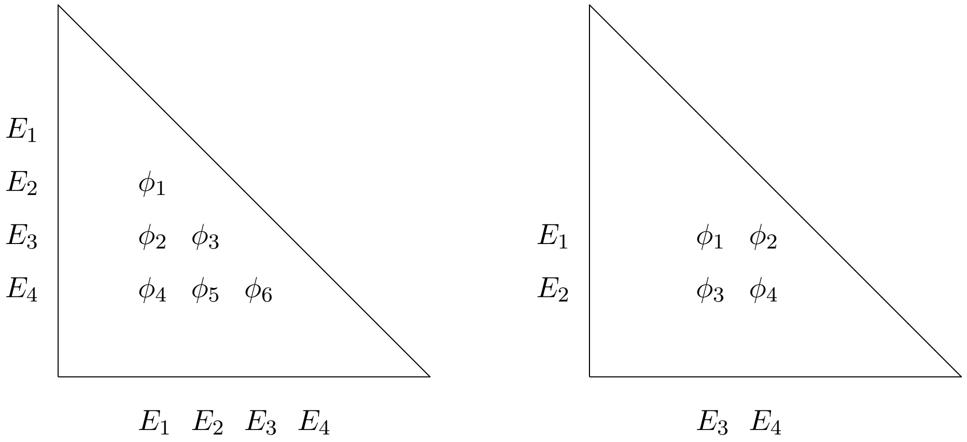
This is equivalent to the following specification:

```
COV E2 E1 = PHI1,
    E3 E1 = PHI2, E3 E2 = PHI3,
    E4 E1 = PHI4, E4 E2 = PHI5, E4 E3 = PHI6;
```

The symmetric elements are generated automatically. When you use prefix names on the right-hand sides, you do not have to count the exact number of parameters. For example,

```
COV E1-E4 = PHI: ;
```

generates the same list of parameter names if the prefix PHI is not used in a previous statement.



Within List Covariances

Between List Covariances

Figure 19.4. Within-List and Between-List Covariances

Between-List Covariances

Using k_1 and k_2 variable names in the two lists (separated by an asterisk) on the left-hand side of an equal sign in a COV statement means that the parameter list on the right-hand side refers to all $k_1 \times k_2$ distinct variable pairs in the $\mathbf{\Phi}$ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1 E2 * E3 E4 = PHI1-PHI4 ;
```

This is equivalent to the following specification:

```
COV  E1 E3 = PHI1, E1 E4 = PHI2,
     E2 E3 = PHI3, E2 E4 = PHI4;
```

The symmetric elements are generated automatically.

Using prefix names on the right-hand sides lets you achieve the same purpose without counting the number of parameters. That is,

```
   COV  E1 E2 * E3 E4 = PHI: ;
```

FACTOR Model Statement

FACTOR < *options* > **;**

You can use the FACTOR statement to specify an exploratory or confirmatory first-order factor analysis of the given covariance or correlation matrix **C**,

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = diag$$

or

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}'$$

where **U** is a diagonal matrix and **P** is symmetric. Within this section, n denotes the number of manifest variables corresponding to the rows and columns of matrix **C**, and m denotes the number of latent variables (factors or components) corresponding to the columns of the loading matrix **F**.

You can specify only one FACTOR statement with each PROC CALIS statement. You can specify higher-order factor analysis problems using a COSAN model specification. PROC CALIS requires more computing time and memory than PROC FACTOR because it is designed for more general structural estimation problems and is unable to exploit the special properties of the unconstrained factor analysis model.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method (if possible). For details, see "Using the FACTOR and MATRIX Statements" on page 608. If neither of the two methods are appropriate, the initial values are set by the START= option.

The unrestricted factor analysis model is not identified because any orthogonal rotated factor loading matrix $\tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}$ is equivalent to the result **F**,

$$\mathbf{C} = \tilde{\mathbf{F}}\tilde{\mathbf{F}}' + \mathbf{U}, \qquad \tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}, \quad \text{where} \quad \boldsymbol{\Theta}'\boldsymbol{\Theta} = \boldsymbol{\Theta}\boldsymbol{\Theta}' = \mathbf{I}$$

To obtain an identified factor solution, the FACTOR statement imposes zero constraints on the $m(m-1)/2$ elements in the upper triangle of **F** by default.

The following options are available in the FACTOR statement.

COMPONENT | COMP

computes a component analysis instead of a factor analysis (the diagonal matrix $\mathbf{U}$ in the model is set to 0). Note that the rank of $\mathbf{FF}'$ is equal to the number m of components in $\mathbf{F}$. If m is smaller than the number of variables in the moment matrix $\mathbf{C}$, the matrix of predicted model values is singular and maximum likelihood estimates for $\mathbf{F}$ cannot be computed. You should compute ULS estimates in this case.

HEYWOOD | HEY

constrains the diagonal elements of $\mathbf{U}$ to be nonnegative; in other words, the model is replaced by

$$\mathbf{C} = \mathbf{FF}' + \mathbf{U}^2, \quad \mathbf{U} = diag$$

N = *m*

specifies the number of first-order factors or components. The number m of factors should not exceed the number n of variables in the covariance or correlation matrix analyzed. For the saturated model, $m = n$, the COMP option should generally be specified for $\mathbf{U} = 0$; otherwise, $df < 0$. For $m = 0$ no factor loadings are estimated, and the model is $\mathbf{C} = \mathbf{U}$, with $\mathbf{U} = diag$. By default, $m = 1$.

NORM

normalizes the rows of the factor pattern for rotation using Kaiser's normalization.

RCONVERGE=*p*

RCONV=*p*

specifies the convergence criterion for rotation cycles. The option is applicable to rotation using either the QUARTIMAX, VARIMAX, EQUAMAX, or PARSIMAX method in the ROTATE= option. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E-9 by default and is modified by the RCONVERGE= value.

RITER=*n*

specifies the maximum number of cycles n for factor rotation using either the QUARTIMAX, VARIMAX, EQUAMAX, or PARSIMAX method in the ROTATE= option. The default n is the maximum between 100 and 10 times of the number of variables.

ROTATE | R = *name*

specifies an orthogonal rotation. By default, ROTATE=NONE. The possible values for *name* are as follows:

PRINCIPAL | PC specifies a principal axis rotation. If ROTATE=PRINCIPAL is used with a factor rather than a component model, the following rotation is performed:

$$\mathbf{F}_{new} = \mathbf{F}_{old}\mathbf{T}, \quad \text{with} \quad \mathbf{F}'_{old}\mathbf{F}_{old} = \mathbf{T}\boldsymbol{\Lambda}\mathbf{T}'$$

where the columns of matrix $\mathbf{T}$ contain the eigenvectors of $\mathbf{F}'_{old}\mathbf{F}_{old}$.

QUARTIMAX | Q specifies quartimax rotation.

VARIMAX | V specifies varimax rotation.

EQUAMAX | E specifies equamax rotation.

PARSIMAX | P specifies parsimax rotation.

NONE performs no rotation (default).

Using the FACTOR and MATRIX Statements

You can specify the MATRIX statement and the FACTOR statement to compute a confirmatory first-order factor or component analysis. You can define the elements of the matrices $\mathbf{F}$, $\mathbf{P}$, and $\mathbf{U}$ of the oblique model,

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = diag$$

To specify the structure for matrix $\mathbf{F}$, $\mathbf{P}$, or $\mathbf{U}$, you have to refer to the matrix _F_ , _P_ , or _U_ in the MATRIX statement. Matrix names automatically set by PROC CALIS always start with an underscore. As you name your own matrices or variables, you should avoid leading underscores.

The default matrix forms are as follows.

_F_ lower triangular matrix (0 upper triangle for problem identification, removing rotational invariance)

_P_ identity matrix (constant)

_U_ diagonal matrix

For details about specifying the elements in matrices, see the section "MATRIX Statement" on page 593. If you are using at least one MATRIX statement in connection with a FACTOR model statement, you can also use the BOUNDS or PARAMETERS statement and program statements to constrain the parameters named in the MATRIX statement. Initial estimates are computed by McDonald's (McDonald and Hartmann 1992) method. McDonald's method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than by setting the loadings of the reference variables equal to 1.

BOUNDS Statement

BOUNDS *constraint* < , *constraint* . . . > **;**

where *constraint* represents
< *number operator* > *parameter-list* < *operator number* >

You can use the BOUNDS statement to define boundary constraints for any parameter that has its name specified in a MATRIX, LINEQS, STD, COV, or RAM statement or that is used in the model of an INRAM= data set. Valid operators are $<=$, $<$, $>=$, $>$, and $=$ or, equivalently, LE, LT, GE, GT, and EQ. The following is an example of the BOUNDS statement:

```
bounds         0.   <= a1-a9 x    <= 1. ,
              -1.   <= c2-c5            ,
                       b1-b10 y   >= 0. ;
```

You must separate boundary constraints with a comma, and you can specify more than one BOUNDS statement. The feasible region for a parameter is the intersection of all boundary constraints specified for that parameter; if a parameter has a maximum lower boundary constraint larger than its minimum upper bound, the parameter is set equal to the minimum of the upper bounds.

If you need to compute the values of the upper or lower bounds, create a TYPE=EST data set containing _TYPE_='UPPERBD' or _TYPE_='LOWERBD' observations and use it as an INEST= or INVAR= input data set in a later PROC CALIS run.

The BOUNDS statement can contain only parameter names and numerical constants. You cannot use the names of variables created in program statements.

The active set strategies made available in PROC CALIS cannot realize the strict inequality constraints $<$ or $>$. For example, you cannot specify `BOUNDS x > 0;` to prevent infinite values for $y = log(x)$. Use `BOUNDS x > 1E-8;` instead.

If the CALIS procedure encounters negative diagonal elements in the central model matrices during the minimization process, serious convergence problems can occur. You can use the BOUNDS statement to constrain these parameters to nonnegative values. Using negative values in these locations can lead to a smaller χ^2 value but uninterpretable estimates.

LINCON Statement

LINCON *constraint* < , *constraint* . . . > **;**

where *constraint* represents
number operator linear-term or
linear-term operator number ,
and *linear-term* is
<+|-><*coefficient*$*$ > *parameter* <<+|-><*coefficient*$*$ > *parameter*. . . >

The LINCON statement specifies a set of linear equality or inequality constraints of the form

$$\sum_{j=1}^{n} a_{ij}x_j \leq b_i, \quad i = 1, \ldots, m$$

The constraints must be separated by commas. Each linear constraint i in the statement consists of a linear combination $\sum_j a_{ij}x_j$ of a subset of the n parameters $x_j, j = 1, \ldots, n$, and a constant value b_i separated by a comparison operator. Valid operators are $<=$, $<$, $>=$, $>$, and $=$ or, equivalently, LE, LT, GE, GT, and EQ. PROC CALIS cannot enforce the strict inequalities $<$ or $>$. Note that the coefficients a_{ij} in the linear combination must be constant numbers and must be followed by an asterisk and the name of a parameter (for example, listed in the PARMS, STD or COV statement). The following is an example of the LINCON statement that sets a linear constraint on parameters x1 and x2:

```
lincon      x1 + 3 * x2 <= 1;
```

Although you can easily express boundary constraints in LINCON statements, for many applications it is much more convenient to specify both the BOUNDS and the LINCON statements in the same PROC CALIS call.

The LINCON statement can contain only parameter names, operators, and numerical constants. If you need to compute the values of the coefficients a_{ij} or right-hand sides b_i, you can run a preliminary DATA step and create a TYPE=EST data set containing _TYPE_='LE', _TYPE_='GE', or _TYPE_='EQ' observations, then specify this data set as an INEST= or INVAR= data set in a following PROC CALIS run.

NLINCON Statement

NLINCON | NLC *constraint* < , *constraint* . . . > **;**

where *constraint* represents
number operator variable-list number operator or
variable-list operator number or
number operator variable-list

You can specify nonlinear equality and inequality constraints with the NLINCON or NLC statement. The QUANEW optimization subroutine is used when you specify nonlinear constraints using the NLINCON statement.

The syntax of the NLINCON statement is similar to that of the BOUNDS statement, except that the NLINCON statement must contain the names of variables that are defined in the program statements and are defined as continuous functions of parameters in the model. They must not be confused with the variables in the data set.

As with the BOUNDS statement, one- or two-sided constraints are allowed in the NLINCON statement; equality constraints must be one sided. Valid operators are $<=$, $<$, $>=$, $>$, and $=$ or, equivalently, LE, LT, GE, GT, and EQ.

PROC CALIS cannot enforce the strict inequalities $<$ or $>$ but instead treats them as $<=$ and $>=$, respectively. The listed nonlinear constraints must be separated by commas. The following is an example of the NLINCON statement that constrains the nonlinear parametric function $x_1 * x_1 + u_1$, which is defined below in a program statement, to a fixed value of 1:

```
nlincon    xx = 1;
xx = x1 * x1 + u1;
```

Note that x1 and u1 are parameters defined in the model. The following three NLINCON statements, which require xx1, xx2, and xx3 to be between zero and ten, are equivalent:

```
nlincon  0. <= xx1-xx3,
               xx1-xx3 <= 10;
nlincon 0. <= xx1-xx3 <= 10.;
nlincon 10. >= xx1-xx3 >= 0.;
```

NLOPTIONS Statement

NLOPTIONS *option(s)* **;**

Many options that are available in PROC NLP can now be specified for the optimization subroutines in PROC CALIS using the NLOPTIONS statement. The NLOPTIONS statement provides more displayed and file output on the results of the optimization process, and it permits the same set of termination criteria as in PROC NLP. These are more technical options that you may not need to specify in most cases. The available options are summarized in Table 19.2 through Table 19.4, and the options are described in detail in the following three sections.

Table 19.2. Options Documented in the PROC CALIS Statement

Option	Short Description
Estimation Methods	
G4=*i*	algorithm for computing STDERR
Optimization Techniques	
TECHNIQUE=*name*	minimization method
UPDATE=*name*	update technique
LINESEARCH=*i*	line-search method
FCONV=*r*	relative change function convergence criterion
GCONV=*r*	relative gradient convergence criterion
INSTEP=*r*	initial step length (SALPHA=, RADIUS=)
LSPRECISION=*r*	line-search precision
MAXFUNC=*i*	maximum number of function calls
MAXITER=*i* <*n*>	maximum number of iterations
Miscellaneous Options	

Table 19.2. Options Documented in the PROC CALIS Statement (continued)

Option	**Short Description**
ASINGULAR=*r*	absolute singularity criterion for inversion of the information matrix
COVSING=*r*	singularity tolerance of the information matrix
MSINGULAR=*r*	relative M singularity criterion for inversion of the information matrix
SINGULAR=*r*	singularity criterion for inversion of the Hessian
VSINGULAR=*r*	relative V singularity criterion for inversion of the information matrix

Table 19.3. Termination Criteria Options

Option	**Short Description**
Options Used by All Techniques	
ABSCONV=*r*	absolute function convergence criterion
MAXFUNC=*i*	maximum number of function calls
MAXITER=*i* <*n*>	maximum number of iterations
MAXTIME=*r*	maximum CPU time
MINITER=*i*	minimum number of iterations
Options for Unconstrained and Linearly Constrained Techniques	
ABSFCONV=*r* <*n*>	absolute change function convergence criterion
ABSGCONV=*r* <*n*>	absolute gradient convergence criterion
ABSXCONV=*r* <*n*>	absolute change parameter convergence criterion
FCONV=*r* <*n*>	relative change function convergence criterion
FCONV2=*r* <*n*>	function convergence criterion
FDIGITS=*r*	precision in computation of the objective function
FSIZE=*r*	parameter for FCONV= and GCONV=
GCONV=*r* <*n*>	relative gradient convergence criterion
GCONV2=*r* <*n*>	relative gradient convergence criterion
XCONV=*r* <*n*>	relative change parameter convergence criterion
XSIZE=*r*	parameter for XCONV=
Options for Nonlinearly Constrained Techniques	
ABSGCONV=*r* <*n*>	maximum absolute gradient of Lagrange function criterion
FCONV2=*r* <*n*>	predicted objective function reduction criterion
GCONV=*r* <*n*>	normalized predicted objective function reduction criterion

Table 19.4. Miscellaneous Options

Option	**Short Description**
Options for the Approximate Covariance Matrix of Parameter Estimates	
CFACTOR=*r*	scalar factor for STDERR

Table 19.4. Miscellaneous Options (continued)

Option	Short Description
NOHLF	use Hessian of the objective function for STDERR
Options for Additional Displayed Output	
PALL	display initial and final optimization values
PCRPJAC	display approximate Hessian matrix
PHESSIAN	display Hessian matrix
PHISTORY	display optimization history
PINIT	display initial values and derivatives (PALL)
PNLCJAC	display Jacobian matrix of nonlinear constraints (PALL)
PRINT	display results of the optimization process
Additional Options for Optimization Techniques	
DAMPSTEP< =*r* >	controls initial line-search step size
HESCAL=*n*	scaling version of Hessian or Jacobian
LCDEACT=*r*	Lagrange multiplier threshold of constraint
LCEPSILON=*r*	range for boundary and linear constraints
LCSINGULAR=*r*	QR decomposition linear dependence criterion
NOEIGNUM	suppress computation of matrices
RESTART=*i*	restart algorithm with a steepest descent direction
VERSION=1 | 2	quasi-Newton optimization technique version

Options Documented in the PROC CALIS Statement

The following options are the same as in the PROC CALIS statement and are documented in the section "PROC CALIS Statement" on page 568.

Estimation Method Option

G4=*i*

specifies the method for computing the generalized (G2 or G4) inverse of a singular matrix needed for the approximate covariance matrix of parameter estimates. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

Optimization Technique Options

TECHNIQUE | TECH=*name*
OMETHOD | OM=*name*

specifies the optimization technique.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

LINESEARCH | LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques.

FCONV | FTOL=*r*

specifies the relative function convergence criterion. For more details, see the section "Termination Criteria Options" on page 615.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion. For more details, see the section "Termination Criteria Options" on page 615.

INSTEP | SALPHA | RADIUS=*r*

restricts the step length of an optimization algorithm during the first iterations.

LSPRECISION | LSP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3.

MAXFUNC | MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. For more details, see the section "Termination Criteria Options" on page 615.

MAXITER | MAXIT=*i* <*n*>

specifies the maximum number i of iterations in the optimization process. For more details, see the section "Termination Criteria Options" on page 615.

Miscellaneous Options

ASINGULAR | ASING=*r*

specifies an absolute singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

COVSING=*r*

specifies a nonnegative threshold r, $r > 0$, that decides whether the eigenvalues of the information matrix are considered to be zero. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

MSINGULAR | MSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

SINGULAR | SING =*r*

specifies the singularity criterion r, $0 \leq r \leq 1$, that is used for the inversion of the Hessian matrix. The default value is 1E$-$8.

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

Termination Criteria Options

Let x^* be the point at which the objective function $f(\cdot)$ is optimized, and let $x^{(k)}$ be the parameter values attained at the kth iteration. All optimization techniques stop at the kth iteration if at least one of a set of termination criteria is satisfied. The specified termination criteria should allow termination in an area of sufficient size around x^*. You can avoid termination respective to any of the following function, gradient, or parameter criteria by setting the corresponding option to zero. There is a default set of termination criteria for each optimization technique; most of these default settings make the criteria ineffective for termination. PROC CALIS may have problems due to rounding errors (especially in derivative evaluations) that prevent an optimizer from satisfying strong termination criteria.

Note that PROC CALIS also terminates if the point $x^{(k)}$ is fully constrained by linearly independent active linear or boundary constraints, and all Lagrange multiplier estimates of active inequality constraints are greater than a small negative tolerance.

The following options are available only in the NLOPTIONS statement (except for FCONV, GCONV, MAXFUNC, and MAXITER), and they affect the termination criteria.

Options Used by All Techniques

The following five criteria are used by all optimization techniques.

ABSCONV | ABSTOL=*r*

specifies an absolute function convergence criterion.

- For minimization, termination requires

$$f^{(k)} = f(x^{(k)}) \leq ABSCONV$$

- For maximization, termination requires

$$f^{(k)} = f(x^{(k)}) \geq ABSCONV$$

The default value of ABSCONV is

- for minimization, the negative square root of the largest double precision value
- for maximization, the positive square root of the largest double precision value

MAXFUNC | MAXFU=*i*

requires the number of function calls to be no larger than i. The default values are listed in the following table.

TECH=	MAXFUNC default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=125
DBLDOG, QUANEW	i=500
CONGRA	i=1000

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT= *i* <*n*>

requires the number of iterations to be no larger than i. The default values are listed in the following table.

TECH=	MAXITER default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=50
DBLDOG, QUANEW	i=200
CONGRA	i=400

The default is used if you specify MAXITER=0 or you omit the MAXITER option.

The optional second value n is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default value is n=20. For example, specifying `MAXITER= .  0` means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

MAXTIME=*r*

requires the CPU time to be no larger than r. The default value of the MAXTIME= option is the largest double floating point number on your computer.

MINITER | MINIT=*i*

specifies the minimum number of iterations. The default value is $i = 0$.

The ABSCONV=, MAXITER=, MAXFUNC=, and MAXTIME= options are useful for dividing a time-consuming optimization problem into a series of smaller problems by using the OUTEST= and INEST= data sets.

Options for Unconstrained and Linearly Constrained Techniques

This section contains additional termination criteria for all unconstrained, boundary, or linearly constrained optimization techniques.

ABSFCONV | ABSFTOL=*r <n>*

specifies the absolute function convergence criterion. Termination requires a small change of the function value in successive iterations,

$$|f(x^{(k-1)}) - f(x^{(k)})| \le r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV | ABSGTOL=*r <n>*

specifies the absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

$$\max_j |g_j^{(k)}| \le r$$

The default value is r=1E$-$5. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: In some applications, the small default value of the ABSGCONV= criterion is too difficult to satisfy for some of the optimization techniques.

ABSXCONV | ABSXTOL=*r <n>*

specifies the absolute parameter convergence criterion. Termination requires a small Euclidean distance between successive parameter vectors,

$$\parallel x^{(k)} - x^{(k-1)} \parallel_2 \le r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV | FTOL=*r <n>*

specifies the relative function convergence criterion. Termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \le r$$

where $FSIZE$ is defined by the FSIZE= option. The default value is $r = 10^{-FDIGITS}$, where $FDIGITS$ either is specified or is set by default to $-log_{10}(\epsilon)$, where ϵ is the machine precision. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2=*r <n>*

specifies another function convergence criterion. For least-squares problems, termination requires a small predicted reduction

$$df^{(k)} \approx f(x^{(k)}) - f(x^{(k)} + s^{(k)})$$

of the objective function.

The predicted reduction

$$\begin{aligned} df^{(k)} &= -g^{(k)'} s^{(k)} - \frac{1}{2} s^{(k)'} G^{(k)} s^{(k)} \\ &= -\frac{1}{2} s^{(k)'} g^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step

$$s^{(k)} = -G^{(k)-1} g^{(k)}$$

The FCONV2 criterion is the unscaled version of the GCONV criterion. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FDIGITS=*r*

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -log_{10}\epsilon$, where ϵ is the machine precision. The value of r is used for the specification of the default value of the FCONV= option.

FSIZE=*r*

specifies the $FSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the FCONV= and GCONV= options.

GCONV | GTOL=*r <n>*

specifies the relative gradient convergence criterion. For all techniques except the CONGRA technique, termination requires that the normalized predicted function reduction is small,

$$\frac{[g^{(k)}]'[G^{(k)}]^{-1} g^{(k)}}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate **G** is not available),

$$\frac{\parallel g^{(k)} \parallel_2^2 \quad \parallel s^{(k)} \parallel_2}{\parallel g^{(k)} - g^{(k-1)} \parallel_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is r=1E-8. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Note: The default setting for the GCONV= option sometimes leads to early termination far from the location of the optimum. This is especially true for the special form of this criterion used in the CONGRA optimization.

GCONV2 | GTOL2=*r <n>*

specifies another relative gradient convergence criterion. For least-squares problems and the TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the criterion of Browne (1982) is used,

$$\max_j \frac{|g_j^{(k)}|}{\sqrt{f(x^{(k)})G_{j,j}^{(k)}}} \leq r$$

This criterion is not used by the other techniques. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XCONV | XTOL=*r <n>*

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

$$\frac{\max_j |x_j^{(k)} - x_j^{(k-1)}|}{\max(|x_j^{(k)}|, |x_j^{(k-1)}|, XSIZE)} \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE=*r*

specifies the $XSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the XCONV= option.

Options for Nonlinearly Constrained Techniques

The non-NMSIMP algorithms available for nonlinearly constrained optimization (currently only TECH=QUANEW) do not monotonically reduce either the value of the objective function or some kind of merit function that combines objective and constraint functions. Furthermore, the algorithm uses the watchdog technique with backtracking (Chamberlain et al., 1982). Therefore, no termination criteria are implemented that are based on the values (x or f) of successive iterations. In addition to the criteria used by all optimization techniques, only three more termination criteria are currently available, and they are based on the Lagrange function

$$L(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x)$$

and its gradient

$$\nabla_x L(x, \lambda) = g(x) - \sum_{i=1}^{m} \lambda_i \nabla_x c_i(x)$$

Here, m denotes the total number of constraints, $g = g(x)$ denotes the gradient of the objective function, and λ denotes the m vector of Lagrange multipliers. The Kuhn-Tucker conditions require that the gradient of the Lagrange function is zero at the optimal point (x^*, λ^*):

$$\nabla_x L(x^*, \lambda^*) = 0$$

The termination criteria available for nonlinearly constrained optimization follow.

ABSGCONV | ABSGTOL=*r <n>*

specifies that termination requires the maximum absolute gradient element of the Lagrange function to be small,

$$\max_j |\{\nabla_x L(x^{(k)}, \lambda^{(k)})\}_j| \le r$$

The default value is r=1E$-$5. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2=*r <n>*

specifies that termination requires the predicted objective function reduction to be small:

$$|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i| \le r$$

The default value is r=1E$-$6. This is the criterion used by the programs VMCWD and VF02AD (Powell 1982b). The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

GCONV | GTOL=*r <n>*

specifies that termination requires the normalized predicted objective function reduction to be small:

$$\frac{|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|}{\max(|f(x^{(k)})|, FSIZE)} \le r$$

where $FSIZE$ is defined by the FSIZE= option. The default value is r=1E$-$8. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Miscellaneous Options

Options for the Approximate Covariance Matrix of Parameter Estimates

You can specify the following options to modify the approximate covariance matrix of parameter estimates.

CFACTOR=*r*

specifies the scalar factor for the covariance matrix of parameter estimates. The scalar $r \geq 0$ replaces the default value c/NM. For more details, see the section "Approximate Standard Errors" on page 648.

NOHLF

specifies that the Hessian matrix of the objective function (rather than the Hessian matrix of the Lagrange function) is used for computing the approximate covariance matrix of parameter estimates and, therefore, the approximate standard errors.

It is theoretically not correct to use the NOHLF option. However, since most implementations use the Hessian matrix of the objective function and not the Hessian matrix of the Lagrange function for computing approximate standard errors, the NOHLF option can be used to compare the results.

Options for Additional Displayed Output

You can specify the following options to obtain additional displayed output.

PALL | ALL

displays information on the starting values and final values of the optimization process.

PCRPJAC | PJTJ

displays the approximate Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected approximate Hessian matrix is also displayed.

PHESSIAN | PHES

displays the Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected Hessian matrix is also displayed.

PHISTORY | PHIS

displays the optimization history. The PHISTORY option is set automatically if the PALL or PRINT option is set.

PINIT | PIN

displays the initial values and derivatives (if available). The PINIT option is set automatically if the PALL option is set.

PNLCJAC

displays the Jacobian matrix of nonlinear constraints specified by the NLINCON statement. The PNLCJAC option is set automatically if the PALL option is set.

PRINT | PRI

displays the results of the optimization process, such as parameter estimates and constraints.

More Options for Optimization Techniques

You can specify the following options, in addition to the options already listed, to fine-tune the optimization process. These options should not be necessary in most applications of PROC CALIS.

DAMPSTEP | DS *<=r>*

specifies that the initial step-size value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP techniques) cannot be larger than r times the step-size value used in the former iteration. If the factor r is not specified, the default value is $r = 2$. The DAMPSTEP option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they can lead to floating point overflows during the computation of objective functions and their derivatives. The DAMPSTEP<=r> option can prevent time-costly function calls during line searches with very small step sizes α of objective functions. For more information on setting the start values of each line search, see the section "Restricting the Step Length" on page 672.

HESCAL | HS = 0 | 1 | 2 | 3

specifies the scaling version of the Hessian or crossproduct Jacobian matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization. If HS is not equal to zero, the first iteration and each restart iteration sets the diagonal scaling matrix $\mathbf{D}^{(0)} = diag(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|\mathbf{G}_{i,i}^{(0)}|, \epsilon)}$$

where $\mathbf{G}_{i,i}^{(0)}$ are the diagonal elements of the Hessian or crossproduct Jacobian matrix. In every other iteration, the diagonal scaling matrix $\mathbf{D}^{(0)} = diag(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling is done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max(d_i^{(k)}, \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)})$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max(0.6 * d_i^{(k)}, \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)})$$

HS=3 specifies that d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|\mathbf{G}_{i,i}^{(k)}|, \epsilon)}$$

In the preceding equations, ϵ is the relative machine precision. The default is HS=1 for LEVMAR minimization and HS=0 otherwise. Scaling of the Hessian or crossproduct Jacobian can be time-consuming in the case where general linear constraints are active.

LCDEACT | LCD = ***r***

specifies a threshold r for the Lagrange multiplier that decides whether an active inequality constraint remains active or can be deactivated. For maximization, r must be greater than zero; for minimization, r must be smaller than zero. The default is

$$r = \pm \min(0.01, \max(0.1 * ABSGCONV, 0.001 * gmax^{(k)}))$$

where "$+$" stands for maximization, "$-$" stands for minimization, $ABSGCONV$ is the value of the absolute gradient criterion, and $gmax^{(k)}$ is the maximum absolute element of the (projected) gradient $g^{(k)}$ or $Z'g^{(k)}$.

LCEPSILON | LCEPS | LCE = ***r***

specifies the range r, $r \geq 0$, for active and violated boundary and linear constraints. If the point $x^{(k)}$ satisfies the condition

$$|\sum_{j=1}^{n} a_{ij}x_j^{(k)} - b_i| \leq r * (|b_i| + 1)$$

the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. The default value is r=1E-8. During the optimization process, the introduction of rounding errors can force PROC NLP to increase the value of r by factors of 10. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR | LCSING | LCS = ***r***

specifies a criterion r, $r \geq 0$, used in the update of the QR decomposition that decides whether an active constraint is linearly dependent on a set of other active constraints. The default is r=1E-8. The larger r becomes, the more the active constraints are recognized as being linearly dependent.

NOEIGNUM

suppresses the computation and displayed output of the determinant and the inertia of the Hessian, crossproduct Jacobian, and covariance matrices. The inertia of a symmetric matrix are the numbers of negative, positive, and zero eigenvalues. For large applications, the NOEIGNUM option can save computer time.

RESTART | REST = ***i***

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after at most i iterations, $i > 0$. Default values are as follows:

- CONGRA: UPDATE=PB: restart is done automatically so specification of i is not used.
- CONGRA: UPDATE≠PB: $i = \min(10n, 80)$, where n is the number of parameters.

- QUANEW: i is the largest integer available.

VERSION | VS = 1 | 2

specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

VS=1 specifies the update of the μ vector as in Powell (1978a, 1978b) (update like VF02AD).

VS=2 specifies the update of the μ vector as in Powell (1982a, 1982b) (update like VMCWD).

The default is VS=2.

PARAMETERS Statement

PARAMETERS | PARMS *parameter(s)* << = > *number(s)* >
<< , > *parameter(s)* << = > *number(s)* > . . . > **;**

The PARAMETERS statement defines additional parameters that are not elements of a model matrix to use in your own program statements. You can specify more than one PARAMETERS statement with each PROC CALIS statement. The *parameters* can be followed by an equal sign and a number list. The values of the *numbers* list are assigned as initial values to the preceding parameters in the *parameters* list. For example, each of the following statements assigns the initial values ALPHA=.5 and BETA=-.5 for the parameters used in program statements:

```
parameters alfa beta=.5 -.5;
parameters alfa beta (.5 -.5);
parameters alfa beta .5 -.5;
parameters alfa=.5 beta (-.5);
```

The number of parameters and the number of values does not have to match. When there are fewer values than parameter names, either the RANDOM= or START= option is used. When there are more values than parameter names, the extra values are dropped. Parameters listed in the PARAMETERS statement can be assigned initial values by program statements or by the START= or RANDOM= option in the PROC CALIS statement.

Caution: The OUTRAM= and INRAM= data sets do not contain any information about the PARAMETERS statement or additional program statements.

STRUCTEQ Statement

STRUCTEQ *variable* < *variable* . . . > **;**

The STRUCTEQ statement is used to list the dependent variables of the structural equations. This statement is ignored if you omit the PDETERM option. This statement is useful because the term *structural equation* is not defined in a unique way, and PROC CALIS has difficulty identifying the structural equations.

If LINEQS statements are used, the names of the left-hand-side (dependent) variables of those equations to be treated as structural equations should be listed in the STRUCTEQ statement.

If the RAM statement is used, variable names in the STRUCTEQ statements depend on the VARNAMES statement:

- If the VARNAMES statement is used, variable names must correspond to those in the VARNAMES statement.
- If the VARNAMES statement is not used, variable names must correspond to the names of manifest variables or latent (F) variables.

The STRUCTEQ statement also defines the names of variables used in the causal coefficient matrix of the structural equations, $\mathbf{B}$, for computing the *Stability Coefficient of Reciprocal Causation* (the largest eigenvalue of the $\mathbf{BB}'$ matrix). If the PROC CALIS option PDETERM is used without the STRUCTEQ statement, the structural equations are defined as described in the PDETERM option. See the PROC CALIS option PDETERM on page 585 for more details.

VARNAMES Statement

VARNAMES | VNAMES *assignment* < *, assignment* . . . > **;**

where *assignment* represents

matrix-id variable-names or *matrix-name* = *matrix-name*

Use the VARNAMES statement in connection with the RAM, COSAN, or FACTOR model statement to allocate names to latent variables including error and disturbance terms. This statement is not needed if you are using the LINEQS statement.

In connection with the RAM model statement, the *matrix-id* must be specified by the integer number as it is used in the RAM list input (1 for matrix $\mathbf{A}$, 2 for matrix $\mathbf{P}$). Because the first variables of matrix $\mathbf{A}$ correspond to the manifest variables in the input data set, you can specify names only for the latent variables following the manifest variables in the rows of $\mathbf{A}$. For example, in the RAM notation of the alienation example, you can specify the latent variables by names F1, F2, F3 and the error variables by names E1, . . . , E6, D1, D2, D3 with the following statement:

```
vnames 1  F1-F3,
       2  E1-E6 D1-D3;
```

If the RAM model statement is not accompanied by a VNAMES statement, default variable names are assigned using the prefixes F, E, and D with numerical suffixes: latent variables are F1, F2, . . . , and error variables are E1, E2, ...

The *matrix-id* must be specified by its name when used with the COSAN or FACTOR statement. The *variable-names* following the matrix name correspond to the columns of this matrix. The variable names corresponding to the rows of this matrix are set automatically by

- the names of the manifest variables for the first matrix in each term
- the column variable names of the same matrix for the central symmetric matrix in each term
- the column variable names of the preceding matrix for each other matrix

You also can use the second kind of name assignment in connection with a COSAN statement. Two matrix names separated by an equal sign allocate the column names of one matrix to the column names of the other matrix. This assignment assumes that the column names of at least one of the two matrices are already allocated. For example, in the COSAN notation of the alienation example, you can specify the variable names by using the following statements to allocate names to the columns of $\mathbf{J}$, $\mathbf{A}$, and $\mathbf{P}$:

```
vnames  J  V1-V6 F1-F3 ,
        A =J ,
        P  E1-E6 D1-D3 ;
```

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CALIS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CALIS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not mentioned in other statements are analyzed. You can use the VAR statement to ensure that the manifest variables appear in correct order for use in the RAM statement. Only one VAR statement can be used with each PROC CALIS statement. If you do not use all manifest variables when you specify the model with a RAM or LINEQS statement, PROC CALIS does automatic variable selection. For more information, see the section "Automatic Variable Selection" on page 662.

PARTIAL Statement

PARTIAL *variables* **;**

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables used to partial out the variables in the analysis. You can specify only one PARTIAL statement with each PROC CALIS statement.

FREQ Statement

FREQ *variable* **;**

If one variable in your data set represents the frequency of occurrence for the other values in the observation, specify the variable's name in a FREQ statement. PROC CALIS then treats the data set as if each observation appears n_i times, where n_i is the value of the FREQ variable for observation i. Only the integer portion of the value is used. If the value of the FREQ variable is less than 1 or is missing, that observation is not included in the analysis. The total number of observations is considered to be the sum of the FREQ values when the procedure computes significance probabilities. You can use only one FREQ statement with each PROC CALIS statement.

WEIGHT Statement

WEIGHT *variable* **;**

To compute weighted covariances or correlations, specify the name of the weighting variable in a WEIGHT statement. This is often done when the error variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. You can use only one WEIGHT statement with each PROC CALIS statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the number of observations unless VARDEF=WGT or VARDEF=WDF. An observation is used in the analysis only if the WEIGHT variable is greater than 0 and is not missing.

SAS Program Statements

This section lists the program statements used to express the linear and nonlinear constraints on the parameters and documents the differences between program statements in PROC CALIS and program statements in the DATA step. The very different use of the ARRAY statement by PROC CALIS is also discussed. Most of the program statements that can be used in the SAS DATA step also can be used in PROC CALIS. Refer to *SAS Language Reference: Dictionary* for a description of the SAS program statements. You can specify the following SAS program statements to compute parameter constraints with the CALIS procedure:

ABORT ;

CALL *name* < (*expression* < , *expression* . . . >) > **;**

DELETE;

DO < *variable* = *expression* < **TO** *expression*> < **BY** *expression*>
<, *expression* < **TO** *expression*> < **BY** *expression*> . . . > >
< **WHILE** *expression*>
< **UNTIL** *expression*> **;**

END;

GOTO *statement-label* **;**

IF *expression***;**

IF *expression* **THEN** *program-statement* **;**
ELSE *program-statement* **;**

variable = *expression* **;**

variable+expression **;**

LINK *statement-label* **;**

PUT <*variable*> <=> < . . . > **;**

RETURN ;

SELECT < (*expression*) > **;**

STOP;

SUBSTR (*variable, index, length*) = *expression* **;**

WHEN *(expression) program-statement* **;**
OTHERWISE *program-statement* **;**

For the most part, the SAS program statements work the same as they do in the SAS DATA step as documented in *SAS Language Reference: Concepts*. However, there are several differences that should be noted.

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus,

  ```
  do I=1,2,3;
  ```

  is supported; however,

  ```
  do I='A','B','C';
  ```

  is not valid in PROC CALIS, although it is supported in the DATA step.

- The PUT statement, used mostly for program debugging in PROC CALIS, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not have:
  - The CALIS procedure PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or $.
  - The CALIS procedure PUT statement does support expressions enclosed in parentheses. For example, the following statement displays the square root of x:

    ```
    put (sqrt(x));
    ```

  - The CALIS procedure PUT statement supports the print item _PDV_ to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the _ALL_ print item:

    ```
    put _pdv_ ;
    ```

- The WHEN and OTHERWISE statements allow more than one target statement. That is, DO/END groups are not necessary for multiple WHEN statements. For example, the following syntax is valid:

  ```
  select;
     when ( expression1 ) statement1;
                          statement2;
     when ( expression2 ) statement3;
                          statement4;
  end;
  ```

You can specify one or more PARMS statements to define parameters used in the program statements that are not defined in the model matrices (MATRIX, RAM, LINEQS, STD, or COV statement).

Parameters that are used only on the right-hand side of your program statements are called independent, and parameters that are used at least once on the left-hand side of an equation in the program code are called dependent parameters. The dependent parameters are used only indirectly in the minimization process. They should be fully defined as functions of the independent parameters. The independent parameters are included in the set $\mathbf{X}$ of parameters used in the minimization. Be sure that all independent parameters used in your program statements are somehow connected to elements of the model matrices. Otherwise the minimization function does not depend on those independent parameters, and the parameters vary without control (since the corresponding derivative is the constant 0). You also can specify the PARMS statement to set the initial values of all independent parameters used in the program statements that are not defined as elements of model matrices.

ARRAY Statement

ARRAY *arrayname* <*(dimensions)*>< *$* ><*variables and constants*> **;**

The ARRAY statement is similar to, but not the same as, the ARRAY statement in the DATA step. The ARRAY statement is used to associate a name with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC CALIS does not support all the features of the DATA step ARRAY statement. With PROC CALIS, the ARRAY statement cannot be used to give initial values to array elements. Implicit indexing variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement supported by PROC CALIS does allow both variables and constants to be used as array elements. Constant array elements cannot be changed. Both the dimension specification and the list of elements are optional, but at least one must be given. When the list of elements is not given or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

Details

Input Data Sets

You can use four different kinds of input data sets in the CALIS procedure, and you can use them simultaneously. The DATA= data set contains the data to be analyzed, and it can be an ordinary SAS data set containing raw data or a special TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, TYPE=SYMATRIX, TYPE=SSCP, or TYPE=FACTOR data set containing previously computed statistics. The INEST= data set specifies an input data set that contains initial estimates for the parameters used in the optimization process, and it can also contain boundary and general linear constraints on the parameters. If the model does not change too much, you can use an OUTEST= data set from a previous PROC CALIS analysis; the initial estimates are taken from the values of the PARMS observation. The INRAM= data set names a third input data set that contains all information needed to specify the analysis model in RAM list form (except for user-written program statements). Often the INRAM= data set can be the OUTRAM= data set from a previous PROC CALIS analysis. See the section "OUTRAM= SAS-data-set" on page 638 for the structure of both OUTRAM= and INRAM= data sets. Using the INWGT= data set enables you to read in the weight matrix **W** that can be used in generalized least-squares, weighted least-squares, or diagonally weighted least-squares estimation.

DATA= SAS-data-set

A TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set can be created by the CORR procedure or various other procedures. It contains means, standard deviations, the sample size, the covariance or correlation matrix, and possibly other statistics depending on which procedure is used.

If your data set has many observations and you plan to run PROC CALIS several times, you can save computer time by first creating a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set and using it as input to PROC CALIS. For example, assuming that PROC CALIS is first run with an OUTRAM=MOD option, you can run

```
* create TYPE=COV data set;
proc corr cov nocorr data=raw outp=cov(type=cov);
run;
* analysis using correlations;
proc calis data=cov inram=mod;
run;
* analysis using covariances;
proc calis cov data=cov inram=mod;
run;
```

Most procedures automatically set the TYPE= option of an output data set appropriately. However, the CORR procedure sets TYPE=CORR unless an explicit TYPE= option is used. Thus, `(TYPE=COV)` is needed in the preceding PROC CORR request, since the output data set is a covariance matrix. If you use a DATA step with a SET statement to modify this data set, you must declare the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR attribute in the new data set.

You can use a VAR statement with PROC CALIS when reading a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set to select a subset of the variables or change the order of the variables.

Caution: Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each covariance or correlation from all observations that have values present for the pair of variables involved ("pairwise deletion"). The resulting covariance or correlation matrix can have negative eigenvalues. A correlation or covariance matrix with negative eigenvalues is recognized as a singular matrix in PROC CALIS, and you cannot compute (default) generalized least-squares or maximum likelihood estimates. You can specify the RIDGE option to ridge the diagonal of such a matrix to obtain a positive definite data matrix. If the NOMISS option is used with the CORR procedure, observations with any missing values are completely omitted from the calculations ("listwise deletion"), and there is no possibility of negative eigenvalues (but still a chance for a singular matrix).

PROC CALIS can also create a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set that includes all the information needed for repeated analyses. If the data set DATA=RAW does not contain missing values, the following statements should give the same PROC CALIS results as the previous example.

```
* using correlations;
proc calis data=raw outstat=cov inram=mod;
run;
* using covariances;
proc calis cov data=cov inram=mod;
run;
```

You can create a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement, and include the _TYPE_ and _NAME_ variables. If you want to analyze the covariance matrix but your DATA= data set is a TYPE=CORR or TYPE=UCORR data set, you should include an observation with _TYPE_=STD giving the standard deviation of each variable. If you specify the COV option, PROC CALIS analyzes the recomputed covariance matrix:

```
data correl(type=corr);
   input _type_ $ _name_ $ X1-X3;
   datalines;
std   .   4.  2.  8.
corr  X1  1.0  .   .
corr  X2   .7 1.0  .
corr  X3   .5  .4 1.0
;
proc calis cov inram=model;
run;
```

If you want to analyze the UCOV or UCORR matrix but your DATA= data set is a TYPE=COV or TYPE=CORR data set, you should include observations with _TYPE_=STD and _TYPE_=MEAN giving the standard deviation and mean of each variable.

INEST= SAS-data-set

You can use the INEST= (or INVAR= or ESTDATA=) input data set to specify the initial values of the parameters used in the optimization and to specify boundary constraints and the more general linear constraints that can be imposed on these parameters.

The variables of the INEST= data set must correspond to

- a character variable _TYPE_ that indicates the type of the observation
- n numeric variables with the parameter names used in the specified PROC CALIS model
- the BY variables that are used in a DATA= input data set
- a numeric variable _RHS_ (right-hand side) (needed only if linear constraints are used)
- additional variables with names corresponding to constants used in the program statements

The content of the _TYPE_ variable defines the meaning of the observation of the INEST= data set. PROC CALIS recognizes observations with the following _TYPE_ specifications.

PARMS specifies initial values for parameters that are defined in the model statements of PROC CALIS. The _RHS_ variable is not used. Additional variables can contain the values of constants that are referred to in program statements. At the beginning of each run of PROC CALIS, the values of the constants are read from the PARMS observation initializing the constants in the program statements.

UPPERBD | UB specifies upper bounds with nonmissing values. The use of a missing value indicates that no upper bound is specified for the parameter. The _RHS_ variable is not used.

LOWERBD | LB specifies lower bounds with nonmissing values. The use of a missing value indicates that no lower bound is specified for the parameter. The _RHS_ variable is not used.

LE | <= | < specifies the linear constraint $\sum_j a_{ij} x_j \leq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

GE | >= | > specifies the linear constraint $\sum_j a_{ij} x_j \geq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

EQ | = specifies the linear constraint $\sum_j a_{ij} x_j = b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

The constraints specified in the INEST=, INVAR=, or ESTDATA= data set are added to the constraints specified in BOUNDS and LINCON statements.

You can use an OUTEST= data set from a PROC CALIS run as an INEST= data set in a new run. However, be aware that the OUTEST= data set also contains the boundary and general linear constraints specified in the previous run of PROC CALIS. When you are using this OUTEST= data set without changes as an INEST= data set, PROC CALIS adds the constraints from the data set to the constraints specified by a BOUNDS and LINCON statement. Although PROC CALIS automatically eliminates multiple identical constraints, you should avoid specifying the same constraint a second time.

INRAM= SAS-data-set

This data set is usually created in a previous run of PROC CALIS. It is useful if you want to reanalyze a problem in a different way such as using a different estimation method. You can alter an existing OUTRAM= data set, either in the DATA step or using the FSEDIT procedure, to create the INRAM= data set describing a modified model. For more details on the INRAM= data set, see the section “OUTRAM= SAS-data-set” on page 638.

In the case of a RAM or LINEQS analysis of linear structural equations, the OUTRAM= data set always contains the variable names of the model specified. These variable names and the model specified in the INRAM= data set are the basis of the automatic variable selection algorithm performed after reading the INRAM= data set.

INWGT= SAS-data-set

This data set enables you to specify a weight matrix other than the default matrix for the generalized, weighted, and diagonally weighted least-squares estimation methods. The specification of any INWGT= data set for unweighted least-squares or maximum likelihood estimation is ignored. For generalized and diagonally weighted least-squares estimation, the INWGT= data set must contain a _TYPE_ and a _NAME_ variable as well as the manifest variables used in the analysis. The value of the _NAME_ variable indicates the row index i of the weight w_{ij}. For weighted least squares, the INWGT= data set must contain _TYPE_, _NAME_, _NAM2_, and _NAM3_ variables as well as the manifest variables used in the analysis. The values of the _NAME_, _NAM2_, and _NAM3_ variables indicate the three indices i, j, k of the weight $w_{ij,kl}$. You can store information other than the weight matrix in the INWGT= data set, but only observations with _TYPE_=WEIGHT are used to specify the weight matrix $\mathbf{W}$. This property enables you to store more than one weight matrix in the INWGT= data set. You can then run PROC CALIS with each of the weight matrices by changing only the _TYPE_ observation in the INWGT= data set with an intermediate DATA step.

For more details on the INWGT= data set, see the section "OUTWGT= SAS-data-set" on page 643.

Output Data Sets

OUTEST= SAS-data-set

The OUTEST= (or OUTVAR=) data set is of TYPE=EST and contains the final parameter estimates, the gradient, the Hessian, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the approximate standard errors, the information matrix (crossproduct Jacobian), and the approximate covariance matrix of the parameter estimates ((generalized) inverse of the information matrix). If there are linear or nonlinear equality or active inequality constraints at the solution, the OUTEST= data set also contains Lagrange multipliers, the projected Hessian matrix, and the Hessian matrix of the Lagrange function.

The OUTEST= data set can be used to save the results of an optimization by PROC CALIS for another analysis with either PROC CALIS or another SAS procedure. Saving results to an OUTEST= data set is advised for expensive applications that cannot be repeated without considerable effort.

The OUTEST= data set contains the BY variables, two character variables _TYPE_ and _NAME_, t numeric variables corresponding to the parameters used in the model, a numeric variable _RHS_ (right-hand side) that is used for the right-hand-side value b_i of a linear constraint or for the value $f = f(x)$ of the objective function at the final point x^* of the parameter space, and a numeric variable _ITER_ that is set to zero for initial values, set to the iteration number for the OUTITER output, and set to missing for the result output.

The _TYPE_ observations in Table 19.5 are available in the OUTEST= data set, depending on the request.

Table 19.5. _TYPE_ Observations in the OUTEST= data set

TYPE	**Description**
ACTBC	If there are active boundary constraints at the solution x^*, three observations indicate which of the parameters are actively constrained, as follows. **_NAME_** **Description** GE indicates the active lower bounds LE indicates the active upper bounds EQ indicates the active masks
COV	contains the approximate covariance matrix of the parameter estimates; used in computing the approximate standard errors.
COVRANK	contains the rank of the covariance matrix of the parameter estimates.
CRPJ_LF	contains the Hessian matrix of the Lagrange function (based on CRPJAC).
CRPJAC	contains the approximate Hessian matrix used in the optimization process. This is the inverse of the information matrix.
EQ	If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j = b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, the _RHS_ variable contains b_i, and _NAME_=ACTLC or _NAME_=LDACTLC.
GE	If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j \geq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.
GRAD	contains the gradient of the estimates.
GRAD_LF	contains the gradient of the Lagrange function. The _RHS_ variable contains the value of the Lagrange function.
HESSIAN	contains the Hessian matrix.
HESS_LF	contains the Hessian matrix of the Lagrange function (based on HESSIAN).

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

<table>
<tr><th>_TYPE_</th><th>Description</th></tr>
<tr><td>INFORMAT</td><td>contains the information matrix of the parameter estimates (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).</td></tr>
<tr><td>INITIAL</td><td>contains the starting values of the parameter estimates.</td></tr>
<tr><td>JACNLC</td><td>contains the Jacobian of the nonlinear constraints evaluated at the final estimates.</td></tr>
<tr><td>JACOBIAN</td><td>contains the Jacobian matrix (only if the OUTJAC option is used).</td></tr>
<tr><td>LAGM BC</td><td>contains Lagrange multipliers for masks and active boundary constraints.
<table>
<tr><th>_NAME_</th><th>Description</th></tr>
<tr><td>GE</td><td>indicates the active lower bounds</td></tr>
<tr><td>LE</td><td>indicates the active upper bounds</td></tr>
<tr><td>EQ</td><td>indicates the active masks</td></tr>
</table></td></tr>
<tr><td>LAGM LC</td><td>contains Lagrange multipliers for linear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.
<table>
<tr><th>_NAME_</th><th>Description</th></tr>
<tr><td>LEC_NUM</td><td>number of the linear equality constraint</td></tr>
<tr><td>LEC_VAL</td><td>corresponding Lagrange multiplier value</td></tr>
<tr><td>LIC_NUM</td><td>number of the linear inequality constraint</td></tr>
<tr><td>LIC_VAL</td><td>corresponding Lagrange multiplier value</td></tr>
</table></td></tr>
<tr><td>LAGM NLC</td><td>contains Lagrange multipliers for nonlinear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.
<table>
<tr><th>_NAME_</th><th>Description</th></tr>
<tr><td>NLEC_NUM</td><td>number of the nonlinear equality constraint</td></tr>
<tr><td>NLEC_VAL</td><td>corresponding Lagrange multiplier value</td></tr>
<tr><td>NLIC_NUM</td><td>number of the linear inequality constraint</td></tr>
<tr><td>NLIC_VAL</td><td>corresponding Lagrange multiplier value</td></tr>
</table></td></tr>
<tr><td>LE</td><td>If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j \leq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.</td></tr>
</table>

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

<table>
<tr><th>_TYPE_</th><th>Description</th></tr>
<tr><td>LOWERBD
| LB</td><td>If boundary constraints are used, this observation contains the lower bounds. Those parameters not subjected to lower bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
<tr><td>NACTBC</td><td>All parameter variables contain the number n_{abc} of active boundary constraints at the solution x^*. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
<tr><td>NACTLC</td><td>All parameter variables contain the number n_{alc} of active linear constraints at the solution x^* that are recognized as linearly independent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
<tr><td>NLC_EQ
NLC_GE
NLC_LE</td><td>contains values and residuals of nonlinear constraints. The _NAME_ variable is described as follows.
<table>
<tr><th>_NAME_</th><th>Description</th></tr>
<tr><td>NLC</td><td>inactive nonlinear constraint</td></tr>
<tr><td>NLCACT</td><td>linear independent active nonlinear constr.</td></tr>
<tr><td>NLCACTLD</td><td>linear dependent active nonlinear constr.</td></tr>
</table></td></tr>
<tr><td>NLDACTBC</td><td>contains the number of active boundary constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
<tr><td>NLDACTLC</td><td>contains the number of active linear constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
<tr><td>_NOBS_</td><td>contains the number of observations.</td></tr>
<tr><td>PARMS</td><td>contains the final parameter estimates. The _RHS_ variable contains the value of the objective function.</td></tr>
<tr><td>PCRPJ_LF</td><td>contains the projected Hessian matrix of the Lagrange function (based on CRPJAC).</td></tr>
<tr><td>PHESS_LF</td><td>contains the projected Hessian matrix of the Lagrange function (based on HESSIAN).</td></tr>
<tr><td>PROJCRPJ</td><td>contains the projected Hessian matrix (based on CRPJAC).</td></tr>
<tr><td>PROJGRAD</td><td>If linear constraints are used in the estimation, this observation contains the $n - n_{act}$ values of the projected gradient $g_Z = Z'g$ in the variables corresponding to the first $n - n_{act}$ parameters. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.</td></tr>
</table>

Table 19.5. _TYPE_ Observations in the OUTEST= data set (continued)

TYPE	Description
PROJHESS	contains the projected Hessian matrix (based on HESSIAN).
SIGSQ	contains the scalar factor of the covariance matrix of the parameter estimates.
STDERR	contains approximate standard errors (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).
TERMINAT	The _NAME_ variable contains the name of the termination criterion.
UPPERBD | UB	If boundary constraints are used, this observation contains the upper bounds. Those parameters not subjected to upper bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

If the technique specified by the TECH= option cannot be performed (for example, no feasible initial values can be computed, or the function value or derivatives cannot be evaluated at the starting point), the OUTEST= data set may contain only some of the observations (usually only the PARMS and GRAD observations).

OUTRAM= SAS-data-set

The OUTRAM= data set is of TYPE=RAM and contains the model specification and the computed parameter estimates. This data set is intended to be reused as an INRAM= data set to specify good initial values in a subsequent analysis by PROC CALIS.

The OUTRAM= data set contains the following variables:

- the BY variables, if any
- the character variable _TYPE_, which takes the values MODEL, ESTIM, VARNAME, METHOD, and STAT
- six additional variables whose meaning depends on the _TYPE_ of the observation

Each observation with _TYPE_ =MODEL defines one matrix in the generalized COSAN model. The additional variables are as follows.

Table 19.6. Additional Variables when _TYPE_=MODEL

Variable	Contents
NAME	name of the matrix (character)
MATNR	number for the term and matrix in the model (numeric)
ROW	matrix row number (numeric)
COL	matrix column number (numeric)
ESTIM	first matrix type (numeric)
STDERR	second matrix type (numeric)

If the generalized COSAN model has only one matrix term, the _MATNR_ variable contains only the number of the matrix in the term. If there is more than one term, then it is the term number multiplied by 10,000 plus the matrix number (assuming that there are no more than 9,999 matrices specified in the COSAN model statement).

Each observation with _TYPE_ =ESTIM defines one element of a matrix in the generalized COSAN model. The variables are used as follows.

Table 19.7. Additional Variables when _TYPE_=ESTIM

Variable	**Contents**
NAME	name of the parameter (character)
MATNR	term and matrix location of parameter (numeric)
ROW	row location of parameter (numeric)
COL	column location of parameter (numeric)
ESTIM	parameter estimate or constant value (numeric)
STDERR	standard error of estimate (numeric)

For constants rather than estimates, the _STDERR_ variable is 0. The _STDERR_ variable is missing for ULS and DWLS estimates if NOSTDERR is specified or if the approximate standard errors are not computed.

Each observation with _TYPE_ =VARNAME defines a column variable name of a matrix in the generalized COSAN model.

The observations with _TYPE_=METHOD and _TYPE_=STAT are not used to build the model. The _TYPE_=METHOD observation contains the name of the estimation method used to compute the parameter estimates in the _NAME_ variable. If METHOD=NONE is not specified, the _ESTIM_ variable of the _TYPE_=STAT observations contains the information summarized in Table 19.8 (described in the section "Assessment of Fit" on page 649).

Table 19.8. _ESTIM_ Contents for _TYPE_=STAT

NAME	_ESTIM_
N	sample size
NPARM	number of parameters used in the model
DF	degrees of freedom
N_ACT	number of active boundary constraints for ML, GLS, and WLS estimation
FIT	fit function
GFI	goodness-of-fit index (GFI)
AGFI	adjusted GFI for degrees of freedom
RMR	root mean square residual
PGFI	parsimonious GFI of Mulaik et al. (1989)
CHISQUAR	overall χ^2
P_CHISQ	probability $> \chi^2$
CHISQNUL	null (baseline) model χ^2
RMSEAEST	Steiger & Lind's (1980) RMSEA index estimate
RMSEALOB	lower range of RMSEA confidence interval
RMSEAUPB	upper range of RMSEA confidence interval
P_CLOSFT	Browne & Cudeck's (1993) probability of close fit

Table 19.8. _ESTIM_ Contents for _TYPE_=STAT (continued)

NAME	_ESTIM_
ECVI_EST	Browne & Cudeck's (1993) ECV index estimate
ECVI_LOB	lower range of ECVI confidence interval
ECVI_UPB	upper range of ECVI confidence interval
COMPFITI	Bentler's (1989) comparative fit index
ADJCHISQ	adjusted χ^2 for elliptic distribution
P_ACHISQ	probability corresponding adjusted χ^2
RLSCHISQ	reweighted least-squares χ^2 (only ML estimation)
AIC	Akaike's information criterion
CAIC	Bozdogan's consistent information criterion
SBC	Schwarz's Bayesian criterion
CENTRALI	McDonald's centrality criterion
PARSIMON	Parsimonious index of James, Mulaik, and Brett
ZTESTWH	z test of Wilson and Hilferty
BB_NONOR	Bentler-Bonett (1980) nonnormed index ρ
BB_NORMD	Bentler-Bonett (1980) normed index Δ
BOL_RHO1	Bollen's (1986) normed index ρ_1
BOL_DEL2	Bollen's (1989a) nonnormed index Δ_2
CNHOELT	Hoelter's critical N index

You can edit the OUTRAM= data set to use its contents for initial estimates in a subsequent analysis by PROC CALIS, perhaps with a slightly changed model. But you should be especially careful for _TYPE_=MODEL when changing matrix types. The codes for the two matrix types are listed in Table 19.9.

Table 19.9. Matrix Type Codes

Code	**First Matrix Type**	**Description**
1:	IDE	identity matrix
2:	ZID	zero:identity matrix
3:	DIA	diagonal matrix
4:	ZDI	zero:diagonal matrix
5:	LOW	lower triangular matrix
6:	UPP	upper triangular matrix
7:		temporarily not used
8:	SYM	symmetric matrix
9:	GEN	general-type matrix
10:	BET	identity minus general-type matrix
11:	PER	selection matrix
12:		first matrix (**J**) in LINEQS model statement
13:		second matrix (β) in LINEQS model statement
14:		third matrix (γ) in LINEQS model statement
Code	**Second Matrix Type**	**Description**
0:		noninverse model matrix
1:	INV	inverse model matrix
2:	IMI	'identity minus inverse' model matrix

OUTSTAT= SAS-data-set

The OUTSTAT= data set is similar to the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set produced by the CORR procedure. The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- two character variables, _TYPE_ and _NAME_
- the variables analyzed, that is, those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement but used in the analysis. (**Caution**: Using the LINEQS or RAM model statements selects variables automatically.)

The OUTSTAT= data set contains the following information (when available):

- the mean and standard deviation
- the skewness and kurtosis (if the DATA= data set is a raw data set and the KURTOSIS option is specified)
- the number of observations
- if the WEIGHT statement is used, sum of the weights
- the correlation or covariance matrix to be analyzed
- the predicted correlation or covariance matrix
- the standardized or normalized residual correlation or covariance matrix
- if the model contains latent variables, the predicted covariances between latent and manifest variables, and the latent variable (or factor) score regression coefficients (see the PLATCOV display option on page 586)

In addition, if the FACTOR model statement is used, the OUTSTAT= data set contains:

- the unrotated factor loadings, the unique variances, and the matrix of factor correlations
- the rotated factor loadings and the transformation matrix of the rotation
- the matrix of standardized factor loadings

Each observation in the OUTSTAT= data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are given in Table 19.10.

Table 19.10. _TYPE_ Observations in the OUTSTAT= data set

TYPE	Contents
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations
SKEWNESS	univariate skewness
KURTOSIS	univariate kurtosis
N	sample size
SUMWGT	sum of weights (if WEIGHT statement is used)
COV	covariances analyzed
CORR	correlations analyzed
UCOV	uncorrected covariances analyzed
UCORR	uncorrected correlations analyzed
ULSPRED	ULS predicted model values
GLSPRED	GLS predicted model values
MAXPRED	ML predicted model values
WLSPRED	WLS predicted model values
DWLSPRED	DWLS predicted model values
ULSNRES	ULS normalized residuals
GLSNRES	GLS normalized residuals
MAXNRES	ML normalized residuals
WLSNRES	WLS normalized residuals
DWLSNRES	DWLS normalized residuals
ULSSRES	ULS variance standardized residuals
GLSSRES	GLS variance standardized residuals
MAXSRES	ML variance standardized residuals
WLSSRES	WLS variance standardized residuals
DWLSSRES	DWLS variance standardized residuals
ULSASRES	ULS asymptotically standardized residuals
GLSASRES	GLS asymptotically standardized residuals
MAXASRES	ML asymptotically standardized residuals
WLSASRES	WLS asymptotically standardized residuals
DWLSASRS	DWLS asymptotically standardized residuals
UNROTATE	unrotated factor loadings
FCORR	matrix of factor correlations
UNIQUE_V	unique variances
TRANSFOR	transformation matrix of rotation
LOADINGS	rotated factor loadings
STD_LOAD	standardized factor loadings
LSSCORE	latent variable (or factor) score regression coefficients for ULS method
SCORE	latent variable (or factor) score regression coefficients other than ULS method

The _NAME_ variable contains the name of the manifest variable corresponding to each row for the covariance, correlation, predicted, and residual matrices and contains the name of the latent variable in case of factor regression scores. For other observations, _NAME_ is blank.

The unique variances and rotated loadings can be used as starting values in more difficult and constrained analyses.

If the model contains latent variables, the OUTSTAT= data set also contains the latent variable score regression coefficients and the predicted covariances between latent and manifest variables. You can use the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If the analyzed matrix is a (corrected or uncorrected) covariance rather than a correlation matrix, the _TYPE_=STD or _TYPE_=USTD observation is not included in the OUTSTAT= data set. In this case, the standard deviations can be obtained from the diagonal elements of the covariance matrix. Dropping the _TYPE_=STD or _TYPE_=USTD observation prevents PROC SCORE from standardizing the observations before computing the factor scores.

OUTWGT= SAS-data-set

You can create an OUTWGT= data set that is of TYPE=WEIGHT and contains the weight matrix used in generalized, weighted, or diagonally weighted least-squares estimation. The *inverse* of the weight matrix is used in the corresponding fit function. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. For unweighted least-squares or maximum likelihood estimation, no OUTWGT= data set can be written. The last weight matrix used in maximum likelihood estimation is the predicted model matrix (observations with _TYPE_ =MAXPRED) that is included in the OUTSTAT= data set.

For generalized and diagonally weighted least-squares estimation, the weight matrices $\mathbf{W}$ of the OUTWGT= data set contain all elements w_{ij}, where the indices i and j correspond to all manifest variables used in the analysis. Let $varnam_i$ be the name of the ith variable in the analysis. In this case, the OUTWGT= data set contains n observations with variables as displayed in the following table.

Table 19.11. Contents of OUTWGT= data set for GLS and DWLS Estimation

Variable	**Contents**
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
$varnam_1$	weight w_{i1} for variable $varnam_1$ (numeric)
$\vdots$	$\vdots$
$varnam_n$	weight w_{in} for variable $varnam_n$ (numeric)

For weighted least-squares estimation, the weight matrix $\mathbf{W}$ of the OUTWGT= data set contains only the nonredundant elements $w_{ij,kl}$. In this case, the OUTWGT= data set contains $n(n+1)(2n+1)/6$ observations with variables as follows.

Table 19.12. Contents of OUTWGT= data set for WLS Estimation

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
NAM2	name of variable $varnam_j$ (character)
NAM3	name of variable $varnam_k$ (character)
$varnam_1$	weight $w_{ij,k1}$ for variable $varnam_1$ (numeric)
$\vdots$	$\vdots$
$varnam_n$	weight $w_{ij,kn}$ for variable $varnam_n$ (numeric)

Symmetric redundant elements are set to missing values.

Missing Values

If the DATA= data set contains raw data (rather than a covariance or correlation matrix), observations with missing values for any variables in the analysis are omitted from the computations. If a covariance or correlation matrix is read, missing values are allowed as long as every pair of variables has at least one nonmissing value.

Estimation Criteria

The following five estimation methods are available in PROC CALIS:

- unweighted least squares (ULS)
- generalized least squares (GLS)
- normal-theory maximum likelihood (ML)
- weighted least squares (WLS, ADF)
- diagonally weighted least squares (DWLS)

An INWGT= data set can be used to specify other than the default weight matrices $\mathbf{W}$ for GLS, WLS, and DWLS estimation.

In each case, the parameter vector is estimated iteratively by a nonlinear optimization algorithm that optimizes a goodness-of-fit function F. When n denotes the number of manifest variables, $\mathbf{S}$ denotes the given sample covariance or correlation matrix for a sample with size N, and $\mathbf{C}$ denotes the predicted moment matrix, then the fit function for unweighted least-squares estimation is

$$F_{ULS} = .5Tr[(\mathbf{S} - \mathbf{C})^2]$$

For normal-theory generalized least-squares estimation, the function is

$$F_{GLS} = .5Tr[(\mathbf{S}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For normal-theory maximum likelihood estimation, the function is

$$F_{ML} = Tr(\mathbf{SC}^{-1}) - n + ln(det(\mathbf{C})) - ln(det(\mathbf{S}))$$

The first three functions can be expressed by the generalized weighted least-squares criterion (Browne 1982):

$$F_{GWLS} = .5Tr[(\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For unweighted least squares, the weight matrix $\mathbf{W}$ is chosen as the identity matrix $\mathbf{I}$; for generalized least squares, the default weight matrix $\mathbf{W}$ is the sample covariance matrix $\mathbf{S}$; and for normal-theory maximum likelihood, $\mathbf{W}$ is the iteratively updated predicted moment matrix $\mathbf{C}$. The values of the normal-theory maximum likelihood function F_{ML} and the generally weighted least-squares criterion F_{GWLS} with $\mathbf{W} = \mathbf{C}$ are asymptotically equivalent.

The goodness-of-fit function that is minimized in weighted least-squares estimation is

$$F_{WLS} = Vec(s_{ij} - c_{ij})'\mathbf{W}^{-1}Vec(s_{ij} - c_{ij})$$

where $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$, and $\mathbf{W} = (w_{ij,kl})$ is a positive definite symmetric matrix with $n(n+1)/2$ rows and columns.

If the moment matrix $\mathbf{S}$ is considered as a covariance rather than a correlation matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the consistent but biased estimators of the asymptotic covariances$\sigma_{ij,kl}$ of the sample covariance s_{ij} with the sample covariance s_{kl}

$$w_{ij,kl} = s_{ij,kl} - s_{ij}s_{kl}$$

where

$$s_{ij,kl} = \frac{1}{N}\sum_{r=1}^{N}(z_{ri} - \overline{z_i})(z_{rj} - \overline{z_j})(z_{rk} - \overline{z_k})(z_{rl} - \overline{z_l})$$

The formula of the asymptotic covariances of uncorrected covariances (using the UCOV or NOINT option) is a straightforward generalization of this expression.

The resulting weight matrix $\mathbf{W}$ is at least positive semidefinite (except for rounding errors). Using the ASYCOV option, you can use Browne's (1984, formula (3.8)) unbiased estimators

$$\begin{aligned} w_{ij,kl} &= \frac{N(N-1)}{(N-2)(N-3)}(s_{ij,kl} - s_{ij}s_{kl}) \\ &\quad - \frac{N}{(N-2)(N-3)}(s_{ik}s_{jl} + s_{il}s_{jk} - \frac{2}{N-1}s_{ij}s_{kl}) \end{aligned}$$

There is no guarantee that this weight matrix is positive semidefinite. However, the second part is of order $O(N^{-1})$ and does not destroy the positive semidefinite first part for sufficiently large N. For a large number of independent observations, default settings of the weight matrix $\mathbf{W}$ result in asymptotically distribution-free parameter estimates with unbiased standard errors and a correct χ^2 test statistic (Browne 1982, 1984).

If the moment matrix $\mathbf{S}$ is a correlation (rather than a covariance) matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the estimators of the asymptotic covariances $\sigma_{ij,kl}$ of the correlations $\mathbf{S} = (s_{ij})$ (Browne and Shapiro 1986; DeLeeuw 1983)

$$\begin{aligned} w_{ij,kl} &= r_{ij,kl} - \frac{1}{2} r_{ij}(r_{ii,kl} + r_{jj,kl}) - \frac{1}{2} r_{kl}(r_{kk,ij} + r_{ll,ij}) \\ &\quad + \frac{1}{4} r_{ij} r_{kl}(r_{ii,kk} + r_{ii,ll} + r_{jj,kk} + r_{jj,ll}) \end{aligned}$$

where

$$r_{ij,kl} = \frac{s_{ij,kl}}{\sqrt{s_{ii}s_{jj}s_{kk}s_{ll}}}$$

The asymptotic variances of the diagonal elements of a correlation matrix are 0. Therefore, the weight matrix computed by Browne and Shapiro's formula is always singular. In this case the goodness-of-fit function for weighted least-squares estimation is modified to

$$F_{WLS} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \sum_{k=2}^{n} \sum_{l=1}^{k-1} w^{ij,kl}(s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_{i}^{n} (s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix $\mathbf{W}$. The second term is a penalty term to fit the diagonal elements of the moment matrix $\mathbf{S}$. The default value of $r = 100$ can be decreased or increased by the WPENALTY= option. The often used value of $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. If your model does not fit the diagonal of the moment matrix $\mathbf{S}$, you can specify the NODIAG option to exclude the diagonal elements from the fit function.

Storing and inverting the huge weight matrix $\mathbf{W}$ in WLS estimation needs considerable computer resources. A compromise is found by implementing the DWLS method that uses only the diagonal of the weight matrix $\mathbf{W}$ from the WLS estimation in the minimization function

$$F_{DWLS} = Vec(s_{ij} - c_{ij})' diag(\mathbf{W})^{-1} Vec(s_{ij} - c_{ij})$$

The statistical properties of DWLS estimates are still not known.

In generalized, weighted, or diagonally weighted least-squares estimation, you can change from the default settings of weight matrices $\mathbf{W}$ by using an INWGT= data set. Because the diagonal elements $w_{ii,kk}$ of the weight matrix $\mathbf{W}$ are interpreted as asymptotic variances of the sample covariances or correlations, they cannot be negative. The CALIS procedure requires a positive definite weight matrix that has positive diagonal elements.

Relationships among Estimation Criteria

The five estimation functions, F_{ULS}, F_{GLS}, F_{ML}, F_{WLS}, and F_{DWLS}, belong to the following two groups:

- The functions F_{ULS}, F_{GLS}, and F_{ML} take into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute twice to F, as lower and as upper triangle elements.
- The functions F_{WLS} and F_{DWLS} take into account only the $n(n+1)/2$ lower triangular elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute to F only once.

The F_{DWLS} function used in PROC CALIS differs from that used by the LISREL 7 program. Formula (1.25) of the LISREL 7 manual (Jöreskog and Sörbom 1988, p. 23) shows that LISREL groups the F_{DWLS} function in the first group by taking into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$.

- Relationship between DWLS and WLS:
  PROC CALIS: The F_{DWLS} and F_{WLS} estimation functions deliver the same results for the special case that the weight matrix $\mathbf{W}$ used by WLS estimation is a diagonal matrix.
  LISREL 7: This is not the case.
- Relationship between DWLS and ULS:
  LISREL 7: The F_{DWLS} and F_{ULS} estimation functions deliver the same results for the special case that the diagonal weight matrix $\mathbf{W}$ used by DWLS estimation is an identity matrix (contains only 1s).
  PROC CALIS: To obtain the same results with F_{DWLS} and F_{ULS} estimation, set the diagonal weight matrix $\mathbf{W}$ used in DWLS estimation to

  $$w_{ik,ik} = \begin{cases} 1. & \text{if } i = k \\ 0.5 & \text{otherwise} \end{cases}$$

  Because the reciprocal elements of the weight matrix are used in the goodness-of-fit function, the off-diagonal residuals are weighted by a factor of 2.

Testing Rank Deficiency in the Approximate Covariance Matrix

The inverse of the information matrix (or approximate Hessian matrix) is used for the covariance matrix of the parameter estimates, which is needed for the computation of approximate standard errors and modification indices. The numerical condition of the information matrix (computed as the crossproduct $\mathbf{J}'\mathbf{J}$ of the Jacobian matrix $\mathbf{J}$) can be very poor in many practical applications, especially for the analysis of unscaled covariance data. The following four-step strategy is used for the inversion of the information matrix.

1. The inversion (usually of a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$) is tried using a modified form of the Bunch and Kaufman (1977) algorithm, which allows the specification of a different singularity criterion for each pivot. The following three criteria for the detection of rank loss in the information matrix are used to specify thresholds:

   - *ASING* specifies absolute singularity.
   - *MSING* specifies relative singularity depending on the whole matrix norm.
   - *VSING* specifies relative singularity depending on the column matrix norm.

   If no rank loss is detected, the inverse of the information matrix is used for the covariance matrix of parameter estimates, and the next two steps are skipped.
2. The linear dependencies among the parameter subsets are displayed based on the singularity criteria.
3. If the number of parameters t is smaller than the value specified by the G4= option (the default value is 60), the Moore-Penrose inverse is computed based on the eigenvalue decomposition of the information matrix. If you do not specify the NOPRINT option, the distribution of eigenvalues is displayed, and those eigenvalues that are set to zero in the Moore-Penrose inverse are indicated. You should inspect this eigenvalue distribution carefully.
4. If PROC CALIS did not set the right subset of eigenvalues to zero, you can specify the COVSING= option to set a larger or smaller subset of eigenvalues to zero in a further run of PROC CALIS.

Approximate Standard Errors

Except for unweighted and diagonally weighted least-squares estimation, approximate standard errors can be computed as the diagonal elements of the matrix

$$\frac{c}{NM}\mathbf{H}^{-1}, \quad \text{where}$$

$$NM = \begin{cases} (N-1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$$

The matrix $\mathbf{H}$ is the approximate Hessian matrix of F evaluated at the final estimates, $c = 1$ for the WLS estimation method, $c = 2$ for the GLS and ML method, and N is the sample size. If a given correlation or covariance matrix is singular, PROC CALIS offers two ways to compute a generalized inverse of the information matrix and, therefore, two ways to compute approximate standard errors of implicitly constrained parameter estimates, *t* values, and modification indices. Depending on the G4= specification, either a Moore-Penrose inverse or a G2 inverse is computed. The expensive Moore-Penrose inverse computes an estimate of the null space using an eigenvalue decomposition. The cheaper G2 inverse is produced by sweeping the linearly independent rows and columns and zeroing out the dependent ones. The information matrix, the approximate covariance matrix of the parameter estimates, and the approximate standard errors are not computed in the cases of unweighted or diagonally weighted least-squares estimation.

Assessment of Fit

This section contains a collection of formulas used in computing indices to assess the goodness of fit by PROC CALIS. The following notation is used:

- N for the sample size
- n for the number of manifest variables
- t for the number of parameters to estimate
- $NM = \begin{cases} (N-1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$
- df for the degrees of freedom
- $\gamma = \mathbf{X}$ for the t vector of optimal parameter estimates
- $\mathbf{S} = (s_{ij})$ for the $n \times n$ input COV, CORR, UCOV, or UCORR matrix
- $\mathbf{C} = (c_{ij}) = \hat{\mathbf{\Sigma}} = \mathbf{\Sigma}(\hat{\gamma})$ for the predicted model matrix
- $\mathbf{W}$ for the weight matrix ($\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for default GLS, and $\mathbf{W} = \mathbf{C}$ for ML estimates)
- $\mathbf{U}$ for the $n^2 \times n^2$ asymptotic covariance matrix of sample covariances
- $\Phi(x|\lambda, df)$ for the cumulative distribution function of the noncentral chi-squared distribution with noncentrality parameter λ

The following notation is for indices that allow testing nested models by a χ^2 difference test:

- f_0 for the function value of the independence model
- df_0 for the degrees of freedom of the independence model
- $f_{min} = F$ for the function value of the fitted model
- $df_{min} = df$ for the degrees of freedom of the fitted model

The degrees of freedom df_{min} and the number of parameters t are adjusted automatically when there are active constraints in the analysis. The computation of many fit statistics and indices are affected. You can turn off the automatic adjustment using the NOADJDF option. See the section "Counting the Degrees of Freedom" on page 676 for more information.

Residuals

PROC CALIS computes four types of residuals and writes them to the OUTSTAT= data set.

- **Raw Residuals**

  $$Res = \mathbf{S} - \mathbf{C}, Res_{ij} = s_{ij} - c_{ij}$$

  The raw residuals are displayed whenever the PALL, the PRINT, or the RESIDUAL option is specified.

- **Variance Standardized Residuals**

  $$VSRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{s_{ii}s_{jj}}}$$

  The variance standardized residuals are displayed when you specify

  - the PALL, the PRINT, or the RESIDUAL option and METHOD=NONE, METHOD=ULS, or METHOD=DWLS
  - RESIDUAL=VARSTAND

  The variance standardized residuals are equal to those computed by the EQS 3 program (Bentler 1989).

- **Asymptotically Standardized Residuals**

  $$ASRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{v_{ij,ij}}}, \quad \text{where}$$

  $$v_{ij,ij} = diag(\mathbf{U} - \mathbf{J}Cov(\gamma)\mathbf{J}')_{ij}$$

  The matrix $\mathbf{J}$ is the $n^2 \times t$ Jacobian matrix $d\mathbf{\Sigma}/d\boldsymbol{\gamma}$, and $Cov(\gamma)$ is the $t \times t$ asymptotic covariance matrix of parameter estimates (the inverse of the information matrix). Asymptotically standardized residuals are displayed when one of the following conditions is met:

  - The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are computed for some other reason.
  - RESIDUAL= ASYSTAND is specified.

The asymptotically standardized residuals are equal to those computed by the LISREL 7 program (Jöreskog and Sörbom 1988) except for the denominator NM in the definition of matrix **U**.

- **Normalized Residuals**

$$NRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{u_{ij,ij}}}$$

where the diagonal elements $u_{ij,ij}$ of the $n^2 \times n^2$ asymptotic covariance matrix **U** of sample covariances are defined for the following methods.

- **GLS** as $u_{ij,ij} = \frac{1}{NM}(s_{ii}s_{jj} + s_{ij}^2)$
- **ML** as $u_{ij,ij} = \frac{1}{NM}(c_{ii}c_{jj} + c_{ij}^2)$
- **WLS** as $\frac{u_{ij,ij}=1}{NMw_{ij,ij}}$

Normalized residuals are displayed when one of the following conditions is met:

- The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are **not** computed for some other reason.
- RESIDUAL=NORM is specified.

The normalized residuals are equal to those computed by the LISREL VI program (Jöreskog and Sörbom 1985) except for the definition of the denominator NM in matrix **U**.

For estimation methods that are not BGLS estimation methods (Browne 1982, 1984), such as METHOD=NONE, METHOD=ULS, or METHOD=DWLS, the assumption of an asymptotic covariance matrix **U** of sample covariances does not seem to be appropriate. In this case, the normalized residuals should be replaced by the more relaxed variance standardized residuals. Computation of asymptotically standardized residuals requires computing the Jacobian and information matrices. This is computationally very expensive and is done only if the Jacobian matrix has to be computed for some other reason, that is, if at least one of the following items is true:

- The default, PRINT, or PALL displayed output is requested, and neither the NOMOD nor NOSTDERR option is specified.
- Either the MODIFICATION (included in PALL), PCOVES, or STDERR (included in default, PRINT, and PALL output) option is requested or RESIDUAL=ASYSTAND is specified.
- The LEVMAR or NEWRAP optimization technique is used.
- An OUTRAM= data set is specified without using the NOSTDERR option.
- An OUTEST= data set is specified without using the NOSTDERR option.

Since normalized residuals use an overestimate of the asymptotic covariance matrix of residuals (the diagonal of $\mathbf{U}$), the normalized residuals cannot be larger than the asymptotically standardized residuals (which use the diagonal of $\mathbf{U} - \mathbf{J}Cov(\gamma)\mathbf{J}'$).

Together with the residual matrices, the values of the average residual, the average off-diagonal residual, and the rank order of the largest values are displayed. The distribution of the normalized and standardized residuals is displayed also.

Goodness-of-Fit Indices Based on Residuals

The following items are computed for all five kinds of estimation:ULS, GLS, ML, WLS, and DWLS. All these indices are written to the OUTRAM= data set. The goodness of fit (GFI), adjusted goodness of fit (AGFI), and root mean square residual (RMR) are computed as in the LISREL VI program of Jöreskog and Sörbom (1985).

- **Goodness-of-Fit Index**
  The goodness-of-fit index for the ULS, GLS, and ML estimation methods is

$$GFI = 1 - \frac{Tr((\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2)}{Tr((\mathbf{W}^{-1}\mathbf{S})^2)}$$

  but for WLS and DWLS estimation, it is

$$GFI = 1 - \frac{Vec(s_{ij} - c_{ij})'\mathbf{W}^{-1}Vec(s_{ij} - c_{ij})}{Vec(s_{ij})'\mathbf{W}^{-1}Vec(s_{ij})}$$

  where $\mathbf{W} = diag$ for DWLS estimation, and $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$. For a constant weight matrix $\mathbf{W}$, the goodness-of-fit index is 1 minus the ratio of the minimum function value and the function value before any model has been fitted. The GFI should be between 0 and 1. The data probably do not fit the model if the GFI is negative or much larger than 1.

- **Adjusted Goodness-of-Fit Index**
  The AGFI is the GFI adjusted for the degrees of freedom of the model

$$AGFI = 1 - \frac{n(n+1)}{2df}(1 - GFI)$$

  The AGFI corresponds to the GFI in replacing the total sum of squares by the mean sum of squares.

  Caution:

  – Large n and small df can result in a negative AGFI. For example, GFI=0.90, n=19, and df=2 result in an AGFI of -8.5.
  – AGFI is not defined for a saturated model, due to division by $df = 0$.
  – AGFI is not sensitive to losses in df.

The AGFI should be between 0 and 1. The data probably do not fit the model if the AGFI is negative or much larger than 1. For more information, refer to Mulaik et al. (1989).

- **Root Mean Square Residual**
  The RMR is the mean of the squared residuals:

$$RMR = \sqrt{\frac{2}{n(n+1)} \sum_i^n \sum_j^i (s_{ij} - c_{ij})^2}$$

- **Parsimonious Goodness-of-Fit Index**
  The PGFI (Mulaik et al. 1989) is a modification of the GFI that takes the parsimony of the model into account:

$$PGFI = \frac{df_{min}}{df_0} GFI$$

  The PGFI uses the same parsimonious factor as the parsimonious normed Bentler-Bonett index (James, Mulaik, and Brett 1982).

Goodness-of-Fit Indices Based on the χ^2

The following items are transformations of the overall χ^2 value and in general depend on the sample size N. These indices are not computed for ULS or DWLS estimates.

- **Uncorrected χ^2**
  The overall χ^2 measure is the optimum function value F multiplied by $N-1$ if a CORR or COV matrix is analyzed, or multiplied by N if a UCORR or UCOV matrix is analyzed. This gives the likelihood ratio test statistic for the null hypothesis that the predicted matrix **C** has the specified model structure against the alternative that **C** is unconstrained. The χ^2 test is valid only if the observations are independent and identically distributed, the analysis is based on the nonstandardized sample covariance matrix **S**, and the sample size N is sufficiently large (Browne 1982; Bollen 1989b; Jöreskog and Sörbom 1985). For ML and GLS estimates, the variables must also have an approximately multivariate normal distribution. The notation Prob>Chi**2 means “the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed.”

$$\chi^2 = NM * F$$

  where F is the function value at the minimum.

- **χ_0^2 Value of the Independence Model**
  The χ_0^2 value of the independence model

$$\chi_0^2 = NM * f_0$$

  and the corresponding degrees of freedom df_0 can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989).

- **RMSEA Index (Steiger and Lind 1980)**
  The Steiger and Lind (1980) root mean squared error approximation (RMSEA) coefficient is

  $$\epsilon_\alpha = \sqrt{\max(\frac{F}{df} - \frac{1}{NM}, 0)}$$

  The lower and upper limits of the confidence interval are computed using the cumulative distribution function of the noncentral chi-squared distribution $\Phi(x|\lambda, df) = \alpha$, with $x = NM * F$, λ_L satisfying $\Phi(x|\lambda_L, df) = 1 - \frac{\alpha}{2}$, and λ_U satisfying $\Phi(x|\lambda_U, df) = \frac{\alpha}{2}$:

  $$(\epsilon_{\alpha_L}; \epsilon_{\alpha_U}) = (\sqrt{\frac{\lambda_L}{NM * df}}; \sqrt{\frac{\lambda_U}{NM * df}})$$

  Refer to Browne and Du Toit (1992) for more details. The size of the confidence interval is defined by the option ALPHARMS=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the RMSEA.

- **Probability for Test of Close Fit (Browne and Cudeck 1993)**
  The traditional exact χ^2 test hypothesis $H_0: \epsilon_\alpha = 0$ is replaced by the null hypothesis of close fit $H_0: \epsilon_\alpha \leq 0.05$ and the exceedance probability P is computed as

  $$P = 1 - \Phi(x|\lambda^*, df)$$

  where $x = NM * F$ and $\lambda^* = 0.05^2 * NM * df$. The null hypothesis of close fit is rejected if P is smaller than a prespecified level (for example, $P < 0.05$).

- **Expected Cross Validation Index (Browne and Cudeck 1993)**
  For GLS and WLS, the estimator c of the ECVI is linearly related to AIC:

  $$c = F(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM}$$

  For ML estimation, c_{ML} is used.

  $$c_{ML} = F_{ML}(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM - n - 1}$$

  The confidence interval $(c_L; c_U)$ for c is computed using the cumulative distribution function $\Phi(x|\lambda, df)$ of the noncentral chi-squared distribution,

  $$(c_L; c_U) = (\frac{\lambda_L + nnt}{NM}; \frac{\lambda_U + nnt}{NM})$$

  with $nnt = n(n+1)/2 + t$, $x = NM * F$, $\Phi(x|\lambda_U, df) = 1 - \frac{\alpha}{2}$, and $\Phi(x|\lambda_L, df) = \frac{\alpha}{2}$. The confidence interval $(c^*_L; c^*_U)$ for c_{ML} is

  $$(c^*_L; c^*_U) = (\frac{\lambda^*_L + nnt}{NM - n - 1}; \frac{\lambda^*_U + nnt}{NM - n - 1})$$

where $nnt = n(n+1)/2 + t$, $x = (NM - n - 1) * F$, $\Phi(x|\lambda_U^*, df) = \frac{\alpha}{2}$ and $\Phi(x|\lambda_L^*, df) = 1 - \frac{\alpha}{2}$. Refer to Browne and Cudeck (1993). The size of the confidence interval is defined by the option ALPHAECV=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the ECVI.

- **Comparative Fit Index (Bentler 1989)**

$$CFI = 1 - \frac{\max(NM * f_{min} - df_{min}, 0)}{\max(NM * f_0 - df_0, 0)}$$

- **Adjusted χ^2 Value (Browne 1982)**
  If the variables are n-variate elliptic rather than normal and have significant amounts of multivariate kurtosis (leptokurtic or platykurtic), the χ^2 value can be adjusted to

$$\chi^2_{ell} = \frac{\chi^2}{\eta_2}$$

where η_2 is the multivariate relative kurtosis coefficient.

- **Normal Theory Reweighted LS χ^2 Value**
  This index is displayed only if METHOD=ML. Instead of the function value F_{ML}, the reweighted goodness-of-fit function F_{GWLS} is used,

$$\chi^2_{GWLS} = NM * F_{GWLS}$$

where F_{GWLS} is the value of the function at the minimum.

- **Akaike's Information Criterion (AIC) (Akaike 1974; Akaike 1987)**
  This is a criterion for selecting the best model among a number of candidate models. The model that yields the smallest value of AIC is considered the best.

$$AIC = \chi^2 - 2df$$

- **Consistent Akaike's Information Criterion (CAIC) (Bozdogan 1987)**
  This is another criterion, similar to AIC, for selecting the best model among alternatives. The model that yields the smallest value of CAIC is considered the best. CAIC is preferred by some people to AIC or the χ^2 test.

$$CAIC = \chi^2 - (ln(N) + 1)df$$

- **Schwarz's Bayesian Criterion (SBC) (Schwarz 1978; Sclove 1987)**
  This is another criterion, similar to AIC, for selecting the best model. The model that yields the smallest value of SBC is considered the best. SBC is preferred by some people to AIC or the χ^2 test.

$$SBC = \chi^2 - ln(N)df$$

- **McDonald's Measure of Centrality (McDonald and Hartmann 1992)**

  $$CENT = exp(-\frac{(\chi^2 - df)}{2N})$$

- **Parsimonious Normed Fit Index (James, Mulaik, and Brett 1982)**
  The PNFI is a modification of Bentler-Bonett's normed fit index that takes parsimony of the model into account,

  $$PNFI = \frac{df_{min}}{df_0} \frac{(f_0 - f_{min})}{f_0}$$

  The PNFI uses the same parsimonious factor as the parsimonious GFI of Mulaik et al. (1989).

- **Z-Test (Wilson and Hilferty 1931)**
  The Z-Test of Wilson and Hilferty assumes an n-variate normal distribution:

  $$Z = \frac{\sqrt[3]{\frac{\chi^2}{df}} - (1 - \frac{2}{9df})}{\sqrt{\frac{2}{9df}}}$$

  Refer to McArdle (1988) and Bishop, Fienberg, and Holland (1977, p. 527) for an application of the Z-Test.

- **Nonnormed Coefficient (Bentler and Bonett 1980)**

  $$\rho = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0 - 1/NM}$$

  Refer to Tucker and Lewis (1973).

- **Normed Coefficient (Bentler and Bonett 1980)**

  $$\Delta = \frac{f_0 - f_{min}}{f_0}$$

  Mulaik et al. (1989) recommend the parsimonious weighted form PNFI.

- **Normed Index ρ_1 (Bollen 1986)**

  $$\rho_1 = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0}$$

  ρ_1 is always less than or equal to 1; $\rho_1 < 0$ is unlikely in practice. Refer to the discussion in Bollen (1989a).

- **Nonnormed Index Δ_2 (Bollen 1989a)**

$$\Delta_2 = \frac{f_0 - f_{min}}{f_0 - \frac{df}{NM}}$$

is a modification of Bentler & Bonett's Δ that uses df and "lessens the dependence" on N. Refer to the discussion in Bollen (1989b). Δ_2 is identical to Mulaik et al.'s (1989) IFI2 index.

- **Critical N Index (Hoelter 1983)**

$$CN = \frac{\chi^2_{crit}}{F} + 1$$

where χ^2_{crit} is the critical chi-square value for the given df degrees of freedom and probability $\alpha = 0.05$, and F is the value of the estimation criterion (minimization function). Refer to Bollen (1989b, p. 277). Hoelter (1983) suggests that CN should be at least 200; however, Bollen (1989b) notes that the CN value may lead to an overly pessimistic assessment of fit for small samples.

Squared Multiple Correlation

The following are measures of the squared multiple correlation for manifest and endogenous variables and are computed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). The DETAE, DETSE, and DETMV determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. These coefficients are displayed only when you specify the PDETERM option with a RAM or LINEQS model.

- **R^2 Values Corresponding to Endogenous Variables**

$$R_i^2 = 1 - \frac{\widehat{var(\boldsymbol{\zeta}_i)}}{\widehat{var(\boldsymbol{\eta}_i)}}$$

- **Total Determination of All Equations**

$$DETAE = 1 - \frac{det(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Psi}})}{det(\widehat{Cov(y, x, \boldsymbol{\eta})})}$$

- **Total Determination of the Structural Equations**

$$DETSE = 1 - \frac{det(\hat{\boldsymbol{\Psi}})}{det(\widehat{Cov(\boldsymbol{\eta})})}$$

- **Total Determination of the Manifest Variables**

$$DETMV = 1 - \frac{det(\hat{\mathbf{\Theta}})}{det(\mathbf{S})}$$

Caution: In the LISREL program, the structural equations are defined by specifying the BETA matrix. In PROC CALIS, a structural equation has a dependent left-hand-side variable that appears at least once on the right-hand side of another equation, or the equation has at least one right-hand-side variable that is the left-hand-side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

Measures of Multivariate Kurtosis

In many applications, the manifest variables are not even approximately multivariate normal. If this happens to be the case with your data set, the default generalized least-squares and maximum likelihood estimation methods are not appropriate, and you should compute the parameter estimates and their standard errors by an asymptotically distribution-free method, such as the WLS estimation method. If your manifest variables are multivariate normal, then they have a zero relative multivariate kurtosis, and all marginal distributions have zero kurtosis (Browne 1982). If your DATA= data set contains raw data, PROC CALIS computes univariate skewness and kurtosis and a set of multivariate kurtosis values. By default, the values of univariate skewness and kurtosis are corrected for bias (as in PROC UNIVARIATE), but using the BIASKUR option enables you to compute the uncorrected values also. The values are displayed when you specify the PROC CALIS statement option KURTOSIS.

- **Corrected Variance for Variable z_j**

$$\sigma_j^2 = \frac{1}{N-1}\sum_i^N (z_{ij} - \overline{z_j})^2$$

- **Corrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N}{(N-1)(N-2)} \frac{\sum_i^N (z_{ij} - \overline{z_j})^3}{\sigma_j^3}$$

- **Uncorrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N\sum_i^N (z_{ij} - \overline{z_j})^3}{\sqrt{N[\sum_i^N (z_{ij} - \overline{z_j})^2]^3}}$$

- **Corrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \frac{\sum_i^N (z_{ij} - \overline{z_j})^4}{\sigma_j^4} - \frac{3(N-1)^2}{(N-2)(N-3)}$$

- **Uncorrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N \sum_i^N (z_{ij} - \overline{z_j})^4}{[\sum_i^N (z_{ij} - \overline{z_j})^2]^2} - 3$$

- **Mardia's Multivariate Kurtosis**

$$\gamma_2 = \frac{1}{N} \sum_i^N [(z_i - \overline{z})' S^{-1} (z_i - \overline{z})]^2 - n(n+2)$$

- **Relative Multivariate Kurtosis**

$$\eta_2 = \frac{\gamma_2 + n(n+2)}{n(n+2)}$$

- **Normalized Multivariate Kurtosis**

$$\kappa_0 = \frac{\gamma_2}{\sqrt{8n(n+2)/N}}$$

- **Mardia Based Kappa**

$$\kappa_1 = \frac{\gamma_2}{n(n+2)}$$

- **Mean Scaled Univariate Kurtosis**

$$\kappa_2 = \frac{1}{3n} \sum_j^n \gamma_{2(j)}$$

- **Adjusted Mean Scaled Univariate Kurtosis**

$$\kappa_3 = \frac{1}{3n} \sum_j^n \gamma_{2(j)}^*$$

with

$$\gamma_{2(j)}^* = \begin{cases} \gamma_{2(j)} \quad , & if \quad \gamma_{2(j)} > \frac{-6}{n+2} \\ \frac{-6}{n+2} \quad , & \text{otherwise} \end{cases}$$

If variable Z_j is normally distributed, the uncorrected univariate kurtosis $\gamma_{2(j)}$ is equal to 0. If Z has an n-variate normal distribution, Mardia's multivariate kurtosis γ_2 is equal to 0. A variable Z_j is called *leptokurtic* if it has a positive value of $\gamma_{2(j)}$ and is called *platykurtic* if it has a negative value of $\gamma_{2(j)}$. The values of κ_1, κ_2, and κ_3 should not be smaller than a lower bound (Bentler 1985):

$$\hat{\kappa} \geq \frac{-2}{n+2}$$

PROC CALIS displays a message if this happens.

If weighted least-squares estimates (METHOD=WLS or METHOD=ADF) are specified and the weight matrix is computed from an input raw data set, the CALIS procedure computes two more measures of multivariate kurtosis.

- **Multivariate Mean Kappa**

  $$\kappa_4 = \frac{1}{m} \sum_i^n \sum_j^i \sum_k^j \sum_l^k \hat{\kappa}_{ij,kl} - 1$$

  where

  $$\hat{\kappa}_{ij,kl} = \frac{s_{ij,kl}}{s_{ij}s_{kl} + s_{ik}s_{jl} + s_{il}s_{jk}}$$

  and $m = n(n+1)(n+2)(n+3)/24$ is the number of elements in the vector $s_{ij,kl}$ (Bentler 1985).

- **Multivariate Least-Squares Kappa**

  $$\kappa_5 = \frac{s_4' s_2}{s_2' s_2} - 1$$

  where

  $$s_{ij,kl} = \frac{1}{N} \sum_{r=1}^N (z_{ri} - \overline{z_i})(z_{rj} - \overline{z_j})(z_{rk} - \overline{z_k})(z_{rl} - \overline{z_l})$$

  s_4 is the vector of the $s_{ij,kl}$, and s_2 is the vector of the elements in the denominator of $\hat{\kappa}$ (Bentler 1985).

The occurrence of significant nonzero values of Mardia's multivariate kurtosis γ_2 and significant amounts of some of the univariate kurtosis values $\gamma_{2(j)}$ indicate that your variables are not multivariate normal distributed. Violating the multivariate normality assumption in (default) generalized least-squares and maximum likelihood estimation usually leads to the wrong approximate standard errors and incorrect fit statistics based on the χ^2 value. In general, the parameter estimates are more stable against violation of the normal distribution assumption. For more details, refer to Browne (1974, 1982, 1984).

Initial Estimates

Each optimization technique requires a set of initial values for the parameters. To avoid local optima, the initial values should be as close as possible to the globally optimal solution. You can check for local optima by running the analysis with several different sets of initial values; the RANDOM= option in the PROC CALIS statement is useful in this regard.

- RAM and LINEQS: There are several default estimation methods available in PROC CALIS for initial values of parameters in a linear structural equation model specified by a RAM or LINEQS model statement, depending on the form of the specified model.
  - two-stage least-squares estimation
  - instrumental variable method (Hägglund 1982; Jennrich 1987)
  - approximative factor analysis method
  - ordinary least-squares estimation
  - estimation method of McDonald (McDonald and Hartmann 1992)
- FACTOR: For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method if possible. McDonald's method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than setting the loadings of the reference variables equal to 1. If none of the two methods seems to be appropriate, the initial values are set by the START= option.
- COSAN: For the more general COSAN model, there is no default estimation method for the initial values. In this case, the START= or RANDOM= option can be used to set otherwise unassigned initial values.

Poor initial values can cause convergence problems, especially with maximum likelihood estimation. You should not specify a constant initial value for all parameters since this would produce a singular predicted model matrix in the first iteration. Sufficiently large positive diagonal elements in the central matrices of each model matrix term provide a nonnegative definite initial predicted model matrix. If maximum likelihood estimation fails to converge, it may help to use METHOD=LSML, which uses the final estimates from an unweighted least-squares analysis as initial estimates for maximum likelihood. Or you can fit a slightly different but better-behaved model and produce an OUTRAM= data set, which can then be modified in accordance with the original model and used as an INRAM= data set to provide initial values for another analysis.

If you are analyzing a covariance or scalar product matrix, be sure to take into account the scales of the variables. The default initial values may be inappropriate when some variables have extremely large or small variances.

Automatic Variable Selection

You can use the VAR statement to reorder the variables in the model and to delete the variables not used. Using the VAR statement saves memory and computation time. If a linear structural equation model using the RAM or LINEQS statement (or an INRAM= data set specifying a RAM or LINEQS model) does not use all the manifest variables given in the input DATA= data set, PROC CALIS automatically deletes those manifest variables not used in the model.

In some special circumstances, the automatic variable selection performed for the RAM and LINEQS statements may be inappropriate, for example, if you are interested in modification indices connected to some of the variables that are not used in the model. You can include such manifest variables as exogenous variables in the analysis by specifying constant zero coefficients.

For example, the first three steps in a stepwise regression analysis of the Werner Blood Chemistry data (Jöreskog and Sörbom 1988, p. 111) can be performed as follows:

```
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=0 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
   std    e=var;
run;
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
   std    e=var;
run;
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+g6 x6+0 x7+e;
   std    e=var;
run;
```

Using the COSAN statement does not automatically delete those variables from the analysis that are not used in the model. You can use the output of the predetermined values in the predicted model matrix (PREDET option) to detect unused variables. Variables that are not used in the model are indicated by 0 in the rows and columns of the predetermined predicted model matrix.

Exogenous Manifest Variables

If there are exogenous manifest variables in the linear structural equation model, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix ($\mathbf{P}$ or $\mathbf{\Phi}$). In general, using exogenous manifest variables reduces the degrees of freedom since the corresponding sample correlations or covariances are not part of the exogenous information provided for the parameter estimation. See the section "Counting the Degrees of Freedom" on page 676 for more information.

If you specify a RAM or LINEQS model statement, or if such a model is recognized in an INRAM= data set, those elements in the central model matrices that correspond to the exogenous manifest variables are reset to the sample values after computing covariances or correlations within the current BY group.

The COSAN statement does not automatically set the covariances in the central model matrices that correspond to manifest exogenous variables.

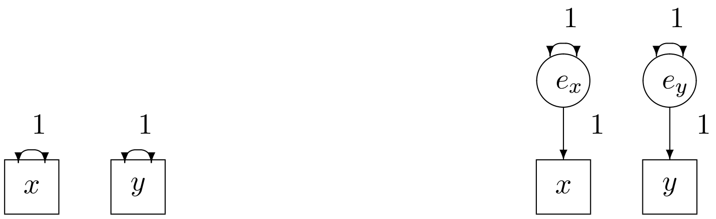
You can use the output of the predetermined values in the predicted model matrix (PREDET option) that correspond to manifest exogenous variables to see which of the manifest variables are exogenous variables and to help you set the corresponding locations of the central model matrices with their covariances.

The following two examples show how different the results of PROC CALIS can be if manifest variables are considered either as endogenous or as exogenous variables. (See Figure 19.5.) In both examples, a correlation matrix **S** is tested against an identity model matrix **C**; that is, no parameter is estimated. The three runs of the first example (specified by the COSAN, LINEQS, and RAM statements) consider the two variables y and x as endogenous variables.

```
title2 'Data: FULLER (1987, p.18)';
data corn;
   input y x;
   datalines;
 86  70
115  97
 90  53
 86  64
110  95
 91  64
 99  50
 96  70
 99  94
104  69
 96  51
;

title3 'Endogenous Y and X';
proc calis data=corn;
   cosan corr(2,ide);
run;
proc calis data=corn;
   lineqs
          y=ey,
          x=ex;
   std    ey ex=2 * 1;
run;
proc calis data=corn;
   ram
       1  1  3  1.,
       1  2  4  1.,
       2  3  3  1.,
       2  4  4  1.;
run;
```

The two runs of the second example (specified by the LINEQS and RAM statements) consider y and x as exogenous variables.

```
title3 'Exogenous Y and X';
proc calis data=corn;
   std y x=2 * 1;
run;
proc calis data=corn;
   ram
       2  1  1  1.,
       2  2  2  1.;
run;
```



Exogenous x, y

Endogenous x, y

Figure 19.5. Exogenous and Endogenous Variables

The LINEQS and the RAM model statements set the covariances (correlations) of exogenous manifest variables in the estimated model matrix and automatically reduce the degrees of freedom.

Use of Optimization Techniques

No algorithm for optimizing general nonlinear functions exists that will always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC CALIS provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC CALIS will find the correct solution. All optimization techniques in PROC CALIS use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters.

The PROC CALIS statement NLOPTIONS can be especially helpful for tuning applications with nonlinear equality and inequality constraints on the parameter estimates. Some of the options available in NLOPTIONS may also be invoked as PROC CALIS options. The NLOPTIONS statement can specify almost the same options as the SAS/OR NLP procedure.

Nonlinear optimization requires the repeated computation of

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)

- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)
- values of linear and nonlinear constraints
- the first-order partial derivatives (Jacobian) of nonlinear constraints

For the criteria used by PROC CALIS, computing the gradient takes more computer time than computing the function value, and computing the Hessian takes *much* more computer time and memory than computing the gradient, especially when there are many parameters to estimate. Unfortunately, optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower. Techniques that do not use the Hessian also tend to be less reliable (for example, they may terminate at local rather than global optima).

The available optimization techniques are displayed in Table 19.13 and can be chosen by the TECH=$name$ option.

Table 19.13. Optimization Techniques

TECH=	**Optimization Technique**
LEVMAR	Levenberg-Marquardt Method
TRUREG	Trust-Region Method
NEWRAP	Newton-Raphson Method with Line Search
NRRIDG	Newton-Raphson Method with Ridging
QUANEW	Quasi-Newton Methods (DBFGS, DDFP, BFGS, DFP)
DBLDOG	Double-Dogleg Method (DBFGS, DDFP)
CONGRA	Conjugate Gradient Methods (PB, FR, PR, CD)

Table 19.14 shows, for each optimization technique, which derivatives are needed (first-order or second-order) and what kind of constraints (boundary, linear, or nonlinear) can be imposed on the parameters.

Table 19.14. Derivatives Needed and Constraints Allowed

	Derivatives		**Constraints**		
TECH=	**First Order**	**Second Order**	**Boundary**	**Linear**	**Nonlinear**
LEVMAR	x	x	x	x	-
TRUREG	x	x	x	x	-
NEWRAP	x	x	x	x	-
NRRIDG	x	x	x	x	-
QUANEW	x	-	x	x	x
DBLDOG	x	-	x	x	-
CONGRA	x	-	x	x	-

The Levenberg-Marquardt, trust-region, and Newton-Raphson techniques are usually the most reliable, work well with boundary and general linear constraints, and generally converge after a few iterations to a precise solution. However, these techniques need to compute a Hessian matrix in each iteration. For HESSALG=1, this means that you need about $4(n(n+1)/2)t$ bytes of work memory ($n =$ the number of manifest variables, $t =$ the number of parameters to estimate) to store the Jacobian and its cross product. With HESSALG=2 or HESSALG=3, you do not need this work memory, but the use of a utility file increases execution time. Computing the approximate

Hessian in each iteration can be very time- and memory-consuming, especially for large problems (more than 60 or 100 parameters, depending on the computer used). For large problems, a quasi-Newton technique, especially with the BFGS update, can be far more efficient.

For a poor choice of initial values, the Levenberg-Marquardt method seems to be more reliable.

If memory problems occur, you can use one of the conjugate gradient techniques, but they are generally slower and less reliable than the methods that use second-order information.

There are several options to control the optimization process. First of all, you can specify various termination criteria. You can specify the GCONV= option to specify a relative gradient termination criterion. If there are active boundary constraints, only those gradient components that correspond to inactive constraints contribute to the criterion. When you want very precise parameter estimates, the GCONV= option is useful. Other criteria that use relative changes in function values or parameter estimates in consecutive iterations can lead to early termination when active constraints cause small steps to occur. The small default value for the FCONV= option helps prevent early termination. Using the MAXITER= and MAXFUNC= options enables you to specify the maximum number of iterations and function calls in the optimization process. These limits are especially useful in combination with the INRAM= and OUTRAM= options; you can run a few iterations at a time, inspect the results, and decide whether to continue iterating.

Nonlinearly Constrained QN Optimization

The algorithm used for nonlinearly constrained quasi-Newton optimization is an efficient modification of Powell's (1978a, 1978b, 1982a, 1982b) *Variable Metric Constrained WatchDog* (VMCWD) algorithm. A similar but older algorithm (VF02AD) is part of the Harwell library. Both VMCWD and VF02AD use Fletcher's VE02AD algorithm (also part of the Harwell library) for positive definite quadratic programming. The PROC CALIS QUANEW implementation uses a quadratic programming subroutine that updates and downdates the approximation of the Cholesky factor when the active set changes. The nonlinear QUANEW algorithm is not a feasible point algorithm, and the value of the objective function need not decrease (minimization) or increase (maximization) monotonically. Instead, the algorithm tries to reduce a linear combination of the objective function and constraint violations, called the *merit function*.

The following are similarities and differences between this algorithm and VMCWD:

- A modification of this algorithm can be performed by specifying VERSION=1, which replaces the update of the Lagrange vector μ with the original update of Powell (1978a, 1978b), which is used in VF02AD. This can be helpful for some applications with linearly dependent active constraints.
- If the VERSION= option is not specified or VERSION=2 is specified, the evaluation of the Lagrange vector μ is performed in the same way as Powell (1982a, 1982b) describes.

- Instead of updating an approximate Hessian matrix, this algorithm uses the dual BFGS (or DFP) update that updates the Cholesky factor of an approximate Hessian. If the condition of the updated matrix gets too bad, a restart is done with a positive diagonal matrix. At the end of the first iteration after each restart, the Cholesky factor is scaled.
- The Cholesky factor is loaded into the quadratic programming subroutine, automatically ensuring positive definiteness of the problem. During the quadratic programming step, the Cholesky factor of the projected Hessian matrix $\mathbf{Z}'_k\mathbf{G}\mathbf{Z}_k$ and the QT decomposition are updated simultaneously when the active set changes. Refer to Gill et al. (1984) for more information.
- The line-search strategy is very similar to that of Powell (1982a, 1982b). However, this algorithm does not call for derivatives during the line search; hence, it generally needs fewer derivative calls than function calls. The VMCWD algorithm always requires the same number of derivative and function calls. It was also found in several applications of VMCWD that Powell's line-search method sometimes uses steps that are too long during the first iterations. In those cases, you can use the INSTEP= option specification to restrict the step length α of the first iterations.
- Also the watchdog strategy is similar to that of Powell (1982a, 1982b). However, this algorithm doesn't return automatically after a fixed number of iterations to a former better point. A return here is further delayed if the observed function reduction is close to the expected function reduction of the quadratic model.
- Although Powell's termination criterion still is used (as FCONV2), the QUANEW implementation uses two additional termination criteria (GCONV and ABSGCONV).

This algorithm is automatically invoked when you specify the NLINCON statement. The nonlinear QUANEW algorithm needs the Jacobian matrix of the first-order derivatives (constraints normals) of the constraints

$$(\nabla c_i) = (\frac{\partial c_i}{\partial x_j}), \quad i = 1, \dots, nc, j = 1, \dots, n$$

where nc is the number of nonlinear constraints for a given point x.

You can specify two update formulas with the UPDATE= option:

- UPDATE=DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
- UPDATE=DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

This algorithm uses its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome.

You can specify the INSTEP= option to impose an upper bound for the step size α during the first five iterations. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the nonlinear constraints QUANEW algorithm.

Optimization and Iteration History

The optimization and iteration histories are displayed by default because it is important to check for possible convergence problems.

The optimization history includes the following summary of information about the initial state of the optimization.

- the number of constraints that are active at the starting point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the starting point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- for the TRUREG and LEVMAR subroutines, the initial radius of the trust region around the starting point

The optimization history ends with some information concerning the optimization result:

- the number of constraints that are active at the final point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the final point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- other information specific to the optimization technique

The iteration history generally consists of one line of displayed output containing the most important information for each iteration. The _LIST_ variable (see the "SAS Program Statements" section on page 628) also enables you to display the parameter estimates and the gradient in some or all iterations.

The iteration history always includes the following (the words in parentheses are the column header output):

- the iteration number (Iter)
- the number of iteration restarts (rest)
- the number of function calls (nfun)
- the number of active constraints (act)
- the value of the optimization criterion (optcrit)
- the difference between adjacent function values (difcrit)
- the maximum of the absolute gradient components corresponding to inactive boundary constraints (maxgrad)

An apostrophe trailing the number of active constraints indicates that at least one of the active constraints is released from the active set due to a significant Lagrange multiplier.

For the Levenberg-Marquardt technique (LEVMAR), the iteration history also includes the following information:

- An asterisk trailing the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value. If all or the last several iterations show a singular Hessian approximation, the problem is not sufficiently identified. Thus, there are other locally optimal solutions that lead to the same optimum function value for different parameter values. This implies that standard errors for the parameter estimates are not computable without the addition of further constraints.
- the value of the Lagrange multiplier (lambda); this is 0 if the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than 0 when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratic constraint optimization is performed). Large values indicate optimization difficulties. For a nonsingular Hessian matrix, the value of lambda should go to 0 during the last iterations, indicating that the objective function can be well approximated by a quadratic function in a small neighborhood of the optimum point. An increasing lambda value often indicates problems in the optimization process.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1 indicate optimization difficulties. The value of the ratio ρ indicates the goodness of the quadratic function approximation; in other words, $\rho << 1$ means that the radius of the trust region has to be reduced. A fairly large value of ρ means that the radius of the trust region need not be changed. And a value close to or larger than 1 means that the radius can be increased, indicating a good quadratic function approximation.

For the Newton-Raphson technique (NRRIDG), the iteration history also includes the following information:

- the value of the ridge parameter. This is 0 when a Newton step can be performed, and it is greater than 0 when either the Hessian approximation is singular or a Newton step fails to reduce the optimization criterion. Large values indicate optimization difficulties.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1.0 indicate optimization difficulties.

For the Newton-Raphson with line-search technique (NEWRAP), the iteration history also includes

- the step size α (alpha) computed with one of the line-search algorithms
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

For the Trust-Region technique (TRUREG), the iteration history also includes the following information.

- An asterisk after the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value.
- the value of the Lagrange multiplier (lambda). This value is zero when the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than zero when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratically constrained optimization is performed). Large values indicate optimization difficulties. As in Gay (1983), a negative lambda value indicates the special case of an indefinite Hessian matrix (the smallest eigenvalue is negative in minimization).
- the value of the radius Δ of the trust region. Small trust region radius values combined with large lambda values in subsequent iterations indicate optimization problems.

For the quasi-Newton (QUANEW) and conjugate gradient (CONGRA) techniques, the iteration history also includes the following information:

- the step size (alpha) computed with one of the line-search algorithms
- the descent of the search direction at the current parameter iterate. This value should be significantly smaller than 0. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

Frequent update restarts (rest) of a quasi-Newton algorithm often indicate numerical problems related to required properties of the approximate Hessian update, and they decrease the speed of convergence. This can happen particularly if the ABSGCONV= termination criterion is too small, that is, when the requested precision cannot be obtained by quasi-Newton optimization. Generally, the number of automatic restarts used by conjugate gradient methods are much higher.

For the nonlinearly constrained quasi-Newton technique, the iteration history also includes the following information:

- the maximum value of all constraint violations,

$$\text{conmax} = \max(|c_i(x)| : c_i(x) < 0)$$

- the value of the predicted function reduction used with the GCONV and FCONV2 termination criteria,

$$\text{pred} = |g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|$$

- the step size α of the quasi-Newton step. Note that this algorithm works with a special line-search algorithm.
- the maximum element of the gradient of the Lagrange function,

$$\begin{aligned} \text{lfgmax} &= \nabla_x L(x^{(k)}, \lambda^{(k)}) \\ &= \nabla_x f(x^{(k)}) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla_x c_i(x^{(k)}) \end{aligned}$$

For the double dogleg technique, the iteration history also includes the following information:

- the parameter λ of the double-dogleg step. A value $\lambda = 0$ corresponds to the full (quasi) Newton step.
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative.

Line-Search Methods

In each iteration k, the (dual) quasi-Newton, hybrid quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of the nonlinear objective function f of n parameters x along a feasible descent search direction $s^{(k)}$

$$f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)})$$

by computing an approximately optimal scalar $\alpha^{(k)} > 0$. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (Fletcher 1987).

Various line-search algorithms can be selected by using the LIS= option (page 580). The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand-side and right-hand-side Goldstein conditions (refer to Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default value for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead to feasible points where the objective function f is no longer defined (resulting in indefinite matrices for ML estimation) or is difficult to compute (resulting in floating point overflows). Therefore, PROC CALIS provides options that restrict the step length or trust region radius, especially during the first main iterations.

The inner product $g's$ of the gradient g and the search direction s is the slope of $f(\alpha) = f(x + \alpha s)$ along the search direction s with step length α. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(x+\alpha s)$) during the main iteration k is computed in three steps.

1. Use either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final stepsize value α^- of the previous iteration $k-1$ to compute a first value $\alpha_1^{(0)}$.

   - Using the DAMPSTEP<=r> option:

     $$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

     The initial value for the new step length can be no larger than r times the final step length α^- of the previous iteration. The default is $r = 2$.
   - Not using the DAMPSTEP option:

     $$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \le step \le 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

     with

     $$step = \begin{cases} df/|g's| & \text{if } |g's| \ge \epsilon \max(100df, 1) \\ 1 & \text{otherwise} \end{cases}$$

This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ using the INSTEP=r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

   After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

   where u is the maximum length of a step inside the feasible region.

The INSTEP=r option lets you specify a smaller or larger radius of the trust region used in the first iteration by the trust-region, double-dogleg, and Levenberg-Marquardt algorithm. The default initial trust region radius is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. This choice is successful in most practical applications of the TRUREG, DBLDOG, and LEVMAR algorithms. However, for bad initial values used in the analysis of a covariance matrix with high variances, or for highly nonlinear constraints (such as using the EXP function) in your programming code, the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r, $0 < r < 1$, until the iteration starts successfully. A small factor r also affects the trust region radius of the next steps because the radius is changed in each iteration by a factor $0 < c \leq 4$ depending on the ρ ratio. Reducing the radius corresponds to increasing the ridge parameter λ that produces smaller steps directed closer toward the gradient direction.

Modification Indices

While fitting structural models, you may want to modify the specified model in order to

- reduce the χ^2 value significantly
- reduce the number of parameters to estimate without increasing the χ^2 value too much

If you specify the MODIFICATION or MOD option, PROC CALIS computes and displays a default set of modification indices:

- **Univariate Lagrange multiplier test indices** for most elements in the model matrices that are constrained to *equal constants*. These are second-order approximations of the decrease in the χ^2 value that would result from allowing the constant matrix element to vary. Besides the value of the Lagrange multiplier, the corresponding probability ($df = 1$) and the approximate change of the parameter value (should the constant be changed to a parameter) are displayed. If allowing the constant to be a free estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Not all elements in the model matrices should be allowed to vary; the diagonal elements of the inverse matrices in the RAM or LINEQS model must be constant ones. The univariate Lagrange multipliers are displayed at the constant locations of the model matrices.

- **Univariate Wald test indices** for those matrix elements that correspond to *parameter estimates* in the model. These are second-order approximations of the increase in the χ^2 value that would result from constraining the parameter to a 0 constant. The univariate Wald test indices are the same as the t values that are displayed together with the parameter estimates and standard errors. The univariate Wald test indices are displayed at the parameter locations of the model matrices.
- **Univariate Lagrange multiplier test indices** that are second-order approximations of the decrease in the χ^2 value that would result from the release of *equality constraints*. Multiple equality constraints containing $n > 2$ parameters are tested successively in n steps, each assuming the release of one of the equality-constrained parameters. The expected change of the parameter values of the separated parameter and the remaining parameter cluster are displayed, too.
- **Univariate Lagrange multiplier test indices** for releasing *active boundary constraints* specified by the BOUNDS statement
- **Stepwise multivariate Wald test indices** for constraining estimated parameters to 0 are computed and displayed. In each step, the parameter that would lead to the smallest increase in the multivariate χ^2 value is set to 0. Besides the multivariate χ^2 value and its probability, the univariate increments are also displayed. The process stops when the univariate probability is smaller than the specified value in the SLMW= option.

All of the preceding tests are approximations. You can often get more accurate tests by actually fitting different models and computing likelihood ratio tests. For more details about the Wald and the Lagrange multiplier test, refer to MacCallum (1986), Buse (1982), Bentler (1986), or Lee (1985).

Note that, for large model matrices, the computation time for the default modification indices can considerably exceed the time needed for the minimization process.

The modification indices are not computed for unweighted least-squares or diagonally weighted least-squares estimation.

Caution: Modification indices are not computed if the model matrix is an identity matrix (IDE or ZID), a selection matrix (PER), or the first matrix **J** in the LINEQS model. If you want to display the modification indices for such a matrix, you should specify the matrix as another type; for example, specify an identity matrix used in the COSAN statement as a diagonal matrix with constant diagonal elements of 1.

Constrained Estimation Using Program Code

The CALIS procedure offers a very flexible way to constrain parameter estimates. You can use your own programming statements to express special properties of the parameter estimates. This tool is also present in McDonald's COSAN implementation but is considerably easier to use in the CALIS procedure. PROC CALIS is able to compute analytic first- and second-order derivatives that you would have to specify using the COSAN program. There are also three PROC CALIS statements you can use:

- the BOUNDS statement, to specify simple bounds on the parameters used in the optimization process
- the LINCON statement, to specify general linear equality and inequality constraints on the parameters used in the optimization process
- the NLINCON statement, to specify general nonlinear equality and inequality constraints on the parameters used in the optimization process. The variables listed in the NLINCON statement must be specified in the program code.

There are some traditional ways to enforce parameter constraints by using parameter transformations (McDonald 1980).

- **One-sided boundary constraints:** For example, the parameter q_k should be at least as large (or at most as small) as a given constant value a (or b),

  $$q_k \geq a \qquad \text{or} \qquad q_k \leq b$$

  This inequality constraint can be expressed as an equality constraint

  $$q_k = a + x_j^2 \qquad \text{or} \qquad q_k = b - x_j^2$$

  in which the fundamental parameter x_j is unconstrained.
- **Two-sided boundary constraints:** For example, the parameter q_k should be located between two given constant values a and b, $a < b$,

  $$a \leq q_k \leq b$$

  This inequality constraint can be expressed as an equality constraint

  $$q_k = a + b\frac{exp(x_j)}{1 + exp(x_j)}$$

  in which the fundamental parameter x_j is unconstrained.
- **One-sided order constraints:** For example, the parameters q_1 , . . . , q_k should be ordered in the form

  $$q_1 \leq q_2, \quad q_1 \leq q_3, \quad \ldots, \quad q_1 \leq q_k$$

These inequality constraints can be expressed as a set of equality constraints

$$q_1 = x_1, \quad q_2 = x_1 + x_2^2, \quad \ldots, \quad q_k = x_1 + x_k^2$$

in which the fundamental parameters x_1 , . . . , x_k are unconstrained.

- **Two-sided order constraints:** For example, the parameters q_1 , . . . , q_k should be ordered in the form

  $$q_1 \leq q_2 \leq q_3 \leq \ldots \leq q_k$$

  These inequality constraints can be expressed as a set of equality constraints

  $$q_1 = x_1, \quad q_2 = q_1 + x_2^2, \quad \ldots, \quad q_k = q_{k-1} + x_k^2$$

  in which the fundamental parameters x_1 , . . . , x_k are unconstrained.

- **Linear equation constraints:** For example, the parameters q_1, q_2, q_3 should be linearly constrained in the form

  $$q_1 + q_2 + q_3 = a$$

  which can be expressed in the form of three explicit equations in which the fundamental parameters x_1 and x_2 are unconstrained:

  $$q_1 = x_1, \quad q_2 = x_2, \quad q_3 = a - x_1 - x_2$$

Refer to McDonald (1980) and Browne (1982) for further notes on reparameterizing techniques. If the optimization problem is not too large to apply the Levenberg-Marquardt or Newton-Raphson algorithm, boundary constraints should be requested by the BOUNDS statement rather than by reparameterizing code. If the problem is so large that you must use a quasi-Newton or conjugate gradient algorithm, reparameterizing techniques may be more efficient than the BOUNDS statement.

Counting the Degrees of Freedom

In a regression problem, the number of degrees of freedom for the error estimate is the number of observations in the data set minus the number of parameters. The NOBS=, DFR= (RDF=), and DFE= (EDF=) options refer to degrees of freedom in this sense. However, these values are not related to the degrees of freedom of a test statistic used in a covariance or correlation structure analysis. The NOBS=, DFR=, and DFE= options should be used in PROC CALIS to specify only the effective number of observations in the input DATA= data set.

In general, the number of degrees of freedom in a covariance or correlation structure analysis is defined as the difference between the number of nonredundant values q in the observed $n \times n$ correlation or covariance matrix $\mathbf{S}$ and the number t of free parameters $\mathbf{X}$ used in the fit of the specified model, $df = q - t$. Both values, q and t, are counted differently in different situations by PROC CALIS.

The number of nonredundant values q is generally equal to the number of lower triangular elements in the $n \times n$ moment matrix $\mathbf{S}$ including all diagonal elements, minus a constant c dependent upon special circumstances,

$$q = n(n + 1)/2 - c$$

The number c is evaluated by adding the following quantities:

- If you specify a linear structural equation model containing exogenous manifest variables by using the RAM or LINEQS statement, PROC CALIS adds to c the number of variances and covariances among these manifest exogenous variables, which are automatically set in the corresponding locations of the central model matrices (see the section "Exogenous Manifest Variables" on page 662).
- If you specify the DFREDUCE=i option, PROC CALIS adds the specified number i to c. The number i can be a negative integer.
- If you specify the NODIAG option to exclude the fit of the diagonal elements of the data matrix $\mathbf{S}$, PROC CALIS adds the number n of diagonal elements to c.
- If all the following conditions hold, then PROC CALIS adds to c the number of the diagonal locations:
  - NODIAG and DFREDUC= options are not specified.
  - A correlation structure is being fitted.
  - The predicted correlation matrix contains constants on the diagonal.

In some complicated models, especially those using programming statements, PROC CALIS may not be able to detect all the constant predicted values. In such cases, you must specify the DFREDUCE= option to get the correct degrees of freedom.

The number t is the number of different parameter names used in constructing the model if you do not use programming statements to impose constraints on the parameters. Using programming statements in general introduces two kinds of parameters:

- independent parameters, which are used only at the right-hand side of the expressions
- dependent parameters, which are used at least once at the left-hand side of the expressions

The independent parameters belong to the parameters involved in the estimation process, whereas the dependent parameters are fully defined by the programming statements and can be computed from the independent parameters. In this case, the number t is the number of different parameter names used in the model specification, but not used in the programming statements, plus the number of independent parameters. The independent parameters and their initial values can be defined in a model specification statement or in a PARMS statement.

The degrees of freedom are automatically increased by the number of active constraints in the solution. Similarly, the number of parameters are decreased by the number of active constraints. This affects the computation of many fit statistics and indices. Refer to Dijkstra (1992) for a discussion of the validity of statistical inferences with active boundary constraints. If the researcher believes that the active constraints will have a small chance of occurrence in repeated sampling, it may be more suitable to turn off the automatic adjustment using the NOADJDF option.

Computational Problems

First Iteration Overflows

Analyzing a covariance matrix including high variances in the diagonal and using bad initial estimates for the parameters can easily lead to arithmetic overflows in the first iterations of the minimization algorithm. The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If this occurs with quasi-Newton or conjugate gradient minimization, you can specify the INSTEP= option to reduce the length of the first step. If an arithmetic overflow occurs in the first iteration of the Levenberg-Marquardt algorithm, you can specify the INSTEP= option to reduce the trust region radius of the first iteration. You also can change the minimization technique or the line-search method. If none of these help, you should consider

- scaling the covariance matrix
- providing better initial values
- changing the model

No Convergence of Minimization Process

If convergence does not occur during the minimization process, perform the following tasks:

- If there are *negative variance estimates* in the diagonal locations of the central model matrices, you can
  - specify the BOUNDS statement to obtain nonnegative variance estimates
  - specify the HEYWOOD option, if the FACTOR model statement is specified
- Change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
  - change to METHOD=LSML
  - run some iterations with METHOD=DWLS or METHOD=GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML

- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
    - change to TECH=QUANEW or to TECH=NEWRAP
    - run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH= technique
- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can
    - change the update formula with the UPDATE= option
    - change the line-search algorithm with the LIS= option
    - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3
- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.
- Change the initial values. For many categories of model specifications done by the LINEQS, RAM, or FACTOR model, PROC CALIS computes an appropriate set of initial values automatically. However, for some of the model specifications (for example, structural equations with latent variables on the left-hand side and manifest variables on the right-hand side), PROC CALIS can generate very obscure initial values. In these cases, you have to set the initial values yourself.
    - Increase the initial values of the parameters located at the diagonal of central matrices
        - manually, by setting the values in the model specification
        - automatically, by using the DEMPHAS= option
    - Use a slightly different, but more stable, model to obtain preliminary estimates.
    - Use additional information to specify initial values, for example, by using other SAS software like the FACTOR, REG, SYSLIN, and MODEL (SYSNLIN) procedures for the modified, unrestricted model case.

- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can
    - change to TECH=QUANEW or to TECH=NEWRAP
    - run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH= technique
- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can

  - change the update formula with the UPDATE= option
  - change the line-search algorithm with the LIS= option
  - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3

- Temporarily change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can
  - change to METHOD=LSML
  - run some iterations with METHOD=DWLS or GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML

- You can allow more iterations and function calls by using the MAXIT= and MAXFU= options.

Unidentified Model

The parameter vector $\mathbf{x}$ in the covariance structure model

$$\mathbf{C} = \mathbf{C}(\mathbf{x})$$

is said to be identified in a parameter space G, if

$$\mathbf{C}(\mathbf{x}) = \mathbf{C}(\tilde{\mathbf{x}}), \quad \tilde{\mathbf{x}} \in G$$

implies $\mathbf{x} = \tilde{\mathbf{x}}$. The parameter estimates that result from an unidentified model can be very far from the parameter estimates of a very similar but identified model. They are usually machine dependent. Don't use parameter estimates of an unidentified model as initial values for another run of PROC CALIS.

Singular Predicted Model Matrix

You can easily specify models with singular predicted model matrices, for example, by fixing diagonal elements of central matrices to 0. In such cases, you cannot compute maximum likelihood estimates (the ML function value F is not defined). Since singular predicted model matrices can also occur temporarily in the minimization process, PROC CALIS tries in such cases to change the parameter estimates so that the predicted model matrix becomes positive definite. In such cases, the following message is displayed:

```
NOTE: Parameter set changed.
```

This process does not always work well, especially if there are fixed instead of variable diagonal elements in the central model matrices. A famous example where you cannot compute ML estimates is a component analysis with fewer components than given manifest variables. See the section "FACTOR Model Statement" on page 606 for more details. If you continue to get a singular predicted model matrix after changing initial values and optimization techniques, then your model is perhaps specified so that ML estimates cannot be computed.

Saving Computing Time

For large models, the most computing time is needed to compute the modification indices. If you don't really need the Lagrange multipliers or multiple Wald test indices (the univariate Wald test indices are the same as the t values), using the NOMOD option can save a considerable amount of computing time.

Central Matrices with Negative Eigenvalues

A covariance matrix cannot have negative eigenvalues, since a negative eigenvalue means that some linear combination of the variables has negative variance. PROC CALIS displays a warning if a central model matrix has negative eigenvalues but does not actually compute the eigenvalues. Sometimes this warning can be triggered by 0 or very small positive eigenvalues that appear negative because of numerical error. If you want to be sure that the central model matrix you are fitting can be considered to be a variance-covariance matrix, you can use the SAS/IML command *VAL=EIGVAL(U)* to compute the vector *VAL* of eigenvalues of matrix **U**.

Negative R^2 *Values*

The estimated squared multiple correlations R^2 of the endogenous variables are computed using the estimated error variances

$$R_i^2 = 1 - \frac{\widehat{var(\zeta_i)}}{\widehat{var(\eta_i)}}$$

If the model is a poor fit, it is possible that $\widehat{var(\zeta_i)} > \widehat{var(\eta_i)}$, which results in $R_i^2 < 0$.

Displayed Output

The output displayed by PROC CALIS depends on the statement used to specify the model. Since an analysis requested by the LINEQS or RAM statement implies the analysis of a structural equation model, more statistics can be computed and displayed than for a covariance structure analysis following the generalized COSAN model requested by the COSAN statement. The displayed output resulting from use of the FACTOR statement includes all the COSAN displayed output as well as more statistics displayed only when you specify the FACTOR statement. Since the displayed output using the RAM statement differs only in its form from that generated by the LINEQS statement, in this section distinctions are made between COSAN and LINEQS output only.

The unweighted least-squares and diagonally weighted least-squares estimation methods do not provide a sufficient statistical basis to provide the following output (neither displayed nor written to an OUTEST= data set):

- most of the fit indices
- approximate standard errors

- normalized or asymptotically standardized residuals
- modification indices
- information matrix
- covariance matrix of parameter estimates

The notation $\mathbf{S} = (s_{ij})$ is used for the analyzed covariance or correlation matrix, $\mathbf{C} = (c_{ij})$ for the predicted model matrix, $\mathbf{W}$ for the weight matrix (for example, $\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for GLS, $\mathbf{W} = \mathbf{C}$ for ML estimates), $\mathbf{X}$ for the vector of optimal parameter estimates, n for the number of manifest variables, t for the number of parameter estimates, and N for the sample size.

The output of PROC CALIS includes the following:

- COSAN and LINEQS: List of the matrices and their properties specified by the generalized COSAN model if you specify at least the PSHORT option.
- LINEQS: List of manifest variables that are not used in the specified model and that are automatically omitted from the analysis. Note that there is no automatic variable reduction with the COSAN or FACTOR statement. If necessary, you should use the VAR statement in these cases.
- LINEQS: List of the endogenous and exogenous variables specified by the LINEQS, STD, and COV statements if you specify at least the PSHORT option.
- COSAN: Initial values of the parameter matrices indicating positions of constants and parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- LINEQS: The set of structural equations containing the initial values and indicating constants and parameters, and output of the initial error variances and covariances. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The weight matrix $\mathbf{W}$ is displayed if GLS, WLS, or DWLS estimation is used and you specify the PWEIGHT or PALL option.
- COSAN and LINEQS: General information about the estimation problem: number of observations (N), number of manifest variables (n), amount of independent information in the data matrix (information, $n(n+1)/2$), number of terms and matrices in the specified generalized COSAN model, and number of parameters to be estimated (parameters, t). If there are no exogenous manifest variables, the difference between the amount of independent information ($n(n+1)/2$) and the number of requested estimates (t) is equal to the degrees of freedom (df). A necessary condition for a model to be identified is that the degrees of freedom are nonnegative. The output, or at least the default output, is displayed if you specify the SIMPLE option.
- COSAN and LINEQS: Mean and Std Dev (standard deviation) of each variable if you specify the SIMPLE option, as well as skewness and kurtosis if the DATA= data set is a raw data set and you specify the KURTOSIS option.

- COSAN and LINEQS: Various coefficients of multivariate kurtosis and the numbers of observations that contribute most to the normalized multivariate kurtosis if the DATA= data set is a raw data set and the KURTOSIS option, or you specify at least the PRINT option. See the section "Measures of Multivariate Kurtosis" on page 658 for more information.

- COSAN and LINEQS: Covariance or correlation matrix to be analyzed and the value of its determinant if you specify the output option PCORR or PALL. A 0 determinant indicates a singular data matrix. In this case, the generalized least-squares estimates with default weight matrix **S** and maximum likelihood estimates cannot be computed.

- LINEQS: If exogenous manifest variables in the linear structural equation model are specified, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix Φ or P. The output indicates which manifest variables are recognized as exogenous, that is, for which variables the entries in the central model matrix are set to fixed parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- COSAN and LINEQS: Vector of parameter names, initial values, and corresponding matrix locations, also indicating dependent parameter names used in your program statements that are not allocated to matrix locations and have no influence on the fit function. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- COSAN and LINEQS: The pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model is displayed if there are significant differences between constant elements in the predicted model matrix and the data matrix and you specify at least the PSHORT option. It is also displayed if you specify the PREDET option. The output indicates the differences between constant values in the predicted model matrix and the data matrix that is analyzed.

- COSAN and LINEQS: Special features of the optimization technique chosen if you specify at least the PSHORT option.

- COSAN and LINEQS: Optimization history if at least the PSHORT option is specified. For more details, see the section "Use of Optimization Techniques" on page 664.

- COSAN and LINEQS: Specific output requested by options in the NLOPTIONS statement; for example, parameter estimates, gradient, gradient of Lagrange function, constraints, Lagrange multipliers, projected gradient, Hessian, projected Hessian, Hessian of Lagrange function, Jacobian of nonlinear constraints.

- COSAN and LINEQS: The predicted model matrix and its determinant, if you specify the output option PCORR or PALL.

- COSAN and LINEQS: Residual and normalized residual matrix if you specify the RESIDUAL, or at least the PRINT option. The variance standardized or asymptotically standardized residual matrix can be displayed also. The average residual and the average off-diagonal residual are also displayed. See the section "Assessment of Fit" on page 649 for more details.

- COSAN and LINEQS: Rank order of the largest normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Bar chart of the normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Value of the fit function F. See the section "Estimation Criteria" on page 644 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Goodness-of-fit index (GFI), adjusted goodness-of-fit index (AGFI), and root mean square residual (RMR) (Jöreskog and Sörbom 1985). See the section "Assessment of Fit" on page 649 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Parsimonious goodness-of-fit index (PGFI) of Mulaik et al. (1989). See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Overall χ^2, df, and Prob>Chi**2 if the METHOD= option is not ULS or DWLS. The χ^2 measure is the optimum function value F multiplied by $(N-1)$ if a CORR or COV matrix is analyzed or multiplied by N if a UCORR or UCOV matrix is analyzed; χ^2 measures the likelihood ratio test statistic for the null hypothesis that the predicted matrix $\mathbf{C}$ has the specified model structure against the alternative that $\mathbf{C}$ is unconstrained. The notation Prob>Chi**2 means "the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed." This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the χ_0^2 value of the independence model and the corresponding degrees of freedom can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989). See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the value of the Steiger & Lind (1980) root mean squared error of approximation (RMSEA) coefficient and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHARMS=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the *probability of close fit* (Browne and Cudeck 1993). See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the Browne & Cudeck (1993) expected cross validation (ECVI) index and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHAECV=α, $0 \leq$

$\alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, Bentler's (1989) Comparative Fit Index. See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If you specify METHOD=ML or METHOD=GLS, the χ^2 value and corresponding probability adjusted by the relative kurtosis coefficient η_2, which should be a close approximation of the χ^2 value for elliptically distributed data (Browne 1982). See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: The Normal Theory Reweighted LS χ^2 Value is displayed if METHOD= ML. Instead of the function value F_{ML}, the reweighted goodness-of-fit function F_{GWLS} is used. See the section "Assessment of Fit" on page 649 for more detail.
- COSAN and LINEQS: Akaike's Information Criterion if the value of the METHOD= option is not ULS or DWLS. See the section "Assessment of Fit" on page 649. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Bozdogan's (1987) Consistent Information Criterion, CAIC. See the section "Assessment of Fit" on page 649. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Schwarz's Bayesian Criterion (SBC) if the value of the METHOD= option is not ULS or DWLS (Schwarz 1978). See the section "Assessment of Fit" on page 649. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the following fit indices based on the overall χ^2 value are displayed:

  - McDonald's (McDonald and Hartmann 1992) measure of centrality
  - Parsimonious index of James, Mulaik, and Brett (1982)
  - Z-Test of Wilson and Hilferty (1931)
  - Bentler and Bonett's (1980) nonnormed coefficient
  - Bentler and Bonett's (1980) normed coefficient
  - Bollen's (1986) normed index ρ_1
  - Bollen's (1989a) nonnormed index Δ_2

  See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Hoelter's (1983) Critical N Index is displayed (Bollen 1989b, p. 277). See the section "Assessment of Fit" on page 649 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Equations of linear dependencies among the parameters used in the model specification if the information matrix is recognized as singular at the final solution.
- COSAN: Model matrices containing the parameter estimates. Except for ULS or DWLS estimates, the approximate standard errors and *t* values are also displayed. This output is displayed if you specify the PESTIM option or at least the PSHORT option.
- LINEQS: Linear equations containing the parameter estimates. Except for ULS and DWLS estimates, the approximate standard errors and *t* values are also displayed. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Variances and covariances of the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT.
- LINEQS: Linear equations containing the standardized parameter estimates. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Table of correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Squared Multiple Correlations table, which displays the error variances of the endogenous variables. These are the diagonal elements of the predicted model matrix. Also displayed is the Total Variance and the R^2 values corresponding to all endogenous variables. See the section "Assessment of Fit" on page 649 for more detail. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: If you specify the PDETERM or the PALL option, the total determination of all equations (DETAE), the total determination of the structural equations (DETSE), and the total determination of the manifest variables (DETMV) are displayed. See the section "Assessment of Fit" on page 649 for more details. If one of the determinants in the formulas is 0, the corresponding coefficient is displayed as a missing value. If there are structural equations, PROC CALIS also displays the Stability Coefficient of Reciprocal Causation, that is, the largest eigenvalue of the $\mathbf{BB}'$ matrix, where $\mathbf{B}$ is the causal coefficient matrix of the structural equations.
- LINEQS: The matrix of estimated covariances among the latent variables if you specify the PLATCOV option, or at least the PRINT option.
- LINEQS: The matrix of estimated covariances between latent and manifest variables used in the model if you specify the PLATCOV option, or at least the PRINT option.
- LINEQS and FACTOR: The matrix $\mathbf{FSR}$ of latent variable scores regression coefficients if you specify the PLATCOV option, or at least the PRINT option.

The **FSR** matrix is a generalization of Lawley and Maxwell's (1971, p.109) factor scores regression matrix,

$$\mathbf{FSR} = \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}$$

where $\mathbf{C}_{xx}$ is the $n \times n$ predicted model matrix (predicted covariances among manifest variables) and $\mathbf{C}_{yx}$ is the $n_{lat} \times n$ matrix of the predicted covariances between latent and manifest variables. You can multiply the manifest observations by this matrix to estimate the scores of the latent variables used in your model.

- LINEQS: The matrix **TEF** of total effects if you specify the TOTEFF option, or at least the PRINT option. For the LINEQS model, the matrix of total effects is

  $$\mathbf{TEF} = (\mathbf{I} - \boldsymbol{\beta})^{-1}\gamma - (\mathbf{O} : I)$$

  (For the LISREL model, refer to Jöreskog and Sörbom 1985) The matrix of indirect effects is displayed also.
- FACTOR: The matrix of rotated factor loadings and the orthogonal transformation matrix if you specify the ROTATE= and PESTIM options, or at least the PSHORT options.
- FACTOR: Standardized (rotated) factor loadings, variance estimates of endogenous variables, R^2 values, correlations among factors, and factor scores regression matrix, if you specify the PESTIM option, or at least the PSHORT option. The determination of manifest variables is displayed only if you specify the PDETERM option.
- COSAN and LINEQS: Univariate Lagrange multiplier and Wald test indices are displayed in matrix form if you specify the MODIFICATION (or MOD) or the PALL option. Those matrix locations that correspond to constants in the model in general contain three values: the value of the Lagrange multiplier, the corresponding probability ($df = 1$), and the estimated change of the parameter value should the constant be changed to a parameter. If allowing the constant to be an estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Those matrix locations that correspond to parameter estimates in the model contain the Wald test index and the name of the parameter in the model. See the section "Modification Indices" on page 673 for more detail.
- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing equality constraints if you specify the MODIFICATION (or MOD) or the PALL option. See the section "Modification Indices" on page 673 for more detail.
- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing active boundary constraints specified by the BOUNDS statement if you specify the MODIFICATION (or MOD) or the PALL option. See the section "Modification Indices" on page 673 for more detail.

- COSAN and LINEQS: If the MODIFICATION (or MOD) or the PALL option is specified, the stepwise multivariate Wald test for constraining estimated parameters to zero constants is performed as long as the univariate probability is larger than the value specified in the PMW= option (default PMW=0.05). See the section "Modification Indices" on page 673 for more details.

ODS Table Names

PROC CALIS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 19.15. ODS Tables Created in PROC CALIS

ODS Table Name	Model[1]	Description	Option[2]
AddParms	C, F, L, R	Additional parameters in the PARAMETERS statement	PINITIAL, or default
AsymStdRes	C, F, L, R	Asymptotically standardized residual matrix	RESIDUAL=, or PRINT
AveAsymStdRes	C, F, L, R	Average absolute asymptotically standardized residuals	RESIDUAL=, or PRINT
AveNormRes	C, F, L, R	Average absolute normalized residuals	RESIDUAL=, or PRINT
AveRawRes	C, F, L, R	Average absolute raw residuals	RESIDUAL=, or PRINT
AveVarStdRes	C, F, L, R	Average absolute variance standardized residuals	RESIDUAL=, or PRINT
ContKurtosis	C, F, L, R	Contributions to kurtosis	KURTOSIS, or PRINT
ConvergenceStatus	C, F, L, R	Convergence status	PSHORT
CorrExog	L	Correlations among exogenous variables	PESTIM, or PSHORT
CorrParm	C, F, L, R	Correlations among parameter estimates	PCOVES, and default
CovMat	C, F, L, R	Assorted cov matrices	PCOVES, and default
DependParms	C, F, L, R	Dependent parameters (if specified by program statements)	PRIVEC, and default
Determination	L, F, R	Coefficients of determination	PDETERM, and default
DistAsymStdRes	C, F, L, R	Distribution of asymptotically standardized residuals	RESIDUAL=, or PRINT
DistNormRes	C, F, L, R	Distribution of normalized residuals	RESIDUAL=, or PRINT
DistVarStdRes	C, F, L, R	Distribution of variance standardized residuals	RESIDUAL=, or PRINT
EndogenousVar	L	Endogenous variables	PESTIM, or PSHORT
EstCovExog	L	Estimated covariances among exogenous variables	PESTIM, or PSHORT
Estimates	C, F, L, R	Vector of estimates	PRIVEC
EstLatentEq	L	Estimated latent variable equations	PESTIM, or PSHORT
EstManifestEq	L	Estimated manifest variable equations	PESTIM, or PSHORT
EstParms	C, F	Estimated parameter matrix	PESTIM, or PSHORT
EstVarExog	L	Estimated variances of exogenous variables	PESTIM, or PSHORT

Table 19.15. (continued)

ODS Table Name	Model[1]	Description	Option[2]
ExogenousVar	L	List of exogenous variables	PESTIM, or PSHORT
FactCorrExog	F	Correlations among factors	PESTIM, or PSHORT
FactScoreCoef	F	Factor score regression coefficients	PESTIM, or PSHORT
Fit	C, F, L, R	Fit statistics	PSUMMARY
GenModInfo	C, F, L, R	General modeling information	PSIMPLE, or default
Gradient	C, F, L, R	First partial derivatives (Gradient)	PRIVEC, and default
InCorr	C, F, L, R	Input correlation matrix	PCORR, or PALL
InCorrDet	C, F, L, R	Determinant of the input correlation matrix	PCORR, or PALL
InCov	C, F, L, R	Input covariance matrix	PCORR, or PALL
InCovDet	C, F, L, R	Determinant of the input covariance matrix	PCORR, or PALL
InCovExog	L	Input covariances among exogenous variables	PESTIM, or PSHORT
Indirect Effects	L, R	Indirect effects	TOTEFF, or PRINT
Information	C, F, L, R	Information matrix	PCOVES, and default
InitEstimates	C, F, L, R	Initial vector of parameter estimates	PINITIAL, or default
InitParms	C, F	Initial matrix of parameter estimates	PINITIAL, or default
InitParms	L, R	Initial matrix of parameter estimates	PRIMAT, and default
InitRAMEstimates	R	Initial RAM estimates	PESTIM, or PSHORT
InLatentEq	L	Input latent variable equations	PESTIM, or PSHORT
InManifestEq	L	Input manifest variable equations	PESTIM, or PSHORT
InSymmetric	C, F, L, R	Input symmetric matrix (SYMATRIX data type)	PCORR, or PALL
InVarExog	L	Input variances of exogenous variables	PESTIM, or PSHORT
IterHist	C, F, L, R	Iteration history	PSHORT
IterStart	C, F, L, R	Iteration start	PSHORT
IterStop	C, F, L, R	Iteration stop	PSHORT
Jacobian	C, F, L, R	Jacobi column pattern	PJACPAT
Kurtosis	C, F, L, R	Kurtosis, with raw data input	KURTOSIS, or PRINT
LagrangeBoundary	C, F, L, R	Lagrange, releasing active boundary constraints	MODIFICATION[3], or PALL
LagrangeEquality	C, F, L, R	Lagrange, releasing equality constraints	MODIFICATION, or PALL
LatentScoreCoef	L, R	Latent variable regression score coefficients	PLATCOV, or PRINT
ModelStatement	C, F, L, R	Model summary	PSHORT
ModIndices	C, F, L, R	Lagrange multiplier and Wald test statistics	MODIFICATION, or PALL
NormRes	C, F, L, R	Normalized residual matrix	RESIDUAL=, or PRINT
PredetElements	C, F, L, R	Predetermined elements	PREDET, or PALL
PredModel	C, F, L, R	Predicted model matrix	PCORR, or PALL
PredModelDet	C, F, L, R	Predicted model determinant	PCORR, or PALL
PredMomentLatent	L, R	Predicted latent variable moments	PLATCOV, or PRINT
PredMomentManLat	L, R	Predicted manifest and latent variable moments	PLATCOV, or PRINT

Table 19.15. (continued)

ODS Table Name	Model[1]	Description	Option[2]
ProblemDescription	C, F, L, R	Problem description	PSHORT
RAMCorrExog	R	Correlations among exogenous variables	PESTIM, or PSHORT
RAMEstimates	R	RAM Final Estimates	PESTIM, or PSHORT
RAMStdEstimates	R	Standardized estimates	PESTIM, or PSHORT
RankAsymStdRes	C, F, L, R	Ranking of the largest asymptotically standardized residuals	RESIDUAL=, or PRINT
RankLagrange	C, F, L, R	Ranking of the largest Lagrange indices	RESIDUAL=, or PRINT
RankNormRes	C, F, L, R	Ranking of the largest normalized residuals	RESIDUAL=, or PRINT
RankRawRes	C, F, L, R	Ranking of the largest raw residuals	RESIDUAL=, or PRINT
RankVarStdRes	C, F, L, R	Ranking of the largest variance standardized residuals	RESIDUAL=, or PRINT
RawRes	C, F, L, R	Raw residual matrix	RESIDUAL=, or PRINT
RotatedLoadings	F	Rotated loadings, with ROTATE= option in FACTOR statement	PESTIM, or PSHORT
Rotation	F	Rotation Matrix, with ROTATE= option in FACTOR statement	PESTIM, or PSHORT
SetCovExog	L, R	Set covariance parameters for manifest exogenous variables	PINITIAL, or default
SimpleStatistics	C, F, L, R	Simple statistics, with raw data input	SIMPLE, or default
SqMultCorr	F, L, R	Squared multiple correlations	PESTIM, or PSHORT
Stability	L, R	Stability of reciprocal causation	PDETERM, and default
StdErrs	C, F, L, R	Vector of standard errors	PRIVEC, and default
StdLatentEq	L	Standardized latent variable equations	PESTIM, or PSHORT
StdLoadings	F	Standardized factor loadings	PESTIM, or PSHORT
StdManifestEq	L	Standardized manifest variable equations	PESTIM, or PSHORT
StructEq	L, R	Variables in the structural equations	PDETERM, and default
SumSqDif	C, F, L, R	Sum of squared differences of predetermined elements	PREDET, or PALL
TotalEffects	L, R	Total effects	TOTEFF, or PRINT
tValues	C, F, L, R	Vector of t values	PRIVEC, and default
VarSelection	L, R	Manifest variables, if not all are used, selected for Modeling	default
VarStdRes	C, F, L, R	Variance standardized residual matrix	RESIDUAL=, or PRINT
WaldTest	C, F, L, R	Wald test	MODIFICATION, or PALL
Weights	C, F, L, R	Weight matrix	PWEIGHT[4], or PALL
WeightsDet	C, F, L, R	Determinant of the weight matrix	PWEIGHT[4], or PALL

1. Most CALIS output tables are specific to the model statement used. Keys: C: COSAN model, F: FACTOR model, L: LINEQS model, R: RAM model.

2. The printing options PALL, PRINT, “default”, PSHORT, and PSUMM form hierarchical levels of output control, with PALL including all the output enabled by the options at the lower levels, and so on. The “default” option means that NOPRINT is not specified. Therefore, in the table, for example,

if PSHORT is the printing option for an output, PALL, PRINT, or “default” will also enable the same output printing.

3. The printing of LagrangeBoundary is effective only if you have set some boundary constraints for parameters.

4. The printing of Weights or WeightsDet is effective only if your estimation method uses the weight matrix (e.g., WLS or LSWLS).

Examples

Example 19.1. Path Analysis: Stability of Alienation

The following covariance matrix from Wheaton, Muthen, Alwin, and Summers (1977) has served to illustrate the performance of several implementations for the analysis of structural equation models. Two different models have been analyzed by an early implementation of LISREL and are mentioned in Jöreskog (1978). You also can find a more detailed discussion of these models in the LISREL VI manual (Jöreskog and Sörbom 1985). A slightly modified model for this covariance matrix is included in the EQS 2.0 manual (Bentler 1985, p. 28). The path diagram of this model is displayed in Figure 19.1. The same model is reanalyzed here by PROC CALIS. However, for the analysis with the EQS implementation, the last variable (V6) is rescaled by a factor of 0.1 to make the matrix less ill-conditioned. Since the Levenberg-Marquardt or Newton-Raphson optimization techniques are used with PROC CALIS, rescaling the data matrix is not necessary and, therefore, is not done here. The results reported here reflect the estimates based on the original covariance matrix.

```
data Wheaton(TYPE=COV);
title "Stability of Alienation";
title2 "Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)";
   _type_ = ’cov’; input _name_ $ v1-v6;
   label v1=’Anomia (1967)’ v2=’Anomia (1971)’ v3=’Education’
         v4=’Powerlessness (1967)’ v5=’Powerlessness (1971)’
         v6=’Occupational Status Index’;
   datalines;
v1   11.834     .        .        .       .        .
v2    6.947    9.364     .        .       .        .
v3    6.819    5.091   12.532     .       .        .
v4    4.783    5.028    7.495    9.986    .        .
v5   -3.839   -3.889   -3.841   -3.625   9.610     .
v6  -21.899  -18.831  -21.748  -18.775  35.522  450.288
;
```

```
proc calis cov data=Wheaton tech=nr edf=931 pall;
   Lineqs
      V1 =         F1                  + E1,
      V2 =    .833 F1                  + E2,
      V3 =         F2                  + E3,
      V4 =    .833 F2                  + E4,
      V5 =         F3                  + E5,
      V6 = Lamb (.5) F3                + E6,
      F1 = Gam1(-.5) F3                + D1,
      F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;
   Std
      E1-E6 = The1-The2 The1-The4 (6 * 3.),
      D1-D2 = Psi1-Psi2 (2 * 4.),
      F3    = Phi (6.) ;
   Cov
      E1 E3 = The5 (.2),
      E4 E2 = The5 (.2);
run;
```

The COV option in the PROC CALIS statement requests the analysis of the covariance matrix. Without the COV option, the correlation matrix would be computed and analyzed. Since no METHOD= option has been used, maximum likelihood estimates are computed by default. The TECH=NR option requests the Newton-Raphson optimization method. The PALL option produces the almost complete set of displayed output, as displayed in Output 19.1.1 through Output 19.1.11. Note that, when you specify the PALL option, you can produce large amounts of output. The PALL option is used in this example to show how you can get a wide spectrum of useful information from PROC CALIS.

Output 19.1.1 displays the model specification in matrix terms, followed by the lists of endogenous and exogenous variables. Equations and initial parameter estimates are also displayed. You can use this information to ensure that the desired model is the model being analyzed.

Output 19.1.1. Model Specification

```
                           Stability of Alienation
            Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                             The CALIS Procedure
          Covariance Structure Analysis: Pattern and Initial Values

                            LINEQS Model Statement


                       Matrix      Rows    Columns    ------Matrix Type-------

Term 1            1    _SEL_          6         17    SELECTION
                  2    _BETA_        17         17    EQSBETA        IMINUSINV
                  3    _GAMMA_       17          9    EQSGAMMA
                  4    _PHI_          9          9    SYMMETRIC


                          The 8 Endogenous Variables

    Manifest        v1  v2  v3  v4  v5  v6
    Latent          F1  F2


                          The 9 Exogenous Variables

    Manifest
    Latent          F3
    Error           E1  E2  E3  E4  E5  E6  D1  D2
```

Output 19.1.1. (continued)

```
                           Stability of Alienation
            Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

          Covariance Structure Analysis: Pattern and Initial Values

                    v1      =   1.0000 F1   +  1.0000 E1
                    v2      =   0.8330 F1   +  1.0000 E2
                    v3      =   1.0000 F2   +  1.0000 E3
                    v4      =   0.8330 F2   +  1.0000 E4
                    v5      =   1.0000 F3   +  1.0000 E5
                    v6      =   0.5000*F3   +  1.0000 E6
                                       Lamb



                           Stability of Alienation
            Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

          Covariance Structure Analysis: Pattern and Initial Values

            F1      =  -0.5000*F3   +  1.0000 D1
                               Gam1
            F2      =   0.5000*F1   + -0.5000*F3   +  1.0000 D2
                               Beta           Gam2
```

Output 19.1.1. (continued)

```
                            Stability of Alienation
                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

              Covariance Structure Analysis: Pattern and Initial Values

                          Variances of Exogenous Variables

                          Variable Parameter      Estimate

                          F3       Phi             6.00000
                          E1       The1            3.00000
                          E2       The2            3.00000
                          E3       The1            3.00000
                          E4       The2            3.00000
                          E5       The3            3.00000
                          E6       The4            3.00000
                          D1       Psi1            4.00000
                          D2       Psi2            4.00000


                       Covariances Among Exogenous Variables

                         Var1 Var2 Parameter      Estimate

                         E1   E3   The5            0.20000
                         E2   E4   The5            0.20000
```

General modeling information and simple descriptive statistics are displayed in Output 19.1.2. Because the input data set contains only the covariance matrix, the means of the manifest variables are assumed to be zero. Note that this has no impact on the estimation, unless a mean structure model is being analyzed. The twelve parameter estimates in the model and their respective locations in the parameter matrices are also displayed. Each of the parameters, The1, The2, and The5, is specified for two elements in the parameter matrix _PHI_.

Output 19.1.2. Modeling Information, Simple Statistics, and Parameter Vector

```
                            Stability of Alienation
                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

             Covariance Structure Analysis: Maximum Likelihood Estimation

                Observations         932    Model Terms             1
                Variables              6    Model Matrices          4
                Informations          21    Parameters             12


                        Variable                     Mean       Std Dev

             v1    Anomia (1967)                        0       3.44006
             v2    Anomia (1971)                        0       3.06007
             v3    Education                            0       3.54006
             v4    Powerlessness (1967)                 0       3.16006
             v5    Powerlessness (1971)                 0       3.10000
             v6    Occupational Status Index            0      21.21999
```

Output 19.1.2. (continued)

```
                                     Stability of Alienation
                       Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                      Covariance Structure Analysis: Maximum Likelihood Estimation

                                           Covariances

                                     v1             v2             v3             v4             v5             v6

v1   Anomia (1967)          11.83400000     6.94700000     6.81900000     4.78300000    -3.83900000    -21.8990000
v2   Anomia (1971)           6.94700000     9.36400000     5.09100000     5.02800000    -3.88900000    -18.8310000
v3   Education               6.81900000     5.09100000    12.53200000     7.49500000    -3.84100000    -21.7480000
v4   Powerlessness (1967)    4.78300000     5.02800000     7.49500000     9.98600000    -3.62500000    -18.7750000
v5   Powerlessness (1971)   -3.83900000    -3.88900000    -3.84100000    -3.62500000     9.61000000     35.5220000
v6   Occupational Status Index   -21.89900000   -18.83100000   -21.74800000   -18.77500000    35.52200000    450.2880000

                                 Determinant       6080570    Ln      15.620609
```

Output 19.1.2. (continued)

```
                                     Stability of Alienation
                       Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                      Covariance Structure Analysis: Maximum Likelihood Estimation

                                   Vector of Initial Estimates

                          Parameter      Estimate    Type

                      1   Beta            0.50000    Matrix Entry: _BETA_[8:7]
                      2   Lamb            0.50000    Matrix Entry: _GAMMA_[6:1]
                      3   Gam1           -0.50000    Matrix Entry: _GAMMA_[7:1]
                      4   Gam2           -0.50000    Matrix Entry: _GAMMA_[8:1]
                      5   Phi             6.00000    Matrix Entry: _PHI_[1:1]
                      6   The1            3.00000    Matrix Entry: _PHI_[2:2]  _PHI_[4:4]
                      7   The2            3.00000    Matrix Entry: _PHI_[3:3]  _PHI_[5:5]
                      8   The5            0.20000    Matrix Entry: _PHI_[4:2]  _PHI_[5:3]
                      9   The3            3.00000    Matrix Entry: _PHI_[6:6]
                     10   The4            3.00000    Matrix Entry: _PHI_[7:7]
                     11   Psi1            4.00000    Matrix Entry: _PHI_[8:8]
                     12   Psi2            4.00000    Matrix Entry: _PHI_[9:9]
```

PROC CALIS examines whether each element in the moment matrix is modeled by the parameters defined in the model. If an element is not structured by the model parameters, it is predetermined by its observed value. This occurs, for example, when there are exogenous manifest variables in the model. If present, the predetermined values of the elements will be displayed. In the current example, the '.' displayed for all elements in the predicted moment matrix (Output 19.1.3) indicates that there are no predetermined elements in the model.

Output 19.1.3. Predetermined Elements

```
                                          Stability of Alienation
                               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                             Covariance Structure Analysis: Maximum Likelihood Estimation

                                Predetermined Elements of the Predicted Moment Matrix

                                         v1             v2             v3             v4             v5             v6

v1   Anomia (1967)                        .              .              .              .              .              .
v2   Anomia (1971)                        .              .              .              .              .              .
v3   Education                            .              .              .              .              .              .
v4   Powerlessness (1967)                 .              .              .              .              .              .
v5   Powerlessness (1971)                 .              .              .              .              .              .
v6   Occupational Status Index            .              .              .              .              .              .

                                      Sum of Squared Differences             0
```

Output 19.1.4 displays the optimization information. You can check this table to determine whether the convergence criterion is satisfied. PROC CALIS displays an error message when problematic solutions are encountered.

Output 19.1.4. Optimization

```
                                          Stability of Alienation
                               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                             Covariance Structure Analysis: Maximum Likelihood Estimation

                                      Parameter Estimates                   12
                                      Functions (Observations)              21

                                                 Optimization Start

       Active Constraints                                 0  Objective Function                      119.33282242
       Max Abs Gradient Element                74.016932345


                                                                                                            Ratio
                                                                                                          Between
                                                                                                           Actual
                                                                       Objective      Max Abs                 and
                         Function        Active        Objective        Function     Gradient           Predicted
      Iter    Restarts      Calls   Constraints         Function          Change      Element     Ridge      Change

         1           0          2             0          0.82689           118.5       1.3507         0      0.0154
         2           0          3             0          0.09859          0.7283       0.2330         0       0.716
         3           0          4             0          0.01581          0.0828      0.00684         0       1.285
         4           0          5             0          0.01449         0.00132     0.000286         0       1.042
         5           0          6             0          0.01448        9.936E-7     0.000045         0       1.053
         6           0          7             0          0.01448        4.227E-9     1.685E-6         0       1.056

                                                Optimization Results

       Iterations                                         6  Function Calls                                     8
       Jacobian Calls                                     7  Active Constraints                                 0
       Objective Function                      0.0144844811  Max Abs Gradient Element                1.6847829E-6
       Ridge                                              0  Actual Over Pred Change                 1.0563204982

            ABSGCONV convergence criterion satisfied.
```

The predicted model matrix is displayed next, followed by a list of model test statistics or fit indices (Output 19.1.5). Depending on your modeling philosophy, some indices may be preferred to others. In this example, all indices and test statistics point to a good fit of the model.

Output 19.1.5. Predicted Model Matrix and Fit Statistics

```
                                           Stability of Alienation
                                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                              Covariance Structure Analysis: Maximum Likelihood Estimation

                                                Predicted Model Matrix

                                         v1             v2             v3             v4             v5             v6

v1   Anomia (1967)              11.90390632     6.91059048     6.83016211     4.93499582    -4.16791157    -22.3768816
v2   Anomia (1971)               6.91059048     9.35145064     4.93499582     5.01664889    -3.47187034    -18.6399424
v3   Education                   6.83016211     4.93499582    12.61574998     7.50355625    -4.06565606    -21.8278873
v4   Powerlessness (1967)        4.93499582     5.01664889     7.50355625     9.84539112    -3.38669150    -18.1826302
v5   Powerlessness (1971)       -4.16791157    -3.47187034    -4.06565606    -3.38669150     9.61000000     35.5219999
v6   Occupational Status Index -22.37688158   -18.63994236   -21.82788734   -18.18263015    35.52199986    450.2879993

                                    Determinant       6169285     Ln      15.635094
```

Output 19.1.5. (continued)

```
                                           Stability of Alienation
                                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                              Covariance Structure Analysis: Maximum Likelihood Estimation

                              Fit Function                                          0.0145
                              Goodness of Fit Index (GFI)                           0.9953
                              GFI Adjusted for Degrees of Freedom (AGFI)            0.9890
                              Root Mean Square Residual (RMR)                       0.2281
                              Parsimonious GFI (Mulaik, 1989)                       0.5972
                              Chi-Square                                           13.4851
                              Chi-Square DF                                              9
                              Pr > Chi-Square                                       0.1419
                              Independence Model Chi-Square                         2131.4
                              Independence Model Chi-Square DF                          15
                              RMSEA Estimate                                        0.0231
                              RMSEA 90% Lower Confidence Limit                           .
                              RMSEA 90% Upper Confidence Limit                      0.0470
                              ECVI Estimate                                         0.0405
                              ECVI 90% Lower Confidence Limit                            .
                              ECVI 90% Upper Confidence Limit                       0.0556
                              Probability of Close Fit                              0.9705
                              Bentler's Comparative Fit Index                       0.9979
                              Normal Theory Reweighted LS Chi-Square               13.2804
                              Akaike's Information Criterion                       -4.5149
                              Bozdogan's (1987) CAIC                              -57.0509
                              Schwarz's Bayesian Criterion                        -48.0509
                              McDonald's (1989) Centrality                          0.9976
                              Bentler & Bonett's (1980) Non-normed Index            0.9965
                              Bentler & Bonett's (1980) NFI                         0.9937
                              James, Mulaik, & Brett (1982) Parsimonious NFI        0.5962
                              Z-Test of Wilson & Hilferty (1931)                    1.0754
                              Bollen (1986) Normed Index Rho1                       0.9895
                              Bollen (1988) Non-normed Index Delta2                 0.9979
                              Hoelter's (1983) Critical N                             1170
```

PROC CALIS can perform a detailed residual analysis. Large residuals may indicate misspecification of the model. In Output 19.1.6 for example, note the table for the 10 largest asymptotically standardized residuals. As the table shows, the specified model performs the poorest concerning the variable V5 and its covariance with V2, V1, and V3. This may be the result of a misspecification of the model equation for V5. However, because the model fit is quite good, such a possible misspecification may have no practical significance and is not a serious concern in the analysis.

Output 19.1.6. Residual Analysis

```
                                               Stability of Alienation
                                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                             Covariance Structure Analysis: Maximum Likelihood Estimation

                                                 Raw Residual Matrix

                                          v1            v2            v3            v4            v5            v6

    v1  Anomia (1967)             -.0699063150  0.0364095216  -.0111621061  -.1519958205  0.3289115712  0.4778815840
    v2  Anomia (1971)             0.0364095216  0.0125493646  0.1560041795  0.0113511059  -.4171296612  -.1910576405
    v3  Education                 -.0111621061  0.1560041795  -.0837499788  -.0085562504  0.2246560598  0.0798873380
    v4  Powerlessness (1967)      -.1519958205  0.0113511059  -.0085562504  0.1406088766  -.2383085022  -.5923698474
    v5  Powerlessness (1971)      0.3289115712  -.4171296612  0.2246560598  -.2383085022  0.0000000000  0.0000000000
    v6  Occupational Status Index 0.4778815840  -.1910576405  0.0798873380  -.5923698474  0.0000000000  0.0000000000

                              Average Absolute Residual                     0.153928
                              Average Off-diagonal Absolute Residual        0.195045
```

Output 19.1.6. (continued)

```
                                               Stability of Alienation
                                Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                             Covariance Structure Analysis: Maximum Likelihood Estimation

                                        Rank Order of the 10 Largest Raw Residuals

                                          Row         Column        Residual

                                          v6          v4            -0.59237
                                          v6          v1             0.47788
                                          v5          v2            -0.41713
                                          v5          v1             0.32891
                                          v5          v4            -0.23831
                                          v5          v3             0.22466
                                          v6          v2            -0.19106
                                          v3          v2             0.15600
                                          v4          v1            -0.15200
                                          v4          v4             0.14061
```

Output 19.1.6. (continued)

```
                                            Stability of Alienation
                               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                            Covariance Structure Analysis: Maximum Likelihood Estimation

                                    Asymptotically Standardized Residual Matrix

                                           v1             v2             v3             v4             v5             v6

   v1  Anomia (1967)            -0.308548787    0.526654452   -0.056188826   -0.865070455    2.553366366    0.464866661
   v2  Anomia (1971)             0.526654452    0.054363484    0.876120855    0.057354415   -2.763708659   -0.170127806
   v3  Education                -0.056188826    0.876120855   -0.354347092   -0.121874301    1.697931678    0.070202664
   v4  Powerlessness (1967)     -0.865070455    0.057354415   -0.121874301    0.584930625   -1.557412695   -0.495982427
   v5  Powerlessness (1971)      2.553366366   -2.763708659    1.697931678   -1.557412695    0.000000000    0.000000000
   v6  Occupational Status Index 0.464866661   -0.170127806    0.070202664   -0.495982427    0.000000000    0.000000000

                             Average Standardized Residual                  0.646622
                             Average Off-diagonal Standardized Residual     0.818457
```

Output 19.1.6. (continued)

```
                                            Stability of Alienation
                               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                            Covariance Structure Analysis: Maximum Likelihood Estimation

                        Rank Order of the 10 Largest Asymptotically Standardized Residuals

                                          Row         Column        Residual

                                          v5          v2            -2.76371
                                          v5          v1             2.55337
                                          v5          v3             1.69793
                                          v5          v4            -1.55741
                                          v3          v2             0.87612
                                          v4          v1            -0.86507
                                          v4          v4             0.58493
                                          v2          v1             0.52665
                                          v6          v4            -0.49598
                                          v6          v1             0.46487
```

Output 19.1.6. (continued)

```
                                 Stability of Alienation
                   Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                Covariance Structure Analysis: Maximum Likelihood Estimation

                    Distribution of Asymptotically Standardized Residuals

                                Each * Represents 1 Residuals

                 ----------Range---------    Freq    Percent

                   -3.00000      -2.75000       1       4.76    *
                   -2.75000      -2.50000       0       0.00
                   -2.50000      -2.25000       0       0.00
                   -2.25000      -2.00000       0       0.00
                   -2.00000      -1.75000       0       0.00
                   -1.75000      -1.50000       1       4.76    *
                   -1.50000      -1.25000       0       0.00
                   -1.25000      -1.00000       0       0.00
                   -1.00000      -0.75000       1       4.76    *
                   -0.75000      -0.50000       0       0.00
                   -0.50000      -0.25000       3      14.29    ***
                   -0.25000             0       3      14.29    ***
                          0       0.25000       6      28.57    ******
                    0.25000       0.50000       1       4.76    *
                    0.50000       0.75000       2       9.52    **
                    0.75000       1.00000       1       4.76    *
                    1.00000       1.25000       0       0.00
                    1.25000       1.50000       0       0.00
                    1.50000       1.75000       1       4.76    *
                    1.75000       2.00000       0       0.00
                    2.00000       2.25000       0       0.00
                    2.25000       2.50000       0       0.00
                    2.50000       2.75000       1       4.76    *
```

Output 19.1.7 displays the equations and parameter estimates. Each parameter estimate is displayed with its standard error and the corresponding *t* ratio. As a general rule, a *t* ratio larger than 2 represents a statistically significant departure from 0. From these results, it is observed that both F1 (Alienation 1967) and F2 (Alienation 1971) are regressed negatively on F3 (Socioeconomic Status), and F1 has a positive effect on F2. The estimates and significance tests for the variance and covariance of the exogenous variables are also displayed.

Output 19.1.7. Equations and Parameter Estimates

```
                           Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

                  v1      =   1.0000 F1       +  1.0000 E1
                  v2      =   0.8330 F1       +  1.0000 E2
                  v3      =   1.0000 F2       +  1.0000 E3
                  v4      =   0.8330 F2       +  1.0000 E4
                  v5      =   1.0000 F3       +  1.0000 E5
                  v6      =   5.3688*F3       +  1.0000 E6
                  Std Err     0.4337 Lamb
                  t Value    12.3788



                           Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

        F1      =  -0.6299*F3       +  1.0000 D1
        Std Err     0.0563 Gam1
        t Value   -11.1809
        F2      =   0.5931*F1       + -0.2409*F3       +  1.0000 D2
        Std Err     0.0468 Beta        0.0549 Gam2
        t Value    12.6788            -4.3885
```

Output 19.1.7. (continued)

```
                           Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

                         Variances of Exogenous Variables

                                                   Standard
            Variable Parameter      Estimate          Error    t Value

            F3       Phi             6.61632        0.63914      10.35
            E1       The1            3.60788        0.20092      17.96
            E2       The2            3.59493        0.16448      21.86
            E3       The1            3.60788        0.20092      17.96
            E4       The2            3.59493        0.16448      21.86
            E5       The3            2.99368        0.49861       6.00
            E6       The4          259.57580       18.31150      14.18
            D1       Psi1            5.67047        0.42301      13.41
            D2       Psi2            4.51480        0.33532      13.46


                       Covariances Among Exogenous Variables

                                                    Standard
            Var1 Var2 Parameter      Estimate          Error    t Value

            E1   E3   The5            0.90580        0.12167       7.44
            E2   E4   The5            0.90580        0.12167       7.44
```

The measurement scale of variables is often arbitrary. Therefore, it can be useful to look at the standardized equations produced by PROC CALIS. Output 19.1.8 displays the standardized equations and predicted moments. From the standardized structural equations for F1 and F2, you can conclude that SES (F3) has a larger impact on earlier Alienation (F1) than on later Alienation (F3).

The squared multiple correlation for each equation, the correlation among the exogenous variables, and the covariance matrices among the latent variables and between the observed and the latent variables help to describe the relationships among all variables.

Output 19.1.8. Standardized Equations and Predicted Moments

```
                          Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

                  v1      =   0.8348 F1   +  0.5505 E1
                  v2      =   0.7846 F1   +  0.6200 E2
                  v3      =   0.8450 F2   +  0.5348 E3
                  v4      =   0.7968 F2   +  0.6043 E4
                  v5      =   0.8297 F3   +  0.5581 E5
                  v6      =   0.6508*F3   +  0.7593 E6
                                   Lamb



                          Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

          F1      =  -0.5626*F3   +  0.8268 D1
                               Gam1
          F2      =   0.5692*F1   + -0.2064*F3   +  0.7080 D2
                               Beta           Gam2
```

Output 19.1.8. (continued)

```
                          Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

                        Squared Multiple Correlations

                                    Error        Total
                 Variable        Variance     Variance    R-Squared

             1    v1              3.60788     11.90391      0.6969
             2    v2              3.59493      9.35145      0.6156
             3    v3              3.60788     12.61575      0.7140
             4    v4              3.59493      9.84539      0.6349
             5    v5              2.99368      9.61000      0.6885
             6    v6            259.57580    450.28800      0.4235
             7    F1              5.67047      8.29603      0.3165
             8    F2              4.51480      9.00787      0.4988


                  Correlations Among Exogenous Variables

                    Var1 Var2 Parameter      Estimate

                    E1   E3   The5            0.25106
                    E2   E4   The5            0.25197
```

Output 19.1.8. (continued)

```
                                   Stability of Alienation
                          Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                       Covariance Structure Analysis: Maximum Likelihood Estimation

                                  Predicted Moments of Latent Variables

                                       F1                F2                F3

                        F1      8.296026985       5.924364730      -4.167911571
                        F2      5.924364730       9.007870649      -4.065656060
                        F3     -4.167911571      -4.065656060       6.616317547

                        Predicted Moments between Manifest and Latent Variables

                                       F1                F2                F3

                        v1       8.29602698        5.92436473       -4.16791157
                        v2       6.91059048        4.93499582       -3.47187034
                        v3       5.92436473        9.00787065       -4.06565606
                        v4       4.93499582        7.50355625       -3.38669150
                        v5      -4.16791157       -4.06565606        6.61631755
                        v6     -22.37688158      -21.82788734       35.52199986
```

Output 19.1.9 displays the latent variable score regression coefficients that produce the latent variable scores. Each latent variable is expressed as a linear combination of the observed variables. See Chapter 64, “The SCORE Procedure,” for more information on the creation of latent variable scores. Note that the total effects and indirect effects of the exogenous variables are also displayed.

Output 19.1.9. Latent Variable Score Regression, Direct and Indirect Effects

```
                              Stability of Alienation
                 Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

              Covariance Structure Analysis: Maximum Likelihood Estimation

                      Latent Variable Score Regression Coefficients

                                              F1                F2                F3

      v1     Anomia (1967)              0.4131113567      0.0482681051      -.0521264408
      v2     Anomia (1971)              0.3454029627      0.0400143300      -.0435560637
      v3     Education                  0.0526632293      0.4306175653      -.0399927539
      v4     Powerlessness (1967)       0.0437036855      0.3600452776      -.0334000265
      v5     Powerlessness (1971)       -.0749215200      -.0639697183      0.5057060770
      v6     Occupational Status Index  -.0046390513      -.0039609288      0.0313127184

                                     Total Effects

                               F3                F1                F2

                v1      -0.629944307       1.000000000       0.000000000
                v2      -0.524743608       0.833000000       0.000000000
                v3      -0.614489258       0.593112208       1.000000000
                v4      -0.511869552       0.494062469       0.833000000
                v5       1.000000000       0.000000000       0.000000000
                v6       5.368847492       0.000000000       0.000000000
                F1      -0.629944307       0.000000000       0.000000000
                F2      -0.614489258       0.593112208       0.000000000

                                    Indirect Effects

                               F3                F1                F2

                v1      -.6299443069      0.0000000000                 0
                v2      -.5247436076      0.0000000000                 0
                v3      -.6144892580      0.5931122083                 0
                v4      -.5118695519      0.4940624695                 0
                v5      0.0000000000      0.0000000000                 0
                v6      0.0000000000      0.0000000000                 0
                F1      0.0000000000      0.0000000000                 0
                F2      -.3736276589      0.0000000000                 0
```

PROC CALIS can display Lagrange multiplier and Wald statistics for model modifications. Modification indices are displayed for each parameter matrix. Only the Lagrange multiplier statistics have significance levels and approximate changes of values displayed. The significance level of the Wald statistic for a given parameter is the same as that shown in the equation output. An insignificant *p*-value for a Wald statistic means that the corresponding parameter can be dropped from the model without significantly worsening the fit of the model.

A significant *p*-value for a Lagrange multiplier test indicates that the model would achieve a better fit if the corresponding parameter is free. To aid in determining significant results, PROC CALIS displays the rank order of the ten largest Lagrange multiplier statistics. For example, [E5:E2] in the _PHI_ matrix is associated with the largest Lagrange multiplier statistic; the associated *p*-value is 0.0067. This means that adding a parameter for the covariance between E5 and E2 will lead to a significantly better fit of the model. However, adding parameters indiscriminately can result in specification errors. An over-fitted model may not perform well with future samples. As always, the decision to add parameters should be accompanied with consideration and knowledge of the application area.

Output 19.1.10. Lagrange Multiplier and Wald Tests

```
                                                Stability of Alienation
                                 Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                              Covariance Structure Analysis: Maximum Likelihood Estimation

                                Lagrange Multiplier and Wald Test Indices _PHI_   [9:9]
                                                   Symmetric Matrix
                                       Univariate Tests for Constant Constraints
                       Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

             F3           E1           E2           E3           E4           E5           E6           D1           D2

F3     107.1619       3.3903       3.3901       0.5752       0.5753        .            .            .            .
          .           0.0656       0.0656       0.4482       0.4482        .            .            .            .
          .           0.5079      -0.4231       0.2090      -0.1741        .            .            .            .
          [Phi]                                                             Sing         Sing         Sing         Sing

E1       3.3903     322.4501       0.1529      55.4237       1.2037       5.8025       0.7398       0.4840       0.0000
         0.0656        .           0.6958        .           0.2726       0.0160       0.3897       0.4866       0.9961
         0.5079        .           0.0900        .          -0.3262       0.5193      -1.2587       0.2276       0.0014
                      [The1]                    [The5]

E2       3.3901       0.1529     477.6768       0.5946      55.4237       7.3649       1.4168       0.4840       0.0000
         0.0656       0.6958        .           0.4406        .           0.0067       0.2339       0.4866       0.9961
        -0.4231       0.0900        .           0.2328        .          -0.5060       1.5431      -0.1896      -0.0011
                                   [The2]                    [The5]

E3       0.5752      55.4237       0.5946     322.4501       0.1528       1.5982       0.0991       1.1825       0.5942
         0.4482        .           0.4406        .           0.6958       0.2062       0.7529       0.2768       0.4408
         0.2090        .           0.2328        .          -0.0900       0.2709      -0.4579       0.2984      -0.2806
                      [The5]                    [The1]

E4       0.5753       1.2037      55.4237       0.1528     477.6768       1.2044       0.0029       1.1825       0.5942
         0.4482       0.2726        .           0.6958        .           0.2724       0.9568       0.2768       0.4408
        -0.1741      -0.3262        .          -0.0900        .          -0.2037       0.0700      -0.2486       0.2338
                                   [The5]                    [The2]

E5        .           5.8025       7.3649       1.5982       1.2044      36.0486        .           0.1033       0.1035
          .           0.0160       0.0067       0.2062       0.2724        .            .           0.7479       0.7477
          .           0.5193      -0.5060       0.2709      -0.2037        .            .          -0.2776       0.1062
           Sing                                                           [The3]         Sing

E6        .           0.7398       1.4168       0.0991       0.0029        .          200.9466       0.1034       0.1035
          .           0.3897       0.2339       0.7529       0.9568        .            .           0.7478       0.7477
          .          -1.2587       1.5431      -0.4579       0.0700        .            .           1.4906      -0.5700
           Sing                                                             Sing       [The4]

D1        .           0.4840       0.4840       1.1825       1.1825       0.1033       0.1034     179.6950        .
          .           0.4866       0.4866       0.2768       0.2768       0.7479       0.7478        .            .
          .           0.2276      -0.1896       0.2984      -0.2486      -0.2776       1.4906        .            .
           Sing                                                                                     [Psi1]         Sing

D2        .           0.0000       0.0000       0.5942       0.5942       0.1035       0.1035        .          181.2787
          .           0.9961       0.9961       0.4408       0.4408       0.7477       0.7477        .            .
          .           0.0014      -0.0011      -0.2806       0.2338       0.1062      -0.5700        .            .
           Sing                                                                                       Sing       [Psi2]
```

Output 19.1.10. (continued)

```
                            Stability of Alienation
               Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

            Covariance Structure Analysis: Maximum Likelihood Estimation

            Rank Order of the 10 Largest Lagrange Multipliers in _PHI_

                  Row         Column      Chi-Square    Pr > ChiSq

                  E5          E2             7.36486        0.0067
                  E5          E1             5.80246        0.0160
                  E1          F3             3.39030        0.0656
                  E2          F3             3.39013        0.0656
                  E5          E3             1.59820        0.2062
                  E6          E2             1.41677        0.2339
                  E5          E4             1.20437        0.2724
                  E4          E1             1.20367        0.2726
                  D1          E3             1.18251        0.2768
                  D1          E4             1.18249        0.2768
```

Output 19.1.10. (continued)

```
                        Stability of Alienation
         Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

      Covariance Structure Analysis: Maximum Likelihood Estimation

        Lagrange Multiplier and Wald Test Indices _GAMMA_ [8:1]
                             General Matrix
               Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

                                          F3

                           v1         3.3903
                                      0.0656
                                      0.0768


                           v2         3.3901
                                      0.0656
                                     -0.0639


                           v3         0.5752
                                      0.4482
                                      0.0316


                           v4         0.5753
                                      0.4482
                                     -0.0263


                           v5            .
                                         .
                                         .
                                        Sing

                           v6       153.2354
                                         .
                                         .
                                      [Lamb]

                           F1       125.0132
                                         .
                                         .
                                      [Gam1]

                           F2        19.2585
                                         .
                                         .
                                      [Gam2]
```

Output 19.1.10. (continued)

```
                              Stability of Alienation
                    Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                 Covariance Structure Analysis: Maximum Likelihood Estimation

                 Rank Order of the 4 Largest Lagrange Multipliers in _GAMMA_

                       Row         Column      Chi-Square    Pr > ChiSq

                       v1          F3             3.39030        0.0656
                       v2          F3             3.39013        0.0656
                       v4          F3             0.57526        0.4482
                       v3          F3             0.57523        0.4482
```

Output 19.1.10. (continued)

```
                                                 Stability of Alienation
                                  Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                               Covariance Structure Analysis: Maximum Likelihood Estimation

                                 Lagrange Multiplier and Wald Test Indices _BETA_  [8:8]
                                                      General Matrix
                                           Identity-Minus-Inverse Model Matrix
                                        Univariate Tests for Constant Constraints
                         Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

                  v1            v2            v3            v4            v5            v6            F1            F2

    v1         .            0.1647        0.0511        0.8029        5.4083        0.1233        0.4047        0.4750
               .            0.6849        0.8212        0.3702        0.0200        0.7255        0.5247        0.4907
               .           -0.0159       -0.0063       -0.0284        0.0697        0.0015       -0.0257       -0.0239
                Sing

    v2        0.5957         .            0.6406        0.0135        5.8858        0.0274        0.4047        0.4750
              0.4402         .            0.4235        0.9076        0.0153        0.8686        0.5247        0.4907
              0.0218         .            0.0185        0.0032       -0.0609       -0.0006        0.0214        0.0199
                              Sing

    v3        0.3839        0.3027         .            0.1446        1.1537        0.0296        0.1588        0.0817
              0.5355        0.5822         .            0.7038        0.2828        0.8634        0.6902        0.7750
              0.0178        0.0180         .           -0.0145        0.0322        0.0007        0.0144       -0.0110
                                            Sing

    v4        0.4487        0.2519        0.0002         .            0.9867        0.1442        0.1588        0.0817
              0.5030        0.6157        0.9877         .            0.3206        0.7041        0.6903        0.7750
             -0.0160       -0.0144       -0.0004         .           -0.0249       -0.0014       -0.0120        0.0092
                                                          Sing

    v5        5.4085        8.6455        2.7123        2.1457         .             .            0.1033        0.1035
              0.0200        0.0033        0.0996        0.1430         .             .            0.7479        0.7476
              0.1242       -0.1454        0.0785       -0.0674         .             .           -0.0490        0.0329
                                                                        Sing          Sing

    v6        0.4209        1.4387        0.3044        0.0213         .             .            0.1034        0.1035
              0.5165        0.2304        0.5811        0.8841         .             .            0.7478        0.7477
             -0.2189        0.3924       -0.1602        0.0431         .             .            0.2629       -0.1765
                                                                        Sing          Sing

    F1        1.0998        1.1021        1.6114        1.6128        0.1032        0.1035         .             .
              0.2943        0.2938        0.2043        0.2041        0.7480        0.7477         .             .
              0.0977       -0.0817        0.0993       -0.0831       -0.0927        0.0057         .             .
                                                                                                    Sing          Sing

    F2        0.0193        0.0194        0.4765        0.4760        0.1034        0.1035      160.7520         .
              0.8896        0.8892        0.4900        0.4902        0.7477        0.7477         .             .
             -0.0104        0.0087       -0.0625        0.0522        0.0355       -0.0022         .             .
                                                                                                  [Beta]          Sing
```

Output 19.1.10. (continued)

```
                              Stability of Alienation
                   Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                Covariance Structure Analysis: Maximum Likelihood Estimation

                Rank Order of the 10 Largest Lagrange Multipliers in _BETA_

                      Row         Column      Chi-Square    Pr > ChiSq

                      v5          v2             8.64546        0.0033
                      v2          v5             5.88576        0.0153
                      v5          v1             5.40848        0.0200
                      v1          v5             5.40832        0.0200
                      v5          v3             2.71233        0.0996
                      v5          v4             2.14572        0.1430
                      F1          v4             1.61279        0.2041
                      F1          v3             1.61137        0.2043
                      v6          v2             1.43867        0.2304
                      v3          v5             1.15372        0.2828
```

When you specify equality constraints, PROC CALIS displays Lagrange multiplier tests for releasing the constraints. In the current example, none of the three constraints achieve a *p*-value smaller than 0.05. This means that releasing the constraints may not lead to a significantly better fit of the model. Therefore, all constraints are retained in the model.

Output 19.1.11. Tests for Equality Constraints

```
                              Stability of Alienation
                   Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                Covariance Structure Analysis: Maximum Likelihood Estimation

           Univariate Lagrange Multiplier Test for Releasing Equality Constraints

            Equality Constraint    -----Changes-----    Chi-Square    Pr > ChiSq

            [E1:E1] = [E3:E3]        0.0293  -0.0308       0.02106        0.8846
            [E2:E2] = [E4:E4]       -0.1342   0.1388       0.69488        0.4045
            [E3:E1] = [E4:E2]        0.2468  -0.1710       1.29124        0.2558
```

The model is specified using the LINEQS, STD, and COV statements. The section “Getting Started” on page 560 also contains the COSAN and RAM specifications of this model. These model specifications would give essentially the same results.

```
   proc calis cov data=Wheaton tech=nr edf=931;
      Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
      Matrix A
              [ ,7] = 1. .833  5 * 0. Beta (.5) ,
              [ ,8] = 2 * 0.  1.  .833 ,
              [ ,9] = 4 * 0.  1.  Lamb Gam1-Gam2 (.5 2 * -.5);
      Matrix P
              [1,1] = The1-The2 The1-The4 (6 * 3.) ,
              [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
              [3,1] = The5 (.2) ,
              [4,2] = The5 (.2) ;
      Vnames J V1-V6 F1-F3 ,
             A = J ,
             P E1-E6 D1-D3 ;
   run;

   proc calis cov data=Wheaton tech=nr edf=931;
      Ram
         1   1   7   1.       ,
         1   2   7   .833     ,
         1   3   8   1.       ,
         1   4   8   .833     ,
         1   5   9   1.       ,
         1   6   9   .5    Lamb ,
         1   7   9   -.5   Gam1 ,
         1   8   7   .5    Beta ,
         1   8   9   -.5   Gam2 ,
         2   1   1   3.    The1 ,
         2   2   2   3.    The2 ,
         2   3   3   3.    The1 ,
         2   4   4   3.    The2 ,
         2   5   5   3.    The3 ,
         2   6   6   3.    The4 ,
         2   1   3   .2    The5 ,
         2   2   4   .2    The5 ,
         2   7   7   4.    Psi1 ,
         2   8   8   4.    Psi2 ,
         2   9   9   6.    Phi ;
      Vnames 1 F1-F3,
             2 E1-E6 D1-D3;
   run;
```

Example 19.2. Simultaneous Equations with Intercept

The demand-and-supply food example of Kmenta (1971, pp. 565, 582) is used to illustrate the use of PROC CALIS for the estimation of intercepts and coefficients of simultaneous equations. The model is specified by two simultaneous equations containing two endogenous variables Q and P and three exogenous variables D, F, and Y,

$$Q_t(demand) = \alpha_1 + \beta_1 P_t + \gamma_1 D_t$$

$$Q_t(supply) = \alpha_2 + \beta_2 P_t + \gamma_2 F_t + \gamma_3 Y_t$$

for $t = 1, \ldots, 20$.

The LINEQS statement requires that each endogenous variable appear on the left-hand side of exactly one equation. Instead of analyzing the system

$$\mathbf{B}^* \boldsymbol{\eta} = \boldsymbol{\Gamma} \boldsymbol{\xi} + \boldsymbol{\zeta}$$

PROC CALIS analyzes the equivalent system

$$\boldsymbol{\eta} = \mathbf{B} \boldsymbol{\eta} + \boldsymbol{\Gamma} \boldsymbol{\xi} + \boldsymbol{\zeta}$$

with $\mathbf{B}^* = \mathbf{I} - \mathbf{B}$. This requires that one of the preceding equations be solved for P_t. Solving the second equation for P_t yields

$$P_t = \frac{1}{\beta_2}Q_t - \frac{\alpha_2}{\beta_2} - \frac{\gamma_2}{\beta_2}F_t - \frac{\gamma_3}{\beta_2}Y_t$$

You can estimate the intercepts of a system of simultaneous equations by applying PROC CALIS on the uncorrected covariance (UCOV) matrix of the data set that is augmented by an additional constant variable with the value 1. In the following example, the uncorrected covariance matrix is augmented by an additional variable INTERCEPT by using the AUGMENT option. The PROC CALIS statement contains the options UCOV and AUG to compute and analyze an augmented UCOV matrix from the input data set FOOD.

```
Data food;
Title 'Food example of KMENTA(1971, p.565 & 582)';
  Input Q P D F Y;
  Label  Q='Food Consumption per Head'
         P='Ratio of Food Prices to General Price'
         D='Disposable Income in Constant Prices'
         F='Ratio of Preceding Years Prices'
         Y='Time in Years 1922-1941';
datalines;
  98.485  100.323   87.4   98.0   1
  99.187  104.264   97.6   99.1   2
 102.163  103.435   96.7   99.1   3
 101.504  104.506   98.2   98.1   4
 104.240   98.001   99.8  110.8   5
 103.243   99.456  100.5  108.2   6
 103.993  101.066  103.2  105.6   7
  99.900  104.763  107.8  109.8   8
 100.350   96.446   96.6  108.7   9
 102.820   91.228   88.9  100.6  10
  95.435   93.085   75.1   81.0  11
  92.424   98.801   76.9   68.6  12
  94.535  102.908   84.6   70.9  13
  98.757   98.756   90.6   81.4  14
 105.797   95.119  103.1  102.3  15
 100.225   98.451  105.1  105.0  16
 103.522   86.498   96.4  110.5  17
  99.929  104.016  104.4   92.5  18
 105.223  105.769  110.7   89.3  19
 106.232  113.490  127.1   93.0  20
;
```

```
proc calis ucov aug data=food pshort;
   Title2 'Compute ML Estimates With Intercept';
   Lineqs
      Q = alf1 Intercept + alf2 P + alf3 D + E1,
      P = gam1 Intercept + gam2 Q + gam3 F + gam4 Y + E2;
   Std
      E1-E2 = eps1-eps2;
   Cov
      E1-E2 = eps3;
   Bounds
      eps1-eps2 >= 0. ;
run;
```

The following, essentially equivalent model definition uses program code to reparameterize the model in terms of the original equations; the output is displayed in Output 19.2.1.

```
proc calis data=food ucov aug pshort;
   Lineqs
      Q = alphal Intercept + beta1 P + gamma1 D + E1,
      P = alpha2_b Intercept + gamma2_b F + gamma3_b Y + _b Q + E2;
   Std
      E1-E2 = eps1-eps2;
   Cov
      E1-E2 = eps3;
   Parameters alpha2 (50.) beta2 gamma2 gamma3 (3*.25);
      alpha2_b = -alpha2 / beta2;
      gamma2_b = -gamma2 / beta2;
      gamma3_b = -gamma3 / beta2;
      _b       = 1 / beta2;
   Bounds
      eps1-eps2 >= 0. ;
run;
```

Output 19.2.1. Food Example of Kmenta

```
                        Food example of KMENTA(1971, p.565 & 582)

                                  The CALIS Procedure
                   Covariance Structure Analysis: Pattern and Initial Values

                                 LINEQS Model Statement


                             Matrix      Rows    Columns    ------Matrix Type-------

      Term 1            1    _SEL_          6          8    SELECTION
                        2    _BETA_         8          8    EQSBETA         IMINUSINV
                        3    _GAMMA_        8          6    EQSGAMMA
                        4    _PHI_          6          6    SYMMETRIC


                                The 2 Endogenous Variables

          Manifest        Q          P
          Latent


                                The 6 Exogenous Variables

          Manifest        D          F          Y          Intercept
          Latent
          Error           E1         E2
```

Output 19.2.1. (continued)

```
                            Covariance Structure Analysis: Maximum Likelihood Estimation

                                      Parameter Estimates                   10
                                      Functions (Observations)              21
                                      Lower Bounds                           2
                                      Upper Bounds                           0

                                                 Optimization Start

    Active Constraints                                  0  Objective Function                         2.350006504
    Max Abs Gradient Element                 203.97414363  Radius                                    62167.829154


                                                                                                            Ratio
                                                                                                          Between
                                                                                                           Actual
                                                                    Objective     Max Abs                     and
                        Function         Active        Objective     Function    Gradient               Predicted
   Iter     Restarts       Calls    Constraints         Function       Change     Element     Lambda       Change

      1            0           2              0          1.19094       1.1591      3.9410          0        0.688
      2            0           5              0          0.32678       0.8642      9.9864    0.00127        2.356
      3            0           7              0          0.19108       0.1357      5.5100    0.00006        0.685
      4            0          10              0          0.16682       0.0243      2.0513    0.00005        0.867
      5            0          12              0          0.16288      0.00393      1.0570    0.00014        0.828
      6            0          13              0          0.16132      0.00156      0.3643    0.00004        0.864
      7            0          15              0          0.16077     0.000557      0.2176    0.00006        0.984
      8            0          16              0          0.16052     0.000250      0.1819    0.00001        0.618
      9            0          17              0          0.16032     0.000201      0.0663          0        0.971
     10            0          18              0          0.16030     0.000011      0.0195          0        1.108
     11            0          19              0          0.16030     6.118E-7     0.00763          0        1.389
     12            0          20              0          0.16030     9.454E-8     0.00301          0        1.389
     13            0          21              0          0.16030     1.462E-8     0.00118          0        1.389
     14            0          22              0          0.16030     2.246E-9    0.000466          0        1.380
     15            0          23              0          0.16030     3.61E-10    0.000183          0        1.436

                                                Optimization Results

    Iterations                                         15  Function Calls                                     24
    Jacobian Calls                                     16  Active Constraints                                  0
    Objective Function                       0.1603035477  Max Abs Gradient Element                 0.0001826654
    Lambda                                              0  Actual Over Pred Change                    1.43562251
    Radius                                   0.0010320614

         GCONV convergence criterion satisfied.
```

Output 19.2.1. (continued)

```
                              Covariance Structure Analysis: Maximum Likelihood Estimation

                              Fit Function                                          0.1603
                              Goodness of Fit Index (GFI)                           0.9530
                              GFI Adjusted for Degrees of Freedom (AGFI)            0.0120
                              Root Mean Square Residual (RMR)                       2.0653
                              Parsimonious GFI (Mulaik, 1989)                       0.0635
                              Chi-Square                                            3.0458
                              Chi-Square DF                                              1
                              Pr > Chi-Square                                       0.0809
                              Independence Model Chi-Square                         534.27
                              Independence Model Chi-Square DF                          15
                              RMSEA Estimate                                        0.3281
                              RMSEA 90% Lower Confidence Limit                           .
                              RMSEA 90% Upper Confidence Limit                      0.7777
                              ECVI Estimate                                         1.8270
                              ECVI 90% Lower Confidence Limit                            .
                              ECVI 90% Upper Confidence Limit                       3.3493
                              Probability of Close Fit                              0.0882
                              Bentler's Comparative Fit Index                       0.9961
                              Normal Theory Reweighted LS Chi-Square                2.8142
                              Akaike's Information Criterion                        1.0458
                              Bozdogan's (1987) CAIC                               -0.9500
                              Schwarz's Bayesian Criterion                          0.0500
                              McDonald's (1989) Centrality                          0.9501
                              Bentler & Bonett's (1980) Non-normed Index            0.9409
                              Bentler & Bonett's (1980) NFI                         0.9943
                              James, Mulaik, & Brett (1982) Parsimonious NFI        0.0663
                              Z-Test of Wilson & Hilferty (1931)                    1.4250
                              Bollen (1986) Normed Index Rho1                       0.9145
                              Bollen (1988) Non-normed Index Delta2                 0.9962
                              Hoelter's (1983) Critical N                               25
```

Output 19.2.1. (continued)

```
                              Covariance Structure Analysis: Maximum Likelihood Estimation

      Q         =  -0.2295*P         +  0.3100*D         + 93.6193*Intercept +  1.0000 E1
                           beta1               gamma1              alphal
      P         =   4.2140*Q         + -0.9305*F         + -1.5579*Y         +  -218.9*Intercept +  1.0000 E2
                           _b                  gamma2_b            gamma3_b            alpha2_b
```

Output 19.2.1. (continued)

```
Covariance Structure Analysis: Maximum Likelihood Estimation

             Variances of Exogenous Variables

             Variable  Parameter      Estimate

             D                           10154
             F                            9989
             Y                       151.05263
             Intercept                 1.05263
             E1        eps1            3.51274
             E2        eps2          105.06746


           Covariances Among Exogenous Variables

        Var1      Var2      Parameter      Estimate

        D         F                            9994
        D         Y                            1101
        F         Y                            1046
        D         Intercept               102.66842
        F         Intercept               101.71053
        Y         Intercept                11.05263
        E1        E2        eps3          -18.87270
```

Output 19.2.1. (continued)

```
                               Covariance Structure Analysis: Maximum Likelihood Estimation

      Q         =  -0.2278*P        +  0.3016*D         +  0.9272*Intercept +  0.0181 E1
                           beta1               gamma1              alphal
      P         =   4.2467*Q        + -0.9048*F         + -0.1863*Y         + -2.1849*Intercept +  0.0997 E2
                           _b                  gamma2_b            gamma3_b            alpha2_b


                                               Squared Multiple Correlations

                                                           Error         Total
                                         Variable       Variance      Variance    R-Square

                                    1    Q               3.51274         10730      0.9997
                                    2    P             105.06746         10565      0.9901


                                         Correlations Among Exogenous Variables

                                       Var1      Var2      Parameter      Estimate

                                       D         F                         0.99237
                                       D         Y                         0.88903
                                       F         Y                         0.85184
                                       D         Intercept                 0.99308
                                       F         Intercept                 0.99188
                                       Y         Intercept                 0.87652
                                       E1        E2        eps3           -0.98237


                                        Additional PARMS and Dependent Parameters

                                         The Number of Dependent Parameters is 4

                                                                  Standard
                                     Parameter      Estimate         Error    t Value

                                     alpha2         51.94453             .        .
                                     beta2           0.23731             .        .
                                     gamma2          0.22082             .        .
                                     gamma3          0.36971             .        .
                                     _b              4.21397             .        .
                                     gamma2_b       -0.93053             .        .
                                     gamma3_b       -1.55794             .        .
                                     alpha2_b     -218.89288             .        .
```

You can obtain almost equivalent results by applying the SAS/ETS procedure SYSLIN on this problem.

Example 19.3. Second-Order Confirmatory Factor Analysis

A second-order confirmatory factor analysis model is applied to a correlation matrix of Thurstone reported by McDonald (1985). Using the LINEQS statement, the three-term second-order factor analysis model is specified in equations notation. The first-order loadings for the three factors, F1, F2, and F3, each refer to three variables, X1-X3, X4-X6, and X7-X9. One second-order factor, F4, reflects the correlations among the three first-order factors. The second-order factor correlation matrix P is defined as a 1×1 identity matrix. Choosing the second-order uniqueness matrix U2 as a diagonal matrix with parameters U21-U23 gives an unidentified model. To compute identified maximum likelihood estimates, the matrix U2 is defined as a 3×3 identity matrix. The following code generates results that are partially displayed in Output 19.3.1.

```
Data Thurst(TYPE=CORR);
Title "Example of THURSTONE resp. McDONALD (1985, p.57, p.105)";
   _TYPE_ = ’CORR’; Input _NAME_ $ Obs1-Obs9;
   Label Obs1=’Sentences’ Obs2=’Vocabulary’ Obs3=’Sentence Completion’
         Obs4=’First Letters’ Obs5=’Four-letter Words’ Obs6=’Suffices’
         Obs7=’Letter series’ Obs8=’Pedigrees’ Obs9=’Letter Grouping’;
   Datalines;
Obs1  1.       .      .      .      .      .      .      .      .
Obs2   .828   1.      .      .      .      .      .      .      .
Obs3   .776   .779   1.      .      .      .      .      .      .
Obs4   .439   .493    .460  1.      .      .      .      .      .
Obs5   .432   .464    .425   .674  1.      .      .      .      .
Obs6   .447   .489    .443   .590   .541  1.      .      .      .
Obs7   .447   .432    .401   .381   .402   .288  1.      .      .
Obs8   .541   .537    .534   .350   .367   .320   .555  1.      .
Obs9   .380   .358    .359   .424   .446   .325   .598   .452  1.
;
```

```
proc calis data=Thurst method=max edf=212 pestim se;
Title2 "Identified Second Order Confirmatory Factor Analysis";
Title3 "C = F1 * F2 * P * F2’ * F1’ + F1 * U2 * F1’ + U1, With P=U2=Ide";
Lineqs
   Obs1 = X1 F1 + E1,
   Obs2 = X2 F1 + E2,
   Obs3 = X3 F1 + E3,
   Obs4 = X4 F2 + E4,
   Obs5 = X5 F2 + E5,
   Obs6 = X6 F2 + E6,
   Obs7 = X7 F3 + E7,
   Obs8 = X8 F3 + E8,
   Obs9 = X9 F3 + E9,
   F1   = X10 F4 + E10,
   F2   = X11 F4 + E11,
   F3   = X12 F4 + E12;
Std
   F4      = 1. ,
   E1-E9   = U11-U19 ,
   E10-E12 = 3 * 1.;
Bounds
   0. <= U11-U19;
run;
```

Output 19.3.1. Second-Order Confirmatory Factor Analysis

```
                        Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
                          Identified Second Order Confirmatory Factor Analysis
                     C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

                                       The CALIS Procedure
                       Covariance Structure Analysis: Maximum Likelihood Estimation

                                  Parameter Estimates                 21
                                  Functions (Observations)            45
                                  Lower Bounds                         9
                                  Upper Bounds                         0

                                        Optimization Start

   Active Constraints                                 0  Objective Function                  0.7151823452
   Max Abs Gradient Element                0.4067179803  Radius                              2.2578762496


                                                                                                  Ratio
                                                                                                Between
                                                                                                 Actual
                                                                   Objective     Max Abs            and
                            Function         Active    Objective    Function    Gradient      Predicted
  Iter     Restarts            Calls    Constraints     Function      Change     Element    Lambda       Change

     1            0                2              0      0.23113      0.4840      0.1299         0        1.363
     2            0                3              0      0.18322      0.0479      0.0721         0        1.078
     3            0                4              0      0.18051     0.00271      0.0200         0        1.006
     4            0                5              0      0.18022    0.000289     0.00834         0        1.093
     5            0                6              0      0.18018    0.000041     0.00251         0        1.201
     6            0                7              0      0.18017    6.523E-6     0.00114         0        1.289
     7            0                8              0      0.18017    1.085E-6    0.000388         0        1.347
     8            0                9              0      0.18017    1.853E-7    0.000173         0        1.380
     9            0               10              0      0.18017    3.208E-8    0.000063         0        1.399
    10            0               11              0      0.18017    5.593E-9    0.000028         0        1.408
    11            0               12              0      0.18017    9.79E-10    0.000011         0        1.414

                                       Optimization Results

   Iterations                                        11  Function Calls                                13
   Jacobian Calls                                    12  Active Constraints                             0
   Objective Function                      0.1801712147  Max Abs Gradient Element            0.0000105805
   Lambda                                             0  Actual Over Pred Change             1.4135921728
   Radius                                  0.0002026368

        GCONV convergence criterion satisfied.
```

Output 19.3.1. (continued)

```
                               Identified Second Order Confirmatory Factor Analysis
                         C = F1 * F2 * P * F2′ * F1′ + F1 * U2 * F1′ + U1, With P=U2=Ide

                           Covariance Structure Analysis: Maximum Likelihood Estimation

                           Fit Function                                          0.1802
                           Goodness of Fit Index (GFI)                           0.9596
                           GFI Adjusted for Degrees of Freedom (AGFI)            0.9242
                           Root Mean Square Residual (RMR)                       0.0436
                           Parsimonious GFI (Mulaik, 1989)                       0.6397
                           Chi-Square                                           38.1963
                           Chi-Square DF                                             24
                           Pr > Chi-Square                                       0.0331
                           Independence Model Chi-Square                         1101.9
                           Independence Model Chi-Square DF                          36
                           RMSEA Estimate                                        0.0528
                           RMSEA 90% Lower Confidence Limit                      0.0153
                           RMSEA 90% Upper Confidence Limit                      0.0831
                           ECVI Estimate                                         0.3881
                           ECVI 90% Lower Confidence Limit                            .
                           ECVI 90% Upper Confidence Limit                       0.4888
                           Probability of Close Fit                              0.4088
                           Bentler’s Comparative Fit Index                       0.9867
                           Normal Theory Reweighted LS Chi-Square               40.1947
                           Akaike’s Information Criterion                       -9.8037
                           Bozdogan’s (1987) CAIC                             -114.4747
                           Schwarz’s Bayesian Criterion                        -90.4747
                           McDonald’s (1989) Centrality                          0.9672
                           Bentler & Bonett’s (1980) Non-normed Index            0.9800
                           Bentler & Bonett’s (1980) NFI                         0.9653
                           James, Mulaik, & Brett (1982) Parsimonious NFI        0.6436
                           Z-Test of Wilson & Hilferty (1931)                    1.8373
                           Bollen (1986) Normed Index Rho1                       0.9480
                           Bollen (1988) Non-normed Index Delta2                 0.9868
                           Hoelter’s (1983) Critical N                              204
```

Output 19.3.1. (continued)

```
                    Identified Second Order Confirmatory Factor Analysis
               C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

                 Covariance Structure Analysis: Maximum Likelihood Estimation

                        Obs1    =   0.5151*F1       +  1.0000 E1
                        Std Err     0.0629 X1
                        t Value     8.1868
                        Obs2    =   0.5203*F1       +  1.0000 E2
                        Std Err     0.0634 X2
                        t Value     8.2090
                        Obs3    =   0.4874*F1       +  1.0000 E3
                        Std Err     0.0608 X3
                        t Value     8.0151
                        Obs4    =   0.5211*F2       +  1.0000 E4
                        Std Err     0.0611 X4
                        t Value     8.5342
                        Obs5    =   0.4971*F2       +  1.0000 E5
                        Std Err     0.0590 X5
                        t Value     8.4213
                        Obs6    =   0.4381*F2       +  1.0000 E6
                        Std Err     0.0560 X6
                        t Value     7.8283
                        Obs7    =   0.4524*F3       +  1.0000 E7
                        Std Err     0.0660 X7
                        t Value     6.8584
                        Obs8    =   0.4173*F3       +  1.0000 E8
                        Std Err     0.0622 X8
                        t Value     6.7135
                        Obs9    =   0.4076*F3       +  1.0000 E9
                        Std Err     0.0613 X9
                        t Value     6.6484




                    Identified Second Order Confirmatory Factor Analysis
               C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

                 Covariance Structure Analysis: Maximum Likelihood Estimation

                        F1      =   1.4438*F4       +  1.0000 E10
                        Std Err     0.2565 X10
                        t Value     5.6282
                        F2      =   1.2538*F4       +  1.0000 E11
                        Std Err     0.2114 X11
                        t Value     5.9320
                        F3      =   1.4065*F4       +  1.0000 E12
                        Std Err     0.2689 X12
                        t Value     5.2307
```

Output 19.3.1. (continued)

```
                   Identified Second Order Confirmatory Factor Analysis
             C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

               Covariance Structure Analysis: Maximum Likelihood Estimation

                            Variances of Exogenous Variables

                                                         Standard
                Variable Parameter      Estimate            Error     t Value

                F4                       1.00000
                E1       U11             0.18150          0.02848        6.37
                E2       U12             0.16493          0.02777        5.94
                E3       U13             0.26713          0.03336        8.01
                E4       U14             0.30150          0.05102        5.91
                E5       U15             0.36450          0.05264        6.93
                E6       U16             0.50642          0.05963        8.49
                E7       U17             0.39032          0.05934        6.58
                E8       U18             0.48138          0.06225        7.73
                E9       U19             0.50509          0.06333        7.98
                E10                      1.00000
                E11                      1.00000
                E12                      1.00000
```

Output 19.3.1. (continued)

```
                   Identified Second Order Confirmatory Factor Analysis
             C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

               Covariance Structure Analysis: Maximum Likelihood Estimation

                          Obs1    =   0.9047*F1   +  0.4260 E1
                                             X1
                          Obs2    =   0.9138*F1   +  0.4061 E2
                                             X2
                          Obs3    =   0.8561*F1   +  0.5168 E3
                                             X3
                          Obs4    =   0.8358*F2   +  0.5491 E4
                                             X4
                          Obs5    =   0.7972*F2   +  0.6037 E5
                                             X5
                          Obs6    =   0.7026*F2   +  0.7116 E6
                                             X6
                          Obs7    =   0.7808*F3   +  0.6248 E7
                                             X7
                          Obs8    =   0.7202*F3   +  0.6938 E8
                                             X8
                          Obs9    =   0.7035*F3   +  0.7107 E9
                                             X9



                   Identified Second Order Confirmatory Factor Analysis
             C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

               Covariance Structure Analysis: Maximum Likelihood Estimation

                          F1      =   0.8221*F4   +  0.5694 E10
                                             X10
                          F2      =   0.7818*F4   +  0.6235 E11
                                             X11
                          F3      =   0.8150*F4   +  0.5794 E12
                                             X12
```

Output 19.3.1. (continued)

```
                    Identified Second Order Confirmatory Factor Analysis
               C = F1 * F2 * P * F2′ * F1′ + F1 * U2 * F1′ + U1, With P=U2=Ide

                 Covariance Structure Analysis: Maximum Likelihood Estimation

                               Squared Multiple Correlations

                                       Error          Total
                    Variable        Variance       Variance     R-Square

               1    Obs1             0.18150        1.00000       0.8185
               2    Obs2             0.16493        1.00000       0.8351
               3    Obs3             0.26713        1.00000       0.7329
               4    Obs4             0.30150        1.00000       0.6985
               5    Obs5             0.36450        1.00000       0.6355
               6    Obs6             0.50642        1.00000       0.4936
               7    Obs7             0.39032        1.00000       0.6097
               8    Obs8             0.48138        1.00000       0.5186
               9    Obs9             0.50509        1.00000       0.4949
              10    F1               1.00000        3.08452       0.6758
              11    F2               1.00000        2.57213       0.6112
              12    F3               1.00000        2.97832       0.6642
```

To compute McDonald’s unidentified model, you would have to change the STD and BOUNDS statements to include three more parameters:

```
   Std
      F4      = 1. ,
      E1-E9   = U11-U19 ,
      E10-E12 = U21-U23 ;
   Bounds
      0. <= U11-U19,
      0. <= U21-U23;
```

The unidentified model is indicated in the output by an analysis of the linear dependencies in the approximate Hessian matrix (not shown). Because the information matrix is singular, standard errors are computed based on a Moore-Penrose inverse. The results computed by PROC CALIS differ from those reported by McDonald (1985). In the case of an unidentified model, the parameter estimates are not unique.

To specify the identified model using the COSAN model statement, you can use the following statements:

```
Title2 "Identified Second Order Confirmatory Factor Analysis Using COSAN";
Title3 "C = F1*F2*P*F2’*F1’ + F1*U2*F1’ + U1, With P=U2=Ide";
proc calis data=Thurst method=max edf=212 pestim se;
   Cosan F1(3) * F2(1) * P(1,Ide) + F1(3) * U2(3,Ide) + U1(9,Dia);
   Matrix F1
          [ ,1] = X1-X3,
          [ ,2] = 3 * 0. X4-X6,
          [ ,3] = 6 * 0. X7-X9;
   Matrix F2
          [ ,1] = X10-X12;

   Matrix U1
          [1,1] = U11-U19;
   Bounds
          0. <= U11-U19;
run;
```

Because PROC CALIS cannot compute initial estimates for a model specified by the general COSAN statement, this analysis may require more iterations than one using the LINEQS statement, depending on the precision of the processor.

Example 19.4. Linear Relations Among Factor Loadings

The correlation matrix from Kinzer and Kinzer (N=326) is used by Guttman (1957) as an example that yields an approximate simplex. McDonald (1980) uses this data set as an example of factor analysis where he supposes that the loadings of the second factor are a linear function of the loadings on the first factor, for example

$$b_{j2} = \alpha + \beta b_{j1}, \quad j = 1, \ldots, n$$

This example is also discussed in Browne (1982). The matrix specification of the model is

$$\mathbf{C} = \mathbf{F}_1 \mathbf{F}_1'$$

with

$$F_1 = \begin{pmatrix} b_{11} & \alpha + \beta b_{11} & u_{11} & & & & & \\ b_{21} & \alpha + \beta b_{21} & & u_{22} & & & & \\ b_{31} & \alpha + \beta b_{31} & & & u_{33} & & & \\ b_{41} & \alpha + \beta b_{41} & & & & u_{44} & & \\ b_{51} & \alpha + \beta b_{51} & & & & & u_{55} & \\ b_{61} & \alpha + \beta b_{61} & & & & & & u_{66} \end{pmatrix}$$

This example is recomputed by PROC CALIS to illustrate a simple application of the COSAN model statement combined with program statements. This example also serves to illustrate the identification problem.

```
Data Kinzer(TYPE=CORR);
Title "Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)";
   _TYPE_ = ’CORR’; INPUT _NAME_ $ Obs1-Obs6;
   Datalines;
Obs1  1.00   .     .     .     .     .
Obs2   .51  1.00   .     .     .     .
Obs3   .46   .51  1.00   .     .     .
Obs4   .46   .47   .54  1.00   .     .
Obs5   .40   .39   .49   .57  1.00   .
Obs6   .33   .39   .47   .45   .56  1.00
  ;
```

In a first test run of PROC CALIS, the same model is used as reported in McDonald (1980). Using the Levenberg-Marquardt optimization algorithm, this example specifies maximum likelihood estimation in the following code:

```
proc calis data=Kinzer method=max outram=ram nobs=326 noprint;
   Title2 "Linearly Related Factor Analysis, (Mcdonald,1980)";
   Title3 "Identification Problem";
   Cosan F(8,Gen) * I(8,Ide);
   Matrix F
          [ ,1]= X1-X6,
          [ ,2]= X7-X12,
          [1,3]= X13-X18;
   Parms Alfa = .5 Beta = -.5;
      X7  = Alfa + Beta * X1;
      X8  = Alfa + Beta * X2;
      X9  = Alfa + Beta * X3;
      X10 = Alfa + Beta * X4;
      X11 = Alfa + Beta * X5;
      X12 = Alfa + Beta * X6;
   Bounds X13-X18 >= 0.;
   Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

The pattern of the initial values is displayed in vector and in matrix form. You should always read this output very carefully, particularly when you use your own programming statements to constrain the matrix elements. The vector form shows the mapping of the model parameters to indices of the vector X that is optimized. The matrix form indicates parameter elements that are constrained by program statements by indices of X in angle brackets (< >). An asterisk trailing the iteration number in the displayed optimization history of the Levenberg-Marquardt algorithm indicates that the optimization process encountered a singular Hessian matrix. When this happens, especially in the last iterations, the model may not be properly identified. The computed χ^2 value of 10.337 for 7 degrees of freedom and the computed unique loadings agree with those reported by McDonald (1980), but the maximum likelihood estimates for the common factor loadings differ to some degree. The common factor loadings can be subjected to transformations that do not increase the value of the optimization criterion because the problem is not identified. An estimation problem that is not fully identified can lead to different solutions caused only by different initial values, different optimization techniques, or computers with different machine precision or floating-point arithmetic.

To overcome the identification problem in the first model, restart PROC CALIS with a simple modification to the model in which the former parameter X1 is fixed to 0. This leads to 8 instead of 7 degrees of freedom. The following code produces results that are partially displayed in Output 19.4.1.

```
Data ram2(TYPE=RAM);
  set ram;
  if _type_ = 'ESTIM' then
  if _name_ = 'X1' then do;
     _name_ = ' '; _estim_ = 0.;
  end;
run;
```

```
proc calis data=Kinzer method=max inram=ram2 nobs=326;
   Title2 "Linearly Related Factor Analysis, (Mcdonald,1980)";
   Title3 "Identified Model";
   Parms Alfa = .5 Beta = -.5;
      X7  = Alfa;
      X8  = Alfa + Beta * X2;
      X9  = Alfa + Beta * X3;
      X10 = Alfa + Beta * X4;
      X11 = Alfa + Beta * X5;
      X12 = Alfa + Beta * X6;
   Bounds X13-X18 >= 0.;
run;
```

Output 19.4.1. Linearly Related Factor Analysis: Identification Problem

```
                         Linearly Related Factor Analysis, (Mcdonald,1980)
                                          Identified Model

                                        The CALIS Procedure
                        Covariance Structure Analysis: Pattern and Initial Values

                                       COSAN Model Statement


                                   Matrix    Rows    Columns    ------Matrix Type-------

                 Term 1       1    F            6          8    GENERAL
                              2    I            8          8    IDENTITY




                         Linearly Related Factor Analysis, (Mcdonald,1980)
                                          Identified Model

                                        The CALIS Procedure
                       Covariance Structure Analysis: Maximum Likelihood Estimation

                              Parameter Estimates                  13
                              Functions (Observations)             21
                              Lower Bounds                          6
                              Upper Bounds                          0

                                         Optimization Start

      Active Constraints                             0  Objective Function                   0.3233206993
      Max Abs Gradient Element            2.2941016639  Radius                               5.9649770297
```

Output 19.4.1. (continued)

```
                     Linearly Related Factor Analysis, (Mcdonald,1980)
                                       Identified Model

                  Covariance Structure Analysis: Maximum Likelihood Estimation

                                                                                          Ratio
                                                                                        Between
                                                                                         Actual
                                                            Objective    Max Abs            and
                         Function       Active   Objective   Function   Gradient      Predicted
 Iter    Restarts           Calls  Constraints    Function     Change    Element  Lambda     Change

    1           0               2            0     0.07869     0.2446     0.3945       0      0.556
    2           0               3            0     0.03326     0.0454     0.0652       0      1.197
    3           0               4            0     0.03185    0.00142    0.00473       0      1.047
    4           0               5            0     0.03181   0.000033    0.00239       0      0.761
    5           0               6            0     0.03181   4.182E-6   0.000790       0      0.551
    6           0               7            0     0.03181   1.007E-6   0.000506       0      0.514
    7           0               8            0     0.03181   2.661E-7   0.000213       0      0.504
    8           0               9            0     0.03181   7.129E-8   0.000134       0      0.497
    9           0              10            0     0.03181   1.921E-8   0.000057       0      0.492
   10           0              11            0     0.03181   5.197E-9   0.000036       0      0.488
   11           0              12            0     0.03181    1.41E-9   0.000015       0      0.485
   12           0              13            0     0.03181   3.83E-10   9.489E-6       0      0.483

                                    Optimization Results

  Iterations                                  12  Function Calls                          14
  Jacobian Calls                              13  Active Constraints                       0
  Objective Function                0.0318073951  Max Abs Gradient Element      9.4889247E-6
  Lambda                                       0  Actual Over Pred Change         0.48329327
  Radius                            0.0002173982

       ABSGCONV convergence criterion satisfied.
```

Output 19.4.1. (continued)

```
                 Linearly Related Factor Analysis, (Mcdonald,1980)
                                  Identified Model

            Covariance Structure Analysis: Maximum Likelihood Estimation

            Fit Function                                          0.0318
            Goodness of Fit Index (GFI)                           0.9897
            GFI Adjusted for Degrees of Freedom (AGFI)            0.9730
            Root Mean Square Residual (RMR)                       0.0409
            Parsimonious GFI (Mulaik, 1989)                       0.5278
            Chi-Square                                           10.3374
            Chi-Square DF                                              8
            Pr > Chi-Square                                       0.2421
            Independence Model Chi-Square                         682.87
            Independence Model Chi-Square DF                          15
            RMSEA Estimate                                        0.0300
            RMSEA 90% Lower Confidence Limit                           .
            RMSEA 90% Upper Confidence Limit                      0.0756
            ECVI Estimate                                         0.1136
            ECVI 90% Lower Confidence Limit                            .
            ECVI 90% Upper Confidence Limit                       0.1525
            Probability of Close Fit                              0.7137
            Bentler’s Comparative Fit Index                       0.9965
            Normal Theory Reweighted LS Chi-Square               10.1441
            Akaike’s Information Criterion                       -5.6626
            Bozdogan’s (1987) CAIC                              -43.9578
            Schwarz’s Bayesian Criterion                        -35.9578
            McDonald’s (1989) Centrality                          0.9964
            Bentler & Bonett’s (1980) Non-normed Index            0.9934
            Bentler & Bonett’s (1980) NFI                         0.9849
            James, Mulaik, & Brett (1982) Parsimonious NFI        0.5253
            Z-Test of Wilson & Hilferty (1931)                    0.7019
            Bollen (1986) Normed Index Rho1                       0.9716
            Bollen (1988) Non-normed Index Delta2                 0.9965
            Hoelter’s (1983) Critical N                              489
```

Output 19.4.1. (continued)

```
                        Linearly Related Factor Analysis, (Mcdonald,1980)
                                        Identified Model

                      Covariance Structure Analysis: Maximum Likelihood Estimation

                                  Estimated Parameter Matrix F[6:8]
                                     Standard Errors and t Values
                                            General Matrix

              Fact1         Fact2         Uvar1         Uvar2         Uvar3         Uvar4         Uvar5         Uvar6

Obs1              0        0.7151        0.7283             0             0             0             0             0
                  0        0.0405        0.0408             0             0             0             0             0
                  0       17.6382       17.8276             0             0             0             0             0
                             <X7>         [X13]

Obs2        -0.0543        0.7294             0        0.6707             0             0             0             0
             0.1042        0.0438             0        0.0472             0             0             0             0
            -0.5215       16.6655             0       14.2059             0             0             0             0
               [X2]          <X8>                       [X14]

Obs3         0.1710        0.6703             0             0        0.6983             0             0             0
             0.0845        0.0396             0             0        0.0324             0             0             0
             2.0249       16.9077             0             0       21.5473             0             0             0
               [X3]          <X9>                                     [X15]

Obs4         0.2922        0.6385             0             0             0        0.6876             0             0
             0.0829        0.0462             0             0             0        0.0319             0             0
             3.5224       13.8352             0             0             0       21.5791             0             0
               [X4]         <X10>                                                   [X16]

Obs5         0.5987        0.5582             0             0             0             0        0.5579             0
             0.1003        0.0730             0             0             0             0        0.0798             0
             5.9665        7.6504             0             0             0             0        6.9937             0
               [X5]         <X11>                                                                 [X17]

Obs6         0.4278        0.6029             0             0             0             0             0        0.7336
             0.0913        0.0586             0             0             0             0             0        0.0400
             4.6844       10.2929             0             0             0             0             0       18.3580
               [X6]         <X12>                                                                               [X18]
```

Output 19.4.1. (continued)

```
                        Linearly Related Factor Analysis, (Mcdonald,1980)
                                        Identified Model

                      Covariance Structure Analysis: Maximum Likelihood Estimation

                                Additional PARMS and Dependent Parameters

                                 The Number of Dependent Parameters is 6

                                                     Standard
                         Parameter      Estimate        Error    t Value

                         Alfa            0.71511      0.04054      17.64
                         Beta           -0.26217      0.12966      -2.02
                         X7              0.71511      0.04054      17.64
                         X8              0.72936      0.04376      16.67
                         X9              0.67027      0.03964      16.91
                         X10             0.63851      0.04615      13.84
                         X11             0.55815      0.07296       7.65
                         X12             0.60295      0.05858      10.29
```

The lambda value of the iteration history indicates that Newton steps can always be performed. Because no singular Hessian matrices (which can slow down the convergence rate considerably) are computed, this example needs just 12 iterations compared to the 17 needed in the previous example. Note that the number of iterations may be machine-dependent. The value of the fit function, the residuals, and the χ^2 value agree with the values obtained in fitting the first model. This indicates that this second model is better identified than the first one. It is fully identified, as indicated by the fact that the Hessian matrix is nonsingular.

Example 19.5. Ordinal Relations Among Factor Loadings

McDonald (1980) uses the same data set to compute a factor analysis with ordinally constrained factor loadings. The results of the linearly constrained factor analysis show that the loadings of the two factors are ordered as 2, 1, 3, 4, 6, 5. McDonald (1980) then tests the hypothesis that the factor loadings are all nonnegative and can be ordered in the following manner:

$$b_{11} \leq b_{21} \leq b_{31} \leq b_{41} \leq b_{51} \leq b_{61}$$

$$b_{12} \geq b_{22} \geq b_{32} \geq b_{42} \geq b_{52} \geq b_{62}$$

This example is recomputed by PROC CALIS to illustrate a further application of the COSAN model statement combined with program statements. The same identification problem as in Example 19.4 on page 725 occurs here. The following model specification describes an unidentified model:

```
proc calis data=Kinzer method=max outram=ram tech=nr nobs=326 noprint;
Title2 "Ordinally Related Factor Analysis, (Mcdonald,1980)";
Title3 "Identification Problem";
Cosan F(8,Gen) * I(8,Ide);
   MATRIX F
      [,1]  = x1-x6,
      [,2]  = x7-x12,
      [1,3] = x13-x18;
   PARAMETERS t1-t10=1.;
   x2  = x1  + t1  * t1;
   x3  = x2  + t2  * t2;
   x4  = x3  + t3  * t3;
   x5  = x4  + t4  * t4;
   x6  = x5  + t5  * t5;
   x11 = x12 + t6  * t6;
   x10 = x11 + t7  * t7;
   x9  = x10 + t8  * t8;
   x8  = x9  + t9  * t9;
   x7  = x8  + t10 * t10;
   Bounds x13-x18 >= 0.;
   Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

You can specify the same model with the LINCON statement:

```
proc calis data=Kinzer method=max tech=lm edf=325;
Title3 "Identified Problem 2";
   cosan f(8,gen)*I(8,ide);
   matrix F
      [,1]  = x1-x6,
      [,2]  = x7-x12,
      [1,3] = x13-x18;
   lincon  x1  <= x2,
           x2  <= x3,
           x3  <= x4,
           x4  <= x5,
           x5  <= x6,
           x7  >= x8,
           x8  >= x9,
           x9  >= x10,
           x10 >= x11,
           x11 >= x12;
Bounds x13-x18 >= 0.;
Vnames F Fact1 Fact2 Uvar1-Uvar6;
run;
```

To have an identified model, the loading, b_{11} (x1), is fixed at 0. The information in the OUTRAM= data set (the data set ram), produced by the unidentified model, can be used to specify the identified model. However, because x1 is now a fixed constant in the identified model, it should not have a parameter name in the new analysis. Thus, the data set ram is modified as follows:

```
data ram2(type=ram);
   set ram;
   if _name_ = ’x1’ then do;
      _name_ = ’ ’; _estim_ = 0.;
   end;
run;
```

The data set ram2 is now an OUTRAM= data set in which x1 is no longer a parameter. PROC CALIS reads the information (that is, the set of parameters and the model specification) in the data set ram2 for the identified model. As displayed in the following code, you can use the PARMS statement to specify the desired ordinal relationships between the parameters.

```
proc calis data=Kinzer method=max inram=ram2 tech=nr nobs=326;
title2 "Ordinally Related Factor Analysis, (Mcdonald,1980)";
title3 "Identified Model with X1=0";
parms t1-t10= 10 * 1.;
      x2  =     + t1  * t1;
      x3  = x2  + t2  * t2;
      x4  = x3  + t3  * t3;
      x5  = x4  + t4  * t4;
      x6  = x5  + t5  * t5;
      x11 = x12 + t6  * t6;
      x10 = x11 + t7  * t7;
      x9  = x10 + t8  * t8;
      x8  = x9  + t9  * t9;
      x7  = x8  + t10 * t10;
bounds x13-x18 >= 0.;
run;
```

Selected output for the identified model is displayed in Output 19.5.1.

Output 19.5.1. Factor Analysis with Ordinal Constraints

```
                    Ordinally Related Factor Analysis, (Mcdonald,1980)
                                Identified Model with X1=0

                                   The CALIS Procedure
                  Covariance Structure Analysis: Maximum Likelihood Estimation

                            Parameter Estimates                    17
                            Functions (Observations)               21
                            Lower Bounds                            6
                            Upper Bounds                            0

                                   Optimization Start

   Active Constraints                               0  Objective Function                 5.2552270182
   Max Abs Gradient Element              0.8821788922


                                                                                              Ratio
                                                                                            Between
                                                                                             Actual
                                                              Objective    Max Abs              and
                    Function        Active      Objective      Function   Gradient        Predicted
  Iter   Restarts      Calls   Constraints       Function        Change    Element  Ridge    Change

     1          0          2             0        3.14901        2.1062     1.0712       0    2.226
     2          0          3             0        1.42725        1.7218     1.0902       0    2.064
     3          0          4             0        0.41661        1.0106     0.7472       0    1.731
     4          0          5             0        0.09260        0.3240     0.3365       0    1.314
     5          0          6             0        0.09186      0.000731     0.3880       0   0.0123
     6          0          8             0        0.04570        0.0462     0.2870  0.0313    0.797
     7          0         10             0        0.03269        0.0130     0.0909  0.0031    0.739
     8          0         16             0        0.02771       0.00498     0.0890  0.0800    0.682
     9          0         17             0        0.02602       0.00168     0.0174  0.0400    0.776
    10          0         19             0        0.02570      0.000323     0.0141  0.0800    0.630
    11          0         21             0        0.02560      0.000103    0.00179   0.160    1.170
    12          0         23             0        0.02559      7.587E-6   0.000670   0.160    1.423
    13          0         24             0        0.02559      2.993E-6   0.000402  0.0400    1.010
    14          0         27             0        0.02559      1.013E-6   0.000206   0.160    1.388
    15          0         28             0        0.02559      1.889E-7   0.000202  0.0400    0.530
    16          0         30             0        0.02559      1.803E-7   0.000097  0.0800    0.630
    17          0         32             0        0.02559      4.845E-8   0.000035   0.160    1.340
    18          0         33             0        0.02559      1.837E-9   0.000049  0.0400    0.125
    19          0         35             0        0.02559       9.39E-9   0.000024  0.0800    0.579
    20          0         37             0        0.02559      2.558E-9   6.176E-6   0.160    1.305

                                  Optimization Results

   Iterations                                      20  Function Calls                               38
   Jacobian Calls                                  21  Active Constraints                            0
   Objective Function                    0.0255871615  Max Abs Gradient Element           6.1764582E-6
   Ridge                                         0.04  Actual Over Pred Change            1.3054368156

        ABSGCONV convergence criterion satisfied.
```

Output 19.5.1. (continued)

```
                    Ordinally Related Factor Analysis, (Mcdonald,1980)
                                Identified Model with X1=0

                Covariance Structure Analysis: Maximum Likelihood Estimation

                Fit Function                                         0.0256
                Goodness of Fit Index (GFI)                          0.9916
                GFI Adjusted for Degrees of Freedom (AGFI)           0.9557
                Root Mean Square Residual (RMR)                      0.0180
                Parsimonious GFI (Mulaik, 1989)                      0.2644
                Chi-Square                                           8.3158
                Chi-Square DF                                             4
                Pr > Chi-Square                                      0.0807
                Independence Model Chi-Square                        682.87
                Independence Model Chi-Square DF                         15
                RMSEA Estimate                                       0.0576
                RMSEA 90% Lower Confidence Limit                          .
                RMSEA 90% Upper Confidence Limit                     0.1133
                ECVI Estimate                                        0.1325
                ECVI 90% Lower Confidence Limit                           .
                ECVI 90% Upper Confidence Limit                      0.1711
                Probability of Close Fit                             0.3399
                Bentler’s Comparative Fit Index                      0.9935
                Normal Theory Reweighted LS Chi-Square               8.2901
                Akaike’s Information Criterion                       0.3158
                Bozdogan’s (1987) CAIC                             -18.8318
                Schwarz’s Bayesian Criterion                       -14.8318
                McDonald’s (1989) Centrality                         0.9934
                Bentler & Bonett’s (1980) Non-normed Index           0.9758
                Bentler & Bonett’s (1980) NFI                        0.9878
                James, Mulaik, & Brett (1982) Parsimonious NFI       0.2634
                Z-Test of Wilson & Hilferty (1931)                   1.4079
                Bollen (1986) Normed Index Rho1                      0.9543
                Bollen (1988) Non-normed Index Delta2                0.9936
                Hoelter’s (1983) Critical N                             372
```

Output 19.5.1. (continued)

```
                       Ordinally Related Factor Analysis, (Mcdonald,1980)
                                  Identified Model with X1=0

                     Covariance Structure Analysis: Maximum Likelihood Estimation

                                Estimated Parameter Matrix F[6:8]
                                  Standard Errors and t Values
                                         General Matrix

              Fact1         Fact2         Uvar1         Uvar2         Uvar3         Uvar4         Uvar5         Uvar6

Obs1              0        0.7101        0.7131             0             0             0             0             0
                  0        0.0435        0.0404             0             0             0             0             0
                  0       16.3317       17.6427             0             0             0             0             0
                             <x7>         [x13]

Obs2         0.0261        0.7101             0        0.6950             0             0             0             0
             0.0875        0.0435             0        0.0391             0             0             0             0
             0.2977       16.3317             0       17.7571             0             0             0             0
               <x2>          <x8>                       [x14]

Obs3         0.2382        0.6827             0             0        0.6907             0             0             0
             0.0851        0.0604             0             0        0.0338             0             0             0
             2.7998       11.3110             0             0       20.4239             0             0             0
               <x3>          <x9>                                     [x15]

Obs4         0.3252        0.6580             0             0             0        0.6790             0             0
             0.0823        0.0621             0             0             0        0.0331             0             0
             3.9504       10.5950             0             0             0       20.5361             0             0
               <x4>         <x10>                                                   [x16]

Obs5         0.5395        0.5528             0             0             0             0        0.6249             0
             0.0901        0.0705             0             0             0             0        0.0534             0
             5.9887        7.8359             0             0             0             0       11.7052             0
               <x5>         <x11>                                                                 [x17]

Obs6         0.5395        0.4834             0             0             0             0             0        0.7005
             0.0918        0.0726             0             0             0             0             0        0.0524
             5.8776        6.6560             0             0             0             0             0       13.3749
               <x6>         [x12]                                                                               [x18]
```

Output 19.5.1. (continued)

```
                 Ordinally Related Factor Analysis, (Mcdonald,1980)
                            Identified Model with X1=0

             Covariance Structure Analysis: Maximum Likelihood Estimation

                      Additional PARMS and Dependent Parameters

                       The Number of Dependent Parameters is 10

                                                   Standard
                  Parameter      Estimate             Error    t Value

                  t1              0.16143           0.27111       0.60
                  t2              0.46060           0.09289       4.96
                  t3              0.29496           0.13702       2.15
                  t4              0.46297           0.10756       4.30
                  t5            0.0000522              1311       0.00
                  t6              0.26347           0.12203       2.16
                  t7              0.32430           0.09965       3.25
                  t8              0.15721           0.21134       0.74
                  t9              0.16543           0.20537       0.81
                  t10          -4.2528E-7           0.47736      -0.00
                  x7              0.71007           0.04348      16.33
                  x2              0.02606           0.08753       0.30
                  x8              0.71007           0.04348      16.33
                  x3              0.23821           0.08508       2.80
                  x9              0.68270           0.06036      11.31
                  x4              0.32521           0.08232       3.95
                  x10             0.65799           0.06210      10.60
                  x5              0.53955           0.09009       5.99
                  x11             0.55282           0.07055       7.84
                  x6              0.53955           0.09180       5.88
```

By fixing the loading b_{11} (x1) to constant 0, you obtain $\chi^2 = 8.316$ on $df = 4$ ($p < .09$). McDonald reports the same χ^2 value, but on $df = 3$, and thus, he obtains a smaller *p*-value. An analysis without the fixed loading shows typical signs of an unidentified problem: after more iterations it leads to a parameter set with a χ^2 value of 8.174 on $df = 3$. A singular Hessian matrix occurs.

The singular Hessian matrix of the unidentified problem slows down the convergence rate of the Levenberg-Marquardt algorithm considerably. Compared to the unidentified problem with 30 iterations, the identified problem needs only 20 iterations. Note that the number of iterations may depend on the precision of the processor.

The same model can also be specified using the LINCON statement for linear constraints:

```
proc calis data=Kinzer method=max tech=lm edf=325;
   Title3 "Identified Model 2";
   cosan f(8,gen)*I(8,ide);
   matrix f
      [,1]  = 0. x2-x6,
      [,2]  = x7-x12,
      [1,3] = x13-x18;
   lincon  x2  <= x3,
           x3  <= x4,
           x4  <= x5,
           x5  <= x6,
           x7  >= x8,
           x8  >= x9,
           x9  >= x10,
           x10 >= x11,
           x11 >= x12;
   bounds x2 x13-x18 >= 0.;
run;
```

Example 19.6. Longitudinal Factor Analysis

The following example (McDonald 1980) illustrates both the ability of PROC CALIS to formulate complex covariance structure analysis problems by the generalized COSAN matrix model and the use of program statements to impose nonlinear constraints on the parameters. The example is a longitudinal factor analysis using the Swaminathan (1974) model. For $m = 3$ tests, $k = 3$ occasions, and $r = 2$ factors the matrix model is formulated in the section "First-Order Autoregressive Longitudinal Factor Model" on page 554 as follows:

$$\mathbf{C} = \mathbf{F}_1\mathbf{F}_2\mathbf{F}_3\mathbf{L}\mathbf{F}_3^{-1}\mathbf{F}_2^{-1}\mathbf{P}(\mathbf{F}_2^{-1})'(\mathbf{F}_3^{-1})'\mathbf{L}'\mathbf{F}_3'\mathbf{F}_2'\mathbf{F}_1' + \mathbf{U}^2$$

$$F_1 = \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix}$$

$$L = \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$S_2 = I_2 - D_2^2, \qquad S_3 = I_2 - D_3^2$$

The Swaminathan longitudinal factor model assumes that the factor scores for each (m) common factor change from occasion to occasion (k) according to a first-order autoregressive scheme. The matrix $\mathbf{F}_1$ contains the k factor loading matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ (each is $n \times m$). The matrices $\mathbf{D}_2, \mathbf{D}_3, \mathbf{S}_2, \mathbf{S}_3$ and $\mathbf{U}_{ij}, i, j = 1, \ldots, k$, are diagonal, and the matrices $\mathbf{D}_i$ and $\mathbf{S}_i, i = 2, \ldots, k$, are subjected to the constraint

$$\mathbf{S}_i + \mathbf{D}_i^2 = \mathbf{I}$$

Since the constructed correlation matrix given in McDonald's (1980) paper is singular, only unweighted least-squares estimates can be computed.

```
data Mcdon(TYPE=CORR);
Title "Swaminathan's Longitudinal Factor Model, Data: McDONALD(1980)";
Title2 "Constructed Singular Correlation Matrix, GLS & ML not possible";
   _TYPE_ = 'CORR'; INPUT _NAME_ $ Obs1-Obs9;
   datalines;
Obs1  1.000    .      .      .      .      .      .      .      .
Obs2   .100  1.000    .      .      .      .      .      .      .
Obs3   .250   .400  1.000    .      .      .      .      .      .
Obs4   .720   .108   .270  1.000    .      .      .      .      .
Obs5   .135   .740   .380   .180  1.000    .      .      .      .
Obs6   .270   .318   .800   .360   .530  1.000    .      .      .
Obs7   .650   .054   .135   .730   .090   .180  1.000    .      .
Obs8   .108   .690   .196   .144   .700   .269   .200  1.000    .
Obs9   .189   .202   .710   .252   .336   .760   .350   .580  1.000
     ;



proc calis data=Mcdon method=ls tech=nr nobs=100;
cosan B(6,Gen) * D1(6,Dia) * D2(6,Dia) * T(6,Low) * D3(6,Dia,Inv)
               * D4(6,Dia,Inv) * P(6,Dia) + U(9,Sym);
   Matrix B
            [ ,1]= X1-X3,
            [ ,2]= 0. X4-X5,
            [ ,3]= 3 * 0. X6-X8,
            [ ,4]= 4 * 0. X9-X10,
            [ ,5]= 6 * 0. X11-X13,
            [ ,6]= 7 * 0. X14-X15;
   Matrix D1
            [1,1]= 2 * 1. X16 X17 X16 X17;
   Matrix D2
            [1,1]= 4 * 1. X18 X19;
   Matrix T
            [1,1]= 6 * 1.,
            [3,1]= 4 * 1.,
            [5,1]= 2 * 1.;
   Matrix D3
            [1,1]= 4 * 1. X18 X19;
   Matrix D4
            [1,1]= 2 * 1. X16 X17 X16 X17;
   Matrix P
            [1,1]= 2 * 1. X20-X23;
   Matrix U
            [1,1]= X24-X32,
            [4,1]= X33-X38,
            [7,1]= X39-X41;
   Bounds 0. <= X24-X32,
         -1. <= X16-X19 <= 1.;
   X20 = 1. - X16 * X16;
   X21 = 1. - X17 * X17;
   X22 = 1. - X18 * X18;
   X23 = 1. - X19 * X19;
run;
```

Because this formulation of Swaminathan's model in general leads to an unidentified problem, the results given here are different from those reported by McDonald (1980). The displayed output of PROC CALIS also indicates that the fitted central model matrices **P** and **U** are not positive definite. The BOUNDS statement constrains the diagonals of the matrices **P** and **U** to be nonnegative, but this cannot prevent **U** from having three negative eigenvalues. The fact that many of the published results for more complex models in covariance structure analysis are connected to unidentified problems implies that more theoretical work should be done to study the general features of such models.

References

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control*, 19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Al-Baali, M. and Fletcher, R. (1985), "Variational Methods for Nonlinear Least Squares," *J. Oper. Res. Soc.*, 36, 405–421.

Al-Baali, M. and Fletcher, R. (1986), "An Efficient Line Search for Nonlinear Least Squares," *J. Optimiz. Theory Appl.*, 48, 359–377.

Anderson, T.W. (1960), "Some Stochastic Process Models for Intelligence Test Scores," in *Mathematical Methods in the Social Sciences*, ed. K.J. Arrow, S. Karlin, and P. Suppes, Stanford: Stanford University Press.

Beale, E.M.L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F.A. Lootsma, London: Academic Press.

Bentler, P.M. (1983), "Some Contributions to Efficient Statistics in Structural Models: Specification and Estimation of Moment Structures," *Psychometrika*, 48, 493–517.

Bentler, P.M. (1985), *Theory and Implementation of EQS: A Structural Equations Program*, Manual for Program Version 2.0, Los Angeles: BMDP Statistical Software, Inc.

Bentler, P.M. (1986), *Lagrange Multiplier and Wald Tests for EQS and EQS/PC*, Los Angeles: BMDP Statistical Software, Inc.

Bentler, P.M. (1989), *EQS, Structural Equations, Program Manual*, Program Version 3.0, Los Angeles: BMDP Statistical Software, Inc.

Bentler, P.M. and Bonett, D.G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bentler, P.M. and Weeks, D.G. (1980), "Linear Structural Equations with Latent Variables," *Psychometrika*, 45, 289–308.

Bentler, P.M. and Weeks, D.G. (1982), "Multivariate Analysis with Latent Variables," in *Handbook of Statistics, Vol. 2*, ed. P.R. Krishnaiah and L.N. Kanal, Amsterdam: North Holland Publishing Company.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1977), *Discrete Multivariate Analysis: Theory and Practice,* Cambridge and London: MIT Press.

Bollen, K.A. (1986), “Sample Size and Bentler and Bonett’s Nonnormed Fit Index,” *Psychometrika*, 51, 375–377.

Bollen, K.A. (1989a), “A New Incremental Fit Index for General Structural Equation Models,” *Sociological Methods and Research*, 17, 303–316.

Bollen, K.A. (1989b), *Structural Equations with Latent Variables*, New York: John Wiley & Sons, Inc.

Bozdogan, H. (1987), “Model Selection and Akaike’s Information Criterion (AIC): The General Theory and its Analytical Extensions,” *Psychometrika*, 52, 345–370.

Browne, M.W. (1974), “Generalized Least Squares Estimators in the Analysis of Covariance Structures,” *South African Statistical Journal*, 8, 1–24.

Browne, M.W. (1982), “Covariance Structures,” in *Topics in Multivariate Analyses*, ed. D.M. Hawkins, New York: Cambridge University Press.

Browne, M. W. (1984), “Asymptotically Distribution-Free Methods for the Analysis of Covariance Structures,” *British Journal of Mathematical and Statistical Psychology*, 37, 62–83.

Browne, M. W. (1992), “Circumplex Models for Correlation Matrices,” Psychometrika, 57, 469–497.

Browne, M. W. and Cudeck, R. (1993), “Alternative Ways of Assessing Model Fit,” in *Testing Structural Equation Models*, ed. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications, Inc.

Browne, M. W. and Du Toit, S.H.C. (1992), “Automated Fitting of Nonstandard Models,” *Multivariate Behavioral Research*, 27, 269–300.

Browne, M.W. and Shapiro, A. (1986), “The Asymptotic Covariance Matrix of Sample Correlation Coefficients under General Conditions,” *Linear Algebra and its Applications*, 82, 169–176.

Bunch, J.R. and Kaufman, K. (1977), “Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems,” *Mathematics of Computation*, 31, 162–179.

Buse, A. (1982), “The Likelihood Ratio, Wald, and Lagrange Multiplier Tests: An Expository Note,” *The American Statistician*, 36, 153–157.

Chamberlain, R.M., Powell, M.J.D., Lemarechal, C., and Pedersen, H.C. (1982), “The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization,” *Mathematical Programming*, 16, 1–17.

Cramer, J.S. (1986), *Econometric Applications of Maximum Likelihood Methods*, Cambridge: Cambridge University Press.

DeLeeuw, J. (1983), “Models and Methods for the Analysis of Correlation Coefficients,” *Journal of Econometrics*, 22, 113–137.

Dennis, J.E., Gay, D.M., and Welsch, R.E. (1981), “An Adaptive Nonlinear Least-Squares Algorithm,” *ACM Trans. Math. Software*, 7, 348–368.

Dennis, J.E. and Mei, H.H.W. (1979), “Two New Unconstrained Optimization Algorithms which use Function and Gradient Values,” *J. Optim. Theory Appl.*, 28, 453–482.

Dennis, J.E. and Schnabel, R.B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, New Jersey: Prentice-Hall.

Dijkstra, T. K. (1992), “On Statistical Inference with Parameter Estimates on the Boundary of the Parameter Space,” *British Journal of Mathematical and Statistical Psychology*, 45, 289–309.

Everitt, B.S. (1984), *An Introduction to Latent Variable Methods*, London: Chapman & Hall.

Fletcher, R. (1980), *Practical Methods of Optimization*, Vol. 1, Chichester: John Wiley & Sons, Inc.

Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons, Inc.

Fletcher, R. and Powell, M.J.D. (1963), “A Rapidly Convergent Descent Method for Minimization,” *Comput.J.*, 6, 163–168.

Fletcher, R. and Xu, C. (1987), “Hybrid Methods for Nonlinear Least Squares,” *J. Numerical Analysis*, 7, 371–389.

Fuller, A.W. (1987), *Measurement Error Models*, New York: John Wiley & Sons, Inc.

Gallant, A. R. (1987), *Nonlinear Statistical Models*, New York: John Wiley & Sons, Inc.

Gay, D.M. (1983), “Subroutines for Unconstrained Minimization,” *ACM Trans. Math. Software*, 9, 503–524.

Gill, E.P., Murray, W., Saunders, M.A., and Wright, M.H. (1983), “Computing Forward-Difference Intervals for Numerical Optimization,” *SIAM J. Sci. Stat. Comput.*, 4, 310–321.

Gill, E.P., Murray, W., and Wright, M.H. (1981), *Practical Optimization*, London: Academic Press.

Gill, E.P., Murray, W., Saunders, M.A., and Wright, M.H. (1984), “Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints,” *ACM Trans. Math. Software*, 10, 282–298.

Goldfeld, S.M., Quandt, R.E., and Trotter, H.F. (1966), “Maximization by Quadratic Hill-Climbing,” *Econometrica*, 34, 541–551.

Guttman, L. (1953), “Image Theory for the Structure of Quantitative Variates,” *Psychometrika*, 18, 277–296.

Guttman, L. (1957), “Empirical Verification of the Radex Structure of Mental Abilities and Personality Traits,” *Educational and Psychological Measurement*, 17, 391–407.

Hägglund, G. (1982), “Factor Analysis by Instrumental Variable Methods,” *Psychometrika*, 47, 209–222.

Hartmann, W.M. (1992), *The NLP Procedure: Extended User’s Guide*, Cary: SAS Institute Inc.

Hartmann, W. M. and Hartwig, R. E. (1995), “Computing the Moore-Penrose Inverse for the Covariance Matrix in Constrained Nonlinear Estimation,” accepted for publication in *SIAM Journal on Optimization*.

Hartmann, W. M. and So, Y. (1995), “Nonlinear Least-Squares and Maximum-Likelihood Estimation Using PROC NLP and SAS/IML,” Computer Technology Workshop, American Statistical Association, Joint Statistical Meeting, Orlando, 1995.

Hoelter, J.W. (1983), “The Analysis of Covariance Structures: Goodness-of-Fit Indices,” *Sociological Methods and Research*, 11, 325–344.

James, L.R., Mulaik, S.A., and Brett, J.M. (1982), *Causal Analysis: Assumptions, Models, and Data*, Beverly Hills: Sage Publications, Inc.

Jennrich, R.I. (1987), “Tableau Algorithms for Factor Analysis by Instrumental Variable Methods,” *Psychometrika*, 52, 469–476.

Jöreskog, K.G. (1963), *Statistical Estimation in Factor Analysis*, Stockholm: Almqvist & Wicksell.

Jöreskog, K.G. (1969), “Efficient Estimation in Image Factor Analysis,” *Psychometrika*, 34, 51–75.

Jöreskog, K.G. (1973), “A General Method for Estimating a Linear Structural Equation System,” in *Structural Equation Models in the Social Sciences*, ed. A.S. Goldberger and O.D. Duncan, New York: Academic Press.

Jöreskog, K.G. (1978), “Structural Analysis of Covariance and Correlation Matrices,” *Psychometrika*, 43, 443–477.

Jöreskog, K.G. (1982), “Analysis of Covariance Structures,” in *A Second Generation of Multivariate Analysis*, ed. C. Fornell, New York: Praeger Publishers.

Jöreskog, K.G. and Sörbom, D. (1979), *Advances in Factor Analysis and Structural Equation Modeling*, Cambridge MA: Abt Books.

Jöreskog, K.G. and Sörbom, D. (1985), *LISREL VI; Analysis of Linear Structural Relationships by Maximum Likelihood, Instrumental Variables, and Least Squares*, Uppsala: University of Uppsala.

Jöreskog, K.G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago, Illinois: SPSS Inc.

Keesling, J.W. (1972), “Maximum Likelihood Approaches to Causal Analysis,” Ph.D. dissertation, Chicago, 1972.

Kmenta, J. (1971), *Elements of Econometrics*, New York: Macmillan Publishing Co.

Krane, W.R. and McDonald, R.P. (1978), “Scale Invariance and the Factor Analysis of Correlation Matrices,” *British Journal of Mathematical and Statistical Psychology*, 31, 218–228.

Lawley, D.N. and Maxwell, A.E. (1971), *Factor Analysis as a Statistical Method*, New York: American Elsevier Publishing Company.

Lee, S.Y. (1985), “On Testing Functional Constraints in Structural Equation Models,” *Biometrika*, 72, 125–131.

Lee, S.Y. and Jennrich, R.I. (1979), “A Study of Algorithms for Covariance Structure Analysis with Specific Comparisons Using Factor Analysis,” *Psychometrika*, 44, 99–113.

Loehlin, J.C. (1987), *Latent Variable Models, An Introduction to Factor, Path, and Structural Analysis*, Hillsdale, NJ: L. Erlbaum Associates.

Long, J.S. (1983), *Covariance Structure Models, an Introduction to LISREL*, Beverly Hills, CA: SAGE Publications, Inc.

MacCallum, R. (1986), “Specification Searches in Covariance Structure Modeling,” *Psychological Bulletin*, 100, 107–120.

Marsh, H.W., Balla, J.R. and McDonald, R.P. (1988), “Goodness-of-Fit Indices in Confirmatory Factor Analysis. The Effect of Sample Size,” *Psychological Bulletin*, 103, 391–410.

McArdle, J.J. (1980), “Causal Modeling Applied to Psychonomic Systems Simulation,” *Behavior Research Methods & Instrumentation*, 12, 193–209.

McArdle, J.J. (1988), “Dynamic but Structural Equation Modeling of Repeated Measures Data,” in *The Handbook of Multivariate Experimental Psychology*, ed. J.R. Nesselroade and R.B. Cattell, New York: Plenum Press.

McArdle, J.J. and Boker, S.M. (1986), *RAMpath - Path Diagram Software*, Denver: DATA Transforms, Inc.

McArdle, J.J. and McDonald, R.P. (1984), “Some Algebraic Properties of the Reticular Action Model,” *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R.P. (1978), “A Simple Comprehensive Model for the Analysis of Covariance Structures,” *British Journal of Mathematical and Statistical Psychology*, 31, 59–72.

McDonald, R.P. (1980), “A Simple Comprehensive Model for the Analysis of Covariance Structures: Some Remarks on Applications,” *British Journal of Mathematical and Statistical Psychology*, 33, 161–183.

McDonald, R.P. (1984), “Confirmatory Models for Nonlinear Structural Analysis,” in *Data Analysis and Informatics*, III, ed. E. Diday et al., North Holland: Elsevier Publishers.

McDonald, R.P. (1985), *Factor Analysis and Related Methods*, Hillsdale, NJ: Lawrence Erlbaum Associates.

McDonald, R.P. (1989), “An Index of Goodness-of-Fit Based on Noncentrality,” *Journal of Classification*, 6, 97–103.

McDonald, R.P. and Hartmann, W. (1992), “A Procedure for Obtaining Initial Values of Parameters in the RAM Model,” *Multivariate Behavioral Research*, 27, 57–176.

McDonald, R.P. and Marsh, H.W. (1988), “Choosing a Multivariate Model: Noncentrality and Goodness of Fit,” distributed paper.

McDonald, R.P., Parker, P.M., and Ishizuka, T. (1993), “A Scale-Invariant Treatment of Recursive Path Models,” *Psychometrika*, 58, 431–443.

Moré, J.J. (1978), “The Levenberg-Marquardt Algorithm: Implementation and Theory,” in *Numerical Analysis*−*Dundee 1977*, ed. G.A. Watson, Lecture Notes in Mathematics 630, Berlin: Springer-Verlag.

Moré, J.J. and Sorensen, D.C. (1983), “Computing a Trust-Region Step,” *SIAM J. Sci. Stat. Comput.*, 4, 553–572.

Mulaik, S.A., James, L.R., Van Alstine, J., Bennett, N., Lind, S. and Stilwell, C.D. (1989), “Evaluation of Goodness-of-Fit Indices for Structural Equation Models,” *Psychological Bulletin*, 105, 430–445.

Muthèn, B.O. (1987), LISCOMP: *Analysis of Linear Structural Relations Using a Comprehensive Measurement Model*, Mooresville IN: Scientific Software, Inc.

Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press, Inc.

Powell, J.M.D. (1977), “Restart Procedures for the Conjugate Gradient Method,” *Math. Prog.*, 12, 241–254.

Powell, J.M.D. (1978a), “A Fast Algorithm for Nonlinearly Constraint Optimization Calculations,” in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin: Springer Verlag, 144–175.

Powell, J.M.D. (1978b), “Algorithms for Nonlinear Constraints that Use Lagrangian Functions,” *Mathematical Programming*, 14, 224–248.

Powell, M.J.D. (1982a), “Extensions to Subroutine VF02AD,” in *Systems Modeling and Optimization, Lecture Notes In Control and Information Sciences 38*, ed. R.F. Drenick and F. Kozin, Berlin: Springer Verlag, 529–538.

Powell, J.M.D. (1982b), “VMCWD: A Fortran Subroutine for Constrained Optimization,” *DAMTP 1982/NA4*, Cambridge, England.

Powell, J.M.D. (1992), “A Direct Search Optimization Method that Models the Objective and Constraint Functions by Linear Interpolation,” *DAMTP/NA5*, Cambridge, England.

Saris, W. (1982), “Linear Structural Relationships,” in *A Second Generation of Multivariate Analysis*, ed. C. Fornell, New York: Praeger Publishers.

Schmid, J. and Leiman, J.M. (1957), “The Development of Hierarchical Factor Solutions,” *Psychometrika*, 22, 53–61.

Schwarz, G. (1978), “Estimating the Dimension of a Model,” *Annals of Statistics*, 6, 461–464.

Sclove, L.S. (1987), “Application of Model-Selection Criteria to Some Problems in Multivariate Analysis,” *Psychometrika*, 52, 333–343.

Steiger, J.H. and Lind, J.C. (1980), “Statistically Based Tests for the Number of Common Factors,” paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Swaminathan, H. (1974), “A General Factor Model for the Description of Change,” Report LR-74-9, Laboratory of Psychometric and Evaluative Research, University of Massachusetts.

Tucker, L.R. and Lewis, C. (1973), “A Reliability Coefficient for Maximum Likelihood Factor Analysis,” *Psychometrika*, 38, 1–10.

Wheaton, B., Muthèn, B., Alwin, D.F., and Summers, G.F. (1977), “Assessing Reliability and Stability in Panel Models,” in *Sociological Methodology*, ed. D.R. Heise, San Francisco: Jossey Bass.

Wiley, D.E. (1973), “The Identification Problem for Structural Equation Models with Unmeasured Variables,” in *Structural Equation Models in the Social Sciences*, ed. A.S. Goldberger and O.D. Duncan, New York: Academic Press.

Wilson, E.B. and Hilferty, M.M. (1931), “The Distribution of Chi-Square,” *Proceeding of the National Academy of Science*, 17, 694.

Chapter 20
The CANCORR Procedure

Chapter Contents

Chapter 20
The CANCORR Procedure

Overview

The CANCORR procedure performs canonical correlation, partial canonical correlation, and canonical redundancy analysis.

Canonical correlation is a generalization of multiple correlation for analyzing the relationship between two sets of variables. In multiple correlation, you examine the relationship between a linear combination of a set of explanatory variables, $\mathbf{X}$, and a *single* response variable, $\mathbf{Y}$. In canonical correlation, you examine the relationship between linear combinations of the set of $\mathbf{X}$ variables and linear combinations of a *set* of $\mathbf{Y}$ variables. These linear combinations are called *canonical variables* or *canonical variates*. Either set of variables can be considered explanatory or response variables, since the statistical model is symmetric in the two sets of variables. Simple and multiple correlation are special cases of canonical correlation in which one or both sets contain a single variable.

The CANCORR procedure tests a series of hypotheses that each canonical correlation and all smaller canonical correlations are zero in the population. PROC CANCORR uses an F approximation (Rao 1973; Kshirsagar 1972) that gives better small sample results than the usual χ^2 approximation. At least one of the two sets of variables should have an approximate multivariate normal distribution in order for the probability levels to be valid.

Both standardized and unstandardized canonical coefficients are computed, as well as the four *canonical structure* matrices showing correlations between the two sets of canonical variables and the two sets of original variables. A canonical redundancy analysis (Stewart and Love 1968; Cooley and Lohnes 1971) can also be done. PROC CANCORR provides multiple regression analysis options to aid in interpreting the canonical correlation analysis. You can examine the linear regression of each variable on the opposite set of variables.

PROC CANCORR can produce a data set containing the scores of each observation on each canonical variable, and you can use the PRINT procedure to list these values. A plot of each canonical variable against its counterpart in the other group is often useful, and you can use PROC PLOT with the output data set to produce these plots. A second output data set contains the canonical correlations, coefficients, and most other statistics computed by the procedure.

Background

Canonical correlation was developed by Hotelling (1935, 1936). The application of canonical correlation is discussed by Cooley and Lohnes (1971), Tatsuoka (1971), and Mardia, Kent, and Bibby (1979). One of the best theoretical treatments is given by Kshirsagar (1972).

Given a set of p $\mathbf{X}$ variables and q $\mathbf{Y}$ variables, the CANCORR procedure finds the linear combinations

$$w_1 = a_{11}x_1 + a_{21}x_2 + \cdots + a_{p1}x_p$$

$$v_1 = b_{11}y_1 + b_{21}y_2 + \cdots + b_{q1}y_q$$

such that the two canonical variables, w_1 and v_1, have the largest possible correlation. This maximized correlation between the two canonical variables is the first canonical correlation. The coefficients of the linear combinations are canonical coefficients or canonical weights. It is customary to normalize the canonical coefficients so that each canonical variable has a variance of 1.

PROC CANCORR continues by finding a second set of canonical variables, uncorrelated with the first pair, that produces the second highest correlation coefficient. That is, the second pair of canonical variables is:

$$w_2 = a_{12}x_1 + a_{22}x_2 + \cdots + a_{p2}x_p$$

$$v_2 = b_{12}y_1 + b_{22}y_2 + \cdots + b_{q2}y_q$$

such that w_2 is uncorrelated with w_1 and v_1, v_2 is uncorrelated with w_1 and v_1, and w_2 and v_2 have the largest possible correlation subject to these constraints. The process of constructing canonical variables continues until the number of pairs of canonical variables is $\min(p, q)$, the number of variables in the smaller group.

Each canonical variable is uncorrelated with all the other canonical variables of either set except for the one corresponding canonical variable in the opposite set. The canonical coefficients are not generally orthogonal, however, so the canonical variables do not represent jointly perpendicular directions through the space of the original variables.

The first canonical correlation is at least as large as the multiple correlation between any variable and the opposite set of variables. It is possible for the first canonical correlation to be very large while all the multiple correlations for predicting one of the original variables from the opposite set of canonical variables are small. Canonical redundancy analysis (Stewart and Love 1968; Cooley and Lohnes 1971; van den Wollenberg 1977), examines how well the original variables can be predicted from the canonical variables.

PROC CANCORR can also perform partial canonical correlation, which is a multivariate generalization of ordinary partial correlation (Cooley and Lohnes 1971; Timm 1975). Most commonly-used parametric statistical methods, ranging from t tests to multivariate analysis of covariance, are special cases of partial canonical correlation.

Getting Started

The following example demonstrates how you can use the CANCORR procedure to calculate and test canonical correlations between two sets of variables.

Suppose you want to determine the degree of correspondence between a set of job characteristics and measures of employee satisfaction. Using a survey instrument for employees, you calculate three measures of job satisfaction. With another instrument designed for supervisors, you calculate the corresponding job characteristics profile.

Your three variables associated with job satisfaction are

- career track satisfaction: employee satisfaction with career direction and the possibility of future advancement, expressed as a percent
- management and supervisor satisfaction: employee satisfaction with supervisor's communication and management style, expressed as a percent
- financial satisfaction: employee satisfaction with salary and other benefits, using a scale measurement from 1 to 10 (1=unsatisfied, 10=satisfied)

The three variables associated with job characteristics are

- task variety: degree of variety involved in tasks, expressed as a percent
- feedback: degree of feedback required in job tasks, expressed as a percent
- autonomy: degree of autonomy required in job tasks, expressed as a percent

The following statements create the SAS data set Jobs and request a canonical correlation analysis:

```
options ls=120;
data Jobs;
   input Career Supervisor Finance Variety Feedback Autonomy;
   label
     Career    ='Career Satisfaction' Variety ='Task Variety'
     Supervisor='Supervisor Satisfaction' Feedback='Amount of Feedback'
     Finance   ='Financial Satisfaction' Autonomy='Degree of Autonomy';
   datalines;
72  26  9         10  11  70
63  76  7         85  22  93
96  31  7         83  63  73
96  98  6         82  75  97
84  94  6         36  77  97
66  10  5         28  24  75
31  40  9         64  23  75
45  14  2         19  15  50
42  18  6         33  13  70
79  74  4         23  14  90
39  12  2         37  13  70
54  35  3         23  74  53
60  75  5         45  58  83
63  45  5         22  67  53
;
```

```
proc cancorr data=Jobs
  vprefix=Satisfaction wprefix=Characteristics
  vname='Satisfaction Areas' wname='Job Characteristics';
  var  Career Supervisor Finance;
  with Variety Feedback Autonomy;
run;
```

The DATA= option in the PROC CANCORR statement specifies Jobs as the SAS data set to be analyzed. The VPREFIX and WPREFIX options specify the prefixes for naming the canonical variables from the VAR statement and the WITH statement, respectively. The VNAME option specifies 'Satisfaction Areas' to refer to the set of variables from the VAR statement. Similarly, the WNAME option specifies 'Job Characteristics' to refer to the set of variables from the WITH statement.

The VAR statement defines the first of the two sets of variables to be analyzed as Career, Supervisor and Finance. The WITH statement defines the second set of variables to be Variety, Feedback, and Autonomy. The results of this analysis are displayed in Figure 20.1 to Figure 20.4.

```
                                  The CANCORR Procedure

                              Canonical Correlation Analysis

                                           Adjusted    Approximate        Squared
                            Canonical     Canonical       Standard      Canonical
                          Correlation   Correlation          Error    Correlation

                      1      0.919412      0.898444       0.042901       0.845318
                      2      0.418649      0.276633       0.228740       0.175267
                      3      0.113366       .             0.273786       0.012852

                                                      Test of H0: The canonical correlations in the
                 Eigenvalues of Inv(E)*H                current row and all that follow are zero
                   = CanRsq/(1-CanRsq)
                                                    Likelihood   Approximate
     Eigenvalue   Difference   Proportion   Cumulative        Ratio       F Value   Num DF    Den DF   Pr > F

 1       5.4649       5.2524       0.9604       0.9604   0.12593148          2.93        9    19.621   0.0223
 2       0.2125       0.1995       0.0373       0.9977   0.81413359          0.49        4        18   0.7450
 3       0.0130                    0.0023       1.0000   0.98714819          0.13        1        10   0.7257
```

Figure 20.1. Canonical Correlations, Eigenvalues, and Likelihood Tests

Figure 20.1 displays the canonical correlation, adjusted canonical correlation, approximate standard error, and squared canonical correlation for each pair of canonical variables. The first canonical correlation (the correlation between the first pair of canonical variables) is 0.9194. This value represents the highest possible correlation between any linear combination of the job satisfaction variables and any linear combination of the job characteristics variables.

Figure 20.1 also lists the likelihood ratio and associated statistics for testing the hypothesis that the canonical correlations in the current row and all that follow are zero.

The first approximate F value of 2.93 corresponds to the test that all three canonical correlations are zero. Since the p-value is small (0.0223), you would reject the null hypothesis at the 0.05 level. The second approximate F value of 0.49 corresponds to the test that both the second and the third canonical correlations are zero. Since the

p-value is large (0.7450), you would fail to reject the hypothesis and conclude that only the first canonical correlation is significant.

Figure 20.2 lists several multivariate statistics and F test approximations for the null hypothesis that all canonical correlations are zero. These statistics are described in the section "Multivariate Tests" in Chapter 2, "Introduction to Regression Procedures."

```
                                 The CANCORR Procedure

                              Canonical Correlation Analysis

                       Multivariate Statistics and F Approximations

                                 S=3    M=-0.5    N=3

      Statistic                        Value    F Value    Num DF    Den DF    Pr > F

      Wilks' Lambda               0.12593148       2.93         9    19.621    0.0223
      Pillai's Trace              1.03343732       1.75         9        30    0.1204
      Hotelling-Lawley Trace      5.69042615       4.76         9    9.8113    0.0119
      Roy's Greatest Root         5.46489324      18.22         3        10    0.0002

               NOTE: F Statistic for Roy's Greatest Root is an upper bound.
```

Figure 20.2. Multivariate Statistics and Approximate F Tests

The small p-values for these tests (< 0.05), except for Pillai's Trace, suggest rejecting the null hypothesis that all canonical correlations are zero in the population, confirming the results of the preceding likelihood ratio test (Figure 20.1). With only one of the tests resulting in a p-value larger than 0.05, you can assume that the first canonical correlation is significant. The next step is to interpret or identify the two canonical variables corresponding to this significant correlation.

Even though canonical variables are artificial, they can often be "identified" in terms of the original variables. This is done primarily by inspecting the standardized coefficients of the canonical variables and the correlations between the canonical variables and their original variables. Since only the first canonical correlation is significant, only the first pair of canonical variables (Satisfaction1 and Characteristics1) need to be identified.

PROC CANCORR calculates and displays the raw canonical coefficients for the job satisfaction variables and the job characteristic variables. However, since the original variables do not necessarily have equal variance and are not measured in the same units, the raw coefficients must be standardized to allow interpretation. The coefficients are standardized by multiplying the raw coefficients with the standard deviation of the associated variable.

The standardized canonical coefficients in Figure 20.3 show that the first canonical variable for the Satisfaction group is a weighted sum of the variables Supervisor (0.7854) and Career (0.3028), with the emphasis on Supervisor. The coefficient for the variable Finance is near 0. Thus, a person satisfied with his or her supervisor and with a large degree of career satisfaction would score high on the canonical variable Satisfaction1.

```
                                   The CANCORR Procedure

                               Canonical Correlation Analysis

                Standardized Canonical Coefficients for the Satisfaction Areas

                                                    Satisfaction1      Satisfaction2      Satisfaction3

       Career         Career Satisfaction                  0.3028            -0.5416             1.0408
       Supervisor     Supervisor Satisfaction              0.7854             0.1305            -0.9085
       Finance        Financial Satisfaction               0.0538             0.9754             0.3329

               Standardized Canonical Coefficients for the Job Characteristics

                                            Characteristics1      Characteristics2      Characteristics3

      Variety       Task Variety                    -0.1108                0.8095                0.9071
      Feedback      Amount of Feedback               0.5520               -0.7722                0.4194
      Autonomy      Degree of Autonomy               0.8403                0.1020               -0.8297
```

Figure 20.3. Standardized Canonical Coefficients from the CANCORR Procedure

The coefficients for the job characteristics variables show that degree of autonomy (Autonomy) and amount of feedback (Feedback) contribute heavily to the Characteristics1 canonical variable (0.8403 and 0.5520, respectively).

Figure 20.4 shows the table of correlations between the canonical variables and the original variables.

```
                                   The CANCORR Procedure

                                   Canonical Structure

          Correlations Between the Satisfaction Areas and Their Canonical Variables

                                                    Satisfaction1      Satisfaction2      Satisfaction3

       Career         Career Satisfaction                  0.7499            -0.2503             0.6123
       Supervisor     Supervisor Satisfaction              0.9644             0.0362            -0.2618
       Finance        Financial Satisfaction               0.2873             0.8814             0.3750

          Correlations Between the Job Characteristics and Their Canonical Variables

                                            Characteristics1      Characteristics2      Characteristics3

      Variety       Task Variety                     0.4863                0.6592                0.5736
      Feedback      Amount of Feedback               0.6216               -0.5452                0.5625
      Autonomy      Degree of Autonomy               0.8459                0.4451               -0.2938

   Correlations Between the Satisfaction Areas and the Canonical Variables of the Job Characteristics

                                               Characteristics1      Characteristics2      Characteristics3

  Career         Career Satisfaction                     0.6895               -0.1048                0.0694
  Supervisor     Supervisor Satisfaction                 0.8867                0.0152               -0.0297
  Finance        Financial Satisfaction                  0.2642                0.3690                0.0425

    Correlations Between the Job Characteristics and the Canonical Variables of the Satisfaction Areas

                                                  Satisfaction1      Satisfaction2      Satisfaction3

          Variety       Task Variety                     0.4471             0.2760             0.0650
          Feedback      Amount of Feedback               0.5715            -0.2283             0.0638
          Autonomy      Degree of Autonomy               0.7777             0.1863            -0.0333
```

Figure 20.4. Canonical Structure Correlations from the CANCORR Procedure

Although these univariate correlations must be interpreted with caution since they do not indicate how the original variables contribute *jointly* to the canonical analysis, they are often useful in the identification of the canonical variables.

Figure 20.4 shows that the supervisor satisfaction variable Supervisor is strongly associated with the Satisfaction1 canonical variable with a correlation of 0.9644. Slightly less influential is the variable Career, which has a correlation with the canonical variable of 0.7499. Thus, the canonical variable Satisfaction1 seems to represent satisfaction with supervisor and career track.

The correlations for the job characteristics variables show that the canonical variable Characteristics1 seems to represent all three measured variables, with degree of autonomy variable (Autonomy) being the most influential (0.8459).

Hence, you can interpret these results to mean that job characteristics and job satisfaction are related—jobs that possess a high degree of autonomy and level of feedback are associated with workers who are more satisfied with their supervisor and their career. While financial satisfaction is a factor in job satisfaction, it is not as important as the other measured satisfaction-related variables.

Syntax

The following statements are available in PROC CANCORR.

PROC CANCORR < *options* > **;**
  WITH *variables* **;**
  BY *variables* **;**
  FREQ *variable* **;**
  PARTIAL *variables* **;**
  VAR *variables* **;**
  WEIGHT *variable* **;**

The PROC CANCORR statement and the WITH statement are required. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC CANCORR statement. The remaining statements are covered in alphabetical order.

PROC CANCORR Statement

PROC CANCORR < *options* > **;**

The PROC CANCORR statement starts the CANCORR procedure and optionally identifies input and output data sets, specifies the analyses performed, and controls displayed output. Table 20.1 summarizes the options.

Table 20.1. PROC CANCORR Statement Options

Task	Options	Description
Specify computational details	EDF=	specify error degrees of freedom if input observations are regression residuals
	NOINT	omit intercept from canonical correlation and regression models
	RDF=	specify regression degrees of freedom if input observations are regression residuals
	SINGULAR=	specify the singularity criterion
Specify input and output data sets	DATA=	specify input data set name
	OUT=	specify output data set name
	OUTSTAT=	specify output data set name containing various statistics
Specify labeling options	VNAME=	specify a name to refer to VAR statement variables
	VPREFIX=	specify a prefix for naming VAR statement canonical variables
	WNAME=	specify a name to refer to WITH statement variables
	WPREFIX=	specify a prefix for naming WITH statement canonical variables
Control amount of output	ALL	produce simple statistics, input variable correlations, and canonical redundancy analysis
	CORR	produce input variable correlations
	NCAN=	specify number of canonical variables for which full output is desired
	NOPRINT	suppress all displayed output
	REDUNDANCY	produce canonical redundancy analysis
	SHORT	suppress default output from canonical analysis
	SIMPLE	produce means and standard deviations
Request regression analyses	VDEP	request multiple regression analyses with the VAR variables as dependents and the WITH variables as regressors
	VREG	request multiple regression analyses with the VAR variables as regressors and the WITH variables as dependents
	WDEP	same as VREG
	WREG	same as VDEP

Table 20.1. (continued)

Task	Options	Description
Specify regression statistics	ALL	produce all regression statistics and includes these statistics in the OUTSTAT= data set
	B	produce raw regression coefficients
	CLB	produce 95% confidence interval limits for the regression coefficients
	CORRB	produce correlations among regression coefficients
	INT	request statistics for the intercept when you specify the B, CLB, SEB, T, or PROBT option
	PCORR	display partial correlations between regressors and dependents
	PROBT	display probability levels for t statistics
	SEB	display standard errors of regression coefficients
	SMC	display squared multiple correlations and F tests
	SPCORR	display semipartial correlations between regressors and dependents
	SQPCORR	display squared partial correlations between regressors and dependents
	SQSPCORR	display squared semipartial correlations between regressors and dependents
	STB	display standardized regression coefficients
	T	display t statistics for regression coefficients

Following are explanations of the options that can be used in the PROC CANCORR statement (in alphabetic order):

ALL

displays simple statistics, correlations among the input variables, the confidence limits for the regression coefficients, and the canonical redundancy analysis. If you specify the VDEP or WDEP option, the ALL option displays all related regression statistics (unless the NOPRINT option is specified) and includes these statistics in the OUTSTAT= data set.

B

produces raw regression coefficients from the regression analyses.

CLB

produces the 95% confidence limits for the regression coefficients from the regression analyses.

CORR

C

produces correlations among the original variables. If you include a PARTIAL statement, the CORR option produces a correlation matrix for all variables in the analysis, the regression statistics (R^2, RMSE), the standardized regression coefficients for both the VAR and WITH variables as predicted from the PARTIAL statement variables, and partial correlation matrices.

CORRB

produces correlations among the regression coefficient estimates.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC CANCORR. It can be an ordinary SAS data set or a TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV data set. By default, the procedure uses the most recently created SAS data set.

EDF=*error-df*

specifies the error degrees of freedom if the input observations are residuals from a regression analysis. The effective number of observations is the EDF= value plus one. If you have 100 observations, then specifying EDF=99 has the same effect as omitting the EDF= option.

INT

requests that statistics for the intercept be included when B, CLB, SEB, T, or PROBT is specified for the regression analyses.

NCAN=*number*

specifies the number of canonical variables for which full output is desired. The *number* must be less than or equal to the number of canonical variables in the analysis.

The value of the NCAN= option specifies the number of canonical variables for which canonical coefficients and canonical redundancy statistics are displayed, and the number of variables shown in the canonical structure matrices. The NCAN= option does not affect the number of displayed canonical correlations.

If an OUTSTAT= data set is requested, the NCAN= option controls the number of canonical variables for which statistics are output. If an OUT= data set is requested, the NCAN= option controls the number of canonical variables for which scores are output.

NOINT

omits the intercept from the canonical correlation and regression models. Standard deviations, variances, covariances, and correlations are not corrected for the mean. If you use a TYPE=SSCP data set as input to the CANCORR procedure and list the variable Intercept in the VAR or WITH statement, the procedure runs as if you also specified the NOINT option. If you use NOINT and also create an OUTSTAT= data set, the data set is TYPE=UCORR.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set to contain all the original data plus scores on the canonical variables. If you want to create a permanent SAS data set, you must specify a two-level name. The OUT= option cannot be used when the DATA= data set is TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV. For details on OUT= data sets, see the section "Output Data Sets" on page 766. Refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets.

OUTSTAT=*SAS-data-set*

creates an output SAS data set containing various statistics, including the canonical correlations and coefficients and the multiple regression statistics you request. If you want to create a permanent SAS data set, you must specify a two-level name. For details on OUTSTAT= data sets, see the section "Output Data Sets" on page 766. Refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets.

PCORR

produces partial correlations between regressors and dependent variables, removing from each dependent variable and regressor the effects of all other regressors.

PROBT

produces probability levels for the t statistics in the regression analyses.

RDF=*regression-df*

specifies the regression degrees of freedom if the input observations are residuals from a regression analysis. The effective number of observations is the actual number minus the RDF= value. The degrees of freedom for the intercept should not be included in the RDF= option.

REDUNDANCY
RED

produces canonical redundancy statistics.

SEB

produces standard errors of the regression coefficients.

SHORT

suppresses all default output from the canonical analysis except the tables of canonical correlations and multivariate statistics.

SIMPLE
S

produces means and standard deviations.

SINGULAR=*p*
SING=*p*

specifies the singularity criterion, where $0 < p < 1$. If a variable in the PARTIAL statement has an R^2 as large as $1-p$ (where p is the value of the SINGULAR= option) when predicted from the variables listed before it in the statement, the variable is assigned a standardized regression coefficient of 0, and the LOG generates a linear dependency warning message. By default, SINGULAR=1E$-$8.

SMC

produces squared multiple correlations and F tests for the regression analyses.

SPCORR

produces semipartial correlations between regressors and dependent variables, removing from each regressor the effects of all other regressors.

SQPCORR

produces squared partial correlations between regressors and dependent variables, removing from each dependent variable and regressor the effects of all other regressors.

SQSPCORR

produces squared semipartial correlations between regressors and dependent variables, removing from each regressor the effects of all other regressors.

STB

produces standardized regression coefficients.

T

produces t statistics for the regression coefficients.

VDEP
WREG

requests multiple regression analyses with the VAR variables as dependent variables and the WITH variables as regressors.

VNAME=*'label'*
VN=*'label'*

specifies a character constant to refer to variables from the VAR statement on the output. Enclose the constant in single quotes. If you omit the VNAME= option, these variables are referred to as the VAR Variables. The number of characters in the label should not exceed the label length defined by the VALIDVARNAME= system option. For more information on the VALIDVARNAME= system option, refer to *SAS Language Reference: Dictionary*.

VPREFIX=*name*
VP=*name*

specifies a prefix for naming canonical variables from the VAR statement. By default, these canonical variables are given the names V1, V2, and so on. If you specify VPREFIX=ABC, the names are ABC1, ABC2, and so forth. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the name length defined by the VALIDVARNAME= system option. For more information on the VALIDVARNAME= system option, refer to *SAS Language Reference: Dictionary*.

WDEP
VREG

requests multiple regression analyses with the WITH variables as dependent variables and the VAR variables as regressors.

WNAME=*'label'*
WN=*'label'*

specifies a character constant to refer to variables in the WITH statement on the output. Enclose the constant in quotes. If you omit the WNAME= option, these variables are referred to as the WITH Variables. The number of characters in the label should not exceed the label length defined by the VALIDVARNAME= system option. For more information, on the VALIDVARNAME= system option, refer to *SAS Language Reference: Dictionary*.

WPREFIX=*name*
WP=*name*

specifies a prefix for naming canonical variables from the WITH statement. By default, these canonical variables are given the names W1, W2, and so on. If you specify WPREFIX=XYZ, then the names are XYZ1, XYZ2, and so forth. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the label length defined by the VALIDVARNAME= system option. For more information, on the VALIDVARNAME= system option, refer to *SAS Language Reference: Dictionary*.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CANCORR to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CANCORR procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC CANCORR then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC CANCORR calculates significance probabilities.

PARTIAL Statement

PARTIAL *variables* **;**

You can use the PARTIAL statement to base the canonical analysis on partial correlations. The variables in the PARTIAL statement are partialled out of the VAR and WITH variables.

VAR Statement

VAR *variables* **;**

The VAR statement lists the variables in the first of the two sets of variables to be analyzed. The variables must be numeric. If you omit the VAR statement, all numeric variables not mentioned in other statements make up the first set of variables. If, however, the DATA= data set is TYPE=SSCP, the default set of variables used as VAR variables does not include the variable Intercept.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to compute weighted product-moment correlation coefficients, specify the name of the weighting variable in a WEIGHT statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the degrees of freedom or number of observations. An observation is used in the analysis only if the WEIGHT variable is greater than zero.

WITH Statement

WITH *variables* **;**

The WITH statement lists the variables in the second set of variables to be analyzed. The variables must be numeric. The WITH statement is required.

Details

Missing Values

If an observation has a missing value for any of the variables in the analysis, that observation is omitted from the analysis.

Formulas

Assume without loss of generality that the two sets of variables, $\mathbf{X}$ with p variables and $\mathbf{Y}$ with q variables, have means of zero. Let n be the number of observations, and m be $n-1$.

Note that the scales of eigenvectors and canonical coefficients are arbitrary. PROC CANCORR follows the usual procedure of rescaling the canonical coefficients so that each canonical variable has a variance of one.

There are several different sets of formulas that can be used to compute the canonical correlations, ρ_i, $i = 1, \ldots, \min(p, q)$, and unscaled canonical coefficients:

1. Let $\mathbf{S}_{XX} = \mathbf{X}'\mathbf{X}/m$ be the covariance matrix of $\mathbf{X}$, $\mathbf{S}_{YY} = \mathbf{Y}'\mathbf{Y}/m$ be the covariance matrix of $\mathbf{Y}$, and $\mathbf{S}_{XY} = \mathbf{X}'\mathbf{Y}/m$ be the covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$. Then the eigenvalues of $\mathbf{S}_{YY}^{-1}\mathbf{S}'_{XY}\mathbf{S}_{XX}^{-1}\mathbf{S}_{XY}$ are the squared canonical correlations, and the right eigenvectors are raw canonical coefficients for the $\mathbf{Y}$ variables. The eigenvalues of $\mathbf{S}_{XX}^{-1}\mathbf{S}_{XY}\mathbf{S}_{YY}^{-1}\mathbf{S}'_{XY}$ are the squared canonical correlations, and the right eigenvectors are raw canonical coefficients for the $\mathbf{X}$ variables.
2. Let $\mathbf{T} = \mathbf{Y}'\mathbf{Y}$ and $\mathbf{H} = \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. The eigenvalues ξ_i of $\mathbf{T}^{-1}\mathbf{H}$ are the squared canonical correlations, ρ_i^2, and the right eigenvectors are raw canonical coefficients for the $\mathbf{Y}$ variables. Interchange $\mathbf{X}$ and $\mathbf{Y}$ in the above formulas, and the eigenvalues remain the same, but the right eigenvectors are raw canonical coefficients for the $\mathbf{X}$ variables.
3. Let $\mathbf{E} = \mathbf{T} - \mathbf{H}$. The eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ are $\lambda_i = 1/(1-\rho_i^2)$. The right eigenvectors of $\mathbf{E}^{-1}\mathbf{H}$ are the same as the right eigenvectors of $\mathbf{T}^{-1}\mathbf{H}$.
4. Canonical correlation can be viewed as a principal component analysis of the predicted values of one set of variables from a regression on the other set of variables, in the metric of the error covariance matrix. For example, regress the $\mathbf{Y}$ variables on the $\mathbf{X}$ variables. Call the predicted values $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ and the residuals $\mathbf{R} = \mathbf{Y} - \mathbf{P} = (\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Y}$. The error covariance matrix is $\mathbf{R}'\mathbf{R}/m$. Choose a transformation $\mathbf{Q}$ that converts the error covariance matrix to an identity, that is, $(\mathbf{RQ})'(\mathbf{RQ}) = \mathbf{Q}'\mathbf{R}'\mathbf{RQ} = m\mathbf{I}$. Apply the same transformation to the predicted values to yield, say, $\mathbf{Z} = \mathbf{PQ}$. Now do a principal component analysis on the covariance matrix of $\mathbf{Z}$, and you get the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Repeat with $\mathbf{X}$ and $\mathbf{Y}$ variables interchanged, and you get the same eigenvalues.

   To show this relationship between canonical correlation and principal components, note that $\mathbf{P}'\mathbf{P} = \mathbf{H}$, $\mathbf{R}'\mathbf{R} = \mathbf{E}$, and $\mathbf{QQ}' = m\mathbf{E}^{-1}$. Let the covariance matrix of $\mathbf{Z}$ be $\mathbf{G}$. Then $\mathbf{G} = \mathbf{Z}'\mathbf{Z}/m = (\mathbf{PQ})'\mathbf{PQ}/m = \mathbf{Q}'\mathbf{P}'\mathbf{PQ}/m = \mathbf{Q}'\mathbf{HQ}/m$. Let $\mathbf{u}$ be an eigenvector of $\mathbf{G}$ and κ be the corresponding eigenvalue. Then by definition, $\mathbf{Gu} = \kappa\mathbf{u}$, hence $\mathbf{Q}'\mathbf{HQu}/m = \kappa\mathbf{u}$. Premultiplying both sides by $\mathbf{Q}$ yields $\mathbf{QQ}'\mathbf{HQu}/m = \kappa\mathbf{Qu}$ and thus $\mathbf{E}^{-1}\mathbf{HQu} = \kappa\mathbf{Qu}$. Hence $\mathbf{Qu}$ is an eigenvector of $\mathbf{E}^{-1}\mathbf{H}$ and κ is also an eigenvalue of $\mathbf{E}^{-1}\mathbf{H}$.

5. If the covariance matrices are replaced by correlation matrices, the formulas above yield standardized canonical coefficients instead of raw canonical coefficients.

The formulas for multivariate test statistics are shown in “Multivariate Tests” in Chapter 2, “Introduction to Regression Procedures.” Formulas for linear regression are provided in other sections of that chapter.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. The number of new variables is twice that specified by the NCAN= option. The names of the new variables are formed by concatenating the values given by the VPREFIX= and WPREFIX= options (the defaults are V and W) with the numbers 1, 2, 3, and so on. The new variables have mean 0 and variance equal to 1. An OUT= data set cannot be created if the DATA= data set is TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV or if a PARTIAL statement is used.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR or TYPE=UCORR data set produced by the CORR procedure, but it contains several results in addition to those produced by PROC CORR.

The new data set contains the following variables:

- the BY variables, if any
- two new character variables, _TYPE_ and _NAME_
- Intercept, if the INT option is used
- the variables analyzed (those in the VAR statement and the WITH statement)

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

TYPE	
MEAN	means
STD	standard deviations

USTD uncorrected standard deviations. When you specify the NOINT option in the PROC CANCORR statement, the OUTSTAT= data set contains standard deviations not corrected for the mean (_TYPE_='USTD').

N number of observations on which the analysis is based. This value is the same for each variable.

SUMWGT sum of the weights if a WEIGHT statement is used. This value is the same for each variable.

CORR correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the correlation matrix.

UCORR uncorrected correlation matrix. When you specify the NOINT option in the PROC CANCORR statement, the OUTSTAT= data set contains a matrix of correlations not corrected for the means.

CORRB correlations among the regression coefficient estimates

STB standardized regression coefficients. The _NAME_ variable contains the name of the dependent variable.

B raw regression coefficients

SEB standard errors of the regression coefficients

LCLB 95% lower confidence limits for the regression coefficients

UCLB 95% upper confidence limits for the regression coefficients

T t statistics for the regression coefficients

PROBT probability levels for the t statistics

SPCORR semipartial correlations between regressors and dependent variables

SQSPCORR squared semipartial correlations between regressors and dependent variables

PCORR partial correlations between regressors and dependent variables

SQPCORR squared partial correlations between regressors and dependent variables

RSQUARED R^2s for the multiple regression analyses

ADJRSQ adjusted R^2s

LCLRSQ approximate 95% lower confidence limits for the R^2s

UCLRSQ approximate 95% upper confidence limits for the R^2s

F F statistics for the multiple regression analyses

PROBF probability levels for the F statistics

CANCORR canonical correlations

SCORE standardized canonical coefficients. The _NAME_ variable contains the name of the canonical variable.

To obtain the canonical variable scores, these coefficients should be multiplied by the standardized data using means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD'.

RAWSCORE raw canonical coefficients.

To obtain the canonical variable scores, these coefficients should be multiplied by the raw data centered by means obtained from the observation with _TYPE_='MEAN'.

USCORE scoring coefficients to be applied without subtracting the mean from the raw variables. These are standardized canonical coefficients computed under a NOINT model.

To obtain the canonical variable scores, these coefficients should be multiplied by the data that are standardized by the uncorrected standard deviations obtained from the observation with _TYPE_='USTD'.

STRUCTUR canonical structure

Computational Resources

Notation

$$\begin{aligned}
n &= \text{number of observations} \\
v &= \text{number of variables} \\
w &= \text{number of WITH variables} \\
p &= \max(v, w) \\
q &= \min(v, w) \\
b &= v + w \\
t &= \text{total number of variables (VAR, WITH, and PARTIAL)}
\end{aligned}$$

Time Requirements

The time required to compute the correlation matrix is roughly proportional to

$$n(p+q)^2$$

The time required for the canonical analysis is roughly proportional to

$$\frac{1}{6}p^3 + p^2q + \frac{3}{2}pq^2 + 5q^3$$

but the coefficient for q^3 varies depending on the number of QR iterations in the singular value decomposition.

Memory Requirements

The minimum memory required is approximately

$$4(v^2 + w^2 + t^2)$$

bytes. Additional memory is required if you request the VDEP or WDEP option.

Displayed Output

If the SIMPLE option is specified, PROC CANCORR produces means and standard deviations for each input variable. If the CORR option is specified, PROC CANCORR produces correlations among the input variables. Unless the NOPRINT option is specified, PROC CANCORR displays a table of canonical correlations containing the following:

- Canonical Correlations. These are always nonnegative.
- Adjusted Canonical Correlations (Lawley 1959), which are asymptotically less biased than the raw correlations and may be negative. The adjusted canonical correlations may not be computable, and they are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.
- Approx Standard Errors, which are the approximate standard errors of the canonical correlations
- Squared Canonical Correlations
- Eigenvalues of INV(E)*H, which are equal to CanRsq/(1$-$CanRsq), where CanRsq is the corresponding squared canonical correlation. Also displayed for each eigenvalue is the Difference from the next eigenvalue, the Proportion of the sum of the eigenvalues, and the Cumulative proportion.
- Likelihood Ratio for the hypothesis that the current canonical correlation and all smaller ones are 0 in the population. The likelihood ratio for all canonical correlations equals Wilks' lambda.
- Approx F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Num DF and Den DF (numerator and denominator degrees of freedom) and Pr $> F$ (probability level) associated with the F statistic

Unless you specify the NOPRINT option, PROC CANCORR produces a table of multivariate statistics for the null hypothesis that all canonical correlations are zero in the population. These statistics, as described in the section "Multivariate Tests" in Chapter 2, "Introduction to Regression Procedures." , are:

- Wilks' Lambda
- Pillai's Trace

- Hotelling-Lawley Trace
- Roy's Greatest Root

For each of the preceding statistics, PROC CANCORR displays

- an F approximation or upper bound
- Num DF, the numerator degrees of freedom
- Den DF, the denominator degrees of freedom
- $\mathrm{Pr} > F$, the probability level

Unless you specify the SHORT or NOPRINT option, PROC CANCORR displays the following:

- both Raw (unstandardized) and Standardized Canonical Coefficients normalized to give canonical variables with unit variance. Standardized coefficients can be used to compute canonical variable scores from the standardized (zero mean and unit variance) input variables. Raw coefficients can be used to compute canonical variable scores from the input variables without standardizing them.
- all four Canonical Structure matrices, giving Correlations Between the canonical variables and the original variables

If you specify the REDUNDANCY option, PROC CANCORR displays

- the Canonical Redundancy Analysis (Stewart and Love 1968; Cooley and Lohnes 1971), including Raw (unstandardized) and Standardized Variance and Cumulative Proportion of the Variance of each set of variables Explained by Their Own Canonical Variables and Explained by The Opposite Canonical Variables
- the Squared Multiple Correlations of each variable with the first m canonical variables of the opposite set, where m varies from 1 to the number of canonical correlations

If you specify the VDEP option, PROC CANCORR performs multiple regression analyses with the VAR variables as dependent variables and the WITH variables as regressors. If you specify the WDEP option, PROC CANCORR performs multiple regression analyses with the WITH variables as dependent variables and the VAR variables as regressors. If you specify the VDEP or WDEP option and also specify the ALL option, PROC CANCORR displays the following items. You can also specify individual options to request a subset of the output generated by the ALL option; or you can suppress the output by specifying the NOPRINT option.

- if you specify the SMC option, Squared Multiple Correlations and F Tests. For each regression model, identified by its dependent variable name, PROC CANCORR displays the R-Squared, Adjusted R-Squared (Wherry 1931), F Statistic, and $\Pr > F$. Also for each regression model, PROC CANCORR displays an Approximate 95% Confidence Interval for the population R^2 (Helland 1987). These confidence limits are valid only when the regressors are random and when the regressors and dependent variables are approximately distributed according to a multivariate normal distribution.

  The average R^2s for the models considered, unweighted and weighted by variance, are also given.
- if you specify the CORRB option, Correlations Among the Regression Coefficient Estimates
- if you specify the STB option, Standardized Regression Coefficients
- if you specify the B option, Raw Regression Coefficients
- if you specify the SEB option, Standard Errors of the Regression Coefficients
- if you specify the CLB option, 95% confidence limits for the regression coefficients
- if you specify the T option, T Statistics for the Regression Coefficients
- if you specify the PROBT option, Probability > |T| for the Regression Coefficients
- if you specify the SPCORR option, Semipartial Correlations between regressors and dependent variables, Removing from Each Regressor the Effects of All Other Regressors
- if you specify the SQSPCORR option, Squared Semipartial Correlations between regressors and dependent variables, Removing from Each Regressor the Effects of All Other Regressors
- if you specify the PCORR option, Partial Correlations between regressors and dependent variables, Removing the Effects of All Other Regressors from Both Regressor and Criterion
- if you specify the SQPCORR option, Squared Partial Correlations between regressors and dependent variables, Removing the Effects of All Other Regressors from Both Regressor and Criterion

ODS Table Names

PROC CANCORR assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 20.2.

For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 20.2. ODS Tables Produced in PROC CANCORR

ODS Table Name	Description	Statement	Option
AvgRSquare	Average R-Squares (weighted and unweighted)	PROC CANCORR	VDEP (or WDEP) SMC (or ALL)
CanCorr	Canonical correlations	PROC CANCORR	default
CanStructureVCan	Correlations between the VAR canonical variables and the VAR and WITH variables	PROC CANCORR	default (unless SHORT)
CanStructureWCan	Correlations between the WITH canonical variables and the WITH and VAR variables	PROC CANCORR	default (unless SHORT)
ConfidenceLimits	95% Confidence limits for the regression coefficients	PROC CANCORR	VDEP (or WDEP) CLB (or ALL)
Corr	Correlations among the original variables	PROC CANCORR	CORR (or ALL)
CorrOnPartial	Partial correlations	PARTIAL	CORR (or ALL)
CorrRegCoefEst	Correlations among the regression coefficient estimates	PROC CANCORR	VDEP (or WDEP) CORRB (or ALL)
MultStat	Multivariate statistics	default	
NObsNVar	Number of observations and variables	PROC CANCORR	SIMPLE (or ALL)
ParCorr	Partial correlations	PROC CANCORR	VDEP (or WDEP) PCORR (or ALL)
ProbtRegCoef	Prob > |t| for the regression coefficients	PROC CANCORR	VDEP (or WDEP) PROBT (or ALL)
RawCanCoefV	Raw canonical coefficients for the var variables	PROC CANCORR	default (unless SHORT)
RawCanCoefW	Raw canonical coefficients for the with variables	PROC CANCORR	default (unless SHORT)
RawRegCoef	Raw regression coefficients	PROC CANCORR	VDEP (or WDEP) B (or ALL)
Redundancy	Canonical redundancy analysis	PROC CANCORR	REDUNDANCY (or ALL)
Regression	Squared multiple correlations and F tests	PROC CANCORR	VDEP (or WDEP) SMC (or ALL)
RSquareRMSEOnPartial	R-Squares and RMSEs on PARTIAL	PARTIAL	CORR (or ALL)
SemiParCorr	Semi-partial correlations	PROC CANCORR	VDEP (or WDEP) SPCORR (or ALL)
SimpleStatistics	Simple statistics	PROC CANCORR	SIMPLE (or ALL)
SqMultCorr	Canonical redundancy analysis: squared multiple correlations	PROC CANCORR	REDUNDANCY (or ALL)
SqParCorr	Squared partial correlations	PROC CANCORR	VDEP (or WDEP) SQPCORR (or ALL)

Table 20.2. (continued)

ODS Table Name	Description	Statement	Option
SqSemiParCorr	Squared semi-partial correlations	PROC CANCORR	VDEP (or WDEP) SQSPCORR (or ALL)
StdCanCoefV	Standardized Canonical coefficients for the VAR variables	PROC CANCORR	default (unless SHORT)
StdCanCoefW	Standardized Canonical coefficients for the WITH variables	PROC CANCORR	default (unless SHORT)
StdErrRawRegCoef	Standard errors of the raw regression coefficients	PROC CANCORR	VDEP (or WDEP) SEB (or ALL)
StdRegCoef	Standardized regression coefficients	PROC CANCORR	VDEP (or WDEP) STB (or ALL)
StdRegCoefOnPartial	Standardized regression coefficients on PARTIAL	PARTIAL	CORR (or ALL)
tValueRegCoef	t values for the regression coefficients	PROC CANCORR	VDEP (or WDEP) T (or ALL)

Example

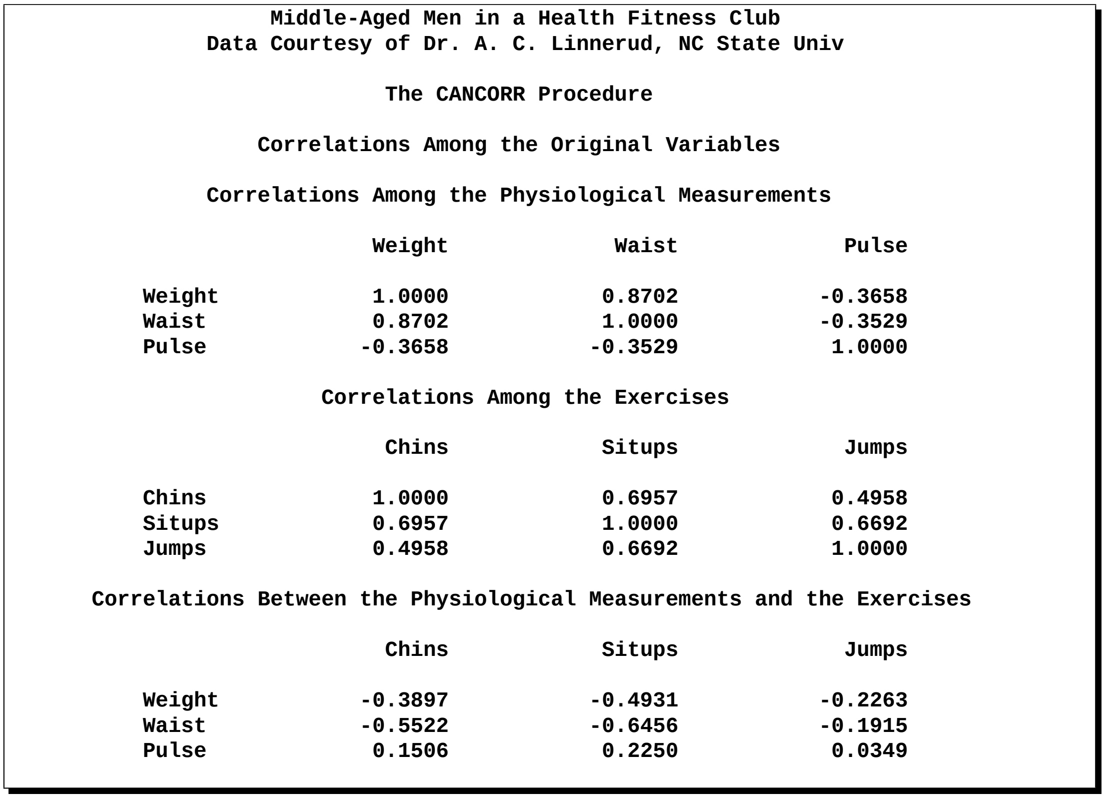
Example 20.1. Canonical Correlation Analysis of Fitness Club Data

Three physiological and three exercise variables are measured on twenty middle-aged men in a fitness club. You can use the CANCORR procedure to determine whether the physiological variables are related in any way to the exercise variables. The following statements create the SAS data set Fit:

```
data Fit;
   input Weight Waist Pulse Chins Situps Jumps;
   datalines;
191  36  50   5  162   60
189  37  52   2  110   60
193  38  58  12  101  101
162  35  62  12  105   37
189  35  46  13  155   58
182  36  56   4  101   42
211  38  56   8  101   38
167  34  60   6  125   40
176  31  74  15  200   40
154  33  56  17  251  250
169  34  50  17  120   38
166  33  52  13  210  115
154  34  64  14  215  105
247  46  50   1   50   50
```

```
193  36  46   6   70   31
202  37  62  12  210  120
176  37  54   4   60   25
157  32  52  11  230   80
156  33  54  15  225   73
138  33  68   2  110   43
;
proc cancorr data=Fit all
     vprefix=Physiological vname='Physiological Measurements'
     wprefix=Exercises wname='Exercises';
   var Weight Waist Pulse;
   with Chins Situps Jumps;
   title 'Middle-Aged Men in a Health Fitness Club';
   title2 'Data Courtesy of Dr. A. C. Linnerud, NC State Univ';
run;
```

Output 20.1.1. Correlations among the Original Variables



```
                    Middle-Aged Men in a Health Fitness Club
               Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                             The CANCORR Procedure

                    Correlations Among the Original Variables

                 Correlations Among the Physiological Measurements

                            Weight             Waist             Pulse

        Weight              1.0000            0.8702           -0.3658
        Waist               0.8702            1.0000           -0.3529
        Pulse              -0.3658           -0.3529            1.0000

                          Correlations Among the Exercises

                             Chins            Situps             Jumps

        Chins               1.0000            0.6957            0.4958
        Situps              0.6957            1.0000            0.6692
        Jumps               0.4958            0.6692            1.0000

    Correlations Between the Physiological Measurements and the Exercises

                             Chins            Situps             Jumps

        Weight             -0.3897           -0.4931           -0.2263
        Waist              -0.5522           -0.6456           -0.1915
        Pulse               0.1506            0.2250            0.0349
```

Output 20.1.1 displays the correlations among the original variables. The correlations between the physiological and exercise variables are moderate, the largest being -0.6456 between Waist and Situps. There are larger within-set correlations: 0.8702 between Weight and Waist, 0.6957 between Chins and Situps, and 0.6692 between Situps and Jumps.

Output 20.1.2. Canonical Correlations and Multivariate Statistics

```
                    Middle-Aged Men in a Health Fitness Club
               Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                             The CANCORR Procedure

                         Canonical Correlation Analysis

                                    Adjusted    Approximate        Squared
                    Canonical      Canonical       Standard      Canonical
                  Correlation    Correlation          Error    Correlation

             1       0.795608       0.754056       0.084197       0.632992
             2       0.200556       -.076399       0.220188       0.040223
             3       0.072570        .             0.228208       0.005266

                                   Eigenvalues of Inv(E)*H
                                     = CanRsq/(1-CanRsq)

                    Eigenvalue    Difference    Proportion    Cumulative

               1        1.7247        1.6828        0.9734        0.9734
               2        0.0419        0.0366        0.0237        0.9970
               3        0.0053                      0.0030        1.0000

                  Test of H0: The canonical correlations in the
                    current row and all that follow are zero

                  Likelihood    Approximate
                       Ratio        F Value    Num DF    Den DF    Pr > F

             1    0.35039053           2.05         9    34.223    0.0635
             2    0.95472266           0.18         4        30    0.9491
             3    0.99473355           0.08         1        16    0.7748



                 Multivariate Statistics and F Approximations

                             S=3    M=-0.5    N=6

Statistic                        Value    F Value    Num DF    Den DF    Pr > F

Wilks’ Lambda               0.35039053       2.05         9    34.223    0.0635
Pillai’s Trace              0.67848151       1.56         9        48    0.1551
Hotelling-Lawley Trace      1.77194146       2.64         9    19.053    0.0357
Roy’s Greatest Root         1.72473874       9.20         3        16    0.0009

         NOTE: F Statistic for Roy’s Greatest Root is an upper bound.
```

As Output 20.1.2 shows, the first canonical correlation is 0.7956, which would appear to be substantially larger than any of the between-set correlations. The probability level for the null hypothesis that all the canonical correlations are 0 in the population is only 0.0635, so no firm conclusions can be drawn. The remaining canonical correlations are not worthy of consideration, as can be seen from the probability levels and especially from the negative adjusted canonical correlations.

Because the variables are not measured in the same units, the standardized coefficients rather than the raw coefficients should be interpreted. The correlations given in the canonical structure matrices should also be examined.

Output 20.1.3. Raw and Standardized Canonical Coefficients

```
                    Middle-Aged Men in a Health Fitness Club
                 Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                              The CANCORR Procedure

                           Canonical Correlation Analysis

          Raw Canonical Coefficients for the Physiological Measurements

                   Physiological1      Physiological2      Physiological3

       Weight        -0.031404688        -0.076319506        -0.007735047
       Waist         0.4932416756        0.3687229894        0.1580336471
       Pulse         -0.008199315        -0.032051994        0.1457322421

                   Raw Canonical Coefficients for the Exercises

                        Exercises1         Exercises2         Exercises3

          Chins       -0.066113986       -0.071041211       -0.245275347
          Situps      -0.016846231       0.0019737454       0.0197676373
          Jumps       0.0139715689       0.0207141063       -0.008167472


                    Middle-Aged Men in a Health Fitness Club
                 Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                              The CANCORR Procedure

                           Canonical Correlation Analysis

      Standardized Canonical Coefficients for the Physiological Measurements

                   Physiological1      Physiological2      Physiological3

       Weight             -0.7754             -1.8844             -0.1910
       Waist               1.5793              1.1806              0.5060
       Pulse              -0.0591             -0.2311              1.0508

              Standardized Canonical Coefficients for the Exercises

                         Exercises1      Exercises2      Exercises3

             Chins          -0.3495         -0.3755         -1.2966
             Situps         -1.0540          0.1235          1.2368
             Jumps           0.7164          1.0622         -0.4188
```

The first canonical variable for the physiological variables, displayed in Output 20.1.3, is a weighted difference of Waist (1.5793) and Weight (-0.7754), with more emphasis on Waist. The coefficient for Pulse is near 0. The correlations between Waist and Weight and the first canonical variable are both positive, 0.9254 for Waist and 0.6206 for Weight. Weight is therefore a suppressor variable, meaning that its coefficient and its correlation have opposite signs.

The first canonical variable for the exercise variables also shows a mixture of signs, subtracting Situps (-1.0540) and Chins (-0.3495) from Jumps (0.7164), with the most weight on Situps. All the correlations are negative, indicating that Jumps is also a suppressor variable.

It may seem contradictory that a variable should have a coefficient of opposite sign from that of its correlation with the canonical variable. In order to understand how this can happen, consider a simplified situation: predicting Situps from Waist and Weight by multiple regression. In informal terms, it seems plausible that fat people should do fewer sit-ups than skinny people. Assume that the men in the sample do not vary much in height, so there is a strong correlation between Waist and Weight (0.8702). Examine the relationships between fatness and the independent variables:

- People with large waists tend to be fatter than people with small waists. Hence, the correlation between Waist and Situps should be negative.
- People with high weights tend to be fatter than people with low weights. Therefore, Weight should correlate negatively with Situps.
- For a fixed value of Weight, people with large waists tend to be shorter and fatter. Thus, the multiple regression coefficient for Waist should be negative.
- For a fixed value of Waist, people with higher weights tend to be taller and skinnier. The multiple regression coefficient for Weight should, therefore, be positive, of opposite sign from the correlation between Weight and Situps.

Therefore, the general interpretation of the first canonical correlation is that Weight and Jumps act as suppressor variables to enhance the correlation between Waist and Situps. This canonical correlation may be strong enough to be of practical interest, but the sample size is not large enough to draw definite conclusions.

The canonical redundancy analysis (Output 20.1.4) shows that neither of the first pair of canonical variables is a good overall predictor of the opposite set of variables, the proportions of variance explained being 0.2854 and 0.2584. The second and third canonical variables add virtually nothing, with cumulative proportions for all three canonical variables being 0.2969 and 0.2767.

Output 20.1.4. Canonical Redundancy Analysis

```
                   Middle-Aged Men in a Health Fitness Club
                Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                            The CANCORR Procedure

                        Canonical Redundancy Analysis

      Standardized Variance of the Physiological Measurements Explained by
                   Their Own                               The Opposite
                Canonical Variables                     Canonical Variables
Canonical
 Variable                  Cumulative     Canonical                 Cumulative
   Number    Proportion    Proportion      R-Square    Proportion    Proportion

        1        0.4508        0.4508        0.6330        0.2854        0.2854
        2        0.2470        0.6978        0.0402        0.0099        0.2953
        3        0.3022        1.0000        0.0053        0.0016        0.2969


              Standardized Variance of the Exercises Explained by
                   Their Own                               The Opposite
                Canonical Variables                     Canonical Variables
Canonical
 Variable                  Cumulative     Canonical                 Cumulative
   Number    Proportion    Proportion      R-Square    Proportion    Proportion

        1        0.4081        0.4081        0.6330        0.2584        0.2584
        2        0.4345        0.8426        0.0402        0.0175        0.2758
        3        0.1574        1.0000        0.0053        0.0008        0.2767
```

```
                   Middle-Aged Men in a Health Fitness Club
                Data Courtesy of Dr. A. C. Linnerud, NC State Univ

                            The CANCORR Procedure

                        Canonical Redundancy Analysis

      Squared Multiple Correlations Between the Physiological Measurements
              and the First M Canonical Variables of the Exercises

                M                  1             2             3

                Weight        0.2438        0.2678        0.2679
                Waist         0.5421        0.5478        0.5478
                Pulse         0.0701        0.0702        0.0749

       Squared Multiple Correlations Between the Exercises and the First
            M Canonical Variables of the Physiological Measurements

                M                  1             2             3

                Chins         0.3351        0.3374        0.3396
                Situps        0.4233        0.4365        0.4365
                Jumps         0.0167        0.0536        0.0539
```

The squared multiple correlations indicate that the first canonical variable of the physiological measurements has some predictive power for Chins (0.3351) and Situps

(0.4233) but almost none for Jumps (0.0167). The first canonical variable of the exercises is a fairly good predictor of Waist (0.5421), a poorer predictor of Weight (0.2438), and nearly useless for predicting Pulse (0.0701).

References

Cooley, W.W. and Lohnes, P.R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons, Inc.

Fisher, R.A. (1938), *Statistical Methods for Research Workers*, Tenth Edition, Edinburgh: Oliver & Boyd.

Hanson, R.J. and Norris, M.J. (1981), "Analysis of Measurements Based on the Singular Value Decomposition," *SIAM Journal of Scientific and Statistical Computing*, 2, 363–373.

Helland, I.S. (1987), "On the Interpretation and Use of R^2 in Regression Analysis," *Biometrics*, 43, 61–69.

Hotelling, H. (1935), "The Most Predictable Criterion," *Journal of Educational Psychology*, 26, 139–142.

Hotelling, H. (1936), "Relations Between Two Sets of Variables," *Biometrika*, 28, 321–377.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker, Inc.

Lawley, D.N. (1959), "Tests of Significance in Canonical Analysis," *Biometrika*, 46, 59–66.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis*, London: Academic Press, Inc.

Mulaik, S.A. (1972), *The Foundations of Factor Analysis*, New York: McGraw-Hill Book Co.

Rao, C.R. (1964), "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya A*, 26, 329–358.

Rao, C.R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons, Inc.

Stewart, D.K. and Love, W.A. (1968), "A General Canonical Correlation Index," *Psychological Bulletin*, 70, 160–163.

Tatsuoka, M.M. (1971), *Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Thompson, B. (1984), "Canonical Correlation Analysis," Sage University Paper series in Quantitative Applications in the Social Sciences, 07-047, Beverly Hills and London: Sage Publications.

Timm, N.H. (1975), *Multivariate Analysis*, Monterey, CA: Brooks-Cole Publishing Co.

van den Wollenberg, A.L. (1977), "Redundancy Analysis—An Alternative to Canonical Correlation Analysis," *Psychometrika*, 42, 207–219.

Wherry, R.J. (1931), “A New Formula for Predicting the Shrinkage of the Coefficient of Multiple Correlation,” *Annals of Mathematical Statistics*, 2, 440–457.

Chapter 21
The CANDISC Procedure

Chapter Contents

Chapter 21
The CANDISC Procedure

Overview

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. The methodology used in deriving the canonical coefficients parallels that of a one-way MANOVA. Whereas in MANOVA the goal is to test for equality of the mean vector across class levels, in a canonical discriminant analysis we find linear combinations of the quantitative variables that provide maximal separation between the classes or groups. Given a classification variable and several quantitative variables, the CANDISC procedure derives *canonical variables*, linear combinations of the quantitative variables that summarize between-class variation in much the same way that principal components summarize total variation.

The CANDISC procedure performs a canonical discriminant analysis, computes squared Mahalanobis distances between class means, and performs both univariate and multivariate one-way analyses of variance. Two output data sets can be produced: one containing the canonical coefficients and another containing, among other things, scored canonical variables. The canonical coefficients output data set can be rotated by the FACTOR procedure. It is customary to standardize the canonical coefficients so that the canonical variables have means that are equal to zero and pooled within-class variances that are equal to one. PROC CANDISC displays both standardized and unstandardized canonical coefficients. Correlations between the canonical variables and the original variables as well as the class means for the canonical variables are also displayed; these correlations, sometimes known as loadings, are called canonical structures. The scored canonical variables output data set can be used in conjunction with the PLOT procedure or the %PLOTIT macro to plot pairs of canonical variables to aid visual interpretation of group differences.

Given two or more groups of observations with measurements on several quantitative variables, canonical discriminant analysis derives a linear combination of the variables that has the highest possible multiple correlation with the groups. This maximal multiple correlation is called the *first canonical correlation*. The coefficients of the linear combination are the *canonical coefficients* or *canonical weights*. The variable defined by the linear combination is the *first canonical variable* or *canonical component*. The second canonical correlation is obtained by finding the linear combination uncorrelated with the first canonical variable that has the highest possible multiple correlation with the groups. The process of extracting canonical variables can be repeated until the number of canonical variables equals the number of original variables or the number of classes minus one, whichever is smaller.

The first canonical correlation is at least as large as the multiple correlation between the groups and any of the original variables. If the original variables have high within-group correlations, the first canonical correlation can be large even if all the multiple correlations are small. In other words, the first canonical variable can show substantial differences between the classes, even if none of the original variables do. Canonical variables are sometimes called *discriminant functions*, but this usage is ambiguous because the DISCRIM procedure produces very different functions for classification that are also called discriminant functions.

For each canonical correlation, PROC CANDISC tests the hypothesis that it and all smaller canonical correlations are zero in the population. An F approximation (Rao 1973; Kshirsagar 1972) is used that gives better small-sample results than the usual chi-square approximation. The variables should have an approximate multivariate normal distribution within each class, with a common covariance matrix in order for the probability levels to be valid.

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the class variable. Canonical discriminant analysis is also equivalent to performing the following steps:

1. Transform the variables so that the pooled within-class covariance matrix is an identity matrix.
2. Compute class means on the transformed variables.
3. Perform a principal component analysis on the means, weighting each mean by the number of observations in the class. The eigenvalues are equal to the ratio of between-class variation to within-class variation in the direction of each principal component.
4. Back-transform the principal components into the space of the original variables, obtaining the canonical variables.

An interesting property of the canonical variables is that they are uncorrelated whether the correlation is calculated from the total sample or from the pooled within-class correlations. The canonical coefficients are not orthogonal, however, so the canonical variables do not represent perpendicular directions through the space of the original variables.

Getting Started

The data in this example are measurements on 159 fish caught in Finland's lake Laengelmavesi. The species, weight, three different length measurements, height, and width of each fish is tallied. The complete data set is displayed in Chapter 67, "The STEPDISC Procedure." The STEPDISC procedure identified all the variables as significant indicators of the differences among the seven fish species.

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
data fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
   symbol = put(Species, specfmt2.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4
1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1
1  363.0 26.3 29.0 33.5 38.0 13.3
 ...[155 more records]
;
```

The following program uses PROC CANDISC to find the three canonical variables that best separate the species of fish in the fish data and creates the output data set outcan. The NCAN= option is used to request that only the first three canonical variables are displayed. The %PLOTIT macro is invoked to create a plot of the first two canonical variables. See Appendix B, "Using the %PLOTIT Macro," for more information on the %PLOTIT macro.

```
proc candisc data=fish ncan=3 out=outcan;
   class Species;
   var Weight Length1 Length2 Length3 Height Width;
run;
%plotit(data=outcan, plotvars=Can2 Can1,
        labelvar=_blank_, symvar=symbol, typevar=symbol,
        symsize=1, symlen=4, tsize=1.5, exttypes=symbol, ls=100,
        plotopts=vaxis=-5 to 15 by 5, vtoh=, extend=close);
```

PROC CANDISC begins by displaying summary information about the variables in the analysis. This information includes the number of observations, the number of quantitative variables in the analysis (specified with the VAR statement), and the number of classes in the classification variable (specified with the CLASS statement). The frequency of each class is also displayed.

```
                              Fish Measurement Data

                              The CANDISC Procedure

           Observations     158          DF Total               157
           Variables          6          DF Within Classes      151
           Classes            7          DF Between Classes       6


                             Class Level Information

                     Variable
        Species      Name          Frequency       Weight    Proportion

        Bream        Bream                34      34.0000      0.215190
        Parkki       Parkki               11      11.0000      0.069620
        Perch        Perch                56      56.0000      0.354430
        Pike         Pike                 17      17.0000      0.107595
        Roach        Roach                20      20.0000      0.126582
        Smelt        Smelt                14      14.0000      0.088608
        Whitefish    Whitefish             6       6.0000      0.037975
```

Figure 21.1. Summary Information

PROC CANDISC performs a multivariate one-way analysis of variance (one-way MANOVA) and provides four multivariate tests of the hypothesis that the class mean vectors are equal. These tests, shown in Figure 21.2, indicate that not all of the mean vectors are equal $(p < .0001)$.

```
                              Fish Measurement Data

                              The CANDISC Procedure

                  Multivariate Statistics and F Approximations

                              S=6    M=-0.5    N=72

Statistic                        Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda               0.00036325      90.71        36    643.89    <.0001
Pillai's Trace              3.10465132      26.99        36       906    <.0001
Hotelling-Lawley Trace     52.05799676     209.24        36    413.64    <.0001
Roy's Greatest Root        39.13499776     984.90         6       151    <.0001

        NOTE: F Statistic for Roy's Greatest Root is an upper bound.
```

Figure 21.2. MANOVA and Multivariate Tests

The first canonical correlation is the greatest possible multiple correlation with the classes that can be achieved using a linear combination of the quantitative variables. The first canonical correlation, displayed in Figure 21.3, is 0.987463.

```
                              Fish Measurement Data

                             The CANDISC Procedure

                                     Adjusted    Approximate        Squared
                     Canonical      Canonical       Standard      Canonical
                   Correlation    Correlation          Error    Correlation

              1       0.987463       0.986671       0.001989       0.975084
              2       0.952349       0.950095       0.007425       0.906969
              3       0.838637       0.832518       0.023678       0.703313
              4       0.633094       0.623649       0.047821       0.400809
              5       0.344157       0.334170       0.070356       0.118444
              6       0.005701        .             0.079806       0.000033
```

Figure 21.3. Canonical Correlations

A likelihood ratio test is displayed of the hypothesis that the current canonical correlation and all smaller ones are zero. The first line is equivalent to Wilks' Lambda multivariate test.

```
                   Test of H0: The canonical correlations in the
                     current row and all that follow are zero

                   Likelihood    Approximate
                        Ratio        F Value    Num DF    Den DF    Pr > F

              1    0.00036325          90.71        36    643.89    <.0001
              2    0.01457896          46.46        25    547.58    <.0001
              3    0.15671134          23.61        16    452.79    <.0001
              4    0.52820347          12.09         9    362.78    <.0001
              5    0.88152702           4.88         4       300    0.0008
              6    0.99996749           0.00         1       151    0.9442
```

Figure 21.4. Likelihood Ratio Test

The first canonical variable, Can1, shows that the linear combination of the centered variables Can1$= -0.0006\times$Weight $-$ $0.33\times$Length1 $-$ $2.49\times$Length2 $+$ $2.60\times$Length3 $+$ $1.12\times$Height $-$ $1.45\times$Width separates the species most effectively (see Figure 21.5).

```
                              Fish Measurement Data

                              The CANDISC Procedure

                            Raw Canonical Coefficients

          Variable              Can1              Can2              Can3

          Weight        -0.000648508      -0.005231659      -0.005596192
          Length1       -0.329435762      -0.626598051      -2.934324102
          Length2       -2.486133674      -0.690253987       4.045038893
          Length3        2.595648437       1.803175454      -1.139264914
          Height         1.121983854      -0.714749340       0.283202557
          Width         -1.446386704      -0.907025481       0.741486686
```

Figure 21.5. Raw Canonical Coefficients

PROC CANDISC computes the means of the canonical variables for each class. The first canonical variable is the linear combination of the variables Weight, Length1, Length2, Length3, Height, and Width that provides the greatest difference (in terms of a univariate F-test) between the class means. The second canonical variable provides the greatest difference between class means while being uncorrelated with the first canonical variable.

```
                              Fish Measurement Data

                              The CANDISC Procedure

                       Class Means on Canonical Variables

         Species               Can1              Can2              Can3

         Bream          10.94142464        0.52078394        0.23496708
         Parkki          2.58903743       -2.54722416       -0.49326158
         Perch          -4.47181389       -1.70822715        1.29281314
         Pike           -4.89689441        8.22140791       -0.16469132
         Roach          -0.35837149        0.08733611       -1.10056438
         Smelt          -4.09136653       -2.35805841       -4.03836098
         Whitefish      -0.39541755       -0.42071778        1.06459242
```

Figure 21.6. Class Means for Canonical Variables

A plot of the first two canonical variables (Figure 21.7) shows that Can1 discriminates between three groups: 1) bream; 2) whitefish, roach, and parkki; and 3) smelt, pike, and perch. Can2 best discriminates between pike and the other species.

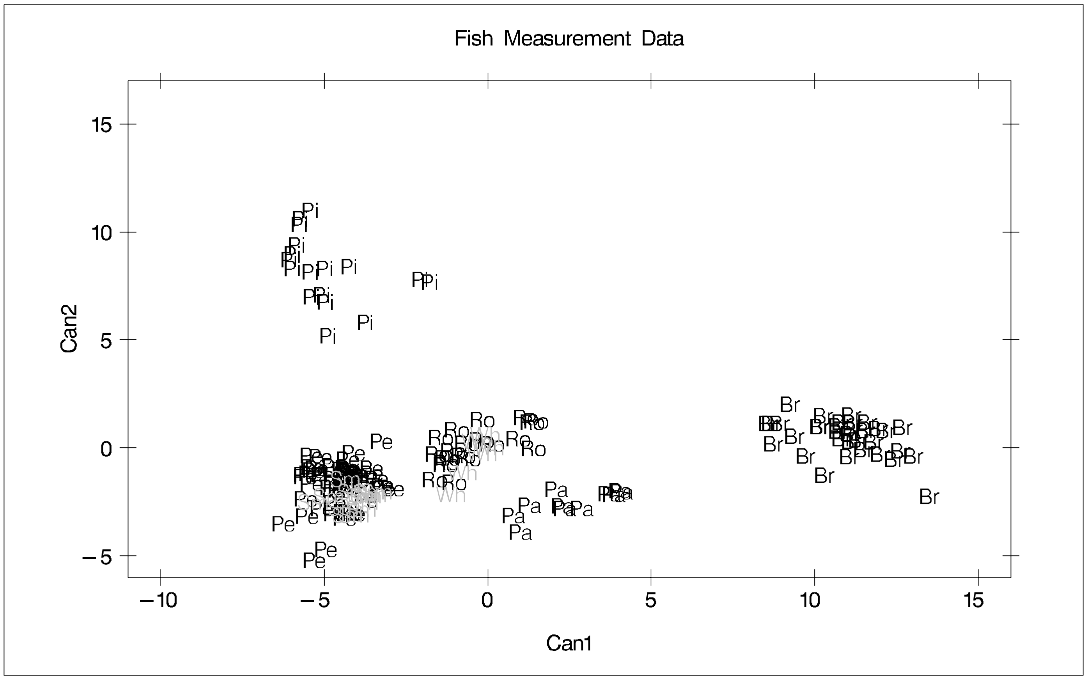


Figure 21.7. Plot of First Two Canonical Variables

Syntax

The following statements are available in PROC CANDISC.

PROC CANDISC < *options* > **;**
  CLASS *variable* **;**
  BY *variables* **;**
  FREQ *variable* **;**
  VAR *variables* **;**
  WEIGHT *variable* **;**

The BY, CLASS, FREQ, VAR, and WEIGHT statements are described after the PROC CANDISC statement.

PROC CANDISC Statement

PROC CANDISC < *options* > **;**

This statement invokes the CANDISC procedure. The options listed in the following table can appear in the PROC CANDISC statement.

Table 21.1. CANDISC Procedure Options

Task	Options
Specify Data Sets	DATA= OUT= OUTSTAT=
Control Canonical Variables	NCAN= PREFIX=
Determine Singularity	SINGULAR=
Control Displayed Correlations	BCORR PCORR TCORR WCORR
Control Displayed Covariances	BCOV PCOV TCOV WCOV
Control Displayed SSCP Matrices	BSSCP PSSCP TSSCP WSSCP
Suppress Output	NOPRINT SHORT
Miscellaneous	ALL ANOVA DISTANCE SIMPLE STDMEAN

ALL

activates all of the display options.

ANOVA

displays univariate statistics for testing the hypothesis that the class means are equal in the population for each variable.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. The between-class covariances should be interpreted in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

DATA=*SAS-data-set*

specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS statistical procedures. These specially structured data sets include TYPE=CORR, COV, CSSCP, and SSCP. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DISTANCE

displays squared Mahalanobis distances between the group means, F statistics, and the corresponding probabilities of greater squared Mahalanobis distances between the group means.

NCAN=*n*

specifies the number of canonical variables to be computed. The value of n must be less than or equal to the number of variables. If you specify NCAN=0, the procedure displays the canonical correlations, but not the canonical coefficients, structures, or means. A negative value suppresses the canonical analysis entirely. Let v be the number of variables in the VAR statement and c be the number of classes. If you omit the NCAN= option, only $\min(v, c-1)$ canonical variables are generated; if you also specify an OUT= output data set, v canonical variables are generated, and the last $v - (c - 1)$ canonical variables have missing values.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*

creates an output SAS data set containing the original data and the canonical variable scores. To create a permanent SAS data set, specify a two-level name (refer to *SAS Language Reference: Concepts*, for more information on permanent SAS data sets).

OUTSTAT=*SAS-data-set*

creates a TYPE=CORR output SAS data set that contains various statistics including class means, standard deviations, correlations, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class. To create a permanent SAS data set, specify a two-level name (refer to *SAS Language Reference: Concepts*, for more information on permanent SAS data sets).

PCORR

displays pooled within-class correlations (partial correlations based on the pooled within-class covariances).

PCOV

displays pooled within-class covariances.

PREFIX=*name*

specifies a prefix for naming the canonical variables. By default the names are Can1, Can2, Can3 and so forth. If you specify PREFIX=Abc, the components are named Abc1, Abc2, and so on. The number of characters in the prefix, plus the number of digits required to designate the canonical variables, should not exceed 32. The prefix is truncated if the combined length exceeds 32.

PSSCP

displays the pooled within-class corrected SSCP matrix.

SHORT

suppresses the display of canonical structures, canonical coefficients, and class means on canonical variables; only tables of canonical correlations and multivariate test statistics are displayed.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the criterion for determining the singularity of the total-sample correlation matrix and the pooled within-class covariance matrix, where $0 < p < 1$. The default is SINGULAR=1E-8.

Let $\mathbf{S}$ be the total-sample correlation matrix. If the R^2 for predicting a quantitative variable in the VAR statement from the variables preceding it exceeds $1 - p$, $\mathbf{S}$ is considered singular. If $\mathbf{S}$ is singular, the probability levels for the multivariate test statistics and canonical correlations are adjusted for the number of variables with R^2 exceeding $1 - p$.

If $\mathbf{S}$ is considered singular and the inverse of $\mathbf{S}$ (Squared Mahalanobis Distances) is required, a quasi-inverse is used instead. For details see the “Quasi-Inverse” section in Chapter 25, “The DISCRIM Procedure.”

STDMEAN

displays total-sample and pooled within-class standardized class means.

TCORR

displays total-sample correlations.

TCOV

displays total-sample covariances.

TSSCP

displays the total-sample corrected SSCP matrix.

WCORR

displays within-class correlations for each class level.

WCOV

displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CANDISC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CANDISC procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The values of the CLASS variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The CLASS variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the name of the variable in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

VAR Statement

VAR *variables* **;**

You specify the quantitative variables to include in the analysis using a VAR statement. If you do not use a VAR statement, the analysis includes all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the WEIGHT variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is assumed.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details

Missing Values

If an observation has a missing value for any of the quantitative variables, it is omitted from the analysis. If an observation has a missing CLASS value but is otherwise complete, it is not used in computing the canonical correlations and coefficients; however, canonical variable scores are computed for that observation for the OUT= data set.

Computational Details

General Formulas

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the class variable. In the following notation the dummy variables will be denoted by $\mathbf{y}$ and the quantitative variables by $\mathbf{x}$. The total sample covariance matrix for the $\mathbf{x}$ and $\mathbf{y}$ variables is

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xy} \\ \mathbf{S}_{yx} & \mathbf{S}_{yy} \end{bmatrix}$$

When c is the number of groups, n_t is the number of observations in group t, and $\mathbf{S}_t$ is the sample covariance matrix for the $\mathbf{x}$ variables in group t, the within-class pooled covariance matrix for the $\mathbf{x}$ variables is

$$\mathbf{S}_p = \frac{1}{\sum n_t - c} \sum (n_t - 1)\mathbf{S}_t$$

The canonical correlations, ρ_i, are the square roots of the eigenvalues, λ_i, of the following matrix. The corresponding eigenvectors are $\mathbf{v}_i$.

$$\mathbf{S}_p{}^{-1/2}\mathbf{S}_{xy}\mathbf{S}_{yy}{}^{-1}\mathbf{S}_{yx}\mathbf{S}_p{}^{-1/2}$$

Let $\mathbf{V}$ be the matrix with the eigenvectors $\mathbf{v}_i$ that correspond to nonzero eigenvalues as columns. The raw canonical coefficients are calculated as follows

$$\mathbf{R} = \mathbf{S}_p{}^{-1/2}\mathbf{V}$$

The pooled within-class standardized canonical coefficients are

$$\mathbf{P} = \mathrm{diag}(\mathbf{S}_p)^{1/2}\mathbf{R}$$

And the total sample standardized canonical coefficients are

$$\mathbf{T} = \mathrm{diag}(\mathbf{S}_{xx})^{1/2}\mathbf{R}$$

Let $\mathbf{X}_c$ be the matrix with the centered $\mathbf{x}$ variables as columns. The canonical scores may be calculated by any of the following

$$\mathbf{X}_c\,\mathbf{R}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_p)^{-1/2}\mathbf{P}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_{xx})^{-1/2}\mathbf{T}$$

For the Multivariate tests based on $\mathbf{E}^{-1}\mathbf{H}$

$$\mathbf{E} = (n-1)(\mathbf{S}_{yy} - \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy})$$

$$\mathbf{H} = (n-1)\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$$

where n is the total number of observations.

Input Data Set

The input DATA= data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. For more information on special types of data sets, see Appendix A, “Special SAS Data Sets.” The BY variable in these data sets becomes the CLASS variable in PROC CANDISC. These specially structured data sets include

- TYPE=CORR data sets created by PROC CORR using a BY statement

- TYPE=COV data sets created by PROC PRINCOMP using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG using both the OUTSSCP= option and a BY statement.

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, PROC CANDISC reads the number of observations for each class from the observations with _TYPE_='N' and the variable means in each class from the observations with _TYPE_='MEAN'. The CANDISC procedure then reads the within-class correlations from the observations with _TYPE_='CORR', the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and crossproducts from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When the data set does not include any observations with _TYPE_='CORR' (data set TYPE=CORR), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CSSCP' (data set TYPE=CSSCP) for each class, PROC CANDISC reads the pooled within-class information from the data set. In this case, PROC CANDISC reads the pooled within-class correlations from the observations with _TYPE_='PCORR', the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR), the pooled within-class covariances from the observations with _TYPE_='PCOV' (data set TYPE=COV), or the pooled within-class corrected SSCP matrix from the observations with_TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, PROC CANDISC reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations from the variable INTERCEPT in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and crossproducts from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_=*variablenames*.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. You determine the number of new variables using the NCAN= option. The names of the new variables are formed as described in the PREFIX= option. The new variables have means equal to zero and pooled within-class variances equal to one. An OUT= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure but contains many results in addition to those produced by the CORR procedure.

The OUTSTAT= data set is TYPE=CORR, and it contains the following variables:

- the BY variables, if any
- the CLASS variable
- _TYPE_, a character variable of length 8 that identifies the type of statistic
- _NAME_, a character variable of length 32 that identifies the row of the matrix or the name of the canonical variable
- the quantitative variables (those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement)

The observations, as identified by the variable _TYPE_, have the following _TYPE_ values:

TYPE	**Contents**
N	number of observations for both the total sample (CLASS variable missing) and within each class (CLASS variable present)
SUMWGT	sum of weights for both the total sample (CLASS variable missing) and within each class (CLASS variable present) if a WEIGHT statement is specified
MEAN	means for both the total sample (CLASS variable missing) and within each class (CLASS variable present)
STDMEAN	total-standardized class means
PSTDMEAN	pooled within-class standardized class means
STD	standard deviations for both the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSTD	pooled within-class standard deviations
BSTD	between-class standard deviations
RSQUARED	univariate R^2s

The following kinds of observations are identified by the combination of the variables _TYPE_ and _NAME_. When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the row of the matrix.

TYPE	**Contents**
CSSCP	corrected SSCP matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSSCP	pooled within-class corrected SSCP matrix

BSSCP	between-class SSCP matrix
COV	covariance matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCOV	pooled within-class covariance matrix
BCOV	between-class covariance matrix
CORR	correlation matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCORR	pooled within-class correlation matrix
BCORR	between-class correlation matrix

When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the canonical variable:

TYPE	**Contents**
CANCORR	canonical correlations
STRUCTUR	canonical structure
BSTRUCT	between canonical structure
PSTRUCT	pooled within-class canonical structure
SCORE	total sample standardized canonical coefficients
PSCORE	pooled within-class standardized canonical coefficients
RAWSCORE	raw canonical coefficients
CANMEAN	means of the canonical variables for each class

You can use this data set with PROC SCORE to get scores on the canonical variables for new data using one of the following forms.

```
* The CLASS variable C is numeric;
proc score data=NewData score=Coef(where=(c = .  )) out=Scores; run;

* The CLASS variable C is character;
proc score data=NewData score=Coef(where=(c = ’ ’)) out=Scores;
run;
```

The WHERE clause is used to exclude the within-class means and standard deviations. PROC SCORE standardizes the new data by subtracting the original variable means that are stored in the _TYPE_=’MEAN’ observations, and dividing by the original variable standard deviations from the _TYPE_=’STD’ observations. Then PROC SCORE multiplies the standardized variables by the coefficients from the _TYPE_=’SCORE’ observations to get the canonical scores.

Computational Resources

In the following discussion, let

n = number of observations
c = number of class levels
v = number of variables in the VAR list
l = length of the CLASS variable

Memory Requirements

The amount of memory in bytes for temporary storage needed to process the data is

$$c(4v^2 + 28v + 4l + 68) + 16v^2 + 96v + 4l$$

With the ANOVA option, the temporary storage must be increased by 16v bytes. The DISTANCE option requires an additional temporary storage of $4v^2 + 4v$ bytes.

Time Requirements

The following factors determine the time requirements of the CANDISC procedure.

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. PROC CANDISC must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n\log(c)$.
- The time for inverting a covariance matrix is proportional to v^3.
- The time required for the canonical discriminant analysis is proportional to v^3.

Each of the preceding factors has a different constant of proportionality.

Displayed Output

The output produced by PROC CANDISC includes

- Class Level Information, including the values of the classification variable, the Frequency and Weight of each value, and its Proportion in the total sample.

Optional output includes

- Within-Class SSCP Matrices for each group
- Pooled Within-Class SSCP Matrix
- Between-Class SSCP Matrix
- Total-Sample SSCP Matrix

- Within-Class Covariance Matrices for each group
- Pooled Within-Class Covariance Matrix
- Between-Class Covariance Matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-Sample Covariance Matrix
- Within-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled Within-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the between-class population correlation coefficients are zero
- Total-Sample Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the total population correlation coefficients are zero
- Simple Statistics including N (the number of observations), Sum, Mean, Variance, and Standard Deviation both for the total sample and within each class
- Total-Sample Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the total sample standard deviation
- Pooled Within-Class Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation
- Pairwise Squared Distances Between Groups
- Univariate Test Statistics, including Total-Sample Standard Deviations, Pooled Within-Class Standard Deviations, Between-Class Standard Deviations, R^2, $R^2/(1-R^2)$, F, and $\mathrm{Pr} > F$ (univariate F values and probability levels for one-way analyses of variance)

By default, PROC CANDISC displays these statistics:

- Multivariate Statistics and F Approximations including Wilks' Lambda, Pillai's Trace, Hotelling-Lawley Trace, and Roy's Greatest Root with F approximations, degrees of freedom (Num DF and Den DF), and probability values $(\mathrm{Pr} > F)$. Each of these four multivariate statistics tests the hypothesis that the class means are equal in the population. See the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures," for more information.
- Canonical Correlations
- Adjusted Canonical Correlations (Lawley 1959). These are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations may not be computable and are displayed as missing values if two

canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.

- Approx Standard Error, approximate standard error of the canonical correlations
- Squared Canonical Correlations
- Eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Each eigenvalue is equal to $\rho^2/(1-\rho^2)$, where ρ^2 is the corresponding squared canonical correlation and can be interpreted as the ratio of between-class variation to pooled within-class variation for the corresponding canonical variable. The table includes Eigenvalues, Differences between successive eigenvalues, the Proportion of the sum of the eigenvalues, and the Cumulative proportion.
- Likelihood Ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for the hypothesis that all canonical correlations equal zero is Wilks' lambda.
- Approx F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Num DF (numerator degrees of freedom), Den DF (denominator degrees of freedom), and $\mathrm{Pr} > F$, the probability level associated with the F statistic

The following statistics can be suppressed with the SHORT option:

- Total Canonical Structure, giving total-sample correlations between the canonical variables and the original variables
- Between Canonical Structure, giving between-class correlations between the canonical variables and the original variables
- Pooled Within Canonical Structure, giving pooled within-class correlations between the canonical variables and the original variables
- Total-Sample Standardized Canonical Coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the total-sample standardized variables
- Pooled Within-Class Standardized Canonical Coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the pooled within-class standardized variables
- Raw Canonical Coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the centered variables
- Class Means on Canonical Variables

ODS Table Names

PROC CANDISC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 21.2. ODS Tables Produced in PROC CANDISC

ODS Table Name	Description	PROC CANDISC Option
ANOVA	Univariate statistics	ANOVA
AveRSquare	Average R-square	ANOVA
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
BStruc	Between canonical structure	default
CanCorr	Canonical correlations	default
CanonicalMeans	Class means on canonical variables	default
Counts	Number of observations, variables, classes, df	default
CovDF	DF for covariance matrices, not printed	any *COV option
Dist	Squared distances	MAHALANOBIS
DistFValues	F statistics based on squared distances	MAHALANOBIS
DistProb	Probabilities for F statistics from squared distances	MAHALANOBIS
Levels	Class level information	default
MultStat	MANOVA	default
PCoef	Pooled standard canonical coefficients	default
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
PStruc	Pooled within canonical structure	default
RCoef	Raw canonical coefficients	default
SimpleStatistics	Simple statistics	SIMPLE
TCoef	Total-sample standard canonical coefficients	default
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
TStruc	Total canonical structure	default
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Example

Example 21.1. Analysis of Iris Data Using PROC CANDISC

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species: *Iris setosa, I. versicolor, and I. virginica*.

This example is a canonical discriminant analysis that creates an output data set containing scores on the canonical variables and plots the canonical variables. The following statements produce Output 21.1.1 through Output 21.1.7:

```
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
run;

data iris;
   title ’Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   symbol = put(Species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
```

```
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
proc candisc data=iris out=outcan distance anova;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

PROC CANDISC first displays information about the observations and the classes in the data set in Output 21.1.1.

Output 21.1.1. Iris Data: Summary Information

```
                                 Fisher (1936) Iris Data

                                 The CANDISC Procedure

                  Observations     150          DF Total               149
                  Variables          4          DF Within Classes      147
                  Classes            3          DF Between Classes       2


                                Class Level Information

                              Variable
              Species         Name           Frequency       Weight    Proportion

              Setosa          Setosa                50      50.0000      0.333333
              Versicolor      Versicolor            50      50.0000      0.333333
              Virginica       Virginica             50      50.0000      0.333333
```

The DISTANCE option in the PROC CANDISC statement displays squared Mahalanobis distances between class means. Results from the DISTANCE option is shown in Output 21.1.2 and Output 21.1.3.

Output 21.1.2. Iris Data: Squared Mahalanobis Distances

```
                          Fisher (1936) Iris Data

                           The CANDISC Procedure

                    Pairwise Squared Distances Between Groups

                       2           _   _          -1  _   _
                      D (i|j) = (X - X )' COV   (X - X )
                                  i   j          i   j


                        Squared Distance to Species

              From
              Species          Setosa    Versicolor     Virginica

              Setosa                0      89.86419     179.38471
              Versicolor     89.86419             0      17.20107
              Virginica     179.38471      17.20107             0
```

Output 21.1.3. Iris Data: Squared Mahalanobis Distance Statistics

```
                          Fisher (1936) Iris Data

                           The CANDISC Procedure

          F Statistics, NDF=4, DDF=144 for Squared Distance to Species

              From
              Species          Setosa    Versicolor     Virginica

              Setosa                0     550.18889          1098
              Versicolor    550.18889             0     105.31265
              Virginica          1098     105.31265             0


           Prob > Mahalanobis Distance for Squared Distance to Species

              From
              Species          Setosa    Versicolor     Virginica

              Setosa           1.0000        <.0001        <.0001
              Versicolor       <.0001        1.0000        <.0001
              Virginica        <.0001        <.0001        1.0000
```

The ANOVA option specifies testing of the hypothesis that the class means are equal using univariate statistics. The resulting R^2 values (see Output 21.1.4) range from 0.4008 for SepalWidth to 0.9414 for PetalLength, and each variable is significant at the 0.0001 level. The multivariate test for differences between the classes (which is displayed by default) is also significant at the 0.0001 level; you would expect this from the highly significant univariate test results.

Output 21.1.4. Iris Data: Univariate and Multivariate Statistics

```
                                    Fisher (1936) Iris Data

                                     The CANDISC Procedure

                                  Univariate Test Statistics

                            F Statistics,    Num DF=2,   Den DF=147

                                    Total    Pooled   Between
                                 Standard  Standard  Standard           R-Square
Variable    Label               Deviation Deviation Deviation R-Square / (1-RSq) F Value Pr > F

SepalLength Sepal Length in mm.    8.2807    5.1479    7.9506   0.6187    1.6226  119.26 <.0001
SepalWidth  Sepal Width in mm.     4.3587    3.3969    3.3682   0.4008    0.6688   49.16 <.0001
PetalLength Petal Length in mm.   17.6530    4.3033   20.9070   0.9414   16.0566 1180.16 <.0001
PetalWidth  Petal Width in mm.     7.6224    2.0465    8.9673   0.9289   13.0613  960.01 <.0001


                                       Average R-Square

                               Unweighted              0.7224358
                               Weighted by Variance    0.8689444


                         Multivariate Statistics and F Approximations

                                     S=2    M=0.5    N=71

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks’ Lambda               0.02343863     199.15         8       288    <.0001
        Pillai’s Trace              1.19189883      53.47         8       290    <.0001
        Hotelling-Lawley Trace     32.47732024     582.20         8     203.4    <.0001
        Roy’s Greatest Root        32.19192920    1166.96         4       145    <.0001

                 NOTE: F Statistic for Roy’s Greatest Root is an upper bound.
                         NOTE: F Statistic for Wilks’ Lambda is exact.
```

The R^2 between Can1 and the class variable, 0.969872, is much larger than the corresponding R^2 for Can2, 0.222027. This is displayed in Output 21.1.5.

Output 21.1.5. Iris Data: Canonical Correlations and Eigenvalues

```
                                    Fisher (1936) Iris Data

                                     The CANDISC Procedure

                                          Adjusted    Approximate        Squared
                          Canonical      Canonical       Standard      Canonical
                        Correlation    Correlation          Error    Correlation

                      1    0.984821       0.984508       0.002468       0.969872
                      2    0.471197       0.461445       0.063734       0.222027

                                                      Test of H0: The canonical correlations in
                                                               the current row and all
                   Eigenvalues of Inv(E)*H                       that follow are zero
                     = CanRsq/(1-CanRsq)
                                                     Likelihood Approximate
         Eigenvalue Difference Proportion Cumulative      Ratio     F Value Num DF Den DF Pr > F

       1    32.1919    31.9065     0.9912     0.9912 0.02343863      199.15      8    288 <.0001
       2     0.2854                0.0088     1.0000 0.77797337       13.79      3    145 <.0001
```

Output 21.1.6. Iris Data: Correlations Between Canonical and Original Variables

```
                               Fisher (1936) Iris Data

                                The CANDISC Procedure

                              Total Canonical Structure

            Variable         Label                            Can1             Can2

            SepalLength      Sepal Length in mm.          0.791888         0.217593
            SepalWidth       Sepal Width in mm.          -0.530759         0.757989
            PetalLength      Petal Length in mm.          0.984951         0.046037
            PetalWidth       Petal Width in mm.           0.972812         0.222902


                             Between Canonical Structure

            Variable         Label                            Can1             Can2

            SepalLength      Sepal Length in mm.          0.991468         0.130348
            SepalWidth       Sepal Width in mm.          -0.825658         0.564171
            PetalLength      Petal Length in mm.          0.999750         0.022358
            PetalWidth       Petal Width in mm.           0.994044         0.108977


                          Pooled Within Canonical Structure

            Variable         Label                            Can1             Can2

            SepalLength      Sepal Length in mm.          0.222596         0.310812
            SepalWidth       Sepal Width in mm.          -0.119012         0.863681
            PetalLength      Petal Length in mm.          0.706065         0.167701
            PetalWidth       Petal Width in mm.           0.633178         0.737242
```

The raw canonical coefficients (shown in Output 21.1.7) for the first canonical variable, Can1, show that the classes differ most widely on the linear combination of the centered variables $-0.0829378 \times$ SepalLength $- 0.153447 \times$ SepalWidth $+ 0.220121 \times$ PetalLength $+ 0.281046 \times$ PetalWidth.

Output 21.1.7. Iris Data: Canonical Coefficients

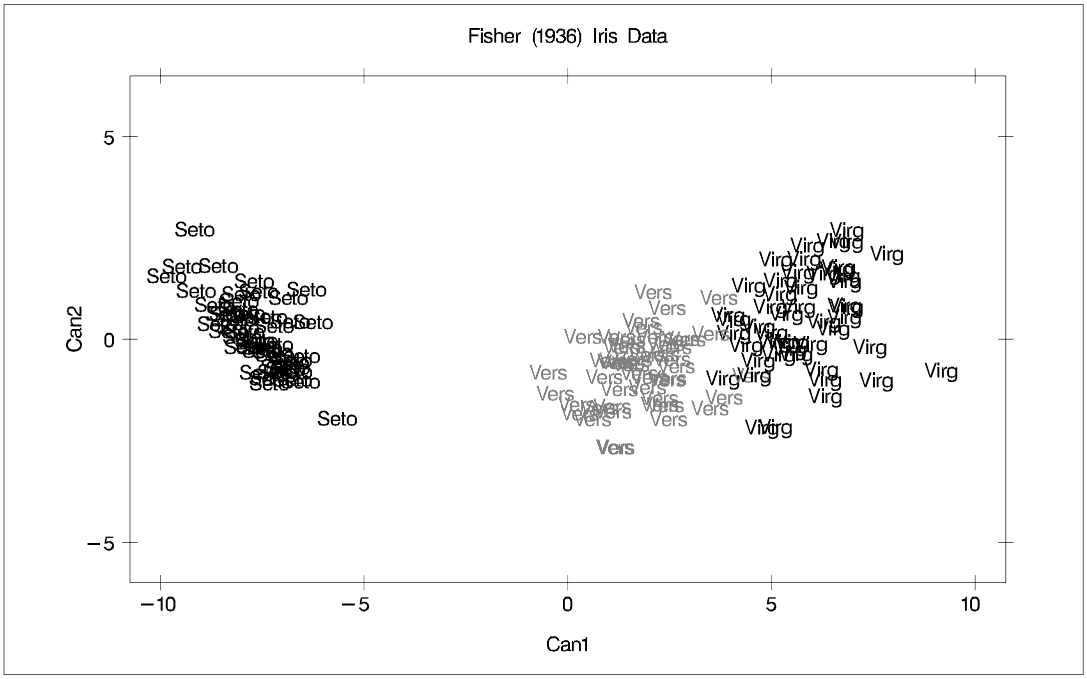
```
                               Fisher (1936) Iris Data

                                The CANDISC Procedure

                    Total-Sample Standardized Canonical Coefficients

            Variable         Label                            Can1             Can2

            SepalLength      Sepal Length in mm.      -0.686779533      0.019958173
            SepalWidth       Sepal Width in mm.       -0.668825075      0.943441829
            PetalLength      Petal Length in mm.       3.885795047     -1.645118866
            PetalWidth       Petal Width in mm.        2.142238715      2.164135931


                Pooled Within-Class Standardized Canonical Coefficients

            Variable         Label                            Can1             Can2

            SepalLength      Sepal Length in mm.      -.4269548486     0.0124075316
            SepalWidth       Sepal Width in mm.       -.5212416758     0.7352613085
            PetalLength      Petal Length in mm.      0.9472572487     -.4010378190
            PetalWidth       Petal Width in mm.       0.5751607719     0.5810398645
```

```
                            Fisher (1936) Iris Data

                             The CANDISC Procedure

                          Raw Canonical Coefficients

Variable         Label                             Can1             Can2

SepalLength      Sepal Length in mm.      -.0829377642      0.0024102149
SepalWidth       Sepal Width in mm.       -.1534473068      0.2164521235
PetalLength      Petal Length in mm.      0.2201211656      -.0931921210
PetalWidth       Petal Width in mm.       0.2810460309      0.2839187853


                      Class Means on Canonical Variables

             Species                   Can1              Can2

             Setosa            -7.607599927       0.215133017
             Versicolor         1.825049490      -0.727899622
             Virginica          5.782550437       0.512766605
```

The plot of canonical variables in Output 21.1.8 shows that of the two canonical variables Can1 has the most discriminatory power. The following invocation of the %PLOTIT macro creates this plot:

```
%plotit(data=outcan, plotvars=Can2 Can1,
        labelvar=_blank_, symvar=symbol, typevar=symbol,
        symsize=1, symlen=4, exttypes=symbol, ls=100,
        tsize=1.5, extend=close);
```

Output 21.1.8. Iris Data: Plot of First Two Canonical Variables



References

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker, Inc.

Lawley, D.N. (1959), “Tests of Significance in Canonical Analysis,” *Biometrika*, 46, 59–66.

Rao, C.R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons, Inc.

Chapter 22
The CATMOD Procedure

Chapter Contents

Chapter 22
The CATMOD Procedure

Overview

The CATMOD procedure performs categorical data modeling of data that can be represented by a contingency table. PROC CATMOD fits linear models to functions of response frequencies, and it can be used for linear modeling, log-linear modeling, logistic regression, and repeated measurement analysis. PROC CATMOD uses

- weighted least-squares (WLS) estimation of parameters for a wide range of general linear models
- maximum likelihood (ML) estimation of parameters for log-linear models and the analysis of generalized logits

The CATMOD procedure provides a wide variety of categorical data analyses, many of which are generalizations of continuous data analysis methods. For example, analysis of variance, in the traditional sense, refers to the analysis of means and the partitioning of variation among the means into various sources. Here, the term *analysis of variance* is used in a generalized sense to denote the analysis of response functions and the partitioning of variation among those functions into various sources. The response functions might be mean scores if the dependent variables are ordinally scaled. But they can also be marginal probabilities, cumulative logits, or other functions that incorporate the essential information from the dependent variables.

Types of Input Data

The data that PROC CATMOD analyzes are usually supplied in one of two ways. First, you can supply raw data, where each observation is a subject. Second, you can supply cell count data, where each observation is a cell in a contingency table. (A third way, which uses direct input of the covariance matrix, is also available; details are given in the "Inputting Response Functions and Covariances Directly" section on page 862.)

Suppose detergent preference is related to three other categorical variables: water softness, water temperature, and previous use of a brand of detergent. In the raw data case, each observation in the input data set identifies a given respondent in the study and contains information on all four variables. The data set contains the same number of observations as the survey had respondents. In the cell count case, each observation identifies a given cell in the four-way table of water softness, water temperature, previous use of brand, and brand preference. A fifth variable contains the number of respondents in the cell. In the analysis, this fifth variable is identified in a WEIGHT statement. The data set contains the same number of observations as the number of cross-classifications formed by the four categorical variables. For more on this particular example, see Example 22.1 on page 901. For additional details, see the section “Input Data Sets” on page 860.

Most of the examples in this chapter use cell counts as input and use a WEIGHT statement.

Types of Statistical Analyses

This section illustrates, by example, the wide variety of categorical data analyses that PROC CATMOD provides. For each type of analysis, a brief description of the statistical problem and the SAS statements to provide the analysis are given. For each analysis, assume that the input data set consists of a set of cell counts from a contingency table. The variable specified in the WEIGHT statement contains these counts. In all these analyses, both the dependent and independent variables are categorical.

Linear Model Analysis

Suppose you want to analyze the relationship between the dependent variables (r1, r2) and the independent variables (a, b). Analyze the marginal probabilities of the dependent variables, and use a main-effects model.

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2=a b;
quit;
```

Log-Linear Model Analysis

Suppose you want to analyze the nominal dependent variables (r1, r2, r3) with a log-linear model. Use maximum likelihood analysis, and include the main effects and the r1*r2 interaction in the model. Obtain the predicted cell frequencies.

```
proc catmod;
   weight wt;
   model r1*r2*r3=_response_ / pred=freq;
   loglin r1|r2 r3;
quit;
```

Logistic Regression

Suppose you want to analyze the relationship between the nominal dependent variable (r) and the independent variables (x1, x2) with a logistic regression analysis. Use maximum likelihood estimation.

```
proc catmod;
   weight wt;
   direct x1 x2;
   model r=x1 x2;
quit;
```

If x1 and x2 are continuous so that each observation has a unique value of these two variables, then it may be more appropriate to use the LOGISTIC, GENMOD, or PROBIT procedure. See the "Logistic Regression" section on page 869.

Repeated Measures Analysis

Suppose the dependent variables (r1, r2, r3) represent the same type of measurement taken at three different times. Analyze the relationship among the dependent variables, the repeated measurement factor (time), and the independent variable (a).

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2*r3=_response_|a;
   repeated time 3 / _response_=time;
quit;
```

Analysis of Variance

Suppose you want to investigate the relationship between the dependent variable (r) and the independent variables (a, b). Analyze the mean of the dependent variable, and include all main effects and interactions in the model.

```
proc catmod;
   weight wt;
   response mean;
   model r=a|b;
quit;
```

Linear Regression

PROC CATMOD can analyze the relationship between the dependent variables (r1, r2) and the independent variables (x1, x2). Use a linear regression analysis to analyze the marginal probabilities of the dependent variables.

```
proc catmod;
   weight wt;
   direct x1 x2;
   response marginals;
   model r1*r2=x1 x2;
quit;
```

Logistic Analysis of Ordinal Data

Suppose you want to analyze the relationship between the ordinally scaled dependent variable (r) and the independent variable (a). Use cumulative logits to take into account the ordinal nature of the dependent variable. Use weighted least-squares estimation.

```
proc catmod;
   weight wt;
   response clogits;
   model r=_response_ a;
quit;
```

Sample Survey Analysis

Suppose the data set contains estimates of a vector of four functions and their covariance matrix, estimated in such a way as to correspond to the sampling process that is used. Analyze the functions with respect to the independent variables (a, b), and use a main-effects model.

```
proc catmod;
   response read b1-b10;
   model _f_=_response_;
   factors  a 2 , b 5 / _response_=a b;
quit;
```

Background: The Underlying Model

The CATMOD procedure analyzes data that can be represented by a two-dimensional contingency table. The rows of the table correspond to populations (or samples) formed on the basis of one or more independent variables. The columns of the table correspond to observed responses formed on the basis of one or more dependent variables. The frequency in the (i, j)th cell is the number of subjects in the ith population that have the jth response. The frequencies in the table are assumed to follow a product multinomial distribution, corresponding to a sampling design in which a simple random sample is taken for each population. The contingency table can be represented as shown in Table 22.1.

Table 22.1. Contingency Table Representation

	Response				
Sample	1	2	$\cdots$	r	**Total**
1	n_{11}	n_{12}	$\cdots$	n_{1r}	n_1
2	n_{21}	n_{22}	$\cdots$	n_{2r}	n_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s	n_{s1}	n_{s2}	$\cdots$	n_{sr}	n_s

For each sample i, the probability of the jth response (π_{ij}) is estimated by the sample proportion, $p_{ij} = n_{ij}/n_i$. The vector (p) of all such proportions is then transformed into a vector of functions, denoted by $\mathbf{F} = \mathbf{F}(\mathbf{p})$. If π denotes the vector of true probabilities for the entire table, then the functions of the true probabilities, denoted by $\mathbf{F}(\pi)$, are assumed to follow a linear model

$$\mathbf{E_A}(\mathbf{F}) = \mathbf{F}(\pi) = \mathbf{X}\boldsymbol{\beta}$$

where $\mathbf{E_A}$ denotes asymptotic expectation, $\mathbf{X}$ is the design matrix containing fixed constants, and $\boldsymbol{\beta}$ is a vector of parameters to be estimated.

PROC CATMOD provides two estimation methods:

- The maximum likelihood method estimates the parameters of the linear model so as to maximize the value of the joint multinomial likelihood function of the responses. Maximum likelihood estimation is available only for the standard response functions, logits and generalized logits, which are used for logistic regression analysis and log-linear model analysis. Two methods of maximization are available: Newton-Raphson and iterative proportional fitting. For details of the theory, refer to Bishop, Fienberg, and Holland (1975).
- The weighted least-squares method minimizes the weighted residual sum of squares for the model. The weights are contained in the inverse covariance matrix of the functions $\mathbf{F}(\mathbf{p})$. According to central limit theory, if the sample sizes within populations are sufficiently large, the elements of $\mathbf{F}$ and $\mathbf{b}$ (the estimate of $\boldsymbol{\beta}$) are distributed approximately as multivariate normal. This allows the computation of statistics for testing the goodness of fit of the model and the significance of other sources of variation. For details of the theory, refer to Grizzle, Starmer, and Koch (1969) or Koch et al. (1977, Appendix 1). Weighted least-squares estimation is available for all types of response functions.

Following parameter estimation, hypotheses about linear combinations of the parameters can be tested. For that purpose, PROC CATMOD computes generalized Wald (1943) statistics, which are approximately distributed as chi-square if the sample sizes are sufficiently large and the null hypotheses are true.

Linear Models Contrasted with Log-Linear Models

Linear model methods (as typified by the Grizzle, Starmer, Koch approach) make a very clear distinction between independent and dependent variables. The emphasis of these methods is estimation and hypothesis testing of the model parameters. Therefore, it is easy to test for differences among probabilities, perform repeated measurement analysis, and test for marginal homogeneity, but it is awkward to test independence and generalized independence. These methods are a natural extension of the usual ANOVA approach for continuous data.

In contrast, log-linear model methods (as typified by the Bishop, Fienberg, Holland approach) do not make an a priori distinction between independent and dependent variables, although model specifications that allow for the distinction can be made. The emphasis of these methods is on model building, goodness-of-fit tests, and estimation of cell frequencies or probabilities for the underlying contingency table. With these methods, it is easy to test independence and generalized independence, but it is awkward to test for differences among probabilities, do repeated measurement analysis, and test for marginal homogeneity.

Using PROC CATMOD Interactively

You can use the CATMOD procedure interactively. After specifying a model with a MODEL statement and running PROC CATMOD with a RUN statement, you can execute any statement without reinvoking PROC CATMOD. You can execute the statements singly or in groups by following the single statement or group of statements with a RUN statement. Note that you can use more than one MODEL statement; this is an important difference from the GLM procedure.

If you use PROC CATMOD interactively, you can end the CATMOD procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you are using PROC CATMOD interactively, additional RUN statements do not end the procedure but tell the procedure to execute additional statements.

When the CATMOD procedure detects a BY statement, it disables interactive processing; that is, once the BY statement and the next RUN statement are encountered, processing proceeds for each BY group in the data set, and no additional statements are accepted by the procedure. For example, the following statements tell PROC CATMOD to do three analyses: one for the entire data set, one for males, and one for females.

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2=a|b;
run;
   by sex;
run;
```

Note that the BY statement may appear after the first RUN statement; this is an important difference from PROC GLM, which requires that the BY statement appear before the first RUN statement.

Getting Started

The CATMOD procedure is a general modeling procedure for categorical data analysis, and it can be used for very sophisticated analyses that require matrix specification of the response function and the design matrix. It can be used to perform very basic analysis-of-variance-type analyses that require very few statements. The following is a basic example.

Weighted-Least-Squares Analysis of Mean Response

Consider the data in the following table (Stokes, Davis, and Koch 2000).

Table 22.2. Colds in Children

		Periods with Colds			
Sex	**Residence**	**0**	**1**	**2**	**Total**
Female	Rural	45	64	71	180
Female	Urban	80	104	116	300
Male	Rural	84	124	82	290
Male	Urban	106	117	87	310

For males and females in rural and urban counties, the number of periods (of two) in which subjects report cold symptoms are recorded. Thus, 45 subjects who were female and in rural counties report no cold symptoms, and 71 subjects who are female and from rural counties report colds in both periods.

The question of interest is whether the mean number of periods with colds reported is associated with gender or type of county. There is no reason to believe that the mean number of periods with colds is normally distributed, so a weighted least-squares analysis of these data is performed with PROC CATMOD instead of an analysis of variance with PROC ANOVA or PROC GLM.

The input data for categorical data is often recorded in frequency form, with the counts for each particular profile being the input values. Thus, for the colds data, the input SAS data set colds is created with the following statements. The variable count contains the frequency of observations that have the particular profile described by the values of the other variables on that input line.

```
data colds;
   input sex $ residence $ periods count @@;
datalines;
female rural 0  45  female rural 1  64  female rural 2  71
female urban 0  80  female urban 1 104  female urban 2 116
male   rural 0  84  male   rural 1 124  male   rural 2  82
male   urban 0 106  male   urban 1 117  male   urban 2  87
;
run;
```

In order to fit a model to the mean number of periods with colds, you have to specify the response function in PROC CATMOD. The default response function is the logit if the response variable has two values, and it is generalized logits if the response variable has more than two values. If you want a different response function, then you request that function in the RESPONSE statement. To request the mean number of periods with colds, you specify the MEANS option in the RESPONSE statement.

You can request a model consisting of the main effects and interaction of the variables sex and residence just as you would in the GLM procedure. Unlike the GLM procedure, you do not need to use a CLASS statement in PROC CATMOD to treat a variable as a classification variable. All variables in the MODEL statement in the

CATMOD procedure are treated as classification variables unless you specify otherwise with a DIRECT statement. To verify that your model is specified correctly, you can specify the DESIGN option in the MODEL statement to display the design matrix.

Thus, the PROC CATMOD statements needed to model mean periods of colds with a main effects and interaction model are

```
proc catmod data=colds;
   weight count;
   response means;
   model periods = sex residence sex*residence / design;
run;
```

The results of this analysis are shown in Figure 22.1 through Figure 22.3.

```
                              The CATMOD Procedure

                                  Data Summary

               Response           periods     Response Levels     3
               Weight Variable    count       Populations         4
               Data Set           COLDS       Total Frequency  1080
               Frequency Missing  0           Observations       12


                              Population Profiles

                   Sample    sex       residence    Sample Size
                   ---------------------------------------------
                       1     female    rural                180
                       2     female    urban                300
                       3     male      rural                290
                       4     male      urban                310


                               Response Profiles

                              Response    periods
                              -------------------
                                  1       0
                                  2       1
                                  3       2
```

Figure 22.1. Model Information and Profile Tables

The CATMOD procedure first displays a summary of the contingency table you are analyzing. The “Population Profiles” table lists the values of the explanatory variables that define each population, or row of the underlying contingency table, and labels each group with a sample number. The number of observations in each population is also displayed. The “Response Profiles” table lists the variable levels that define the response, or columns of the underlying contingency table.

```
                      Response Functions and Design Matrix

                     Response                Design Matrix
            Sample      Function         1         2         3         4
            --------------------------------------------------------
                1        1.14444         1         1         1         1
                2        1.12000         1         1        -1        -1
                3        0.99310         1        -1         1        -1
                4        0.93871         1        -1        -1         1
```

Figure 22.2. Observed Response Functions and Design Matrix

The “Design Matrix” table contains the observed response functions—in this case, the mean number of periods with colds for each of the populations—and the design matrix. The first column of the design matrix contains the coefficients for the intercept parameter, the second column coefficients are for the sex parameter (note that the sum-to-zero constraint of a full-rank parameterization implies that the coefficient for males is the negative of that for females. The parameter is called the *differential effect* for females), the third column is similarly set up for residence, and the last column is for the interaction.

```
                          Analysis of Variance

                Source             DF   Chi-Square    Pr > ChiSq
                -----------------------------------------------
                Intercept           1      1841.13        <.0001
                sex                 1        11.57        0.0007
                residence           1         0.65        0.4202
                sex*residence       1         0.09        0.7594

                Residual            0          .             .
```

Figure 22.3. ANOVA Table for the Saturated Model

The model-fitting results are displayed in the “Analysis of Variance” table (Figure 22.3), which is similar to an ANOVA table. The effects from the right-hand side of the MODEL statement are listed under the “Source” column.

The interaction effect is nonsignificant, so the data are reanalyzed using a main-effects model. Since PROC CATMOD is an interactive procedure, you can analyze the main-effects model by simply submitting the new MODEL statement as follows. The resulting tables are displayed in Figure 22.4 through Figure 22.7.

```
proc catmod data=colds;
   weight count;
   response means;
   model periods = sex residence / design;
run;
```

```
                              The CATMOD Procedure

                                  Data Summary

               Response           periods     Response Levels     3
               Weight Variable    count       Populations         4
               Data Set           COLDS       Total Frequency  1080
               Frequency Missing  0           Observations       12


                              Population Profiles

                   Sample    sex       residence    Sample Size
                   --------------------------------------------
                       1     female    rural                180
                       2     female    urban                300
                       3     male      rural                290
                       4     male      urban                310


                               Response Profiles

                              Response    periods
                              -------------------
                                  1       0
                                  2       1
                                  3       2
```

Figure 22.4. Population and Response Profiles, Main-Effects Model

```
                     Response Functions and Design Matrix

                         Response         Design Matrix
                 Sample      Function        1        2        3
                 -----------------------------------------------
                    1        1.14444        1        1        1
                    2        1.12000        1        1       -1
                    3        0.99310        1       -1        1
                    4        0.93871        1       -1       -1
```

Figure 22.5. Design Matrix for the Main-Effects Model

```
                             Analysis of Variance

                   Source        DF   Chi-Square    Pr > ChiSq
                   -------------------------------------------
                   Intercept      1      1882.77        <.0001
                   sex            1        12.08        0.0005
                   residence      1         0.76        0.3839

                   Residual       1         0.09        0.7594
```

Figure 22.6. ANOVA Table for the Main-Effects Model

The goodness-of-fit chi-square statistic is 0.09 with one degree of freedom and a p-value of 0.7594; hence, the model fits the data. Note that the chi-square tests in Figure

22.6 test whether all the parameters for a given effect are zero. In this model, each effect has only one parameter, and therefore only one degree of freedom.

```
                 Analysis of Weighted Least Squares Estimates

                                      Standard        Chi-
     Parameter           Estimate       Error       Square    Pr > ChiSq
     --------------------------------------------------------------------
     Intercept             1.0501      0.0242      1882.77        <.0001
     sex       female      0.0842      0.0242        12.08        0.0005
     residence rural       0.0210      0.0241         0.76        0.3839
```

Figure 22.7. Parameter Estimates for the Main-Effects Model

The "Analysis of Weighted-Least-Squares Estimates" table lists the parameters and their estimates for the model, as well as the standard errors, Wald statistics, and p-values. These chi-square tests are single degree-of-freedom tests that the individual parameter is equal to zero. They are equal to the tests shown in Figure 22.6 since each effect is composed of exactly one parameter.

You can compute the mean number of periods of colds for the first population (Sample 1, females in rural residences) from Table 22.2 as follows.

$$\text{mean colds} = 0 \times \frac{45}{180} + 1 \times \frac{64}{180} + 2 \times \frac{71}{180} = 1.1444$$

This is the same value as reported for the Response Function for Sample 1 in Figure 22.5.

PROC CATMOD is fitting a model to the mean number of colds in each population as follows:

$$\begin{bmatrix} \text{Expected number of colds for rural females} \\ \text{urban females} \\ \text{rural males} \\ \text{urban males} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

where the design matrix is the same one displayed in Figure 22.5, β_0 is the mean number of colds averaged over all the populations, β_1 is the differential effect for females, and β_2 is the differential effect for rural residences. The parameter estimates are shown in Figure 22.7; thus, the expected number of periods with colds for rural females from this model is

$$1 \times 1.0501 + 1 \times 0.0842 + 1 \times 0.0210 = 1.1553$$

and the expected number for rural males from this model is

$$1 \times 1.0501 - 1 \times 0.0842 + 1 \times 0.0210 = 0.9869$$

Notice also, in Figure 22.7, that the differential effect for residence is nonsignificant ($p = 0.3839$): If you continue the analysis by fitting a single effect model (sex), you need to include a POPULATION statement to maintain the same underlying contingency table.

```
      population sex residence;
      model periods = sex;
   run;
```

Generalized Logits Model

Over the course of one school year, third-graders from three different schools are exposed to three different styles of mathematics instruction: a self-paced computer-learning style, a team approach, and a traditional class approach. The students are asked which style they prefer, and their responses, classified by the type of program they are in (a regular school day versus a regular day supplemented with an afternoon school program), are displayed in Table 22.3. The data set is from Stokes, Davis, and Koch (2000), and it is also analyzed in Example 42.4 on page 2416 of Chapter 42, "The LOGISTIC Procedure," .

Table 22.3. School Program Data

		Learning Style Preference		
School	**Program**	**Self**	**Team**	**Class**
1	Regular	10	17	26
1	Afternoon	5	12	50
2	Regular	21	17	26
2	Afternoon	16	12	36
3	Regular	15	15	16
3	Afternoon	12	12	20

The levels of the response variable (self, team, and class) have no essential ordering, hence a logistic regression is performed on the generalized logits. The model to be fit is

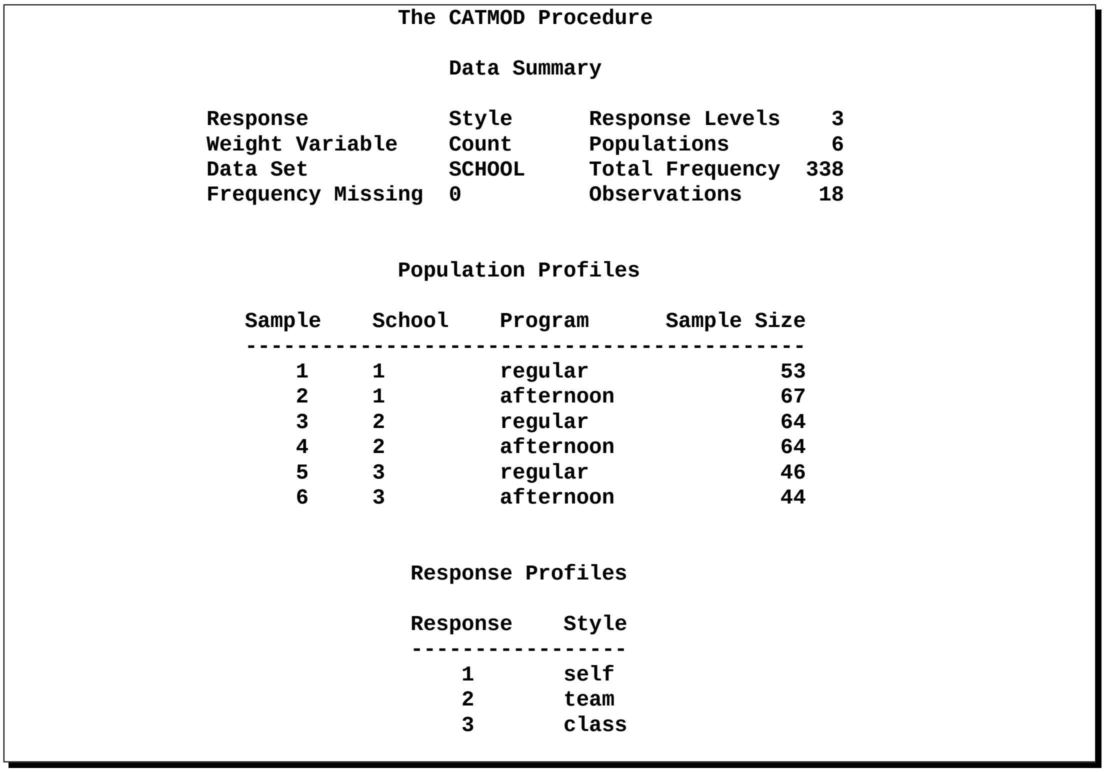
$$\log\left(\frac{\pi_{hij}}{\pi_{hir}}\right) = \alpha_j + \mathbf{x}'_{hi}\beta_j$$

where π_{hij} is the probability that a student in school h and program i prefers teaching style j, $j \neq r$, and style r is the class style. There are separate sets of intercept parameters α_j and regression parameters β_j for each logit, and the matrix $\mathbf{x}_{hi}$ is the set of explanatory variables for the hith population. Thus, two logits are modeled for each school and program combination (population): the logit comparing self to class and the logit comparing team to class.

The following statements create the data set school and request the analysis. Generalized logits are the default response functions, and maximum likelihood estimation is the default method for analyzing generalized logits, so only the WEIGHT and MODEL statements are required. The option ORDER=DATA means that the response variable levels are ordered as they exist in the data set: self, team, and class; thus the logits are formed by comparing self to class and by comparing team to class. The results of this analysis are shown in Figure 22.8 and Figure 22.9.

```
data school;
   length Program $ 9;
   input School Program $ Style $ Count @@;
   datalines;
1 regular   self 10  1 regular   team 17  1 regular   class 26
1 afternoon self  5  1 afternoon team 12  1 afternoon class 50
2 regular   self 21  2 regular   team 17  2 regular   class 26
2 afternoon self 16  2 afternoon team 12  2 afternoon class 36
3 regular   self 15  3 regular   team 15  3 regular   class 16
3 afternoon self 12  3 afternoon team 12  3 afternoon class 20
;
```

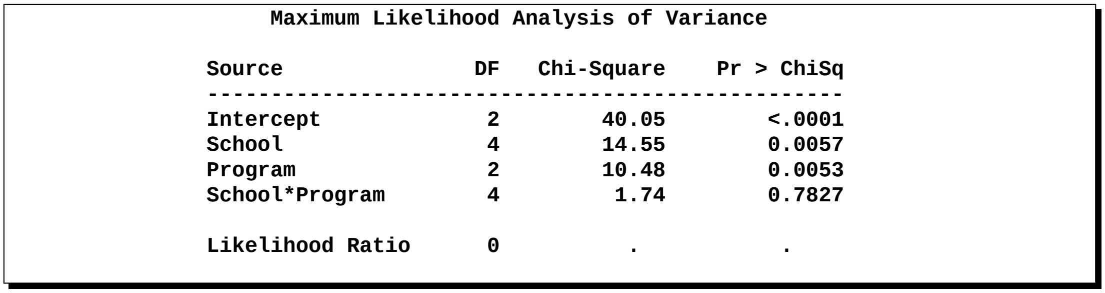
```
proc catmod order=data;
   weight Count;
   model Style=School Program School*Program;
run;
```



```
                         The CATMOD Procedure

                             Data Summary

          Response            Style      Response Levels    3
          Weight Variable     Count      Populations        6
          Data Set            SCHOOL     Total Frequency  338
          Frequency Missing   0          Observations      18


                         Population Profiles

             Sample    School    Program      Sample Size
             -------------------------------------------
                 1     1         regular               53
                 2     1         afternoon             67
                 3     2         regular               64
                 4     2         afternoon             64
                 5     3         regular               46
                 6     3         afternoon             44


                          Response Profiles

                          Response    Style
                          -----------------
                              1       self
                              2       team
                              3       class
```

Figure 22.8. Model Information and Profile Tables

A summary of the data set is displayed in Figure 22.8; the variable levels that form the three responses and six populations are listed in the “Response Profiles” and “Population Profiles” table, respectively.



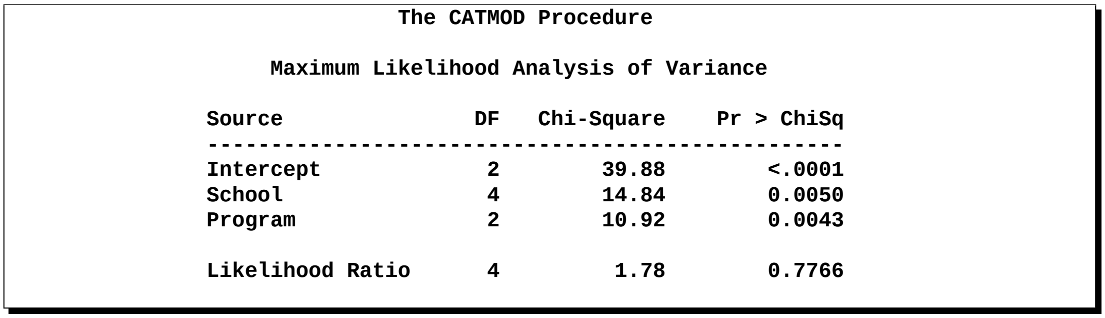
```
                    Maximum Likelihood Analysis of Variance

                Source              DF   Chi-Square    Pr > ChiSq
                --------------------------------------------------
                Intercept            2        40.05        <.0001
                School               4        14.55        0.0057
                Program              2        10.48        0.0053
                School*Program       4         1.74        0.7827

                Likelihood Ratio     0          .            .
```

Figure 22.9. ANOVA Table

The analysis of variance table is displayed in Figure 22.9. Since this is a saturated model, there are no degrees of freedom remaining for a likelihood ratio test, and missing values are displayed in the table. The interaction effect is clearly nonsignificant, so a main effects model is fit.

Since PROC CATMOD is an interactive procedure, you can analyze the main effects model by simply submitting the new MODEL statement as follows.

```
model Style=School Program;
run;
```



```
                              The CATMOD Procedure

                    Maximum Likelihood Analysis of Variance

                Source              DF   Chi-Square    Pr > ChiSq
                --------------------------------------------------
                Intercept            2        39.88        <.0001
                School               4        14.84        0.0050
                Program              2        10.92        0.0043

                Likelihood Ratio     4         1.78        0.7766
```

Figure 22.10. ANOVA Table

You can check the population and response profiles (not shown) to confirm that they are the same as those in Figure 22.8. The analysis of variance table is shown in Figure 22.10. The likelihood ratio chi-square statistic is 1.78 with a p-value of 0.7766, indicating a good fit; the Wald chi-square tests for the school and program effects are also significant. Since School has three levels, two parameters are estimated for each of the two logits they modeled, for a total of four degrees of freedom. Since Program has two levels, one parameter is estimated for each of the two logits, for a total of two degrees of freedom.

```
                    Analysis of Maximum Likelihood Estimates

                     Function                Standard        Chi-
Parameter             Number      Estimate      Error       Square    Pr > ChiSq
--------------------------------------------------------------------------------
Intercept                1         -0.7979     0.1465        29.65        <.0001
                         2         -0.6589     0.1367        23.23        <.0001
School     1             1         -0.7992     0.2198        13.22        0.0003
           1             2         -0.2786     0.1867         2.23        0.1356
           2             1          0.2836     0.1899         2.23        0.1352
           2             2         -0.0985     0.1892         0.27        0.6028
Program    regular       1          0.3737     0.1410         7.03        0.0080
           regular       2          0.3713     0.1353         7.53        0.0061
```

Figure 22.11. Parameter Estimates

The parameter estimates and tests for individual parameters are displayed in Figure 22.11. The ordering of the parameters corresponds to the order of the population and response variables as shown in the profile tables (see Figure 22.8), with the levels of the response variables varying most rapidly. So, for the first response function, which is the logit that compares self to class, Parameter 1 is the intercept, Parameter 3 is the parameter for the differential effect for School=1, Parameter 5 is the parameter for the differential effect for School=2, and Parameter 7 is the parameter for the differential effect for Program=regular. The even parameters are interpreted similarly for the second logit, which compares team to class.

The Program variable (Parameters 7 and 8) has nearly the same effect on both logits, while School=1 (Parameters 3 and 4) has the largest effect of the schools.

Syntax

The following statements are available in PROC CATMOD.

PROC CATMOD < *options* > ;
DIRECT < *variables* > ;
MODEL *response-effect=design-effects* < / *options* > ;
CONTRAST *'label' row-description* <, . . . , *row-description* >
< / *options* > ;
BY *variables* ;
FACTORS *factor-description* <, . . . , *factor-description* >
< / *options* > ;
LOGLIN *effects* ;
POPULATION *variables* ;
REPEATED *factor-description* <, . . . , *factor-description* >
< / *options* > ;
RESPONSE *function* <, . . . , *function* >< / *options* > ;
RESTRICT *parameter=value* < . . . *parameter=value* > ;
WEIGHT *variable* ;

You can use all of the statements in PROC CATMOD interactively. The first RUN statement executes all of the previous statements. Any subsequent RUN statement executes only those statements that appear between the previous RUN statement and the current one. However, if you specify a BY statement, interactive processing is disabled. That is, all statements through the following RUN statement are processed for each BY group in the data set, but no additional statements are accepted by the procedure.

If more than one CONTRAST statement appears between two RUN statements, all the CONTRAST statements are processed. If more than one RESPONSE statement appears between two RUN statements, then analyses associated with each RESPONSE statement are produced. For all other statements, there can be only one occurrence of the statement between any two RUN statements. For example, if there are two LOGLIN statements between two RUN statements, the first LOGLIN statement is ignored.

The PROC CATMOD and MODEL statements are required. If specified, the DIRECT statement must precede the MODEL statement. As a result, if you use the DIRECT statement interactively, you need to specify a MODEL statement in the same RUN group. See the section "DIRECT Statement" on page 835 for an example.

The CONTRAST statements, if any, must follow the MODEL statement.

You can specify only one of the LOGLIN, REPEATED, and FACTORS statements between any two RUN statements, because they all specify the same information: how to partition the variation among the response functions within a population.

A QUIT statement executes any statements that have not been processed and then ends the CATMOD procedure.

The purpose of each statement, other than the PROC CATMOD statement, are summarized in the following list:

BY determines groups in which data are to be processed separately.

CONTRAST specifies a hypothesis to test.

DIRECT specifies independent variables that are to be treated quantitatively (like continuous variables) rather than qualitatively (like class or discrete variables). These variables also help to determine the rows of the contingency table and distinguish response functions in one population from those in other populations.

FACTORS specifies (1) the factors that distinguish response functions from others in the same population and (2) model effects, based on these factors, which help to determine the design matrix.

LOGLIN specifies log-linear model effects.

MODEL specifies (1) dependent variables, which determine the columns of the contingency table, (2) independent variables, which distinguish response functions in one population from those in other populations, and (3) model effects, which determine the design matrix and the way in which total variation among the response functions is partitioned.

POPULATION specifies variables which determine the rows of the contingency table and distinguish response functions in one population from those in other populations.

REPEATED specifies (1) the repeated measurement factors that distinguish response functions from others in the same population and (2) model effects, based on these factors, which help to determine the design matrix.

RESPONSE determines the response functions that are to be modeled.

RESTRICT restricts values of parameters to the values you specify.

WEIGHT specifies a variable containing frequency counts.

PROC CATMOD Statement

PROC CATMOD < *options* > **;**

The PROC CATMOD statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. By default, the CATMOD procedure uses the most recently created SAS data set. For details, see the section "Input Data Sets" on page 860.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 24 and 200 characters. The default length is 24 characters.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you only want to create output data sets with the OUT= or OUTEST= option in the RESPONSE statement. A NOPRINT option is also available in the MODEL statement. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This affects the ordering of the populations, responses, and parameters, as well as the definitions of the parameters. The default, ORDER=INTERNAL, orders the variable levels by their unformatted values (for example, numeric order or alphabetical order).

The following table shows how PROC CATMOD interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. See the section "Ordering of Populations and Responses" on page 863 for more information and examples. For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CATMOD to obtain separate analyses of groups determined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CATMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

When you specify a BY statement with PROC CATMOD, no further interactive processing is possible. In other words, once the BY statement appears, all statements up to the associated RUN statement are executed for each BY group in the data set. After the RUN statement, no further statements are accepted by the procedure.

CONTRAST Statement

CONTRAST *'label' row-description* < , . . . , *row-description* >< / *options* > **;**

where a *row-description* is

< @*n* > *effect values* < . . . < @*n* > *effect values* >

The CONTRAST statement constructs and tests linear functions of the parameters in the MODEL statement or effects listed in the LOGLIN statement. Each set of effects (separated by commas) specifies one row or set of rows of the matrix $\mathbf{C}$ that PROC CATMOD uses to test the hypothesis $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$.

CONTRAST statements must be preceded by the MODEL statement, and by the LOGLIN statement, if one is used. You can specify the following terms in the CONTRAST statement.

'label' specifies up to 256 characters of identifying information displayed with the test. The '*label*' is required.

effect is one of the effects specified in the MODEL or LOGLIN statement, INTERCEPT (for the intercept parameter), or ALL_PARMS (for the complete set of parameters).

The ALL_PARMS option is regarded as an effect with the same number of parameters as the number of columns in the design matrix. This is particularly useful when the design matrix is input directly, as in the following example:

```
model y=(1 0 0 0,
         1 0 1 0,
         1 1 0 0,
         1 1 1 1);
contrast 'Main Effect of B' all_parms 0 1 0 0;
contrast 'Main Effect of C' all_parms 0 0 1 0;
contrast 'B*C Interaction ' all_parms 0 0 0 1;
```

values are numbers that form the coefficients of the parameters associated with the given effect. If there are fewer values than parameters for an effect, the remaining coefficients become zero. For example, if you specify two values and the effect actually has five parameters, the final three are set to zero.

@n points to the parameters in the nth set when the model has a separate set of parameters for each of the response functions. The @n notation is seldom needed. It enables you to test the variation among response functions in the same population. However, it is usually easier to model and test such variation by using the _RESPONSE_ effect in the MODEL statement or by using the ALL_PARMS designation. Usually, contrasts are performed with respect to all of the response functions, and this is what the CONTRAST statement does by default (in this case, do not use the @n notation).

For example, if there are three response functions per population, then

```
contrast 'Level 1 vs. Level 2' A 1 -1 0;
```

results in a three-degree-of-freedom test comparing the first two levels of A simultaneously on the three response functions.

If, however, you want to specify a contrast with respect to the parameters in the nth set only, then use a single $@n$ in a *row-description*. For example, to test that the first parameter of A and the first parameter of B are zero in the third response function, specify

```
contrast 'A=0, B=0, Function 3'  @3  A 1  B 1;
```

To specify a contrast with respect to parameters in two or more different sets of effects, use $@n$ with each effect. For example,

```
contrast 'Average over Functions' @1 A 1 0 -1
                                  @2 A 1 1 -2;
```

When the model does not have a separate set of parameters for each of the response functions, the $@n$ notation is invalid. This type of model is called AVERAGED. For details, see the description of the AVERAGED option on page 842 and the "Generation of the Design Matrix" section on page 876.

You can specify the following options in the CONTRAST statement after a slash.

ALPHA= *value*

specifies the significance level of the confidence interval for each contrast when the ESTIMATE= option is specified. The default is ALPHA=0.05, resulting in a 95% confidence interval for each contrast.

ESTIMATE=*keyword*
EST=*keyword*

requests that each individual contrast (that is, each row, $c_i\boldsymbol{\beta}$, of $\mathbf{C}\boldsymbol{\beta}$) or exponentiated contrast $(\exp(c_i\boldsymbol{\beta}))$ be estimated and tested. PROC CATMOD displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option.

You can estimate the contrast or the exponentiated contrast, or both, by specifying one of the following keywords:

PARM	specifies that the contrast itself be estimated.
EXP	specifies that the exponentiated contrast be estimated.
BOTH	specifies that both the contrast and the exponentiated contrast be estimated.

Specifying Contrasts

PROC CATMOD is parameterized differently than PROC GLM, so you must be careful not to use the same contrasts that you would with PROC GLM. Since PROC CATMOD uses a full-rank parameterization, all estimable parameters are directly estimable without involving other parameters.

For example, suppose a class variable A has four levels. Then there are four parameters ($\alpha_1, \alpha_2, \alpha_3, \alpha_4$), of which PROC CATMOD uses only the first three. The fourth parameter is related to the others by the equation

$$\alpha_4 = -\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test $\alpha_1 = \alpha_4$, which is

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements are constructed similarly; for example,

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

The actual form of the **C** matrix depends on the effects in the model. The following examples assume a single response function for each population.

```
proc catmod;
   model y=a;
   contrast '1 vs. 4' A 2 1 1;
run;
```

The **C** matrix for the preceding statements is

$$\mathbf{C} = [\,0\;2\;1\;1\,]$$

since the first parameter corresponds to the intercept.

But if there is a variable B with three levels and you use the following statements,

```
proc catmod;
   model y=b a;
   contrast '1 vs. 4' A 2 1 1;
run;
```

then the CONTRAST statement induces the **C** matrix

$$\mathbf{C} = [\,0\;0\;0\;2\;1\;1\,]$$

since the first parameter corresponds to the intercept and the next two correspond to the B main effect.

You can also use the CONTRAST statement to test the joint effect of two or more effects in the MODEL statement. For example, the joint effect of A and B in the previous model has five degrees of freedom and is obtained by specifying

```
contrast 'Joint Effect of A&B' A 1 0 0,
                               A 0 1 0,
                               A 0 0 1,
                               B 1 0,
                               B 0 1;
```

The ordering of variable levels is determined by the ORDER= option in the PROC CATMOD statement. Whenever you specify a contrast that depends on the order of the variable levels, you should verify the order from the “Population Profiles” table, the “Response Profiles” table, or the “One-Way Frequencies” table.

DIRECT Statement

DIRECT *variables* **;**

The DIRECT statement lists numeric independent variables to be treated in a quantitative, rather than qualitative, way. The DIRECT statement is useful for logistic regression, which is described in the "Logistic Regression" section on page 869. For limitations of models involving continuous variables, see the "Continuous Variables" section on page 870.

If a DIRECT variable is formatted, then the unformatted (internal) values are used in the analysis and the formatted values are displayed. **CAUTION:** If you use a format to group the internal values into one formatted value, then the first internal value is used in the analysis.

If specified, the DIRECT statement must precede the MODEL statement. For example,

```
proc catmod;
   direct X;
   model Y=X;
run;
```

Suppose X has five levels. Then the main effect X induces only one column in the design matrix, rather than four. The values inserted into the design matrix are the actual values of X.

You can interactively change the variables declared as DIRECT variables by using the statement without listing any variables. The following statements are valid:

```
proc catmod;
   direct X;
   model Y=X;
   weight wt;
run;
   direct;
   model Y=X;
run;
```

The first MODEL statement uses the actual values of X, and the second MODEL statement uses the four variables created when PROC CATMOD generates the design matrix. Note that the preceding statements can be run without a WEIGHT statement if the input data are raw data rather than cell counts.

For more details, see the discussions of main and direct effects in the section "Generation of the Design Matrix" on page 876.

FACTORS Statement

FACTORS *factor-description* <, . . . , *factor-description* >< / *options* > **;**

where a *factor-description* is

factor-name < *$* >< *levels* >

and *factor-description*s are separated from each other by a comma. The $ is required for character-valued factors. The value of *levels* provides the number of levels of the factor identified by a given *factor-name*. For only one factor, *levels* is optional; for two or more factors, it is required.

The FACTORS statement identifies factors that distinguish response functions from others in the same population. It also specifies how those factors are incorporated into the model. You can use the FACTORS statement whenever there is more than one response function per population and the keyword _RESPONSE_ is specified in the MODEL statement. You can specify the name, type, and number of levels of each factor and the identification of each level.

The FACTORS statement is most useful when the response functions and their covariance matrix are read directly from the input data set. In this case, PROC CATMOD reads the response functions as though they are from one population (this poses no problem in the multiple-population case because the appropriately constructed covariance matrix is also read directly). Thus, you can use the FACTORS statement to partition the variation among the response functions into appropriate sources, even when the functions actually represent separate populations.

The format of the FACTORS statement is identical to that of the REPEATED statement. In fact, repeated measurement factors are simply special cases of factors in which some of the response functions correspond to multiple dependent variables that are measurements on the same experimental (or sampling) units.

You cannot specify the FACTORS statement for an analysis that also contains the REPEATED or LOGLIN statement since all of them specify the same information: how to partition the variation among the response functions within a population.

In the FACTORS statement,

factor-name names a factor that corresponds to two or more response functions. This name must be a valid SAS variable name, and it should not be the same as the name of a variable that already exists in the data set being analyzed.

$ indicates that the factor is character-valued. If the $ is omitted, then PROC CATMOD assumes that the factor is numeric. The type of the factor is relevant only when you use the PROFILE= option or when the _RESPONSE_= option (described later in this section) specifies nested-by-value effects.

levels specifies the number of levels of the corresponding factor. If there is only one such factor, and the number is omitted, then PROC

CATMOD assumes that the number of levels is equal to the number of response functions per population (q). Unless you specify the PROFILE= option, the number q must either be equal to or be a multiple of the product of the number of levels of all the factors.

You can specify the following options in the FACTORS statement after a slash.

PROFILE=(*matrix*)

specifies the values assumed by the factors for each response function. There should be one column for each factor, and the values in a given column (character or numeric) should match the type of the corresponding factor. Character values are restricted to 16 characters or less. If there are q response functions per population, then the matrix must have i rows, where q must either be equal to or be a multiple of i. Adjacent rows of the matrix should be separated by a comma.

The values in the PROFILE matrix are useful for specifying models in those situations where the study design is not a full factorial with respect to the factors. They can also be used to specify nested-by-value effects in the _RESPONSE_= option. If you specify character values in both places (the PROFILE= option and the _RESPONSE_= option), then the values must match with respect to whether or not they are enclosed in quotes (that is, enclosed in quotes in both places or in neither place).

For an example of using the PROFILE= option, see Example 22.10 on page 944.

RESPONSE=*effects*

specifies design effects. The variables named in the effects must be *factor-names* that appear in the FACTORS statement. If the _RESPONSE_= option is omitted, then PROC CATMOD builds a full factorial _RESPONSE_ effect with respect to the factors.

TITLE=*'title'*

displays the *title* at the top of certain pages of output that correspond to the current FACTORS statement.

For an example of how the FACTORS statement is useful, consider the case where the response functions and their covariance matrix are read directly from the input data set. The TYPE=EST data set might be created in the following manner:

```
data direct(type=est);
   input b1-b4 _type_ $ _name_ $8.;
   datalines;
0.590463   0.384720   0.273269   0.136458   parms     .
0.001690   0.000911   0.000474   0.000432   cov       b1
0.000911   0.001823   0.000031   0.000102   cov       b2
0.000474   0.000031   0.001056   0.000477   cov       b3
0.000432   0.000102   0.000477   0.000396   cov       b4
;
```

Suppose the response functions correspond to four populations that represent the cross-classification of age (two groups) by sex. You can use the FACTORS statement to identify these two factors and to name the effects in the model. The statements required to fit a main-effects model to these data are

```
proc catmod data=direct;
   response read b1-b4;
   model _f_=_response_;
   factors age 2, sex 2 / _response_=age sex;
run;
```

If you want to specify some nested-by-value effects, you can change the FACTORS statement to

```
factors age $ 2, sex $ 2 /
       _response_=age sex(age='under 30') sex(age='30 & over')
        profile=('under 30'   male,
                 'under 30'   female,
                 '30 & over'  male,
                 '30 & over'  female);
```

If, by design or by chance, the study contains no male subjects under 30 years of age, then there are only three response functions, and you can specify a main-effects model as

```
proc catmod data=direct;
   response read b2-b4;
   model _f_=_response_;
   factors age $ 2, sex $ 2 / _response_=age sex
          profile=('under 30'   female,
                   '30 & over'  male,
                   '30 & over'  female);
run;
```

When you specify two or more factors and omit the PROFILE= option, PROC CATMOD presumes that the response functions are ordered so that the levels of the rightmost factor change most rapidly. For the preceding example, the order implied by the FACTORS statement is as follows.

Response Function	Dependent Variable	Age	Sex
1	b1	1	1
2	b2	1	2
3	b3	2	1
4	b4	2	2

For additional examples of how to use the FACTORS statement, see the section "Repeated Measures Analysis" on page 873. All of the examples in that section are applicable, with the REPEATED statement replaced by the FACTORS statement.

LOGLIN Statement

LOGLIN *effects* < / *option* > **;**

The LOGLIN statement is used to define log-linear model effects. It can be used whenever the default response functions (generalized logits) are used.

In the LOGLIN statement, *effects* are design effects that contain dependent variables in the MODEL statement, including interaction, nested, and nested-by-value effects. You can use the bar (|) and at (@) operators as well. The following lists of effects are equivalent:

```
a b c a*b a*c b*c
```

and

```
a|b|c @2
```

When you use the LOGLIN statement, the keyword _RESPONSE_ should be specified in the MODEL statement. For further information on log-linear model analysis, see the "Log-Linear Model Analysis" section on page 870.

You cannot specify the LOGLIN statement for an analysis that also contains the REPEATED or FACTORS statement since all of them specify the same information: how to partition the variation among the response functions within a population.

You can specify the following option in the LOGLIN statement after a slash.

TITLE=*'title'*
displays the *title* at the top of certain pages of output that correspond to this LOGLIN statement.

The following statements give an example of how to use the LOGLIN statement.

```
proc catmod;
   model a*b*c=_response_;
   loglin a|b|c @ 2;
run;
```

These statements yield a log-linear model analysis that contains all main effects and two-variable interactions. For more examples of log-linear model analysis, see the "Log-Linear Model Analysis" section on page 870.

MODEL Statement

MODEL *response-effect*=< *design-effects* >< / *options* > **;**

PROC CATMOD requires a MODEL statement. You can specify the following in a MODEL statement:

response-effect can be either a single variable, a crossed effect with two or more variables joined by asterisks, or _F_. The _F_ specification indicates that the response functions and their estimated covariance matrix are to be read directly into the procedure (see the "Inputting Response Functions and Covariances Directly" section on page 862 for details). The *response-effect* indicates the dependent variables that determine the response categories (the columns of the underlying contingency table).

design-effects specify potential sources of variation (such as main effects and interactions) in the model. Thus, these effects determine the number of model parameters, as well as the interpretation of such parameters. In addition, if there is no POPULATION statement, PROC CATMOD uses these variables to determine the populations (the rows of the underlying contingency table). When fitting the model, PROC CATMOD adjusts the independent effects in the model for all other independent effects in the model.

Design-effects can be any of those described in the section "Specification of Effects" on page 864, or they can be defined by specifying the actual design matrix, enclosed in parentheses (see the "Specifying the Design Matrix Directly" section on page 847). In addition, you can use the keyword _RESPONSE_ alone or as part of an effect. Effects cannot be nested within _RESPONSE_, so effects of the form A(_RESPONSE_) are invalid.

For more information, see the "Log-Linear Model Analysis" section on page 870 and the "Repeated Measures Analysis" section on page 873.

Some examples of MODEL statements are

`model r=a b;`	main effects only
`model r=a b a*b;`	main effects with interaction
`model r=a b(a);`	nested effect
`model r=a|b;`	complete factorial
`model r=a b(a=1) b(a=2);`	nested-by-value effects
`model r*s=_response_;`	log-linear model
`model r*s=a _response_(a);`	nested repeated measurement factor
`model _f_=_response_;`	direct input of the response functions

The relationship between these specifications and the structure of the design matrix **X** is described in the "Generation of the Design Matrix" section on page 876.

The following table summarizes the options available in the MODEL statement.

Task	Options
Specify details of computation	
Generates maximum likelihood estimates	ML=
Generates weighted least-squares estimates	GLS
	WLS
Omits intercept term from the model	NOINT
Specifies parameterization of classification variables	PARAM=
Adds a number to each cell frequency	ADDCELL=
Averages main effects across response functions	AVERAGED
Specifies the convergence criterion for maximum likelihood	EPSILON=
Specifies the number of iterations for maximum likelihood	MAXITER=
Specifies how missing cells are treated	MISSING=
Specifies how zero cells are treated	ZERO=
Request additional computation and tables	
Significance level of confidence intervals	ALPHA=
Wald confidence intervals of estimates	CLPARM
Estimated correlation matrix of estimates	CORRB
Covariance matrix of response functions	COV
Estimated covariance matrix of estimates	COVB
Design and _RESPONSE_ matrix	DESIGN
Two-way frequency tables	FREQ
Iterations for maximum likelihood	ITPRINT
One-way frequency tables	ONEWAY
Predicted values	PRED=
	PREDICT
Probability estimates	PROB
Population profiles	PROFILE
Crossproducts matrix	XPX
Title	TITLE=
Suppress output	
Design matrix	NODESIGN
Parameter estimates	NOPARM
Variable levels	NOPREDVAR
Population and response profiles	NOPROFILE
RESPONSE matrix	NORESPONSE

The following list describes these options in alphabetical order.

ADDCELL=*number*

adds *number* to the frequency count in each cell, where *number* is any positive number. This option has no effect on maximum likelihood analysis; it is used only for weighted least-squares analysis.

ALPHA=*number*

sets the significance level for the Wald confidence intervals for parameter estimates. The value must be between 0 and 1. The default value of 0.05 results in the calculation of a 95% confidence interval. This option has no effect unless the CLPARM option is also specified.

AVERAGED

specifies that dependent variable effects can be modeled and that independent variable main effects are averaged across the response functions in a population. For further information on the effect of using (or not using) the AVERAGED option, see the "Generation of the Design Matrix" section on page 876. Direct input of the design matrix or specification of the _RESPONSE_ keyword in the MODEL statement automatically induces an AVERAGED model type.

CLPARM

produces Wald confidence limits for the parameter estimates. The confidence coefficient can be specified with the ALPHA= option.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COV

displays $\mathbf{S}_i$, which is the covariance matrix of the response functions for each population.

COVB

displays the estimated covariance matrix of the parameter estimates.

DESIGN

displays the design matrix $\mathbf{X}$ for WLS and ML analyses, and also displays the _RESPONSE_ matrix for log-linear models. For further information, see the "Generation of the Design Matrix" section on page 876.

EPSILON=*number*

specifies the convergence criterion for the maximum likelihood estimation of the parameters. The iterative estimation process stops when the proportional change in the log likelihood is less than *number*, or after the number of iterations specified by the MAXITER= option, whichever comes first. By default, EPSILON=1E−8.

FREQ

produces the two-way frequency table for the cross-classification of populations by responses.

ITPRINT

displays parameter estimates and other information at each iteration of a maximum likelihood analysis.

MAXITER=*number*

specifies the maximum number of iterations used for the maximum likelihood estimation of the parameters. By default, MAXITER=20.

ML < **= NR | IPF**< **(** *ipf-options* **)** >>

computes maximum likelihood estimates (MLE) using either a Newton-Raphson algorithm (NR) or an iterative proportional fitting algorithm (IPF).

The option ML=NR (or simply ML) is available when you use generalized logits, and also when you perform binary logistic regression with logits, cumulative logits, or adjacent category logits. For generalized logits (the default response functions), ML=NR is the default estimation method.

The option ML=IPF is available for fitting a hierarchical log-linear model with one population (no independent variables and no population variables). The use of bar notation to express the log-linear effects guarantees that the model is hierarchical (the presence of any interaction term in the model requires the presence of all its lower-order terms). If your table is *incomplete* (that is, your table has a zero or missing entry in at least one cell), then all missing cells and all cells with zero weight are treated as structural zeros by default; this behavior can be modified with the ZERO= and MISSING= options in the MODEL statement.

You can control the convergence of the two algorithms with the EPSILON= and MAXITER= options in the MODEL statement. You can select the convergence criterion for the IPF algorithm with the CONVCRIT= option. **Note:** The RESTRICT statement is not available with the ML=IPF option.

You can specify the following *ipf-options* within parentheses after the ML=IPF option.

CONV=*keyword*
CONVCRIT=*keyword*

specifies the method that determines when convergence of the IPF algorithm occurs. You can specify one of the following *keywords*:

CELL termination requires the maximum absolute difference between consecutive cell estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

LOGL termination requires the relative difference between consecutive estimates of the log-likelihood to be less than 1E-8 (or the value of the EPSILON= option, if specified). This is the default.

MARGIN termination requires the maximum absolute difference between consecutive margin estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

DF=*keyword*

specifies the method used to compute the degrees of freedom for the goodness of fit G^2 test (labeled "Likelihood Ratio" in the "Estimates" table).

For a *complete* table (a table having nonzero entries in every cell), the degrees of freedom are calculated as the number of cells in the table (n_c) minus the number of independent parameters specified in the model (n_p). For incomplete tables, these degrees of freedom may be adjusted by the number of fitted zeros (n_z, which includes the number of structural zeros) and the number of nonestimable parameters due to the zeros (n_n). If you are analyzing an incomplete table, you should verify that the degrees of freedom are correct.

You can specify one of the following *keywords*:

UNADJ	computes the unadjusted degrees of freedom as $n_c - n_p$. These are the same degrees of freedom you would get if all cells in the table were positive.
ADJ	computes the degrees of freedom as $(n_c - n_p) - (n_z - n_n)$ (Bishop, Fienberg, and Holland 1975), which adjusts for fitted zeros and nonestimable parameters. This is the default, and for complete tables gives the same results as the UNADJ option.
ADJEST	computes the degrees of freedom as $(n_c - n_p) - n_z$, which adjusts for fitted zeros only. This gives a lower bound on the true degrees of freedom.

PARM

computes parameter estimates, generates the "ANOVA," "Parameter Estimates," and "Predicted Values of Response Functions" tables, and includes the predicted standard errors in the "Predicted Values of Frequencies and Probabilities" tables.

When you specify the PARM option, the algorithm used to obtain the maximum likelihood parameter estimates is weighted least squares on the IPF-predicted frequencies. This algorithm can be much faster than the Newton-Raphson algorithm used if you just specify the ML=NR option. In the resulting ANOVA table, the likelihood ratio is computed from the initial IPF fit while the degrees of freedom are generated from the WLS analysis; the DF= option can override this. Also, the initial response function, which the WLS method usually computes from the raw data, is computed from the IPF fitted frequencies.

If there are any zero marginals in the configurations that define the model, then there are predicted cell frequencies of zero and WLS cannot be used to compute the estimates. In this case, PROC CATMOD automatically changes the algorithm from ML=IPF to ML=NR and prints a note in the log.

MISSING=*keyword*
MISS=*keyword*

specifies whether a missing cell is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells may have nonzero expected value and may be estimable. For a single population, the missing cells are treated as structural zeros by default. For multiple populations, as long as some population has a nonzero count for a given population and response profile, the missing values are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats missing values for one or more populations.

MISSING=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets missing weights and cells to *value*	sets missing weights and cells to *value*

NODESIGN

suppresses the display of the design matrix $\mathbf{X}$ when the DESIGN option is also specified. This enables you to display only the _RESPONSE_ matrix for log-linear models.

NOINT

suppresses the intercept term in the model.

NOITER

suppresses the display of parameter estimates and other information at each iteration of a maximum likelihood analysis.

NOPARM

suppresses the display of the estimated parameters and the statistics for testing that each parameter is zero.

NOPREDVAR

suppresses the display of the variable levels in tables requested with the PRED= option and in the "Estimates" table. Population profiles are replaced with the sample number, class variable levels are suppressed, and response profiles are replaced with a function number.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you only want to create output data sets with the OUT= or OUTEST= option in the RESPONSE statement. A NOPRINT option is also available in the PROC CATMOD statement. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

NOPROFILE

suppresses the display of the population profiles and the response profiles.

NORESPONSE

suppresses the display of the _RESPONSE_ matrix for log-linear models when the DESIGN option is also specified. This enables you to display only the design matrix for log-linear models.

ONEWAY

produces a one-way table of frequencies for each variable used in the analysis. This table is useful in determining the order of the observed levels for each variable.

PARAM = EFFECT | REFERENCE

specifies the parameterization method for the classification variable or variables. The default is PARAM=EFFECT. Both the effect and reference parameterizations are full rank. See the "Generation of the Design Matrix" section on page 876 for further details.

PREDICT
PRED=FREQ | PROB

displays the observed and predicted values of the response functions for each population, together with their standard errors and the residuals (observed − predicted). In addition, if the response functions are the standard ones (generalized logits), then the

PRED=FREQ option specifies the computation and display of predicted cell frequencies, while PRED=PROB (or just PREDICT) specifies the computation and display of predicted cell probabilities.

The OUT= data set always contains the predicted probabilities. If the response functions are the generalized logits, the predicted cell probabilities are output unless the option PRED=FREQ is specified, in which case the predicted cell frequencies are output.

PROB

produces the two-way table of probability estimates for the cross-classification of populations by responses. These estimates sum to one across the response categories for each population.

PROFILE

displays all of the population profiles. If you have more than 60 populations, then by default only the first 40 profiles are displayed; the PROFILE option overrides this default behavior.

TITLE='*title*'

displays the *title* at the top of certain pages of output that correspond to this MODEL statement.

WLS
GLS

computes weighted least-squares estimates. This type of estimation is also called generalized-least-squares estimation. For response functions other than the default (of generalized logits), WLS is the default estimation method.

XPX

displays $\mathbf{X}'\mathbf{S}^{-1}\mathbf{X}$, the crossproducts matrix for the normal equations.

ZERO=*keyword*
ZEROS=*keyword*
ZEROES=*keyword*

specifies whether a non-missing cell with zero weight in the data set is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells have nonzero expected value and may be estimable. For a single population, the zero cells are treated as structural zeros by default; with multiple populations, as long as some population has a nonzero count for a given population and response profile, the zeros are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats zeros for one or more populations.

ZERO=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets zero weights to *value*	sets zero weights to *value*

Specifying the Design Matrix Directly

If you specify the design matrix directly, adjacent rows of the matrix must be separated by a comma, and the matrix must have $q \times s$ rows, where s is the number of populations and q is the number of response functions per population. The first q rows correspond to the response functions for the first population, the second set of q rows corresponds to the functions for the second population, and so forth. The following is an example using direct specification of the design matrix.

```
proc catmod;
   model R=(1 0,
            1 1,
            1 2,
            1 3);
run;
```

These statements are appropriate for the case of one population and for R with five levels (generating four response functions), so that $4 \times 1 = 4$. These statements are also appropriate for a situation with two populations and two response functions per population; giving $2 \times 2 = 4$ rows of the design matrix. (To induce more than one population, the POPULATION statement is needed.)

When you input the design matrix directly, you also have the option of specifying that any subsets of the parameters be tested for equality to zero. Indicate each subset by specifying the appropriate column numbers of the design matrix, followed by an equal sign and a label (24 characters or less, in quotes) that describes the subset. Adjacent subsets are separated by a comma, and the entire specification is enclosed in parentheses and placed after the design matrix. For example,

```
proc catmod;
   population Group Time;
   model R=(1  1  0  0,
            1  1  0  1,
            1  1  0  2,
            1  0  1  0,
            1  0  1  1,
            1  0  1  2,
            1 -1 -1  0,
            1 -1 -1  1,
            1 -1 -1  2) (1  ='Intercept',
                         2 3='Group main effect',
                         4  ='Linear effect of Time');
run;
```

The preceding statements are appropriate when Group and Time each have three levels, and R is dichotomous. The POPULATION statement induces nine populations, and $q = 1$ (since R is dichotomous), so $q \times s = 1 \times 9 = 9$.

If you input the design matrix directly but do not specify any subsets of the parameters to be tested, then PROC CATMOD tests the effect of MODEL $|$ MEAN, which represents the significance of the model beyond what is explained by an overall mean. For the previous example, the MODEL | MEAN effect is the same as that obtained by specifying

```
(2 3 4='model|mean');
```

at the end of the MODEL statement.

POPULATION Statement

POPULATION *variables* **;**

The POPULATION statement specifies that populations are to be based only on cross-classifications of the specified *variables*. If you do not specify the POPULATION statement, then populations are based only on cross-classifications of the independent variables in the MODEL statement.

The POPULATION statement has two major uses:

- When you enter the design matrix directly, there are no independent variables in the MODEL statement; therefore, the POPULATION statement is the only way of inducing more than one population.
- When you fit a reduced model, the POPULATION statement may be necessary if you want to form the same number of populations as there are for the saturated model.

To illustrate the first use, suppose that you specify the following statements:

```
data one;
   input A $ B $ wt @@;
   datalines;
yes yes 23   yes no 31   no yes 47   no no 50
;
proc catmod;
   weight wt;
   population B;
   model A=(1 0,
            1 1);
run;
```

Since the dependent variable A has two levels, there is one response function per population. Since the variable B has two levels, there are two populations. Thus, the MODEL statement is valid since the number of rows in the design matrix (2) is the same as the total number of response functions. If the POPULATION statement is omitted, there would be only one population and one response function, and the MODEL statement would be invalid.

To illustrate the second use, suppose that you specify

```
data two;
   input A $ B $ Y wt @@;
   datalines;
yes  yes  1  23       yes  yes  2  63
yes  no   1  31       yes  no   2  70
no   yes  1  47       no   yes  2  80
no   no   1  50       no   no   2  84
;
proc catmod;
   weight wt;
   model Y=A B A*B / wls;
run;
```

These statements form four populations and produce the following design matrix and analysis of variance table.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Source	**DF**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	48.10	<.0001
A	1	3.47	0.0625
B	1	0.25	0.6186
A*B	1	0.19	0.6638
Residual	0		

Since the B and A*B effects are nonsignificant ($p > 0.10$), you may want to fit the reduced model that contains only the A effect. If your new statements are

```
proc catmod;
   weight wt;
   model Y=A / wls;
run;
```

then only two populations are formed, and the design matrix and the analysis of variance table are as follows.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Source	**DF**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	47.94	<.0001
A	1	3.33	0.0678
Residual	0		

However, if the new statements are

```
proc catmod;
   weight wt;
   population A B;
   model Y=A / wls;
run;
```

then four populations are formed, and the design matrix and the analysis of variance table are as follows.

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}$$

Source	**DF**	**Chi-Square**	**Pr > ChiSq**
Intercept	1	47.76	<.0001
A	1	3.30	0.0694
Residual	2	0.35	0.8374

The advantage of the latter analysis is that it retains four populations for the reduced model, thereby creating a built-in goodness-of-fit test: the residual chi-square. Such a test is important because the cumulative (or joint) effect of deleting two or more effects from the model may be significant, even if the individual effects are not.

The resulting differences between the two analyses are due to the fact that the latter analysis uses pure weighted least-squares estimates with respect to the four populations that are actually sampled. The former analysis pools populations and therefore uses parameter estimates that can be regarded as weighted least-squares estimates of maximum likelihood predicted cell frequencies. In any case, the estimation methods are asymptotically equivalent; therefore, the results are very similar. If you specify the ML option (instead of the WLS option) in the MODEL statements, then the parameter estimates are identical for the two analyses.

CAUTION: if your model has different covariate profiles within any population, then the first profile is used in the analysis.

REPEATED Statement

REPEATED *factor-description* < *, . . . , factor-description* >< */ options* > **;**

where a *factor-description* is

factor-name < *$* >< *levels* >

and *factor-description*s are separated from each other by a comma. The $ is required for character-valued factors. The value of *levels* provides the number of levels of the repeated measurement factor identified by a given *factor-name*. For only one repeated measurement factor, *levels* is optional; for two or more repeated measurement factors, it is required.

The REPEATED statement incorporates repeated measurement factors into the model. You can use this statement whenever there is more than one dependent variable and the keyword _RESPONSE_ is specified in the MODEL statement. If the dependent variables correspond to one or more repeated measurement factors, you can use the REPEATED statement to define _RESPONSE_ in terms of those factors. You can specify the name, type, and number of levels of each factor, as well as the identification of each level.

You cannot specify the REPEATED statement for an analysis that also contains the FACTORS or LOGLIN statement since all of them specify the same information: how to partition the variation among the response functions within a population.

In the REPEATED statement,

factor-name names a repeated measurement factor that corresponds to two or more response functions. This name must be a valid SAS variable name, and it should not be the same as the name of a variable that already exists in the data set being analyzed.

$ indicates that the factor is character-valued. If the $ is omitted, then PROC CATMOD assumes that the factor is numeric. The type of the factor is relevant only when you use the PROFILE= option or when the _RESPONSE_= option specifies nested-by-value effects.

levels specifies the number of levels of the corresponding repeated measurement factor. If there is only one such factor and the number is omitted, then PROC CATMOD assumes that the number of levels is equal to the number of response functions per population (q). Unless you specify the PROFILE= option, the number q must either be equal to or be a multiple of the product of the number of levels of all the factors.

You can specify the following options in the REPEATED statement after a slash.

PROFILE=(*matrix*)

specifies the values assumed by the factors for each response function. There should be one column for each factor, and the values in a given column should match the type (character or numeric) of the corresponding factor. Character values are restricted to 16 characters or less. If there are q response functions per population, then the matrix must have i rows, where q must either be equal to or be a multiple of i. Adjacent rows of the matrix should be separated by a comma.

The values in the PROFILE matrix are useful for specifying models in those situations where the study design is not a full factorial with respect to the factors. They can also be used to specify nested-with-value effects in the _RESPONSE_= option. If you specify character values in both the PROFILE= option and the _RESPONSE_= option, then the values must match with respect to whether or not they are enclosed in quotes (that is, enclosed in quotes in both places or in neither place).

RESPONSE=*effects*

specifies design effects. The variables named in the effects must be *factor-names* that appear in the REPEATED statement. If the _RESPONSE_= option is omitted, then PROC CATMOD builds a full factorial _RESPONSE_ effect with respect to the repeated measurement factors. For example, the following two statements are equivalent in that they produce the same parameter estimates.

```
repeated Time 2, Treatment 2;
repeated Time 2, Treatment 2 / _response_=Time|Treatment;
```

However, the second statement produces tests of the Time, Treatment, and Time*Treatment effects in the "Analysis of Variance" table, whereas the first statement produces a single test for the combined effects in _RESPONSE_.

TITLE=*'title'*

displays the *title* at the top of certain pages of output that correspond to this REPEATED statement.

For further information and numerous examples of the REPEATED statement, see the section "Repeated Measures Analysis" on page 873.

RESPONSE Statement

RESPONSE < *function* >< */ options* > **;**

The RESPONSE statement specifies functions of the response probabilities. The procedure models these response functions as linear combinations of the parameters.

By default, PROC CATMOD uses the standard response functions (generalized logits, which are explained in detail in the "Understanding the Standard Response Functions" section on page 859). With these standard response functions, the default estimation method is maximum likelihood, but you can use the WLS option in the MODEL statement to request weighted least-squares estimation. With other response functions (specified in the RESPONSE statement), the default (and only) estimation method is weighted least squares.

You can specify more than one RESPONSE statement, in which case each RESPONSE statement produces a separate analysis. If the computed response functions for any population are linearly dependent (yielding a singular covariance matrix), then PROC CATMOD displays an error message and stops processing. See the "Cautions" section on page 887 for methods of dealing with this.

The *function* specification can be any of the items in the following list. For an example of response functions generated and formulas for q (the number of response functions), see the "More on Response Functions" section on page 854.

ALOGIT
ALOGITS

specifies response functions as adjacent-category logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent adjacent-category logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of the kth ratio is the marginal probability corresponding to the kth level of the variable, and the numerator is the marginal probability corresponding to the $(k + 1)$th level. If a dependent variable has two levels, then the adjacent-category logit is the negative of the generalized logit.

CLOGIT
CLOGITS
specifies that the response functions are cumulative logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent cumulative logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of the kth ratio is the cumulative probability, c_k, corresponding to the kth level of the variable, and the numerator is $1 - c_k$ (Agresti 1984, 113–114). If a dependent variable has two levels, then PROC CATMOD computes its cumulative logit as the negative of its generalized logit. You should use cumulative logits only when the dependent variables are ordinally scaled.

JOINT
specifies that the response functions are the joint response probabilities. A linearly independent set is created by deleting the last response probability. For the case of one dependent variable, the JOINT and MARGINALS specifications are equivalent.

LOGIT
LOGITS
specifies that the response functions are generalized logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent generalized logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of each ratio is the marginal probability corresponding to the last observed level of the variable, and the numerators are the marginal probabilities corresponding to each of the other levels. If there is one dependent variable, then specifying LOGIT is equivalent to using the standard response functions.

MARGINAL
MARGINALS
specifies that the response functions are marginal probabilities for each of the dependent variables in the MODEL statement. For each dependent variable, the response functions are a set of linearly independent marginals, obtained by deleting the marginal probability corresponding to the last level.

MEAN
MEANS
specifies that the response functions are the means of the dependent variables in the MODEL statement. This specification requires that all of the dependent variables be numeric.

READ *variables*
specifies that the response functions and their covariance matrix are to be read directly from the input data set with one response function for each variable named. See the section "Inputting Response Functions and Covariances Directly" on page 862 for more information.

transformation
specifies response functions that can be expressed by using successive applications of the four operations: $\mathbf{LOG}$, $\mathbf{EXP}$, $*$ matrix literal, or $+$ matrix literal. The operations are described in detail in the "Using a Transformation to Specify Response Functions" section on page 856.

You can specify the following options in the RESPONSE statement after a slash.

OUT=*SAS-data-set*

produces a SAS data set that contains, for each population, the observed and predicted values of the response functions, their standard errors, and the residuals. Moreover, if you use the standard response functions, the data set also includes observed and predicted values of the cell frequencies or the cell probabilities. For further information, see the "Output Data Sets" section on page 866.

OUTEST=*SAS-data-set*

produces a SAS data set that contains the estimated parameter vector and its estimated covariance matrix. For further information, see the "Output Data Sets" section on page 866.

TITLE=*'title'*

displays the *title* at the top of certain pages of output that correspond to this RESPONSE statement.

More on Response Functions

Suppose the dependent variable A has 3 levels and is the only *response-effect* in the MODEL statement. The following table shows the proportions upon which the response functions are defined.

Value of A:	1	2	3
proportions:	p_1	p_2	p_3

Note that $\sum_j p_j = 1$. The following table shows the response functions generated for each population.

Function **Specification**	**Value** **of** q	**Response Function**
none*	2	$\ln\left(\frac{p_1}{p_3}\right),\ \ln\left(\frac{p_2}{p_3}\right)$
ALOGITS	2	$\ln\left(\frac{p_2}{p_1}\right),\ \ln\left(\frac{p_3}{p_2}\right)$
CLOGITS	2	$\ln\left(\frac{1-p_1}{p_1}\right),\ \ln\left(\frac{1-(p_1+p_2)}{p_1+p_2}\right)$
JOINT	2	$p_1,\ p_2$
LOGITS	2	$\ln\left(\frac{p_1}{p_3}\right),\ \ln\left(\frac{p_2}{p_3}\right)$
MARGINAL	2	$p_1,\ p_2$
MEAN	1	$1p_1 + 2p_2 + 3p_3$

*Without a function specification, the default response functions are generalized logits.

Now, suppose the dependent variables A and B each have 3 levels (valued 1, 2, and 3 each) and the *response-effect* in the MODEL statement is A*B. The following table shows the proportions upon which the response functions are defined.

Value of A:	1	1	1	2	2	2	3	3	3
Value of B:	1	2	3	1	2	3	1	2	3
proportions:	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9

The marginal totals for the preceding table are defined as follows,

$$\begin{aligned} p_{1\cdot} &= p_1 + p_2 + p_3 & \qquad p_{\cdot 1} &= p_1 + p_4 + p_7 \\ p_{2\cdot} &= p_4 + p_5 + p_6 & p_{\cdot 2} &= p_2 + p_5 + p_8 \\ p_{3\cdot} &= p_7 + p_8 + p_9 & p_{\cdot 3} &= p_3 + p_6 + p_9 \end{aligned}$$

where $\sum_j p_j = 1$. The following table shows the response functions generated for each population.

Function Specification	**Value of** q	**Response Function**
none*	8	$\ln\left(\frac{p_1}{p_9}\right),\ \ln\left(\frac{p_2}{p_9}\right),\ \ln\left(\frac{p_3}{p_9}\right),\ldots,\ \ln\left(\frac{p_8}{p_9}\right)$
ALOGITS	4	$\ln\left(\frac{p_{2\cdot}}{p_{1\cdot}}\right),\ \ln\left(\frac{p_{3\cdot}}{p_{2\cdot}}\right),\ \ln\left(\frac{p_{\cdot 2}}{p_{\cdot 1}}\right),\ \ln\left(\frac{p_{\cdot 3}}{p_{\cdot 2}}\right)$
CLOGITS	4	$\ln\left(\frac{1-p_{1\cdot}}{p_{1\cdot}}\right),\ \ln\left(\frac{1-(p_{1\cdot}+p_{2\cdot})}{p_{1\cdot}+p_{2\cdot}}\right),\ \ln\left(\frac{1-p_{\cdot 1}}{p_{\cdot 1}}\right),\ \ln\left(\frac{1-(p_{\cdot 1}+p_{\cdot 2})}{p_{\cdot 1}+p_{\cdot 2}}\right)$
JOINT	8	$p_1,\ p_2,\ p_3,\ p_4,\ p_5,\ p_6,\ p_7,\ p_8$
LOGITS	4	$\ln\left(\frac{p_{1\cdot}}{p_{3\cdot}}\right),\ \ln\left(\frac{p_{2\cdot}}{p_{3\cdot}}\right),\ \ln\left(\frac{p_{\cdot 1}}{p_{\cdot 3}}\right),\ \ln\left(\frac{p_{\cdot 2}}{p_{\cdot 3}}\right)$
MARGINAL	4	$p_{1\cdot},\ p_{2\cdot},\ p_{\cdot 1},\ p_{\cdot 2}$
MEAN	2	$1p_{1\cdot} + 2p_{2\cdot} + 3p_{3\cdot},\ 1p_{\cdot 1} + 2p_{\cdot 2} + 3p_{\cdot 3}$

[*] Without a function specification, the default response functions are generalized logits.

The READ and *transformation* function specifications are not shown in the preceding table. For these two situations, there is not a general response function; the response functions generated depend on what you specify.

Another important aspect of the function specification is the number of response functions generated per population, q. Let m_i represent the number of levels for the ith dependent variable in the MODEL statement, and let d represent the number of dependent variables in the MODEL statement. Then, if the function specification is ALOGITS, CLOGITS, LOGITS, or MARGINALS, the number of response functions is

$$q = \sum_{i=1}^{d} (m_i - 1)$$

If the function specification is JOINT or the default (generalized logits), the number of response functions per population is

$$q = r - 1$$

where r is the number of response profiles. If every possible cross-classification of the dependent variables is observed in the samples, then

$$r = \prod_{i=1}^{d} m_i$$

Otherwise, r is the number of cross-classifications actually observed.

If the function specification is MEANS, the number of response functions per population is $q = d$.

Response Statement Examples

Some example response statements are shown in the following table.

Example	Result
`response marginals;`	marginals for each dependent variable
`response means;`	the mean of each dependent variable
`response logits;`	generalized logits of the marginal probabilities
`response clogits;`	cumulative logits of the marginal probabilities
`response alogits;`	adjacent-category logits of the marginal probabilities
`response joint;`	the joint probabilities
`response 1 -1 log;`	the logit
`response;`	generalized logits
`response 1 2 3;`	the mean score, with scores of 1, 2, and 3 corresponding to the three response levels
`response read b1-b4;`	four response functions and their covariance matrix, read directly from the input data set

Using a Transformation to Specify Response Functions

If you specify a *transformation*, it is applied to the vector that contains the sample proportions in each population. The *transformation* can be any combination of the following four operations.

Operation	Specification
linear combination	$*$ matrix literal matrix literal
logarithm	**LOG**
exponential	**EXP**
adding constant	$+$ matrix literal

If more than one operation is specified, then PROC CATMOD applies the operations consecutively from right to left.

A matrix literal is a matrix of numbers with each row of the matrix separated from the next by a comma. If you specify a linear combination, in most cases the $*$ is not needed. The following statement defines the response function $p_1 + 1$. The $*$ is needed to separate the two matrix literals '1' and '1 0'.

```
response + 1 * 1 0;
```

The **LOG** of a vector transforms each element of the vector into its natural logarithm; the **EXP** of a vector transforms each element into its exponential function (antilogarithm).

In order to specify a linear response function for data that have $r = 3$ response categories, you could specify either of the following RESPONSE statements:

```
response  * 1 0 0 , 0 1 0;
response    1 0 0 , 0 1 0;
```

The matrix literal in the preceding statements specifies a 2×3 matrix, which is applied to each population as follows:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

where p_1, p_2, and p_3 are sample proportions for the three response categories in a population, and F_1 and F_2 are the two response functions computed for that population. This response function, therefore, sets $F1 = p_1$ and $F2 = p_2$ in each population.

As another example of the linear response function, suppose you have two dependent variables corresponding to two observers who evaluate the same subjects. If the observers grade on the same three-point scale and if all nine possible responses are observed, then the following RESPONSE statement would compute the probability that the observers agree on their assessments:

```
response 1 0 0 0 1 0 0 0 1;
```

This response function is then computed as

$$F = p_{11} + p_{22} + p_{33} = \left[\begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array} \right] * \left[\begin{array}{c} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{array} \right]$$

where p_{ij} denotes the probability that a subject gets a grade of i from the first observer and j from the second observer.

If the function is a compound function, requiring more than one operation to specify it, then the operations should be listed so that the first operation to be applied is on the right and the last operation to be applied is on the left. For example, if there are two response levels, the response function

```
response 1 -1 log;
```

is equivalent to the matrix expression:

$$F = \left[\begin{array}{cc} 1 & -1 \end{array} \right] * \left[\begin{array}{c} \log(p_1) \\ \log(p_2) \end{array} \right] = \log(p_1) - \log(p_2) = \log\left(\frac{p_1}{p_2}\right)$$

which is the logit response function since $p_2 = 1 - p_1$ when there are only two response levels.

Another example of a compound response function is

```
response exp 1 -1 * 1 0 0 1, 0 1 1 0 log;
```

which is equivalent to the matrix expression

$$F = \mathbf{EXP}(\mathbf{A} * \mathbf{B} * \mathbf{LOG}(\mathbf{P}))$$

where $\mathbf{P}$ is the vector of sample proportions for some population,

$$\mathbf{A} = \left[\begin{array}{cc} 1 & -1 \end{array} \right] \text{ and } \mathbf{B} = \left[\begin{array}{cccc} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} \right]$$

If the four responses are based on two dependent variables, each with two levels, then the function can also be written as

$$F = \frac{p_{11}p_{22}}{p_{12}p_{21}}$$

which is the odds (crossproduct) ratio for a 2 × 2 table.

Understanding the Standard Response Functions

If no RESPONSE statement is specified, PROC CATMOD computes the standard response functions, which contrast the log of each response probability with the log of the probability for the last response category. If there are r response categories, then there are $r - 1$ standard response functions. For example, if there are four response categories, using no RESPONSE statement is equivalent to specifying

```
response  1 0 0 -1,
          0 1 0 -1,
          0 0 1 -1  log;
```

This results in three response functions:

$$F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \log(p_1/p_4) \\ \log(p_2/p_4) \\ \log(p_3/p_4) \end{bmatrix}$$

If there are only two response levels, the resulting response function would be a logit. Thus, the standard response functions are called generalized logits. They are useful in dealing with the log-linear model:

$$\pi = \mathbf{EXP}(\mathbf{X}\boldsymbol{\beta})$$

If $\mathbf{C}$ denotes the matrix in the preceding RESPONSE statement, then because of the restriction that the probabilities sum to 1, it follows that an equivalent model is

$$\mathbf{C} * \mathbf{LOG}(\pi) = (\mathbf{CX})\boldsymbol{\beta}$$

But $\mathbf{C}{*}\mathbf{LOG}(\mathbf{P})$ is simply the vector of standard response functions. Thus, fitting a log-linear model on the cell probabilities is equivalent to fitting a linear model on the generalized logits.

RESTRICT Statement

RESTRICT *parameter=value* < . . . *parameter=value* > **;**

where *parameter* is the letter B followed by a number; for example, B3 specifies the third parameter in the model. The *value* is the value to which the parameter is restricted. The RESTRICT statement restricts values of parameters to the values you specify, so that the estimation of the remaining parameters is subject to these restrictions. Consider the following statement:

```
restrict b1=1 b4=0 b6=0;
```

This restricts the values of three parameters. The first parameter is set to 1, and the fourth and sixth parameters are set to zero.

The RESTRICT statement is interactive. A new RESTRICT statement replaces any previous ones. In addition, if you submit two or more MODEL, LOGLIN, FACTORS, or REPEATED statements, then the subsequent occurrences of these statements also delete the previous RESTRICT statement.

WEIGHT Statement

WEIGHT *variable* **;**

You can use a WEIGHT statement to refer to a variable containing the cell frequencies, which need not be integers. The WEIGHT statement lets you use summary data sets containing a count variable. See the "Input Data Sets" section on page 860 for further information concerning the WEIGHT statement.

Details

Missing Values

Observations with missing values for any variable listed in the MODEL or POPULATION statement are omitted from the analysis.

If the WEIGHT variable for an observation has a missing value, the observation is by default omitted from the analysis. You can modify this behavior by specifying the MISSING= option in the MODEL statement. The option MISSING=*value* sets all missing weights to *value* and all missing cells to *value*. The option MISSING=SAMPLING causes all missing cells in a contingency table to be treated as sampling zeros.

Any observation with nonpositive weight is also, by default, omitted from the analysis. If it has zero weight, then you can specify the ZERO= option in the MODEL statement.

Input Data Sets

Data to be analyzed by PROC CATMOD must be in a SAS data set containing one of the following:

- raw data values (variable values for every subject)
- frequency counts and the corresponding variable values
- response function values and their covariance matrix

If you specify a WEIGHT statement, then PROC CATMOD uses the values of the WEIGHT variable as the frequency counts. If the READ function is specified in the RESPONSE statement, then the procedure expects the input data set to contain the values of response functions and their covariance matrix. Otherwise, PROC CATMOD assumes that the SAS data set contains raw data values.

Raw Data Values

If you use raw data, PROC CATMOD first counts the number of observations having each combination of values for all variables specified in the MODEL or POPULATION statements. For example, suppose the variables A and B each take on the values 1 and 2, and their frequencies can be represented as follows.

	A=1	A=2
B=1	2	1
B=2	3	1

The SAS data set Raw containing the raw data might be as follows.

Observation	A	B
1	1	1
2	1	1
3	1	2
4	1	2
5	1	2
6	2	1
7	2	2

And the statements for PROC CATMOD would be

```
proc catmod data=Raw;
   model A=B;
run;
```

For discussions of how to handle structural and random zeros with raw data as input data, see the "Zero Frequencies" section on page 888 and Example 22.5 on page 919.

Frequency Counts

If your data set contains frequency counts, then use the WEIGHT statement in PROC CATMOD to specify the variable containing the frequencies. For example, you could create the Summary data set as follows.

```
data Summary;
   input A B Count;
   datalines;
1 1 2
1 2 3
2 1 1
2 2 1
;
```

In this case, the corresponding statements would be

```
proc catmod data=Summary;
   weight Count;
   model A=B;
run;
```

The data set Summary can also be created from data set Raw by using the FREQ procedure:

```
proc freq data=Raw;
   tables A*B / out=Summary;
run;
```

Inputting Response Functions and Covariances Directly

If you want to read in the response functions and their covariance matrix, rather than have PROC CATMOD compute them, create a TYPE=EST data set. In addition to having one variable name for each function, the data set should have two additional variables: _TYPE_ and _NAME_, both character variables of length 8. The variable _TYPE_ should have the value 'PARMS' when the observation contains the response functions; it should have the value 'COV' when the observation contains elements of the covariance matrix of the response functions. The variable _NAME_ is used only when _TYPE_=COV, in which case it should contain the name of the variable that has its covariance elements stored in that observation. In the following data set, for example, the covariance between the second and fourth response functions is 0.000102.

```
data direct(type=est);
   input b1-b4 _type_ $ _name_ $8.;
   datalines;
0.590463   0.384720   0.273269   0.136458   PARMS     .
0.001690   0.000911   0.000474   0.000432   COV       B1
0.000911   0.001823   0.000031   0.000102   COV       B2
0.000474   0.000031   0.001056   0.000477   COV       B3
0.000432   0.000102   0.000477   0.000396   COV       B4
;
```

In order to tell PROC CATMOD that the input data set contains the values of response functions and their covariance matrix,

- specify the READ function in the RESPONSE statement
- specify _F_ as the dependent variable in the MODEL statement

For example, suppose the response functions correspond to four populations that represent the cross-classification of two age groups by two race groups. You can use the FACTORS statement to identify these two factors and to name the effects in the model. The statements required to fit a main-effects model to these data are

```
proc catmod data=direct;
   response read b1-b4;
   model _f_=_response_;
   factors age 2, race 2 / _response_=age race;
run;
```

Ordering of Populations and Responses

By default, populations and responses are sorted in standard SAS order as follows:

- alphabetic order for character variables
- increasing numeric order for numeric variables

Suppose you specify the following statements:

```
data one;
   length A B $ 6;
   input A $ B $ wt @@;
   datalines;
low       low   23  low    medium  31 low    high  38
medium    low   40  medium medium  42 medium high  50
high      low   52  high   medium  54 high   high  61
;

proc catmod;
   weight wt;
   model A=B / oneway;
run;
```

The ordering of populations and responses corresponds to the alphabetical order of the levels of the character variables. You can specify the ONEWAY option to display the ordering of the variables, while the “Population Profiles” and “Response Profiles” tables display the ordering of the populations and the responses, respectively.

Population Profiles		**Response Profiles**	
Sample	**B**	**Response**	**A**
1	high	1	high
2	low	2	low
3	medium	3	medium

However, in this example, you may want to have the levels ordered in the natural order of ‘low,’ ‘medium,’ ‘high.’ If you specify the ORDER=DATA option

```
proc catmod order=data;
   weight wt;
   model a=b / oneway;
run;
```

then the ordering of populations and responses is as follows.

Population Profiles		**Response Profiles**	
Sample	**B**	**Response**	**A**
1	low	1	low
2	medium	2	medium
3	high	3	high

Thus, you can use the ORDER=DATA option to ensure that populations and responses are ordered in a specific way. But since this also affects the definitions and the ordering of the parameters, you must exercise caution when using the _RESPONSE_ effect, the CONTRAST statement, or direct input of the design matrix.

An alternative method of ensuring that populations and responses are ordered in a specific way is to assign a format to your variables and specify the ORDER=FORMATTED option. The levels will be ordered according to their formatted values.

Another method is to replace any character variables with numeric variables and to assign formatted values such as 'yes' and 'no' to the numeric levels. Since ORDER=INTERNAL is the default ordering, PROC CATMOD orders the populations and responses according to the numeric values but displays the formatted values.

Specification of Effects

By default, the CATMOD procedure treats all variables as classification variables. As a result, there is no CLASS statement in PROC CATMOD. The values of a classification variable can be numeric or character. PROC CATMOD builds a set of effects-coded variables to represent the levels of the classification variable and then uses these to fit the model (for details, see the "Generation of the Design Matrix" section on page 876). You can modify the default by using the DIRECT statement to treat numeric independent continuous variables as continuous variables. The classification variables, combinations of classification variables, and continuous variables are then used in fitting linear models to data.

The parameters of a linear model are generally divided into subsets that correspond to meaningful sources of variation in the response functions. These sources, called *effects*, can be specified in the MODEL, LOGLIN, FACTORS, REPEATED, and CONTRAST statements. Effects can be specified in any of the following ways:

- A main effect is a single class variable (that is, it induces classification levels): A B C.
- A crossed effect (or interaction) is two or more class variables joined by asterisks, for example: A*B A*B*C.
- A nested effect is a main effect or an interaction, followed by a parenthetical field containing a main effect or an interaction. Multiple variables within the parentheses are assumed to form a crossed effect even when the asterisk is absent. Thus, the last two effects are identical: B(A) C(A*B) A*B(C*D) A*B(C D).

- A nested-by-value effect is the same as a nested effect except that any variable in the parentheses can be followed by an equal sign and a value: B(A=1) C(A B=1) C*D(A=1 B=1) A(C='low').
- A direct effect is a variable specified in a DIRECT statement: X Y.
- Direct effects can be crossed with other effects: X*Y X*X*X X*A*B(C D=1).

The variables for crossed and nested effects remain in the order in which they are first encountered. For example, in the model

```
model R=B A A*B C(A B);
```

the effect A*B is reported as B*A since B appeared before A in the statement. Also, C(A B) is interpreted as C(A*B) and is therefore reported as C(B*A).

Bar Notation

You can shorten the specification of multiple effects by using bar notation. For example, two methods of writing a full three-way factorial model are

```
proc catmod;
   model y=a b c a*b a*c b*c a*b*c;
run;
```

and

```
proc catmod;
   model y=a|b|c;
run;
```

When you use the bar (|) notation, the right- and left-hand sides become effects, and the interaction between them becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules 1 through 4 given in Searle (1971, p. 390):

- Multiple bars are evaluated left to right. For example, A|B|C is evaluated as follows:

$$\begin{aligned} \mathrm{A} \mid \mathrm{B} \mid \mathrm{C} &\rightarrow \{\mathrm{A} \mid \mathrm{B}\} \mid \mathrm{C} \\ &\rightarrow \{\mathrm{A\ B\ A{*}B}\} \mid \mathrm{C} \\ &\rightarrow \mathrm{A\ B\ A{*}B\ C\ A{*}C\ B{*}C\ A{*}B{*}C} \end{aligned}$$

- Crossed and nested groups of variables are combined. For example, A(B) | C(D) generates A*C(B D), among other terms.
- Duplicate variables are removed. For example, A(C) | B(C) generates A*B(C C), among other terms, and the extra C is removed.

- Effects are discarded if a variable occurs on both the crossed and nested sides of an effect. For instance, A(B) | B(D E) generates A*B(B D E), but this effect is deleted.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect. For example, the specification A | B | C @ 2 would result in only those effects that contain 2 or fewer variables; in this case, the effects A, B, A*B, C, A*C, and B*C are generated.

Other examples of the bar notation are

A | C(B)	is equivalent to	A C(B) A*C(B)
A(B) | C(B)	is equivalent to	A(B) C(B) A*C(B)
A(B) | B(D E)	is equivalent to	A(B) B(D E)
A | B(A) | C	is equivalent to	A B(A) C A*C B*C(A)
A | B(A) | C@2	is equivalent to	A B(A) C A*C
A | B | C | D@2	is equivalent to	A B A*B C A*C B*C D A*D B*D C*D

For details on how the effects specified lead to a design matrix, see the "Generation of the Design Matrix" section on page 876.

Output Data Sets

OUT= Data Set

For each population, the OUT= data set contains the observed and predicted values of the response functions, their standard errors, the residuals, and variables that describe the population and response profiles. In addition, if you use the standard response functions, the data set includes observed and predicted values for the cell frequencies or the cell probabilities, together with their standard errors and residuals.

Number of Observations

For the standard response functions, there are $s \times (2q - 1)$ observations in the data set for each BY group, where s is the number of populations, and q is the number of response functions per population. Otherwise, there are $s \times q$ observations in the data set for each BY group.

Variables in the OUT= Data Set

The data set contains the following variables:

BY variables   If you use a BY statement, the BY variables are included in the OUT= data set.

dependent variables
If the response functions are the default ones (generalized logits), then the dependent variables, which describe the response profiles, are included in the OUT= data set. When _TYPE_=FUNCTION, the values of these variables are missing.

independent variables
The independent variables, which describe the population profiles, are included in the OUT= data set.

NUMBER
the sequence number of the response function or the cell probability or the cell frequency

OBS
the observed value

PRED
the predicted value

RESID
the residual (observed − predicted)

SAMPLE
the population number. This matches the sample number in the Population Profile section of the output.

SEOBS
the standard error of the observed value

SEPRED
the standard error of the predicted value

TYPE
specifies a character variable with three possible values. When _TYPE_=FUNCTION, the observed and predicted values are values of the response functions. When _TYPE_=PROB, they are values of the cell probabilities. When _TYPE_=FREQ, they are values of the cell frequencies. Cell probabilities or frequencies are provided only when the default response functions are modeled. In this case, cell probabilities are provided by default, and cell frequencies are provided if you specify the option PRED=FREQ.

OUTEST= Data Set

This TYPE=EST output data set contains the estimated parameter vector and its estimated covariance matrix. If you specify both the ML and WLS options in the MODEL statement, the OUTEST= data set contains both sets of estimates. For each BY group, there are $p + 1$ observations in the data set for each estimation method, where p is the number of estimated parameters. The data set contains the following variables.

B1, B2, and so on
variables for the estimated parameters. The OUTEST= data set contains one variable for each estimated parameter.

BY variables
If you use a BY statement, the BY variables are included in the OUT= data set.

METHOD
the method used to obtain parameter estimates. For weighted least-squares estimation, _METHOD_=WLS, and for maximum likelihood estimation, _METHOD_=ML.

NAME identifies parameter names. When _TYPE_=PARMS, _NAME_ is blank, but when _TYPE_=COV, _NAME_ has one of the values B1, B2, and so on, corresponding to the parameter names.

STATUS indicates whether the estimates have converged

TYPE identifies the statistics contained in the variables for parameter estimates (B1, B2, and so on). When _TYPE_=PARMS, the variables contain parameter estimates; when _TYPE_=COV, they contain covariance estimates.

The variables _METHOD_, _NAME_, and _TYPE_ are character variables; the BY variables can be either character or numeric; and the variables for estimated parameters are numeric.

See Appendix A, "Special SAS Data Sets," for more information on special SAS data sets.

Logistic Analysis

In a logistic analysis, the response functions are the logits of the dependent variable.

PROC CATMOD can compute three different types of logits with the use of keywords in the RESPONSE statement. Other types of response functions can be generated by specifying appropriate transformations in the RESPONSE statement.

- Generalized logits are used primarily for nominally scaled dependent variables, but they can also be used for ordinal data modeling. Maximum likelihood estimation is available for the analysis of these logits.
- Cumulative logits are used for ordinally scaled dependent variables. Except for dependent variables with two response levels, only weighted least-squares estimation is available for the analysis of these logits.
- Adjacent-category logits are equivalent to generalized logits, but they have some advantages for ordinal data analysis because they automatically incorporate integer scores for the levels of the dependent variable. Except for dependent variables with two response levels, only weighted least-squares estimation is available for the analysis of these logits.

If the dependent variable has only two responses, then the cumulative logit and the adjacent-category logit are the negative of the generalized logit, as computed by PROC CATMOD. Consequently, parameter estimates obtained using these logits are the negative of those obtained using generalized logits. A simple logistic analysis of variance uses statements like the following:

```
proc catmod;
   model r=a|b;
run;
```

Logistic Regression

If the independent variables are treated quantitatively (like continuous variables), then a logistic analysis is known as a *logistic regression*. If you want PROC CATMOD to treat the independent variables as quantitative variables, specify them in both the DIRECT and MODEL statements, as follows.

```
proc catmod;
   direct x1 x2 x3;
   model r=x1 x2 x3;
run;
```

Since the preceding statements do not include a RESPONSE statement, generalized logits are computed. See Example 22.3 for another example.

The parameter estimates from the CATMOD procedure are the same as those from a logistic regression program such as PROC LOGISTIC (see Chapter 42, “The LOGISTIC Procedure”). The chi-square statistics and the predicted values are also identical. In the binary response case, PROC CATMOD can be made to model the probability of the maximum value by either (1) organizing the input data so that the maximum value occurs first and specifying ORDER=DATA in the PROC CATMOD statement or (2) specifying cumulative logits (CLOGITS) in the RESPONSE statement.

CAUTION: Computational difficulties may occur if you use a continuous variable with a large number of unique values in a DIRECT statement. See the “Continuous Variables” section on page 870 for more details.

Cumulative Logits

If your dependent variable is ordinally scaled, you can specify the analysis of cumulative logits that take into account the ordinal nature of the dependent variable:

```
proc catmod;
   response clogits;
   direct x;
   model r=a x;
run;
```

The preceding statements correspond to a simple analysis that addresses the question of existence of an association between the independent variables and the ordinal dependent variable. However, there are some commonly used models for the analysis of ordinal data (Agresti 1984) that address the structure of association (in terms of odds ratios), as well as its existence.

If the independent variables are class variables, a typical analysis for such a model uses the following statements:

```
proc catmod;
   weight wt;
   response clogits;
   model r=_response_ a b;
run;
```

On the other hand, if the independent variables are ordinally scaled, you might specify numeric scores in variables x1 and x2, and use the following statements:

```
proc catmod;
   weight wt;
   direct x1 x2;
   response clogits;
   model r=_response_ x1 x2;
run;
```

Refer to Agresti (1984) for additional details of estimation, testing, and interpretation.

Continuous Variables

Computational difficulties may occur if you have a continuous variable with a large number of unique values and you use this variable in a DIRECT statement, since an observation often represents a separate population of size one. At this extreme of sparseness, the weighted least-squares method is inappropriate since there are too many zero frequencies. Therefore, you should use the maximum likelihood method. PROC CATMOD is not designed optimally for continuous variables; therefore, it may be less efficient and unable to allocate sufficient memory to handle this problem, as compared with a procedure designed specifically to handle continuous data. In these situations, consider using the LOGISTIC, GENMOD, or PROBIT procedure to analyze your data.

Log-Linear Model Analysis

When the response functions are the default generalized logits, then inclusion of the keyword _RESPONSE_ in every effect in the right-hand side of the MODEL statement induces a log-linear model. The keyword _RESPONSE_ tells PROC CATMOD that you want to model the variation among the dependent variables. You then specify the actual model in the LOGLIN statement.

When you perform log-linear model analysis, you can request weighted least-squares estimates, maximum likelihood estimates, or both. By default, PROC CATMOD calculates maximum likelihood estimates when the default response functions are used. The following table provides appropriate MODEL statements for the combinations of types of estimates.

Estimation Desired	**MODEL Statement**
Maximum likelihood (Newton-Raphson)	`model a*b=_response_;`
Maximum likelihood (Iterative Proportional Fitting)	`model a*b=_response_ / ml=ipf;`
Weighted least squares	`model a*b=_response_ / wls;`
Maximum likelihood and weighted least squares	`model a*b=_response_ / wls ml;`

CAUTION: sampling zeros in the input data set should be specified with the ZERO= option to ensure that these sampling zeros are not treated as structural zeros. Alternatively, you can replace cell counts for sampling zeros by some positive number close to zero (such as 1E-20) in a DATA step. Data containing sampling zeros should be analyzed with maximum likelihood estimation. See the "Cautions" section on page 887 and Example 22.5 on page 919 for further information and an illustration for both cell count data and raw data.

One Population

The usual log-linear model analysis has one population, which means that all of the variables are dependent variables. For example, the statements

```
proc catmod;
   weight wt;
   model r1*r2=_response_;
   loglin r1|r2;
run;
```

yield a maximum likelihood analysis of a saturated log-linear model for the dependent variables r1 and r2.

If you want to fit a reduced model with respect to the dependent variables (for example, a model of independence or conditional independence), specify the reduced model in the LOGLIN statement. For example, the statements

```
proc catmod;
   weight wt;
   model r1*r2=_response_ / pred;
   loglin r1 r2;
run;
```

yield a main-effects log-linear model analysis of the factors r1 and r2. The output includes Wald statistics for the individual effects r1 and r2, as well as predicted cell probabilities. Moreover, the goodness-of-fit statistic is the likelihood ratio test for the hypothesis of independence between r1 and r2 or, equivalently, a test of r1*r2.

Multiple Populations

You can do log-linear model analysis with multiple populations by using a POPULATION statement or by including effects on the right-hand side of the MODEL statement that contain independent variables. Each effect must include the _RESPONSE_ keyword.

For example, suppose the dependent variables r1 and r2 are dichotomous, and the independent variable group has three levels. Then

```
proc catmod;
   weight wt;
   model r1*r2=_response_ group*_response_;
   loglin r1|r2;
run;
```

specifies a saturated model (three degrees of freedom for _RESPONSE_ and six degrees of freedom for the interaction between _RESPONSE_ and group). From another point of view, _RESPONSE_*group can be regarded as a main effect for group with respect to the three response functions, while _RESPONSE_ can be regarded as an intercept effect with respect to the functions. In other words, these statements give essentially the same results as the logistic analysis:

```
proc catmod;
   weight wt;
   model r1*r2=group;
run;
```

The ability to model the interaction between the independent and the dependent variables becomes particularly useful when a reduced model is specified for the dependent variables. For example,

```
proc catmod;
   weight wt;
   model r1*r2=_response_ group*_response_;
   loglin r1 r2;
run;
```

specifies a model with two degrees of freedom for _RESPONSE_ (one for r1 and one for r2) and four degrees of freedom for the interaction of _RESPONSE_*group. The likelihood ratio goodness-of-fit statistic (three degrees of freedom) tests the hypothesis that r1 and r2 are independent in each of the three groups.

Iterative Proportional Fitting

You can use the iterative proportional fitting (IPF) algorithm to fit a hierarchical log-linear model with no independent variables and no population variables.

The advantage of IPF over the Newton-Raphson (NR) algorithm and over the weighted least squares (WLS) method is that, when the contingency table has several dimensions and the parameter vector is large, you can obtain the log-likelihood, the goodness-of-fit G^2, and the predicted frequencies or probabilities without performing potentially expensive parameter estimation and covariance matrix calculations. This enables you to

- compare two models by computing the likelihood ratio statistics to test the significance of the contribution of the variables in one model that are not in the other model.
- compute predicted values of the cell probabilities or frequencies for the final model.

Each iteration of the IPF algorithm is generally faster than an iteration of the NR algorithm; however, the IPF algorithm converges to the MLEs more slowly than the NR algorithm. Both NR and WLS are more general methods that are able to perform more complex analyses than IPF can.

Repeated Measures Analysis

If there are multiple dependent variables and the variables represent repeated measurements of the same observational unit, then the variation among the dependent variables can be attributed to one or more repeated measurement factors. The factors can be included in the model by specifying _RESPONSE_ on the right-hand side of the MODEL statement and using a REPEATED statement to identify the factors.

To perform a repeated measures analysis, you also need to specify a RESPONSE statement, since the standard response functions (generalized logits) cannot be used. Typically, the MEANS or MARGINALS response functions are specified in a repeated measures analysis, but other response functions may also be reasonable.

One Population

Consider an experiment in which each subject is measured at three times, and the response functions are marginal probabilities for each of the dependent variables. If the dependent variables each has k levels, then PROC CATMOD computes $k-1$ response functions for each time. Differences among the response functions with respect to these times could be attributed to the repeated measurement factor Time. To incorporate the Time variation into the model, specify

```
proc catmod;
   response marginals;
   model t1*t2*t3=_response_;
   repeated Time 3 / _response_=Time;
run;
```

These statements induce a Time effect that has $2(k-1)$ degrees of freedom since there are $k-1$ response functions at each time point. Thus, for a dichotomous variable, the Time effect has two degrees of freedom.

Now suppose that at each time point, each subject has X-rays taken, and the X-rays are read by two different radiologists. This creates six dependent variables that represent the 3×2 cross-classification of the repeated measurement factors Time and Reader. A saturated model with respect to these factors can be obtained by specifying

```
proc catmod;
   response marginals;
   model r11*r12*r21*r22*r31*r32=_response_;
   repeated Time 3, Reader 2
      / _response_=Time Reader Time*Reader;
run;
```

If you want to fit a main-effects model with respect to Time and Reader, then change the REPEATED statement to

```
      repeated Time 3, Reader 2 / _response_=Time Reader;
```

If you want to fit a main-effects model for Time but for only one of the readers, the REPEATED statement might look like

```
      repeated Time $ 3, Reader $ 2
               /_response_=Time(Reader=Smith)
                 profile  =('1'  Smith,
                            '1'  Jones,
                            '2'  Smith,
                            '2'  Jones,
                            '3'  Smith,
                            '3'  Jones);
```

If Jones had been unavailable for a reading at time 3, then there would be only $5(k-1)$ response functions, even though PROC CATMOD would be expecting some multiple of 6 $(= 3 \times 2)$. In that case, the PROFILE= option would be necessary to indicate which repeated measurement profiles were actually represented:

```
      repeated Time $ 3, Reader $ 2
               /_response_=Time(Reader=Smith)
                 profile  =('1'  Smith,
                            '1'  Jones,
                            '2'  Smith,
                            '2'  Jones,
                            '3'  Smith);
```

When two or more repeated measurement factors are specified, PROC CATMOD presumes that the response functions are ordered so that the levels of the rightmost factor change most rapidly. This means that the dependent variables should be specified in the same order. For this example, the order implied by the REPEATED statement is as follows, where the variable r$_{ij}$ corresponds to Time i and Reader j.

Response Function	Dependent Variable	Time	Reader
1	r_{11}	1	1
2	r_{12}	1	2
3	r_{21}	2	1
4	r_{22}	2	2
5	r_{31}	3	1
6	r_{32}	3	2

Thus, the order of dependent variables in the MODEL statement must agree with the order implied by the REPEATED statement.

Multiple Populations

When there are variables specified in the POPULATION statement or in the right-hand side of the MODEL statement, these variables induce multiple populations. PROC CATMOD can then model these independent variables, the repeated measurement factors, and the interactions between the two.

For example, suppose that there are five groups of subjects, that each subject in the study is measured at three different times, and that the dichotomous dependent variables are labeled t1, t2, and t3. The following statements induce the computation of three response functions for each population:

```
proc catmod;
   weight wt;
   population Group;
   response marginals;
   model t1*t2*t3=_response_;
   repeated Time / _response_=Time;
run;
```

PROC CATMOD then regards _RESPONSE_ as a variable with three levels corresponding to the three response functions in each population and forms an effect with two degrees of freedom. The MODEL and REPEATED statements tell PROC CATMOD to fit the main effect of Time.

In general, the MODEL statement tells PROC CATMOD how to integrate the independent variables and the repeated measurement factors into the model. For example, again suppose that there are five groups of subjects, that each subject is measured at three times, and that the dichotomous independent variables are labeled t1, t2, and t3. If you use the same WEIGHT, POPULATION, RESPONSE, and REPEATED statements as in the preceding program, the following MODEL statements result in the indicated analyses:

`model t1*t2*t3=Group / averaged;` specifies the Group main effect (with four degrees of freedom).

`model t1*t2*t3=_response_;` specifies the Time main effect (with two degrees of freedom).

`model t1*t2*t3=_response_*Group;` specifies the interaction between Time and Group (with eight degrees of freedom).

`model t1*t2*t3=_response_|Group;` specifies both main effects, and the interaction between Time and Group (with a total of fourteen degrees of freedom).

`model t1*t2*t3=_response_(Group);` specifies a Time main effect within each Group (with ten degrees of freedom).

However, the following MODEL statement is invalid since effects cannot be nested within _RESPONSE_:

```
model t1*t2*t3=Group(_response_);
```

Generation of the Design Matrix

Each row of the design matrix (corresponding to a population) is generated by a unique combination of independent variable values. Each column of the design matrix corresponds to a model parameter. The columns are produced from the effect specifications in the MODEL, LOGLIN, FACTORS, and REPEATED statements. For details on effect specifications, see the "Specification of Effects" section on page 864.

This section is divided into three parts:

- one response function per population
- two or more response functions per population (excluding log-linear models), beginning on page 879
- log-linear models, beginning on page 884

This section assumes that the default effect parameterization is used. Specifying the reference parameterization replaces the "-1"s with zeros in the design matrix for the main effects of classification variables, and makes appropriate changes to interaction terms.

You can display the design matrix by specifying the DESIGN option in the MODEL statement.

One Response Function Per Population

Intercept

When there is one response function per population, all design matrices start with a column of 1s for the intercept unless the NOINT option is specified or the design matrix is input directly.

Main Effects

If a class variable A has k levels, then its main effect has $k-1$ degrees of freedom, and the design matrix has $k-1$ columns that correspond to the first $k-1$ levels of A. The ith column contains a 1 in the ith row, a -1 in the last row, and 0s everywhere else. If α_i denotes the parameter that corresponds to the ith level of variable A, then the $k-1$ columns yield estimates of the independent parameters, $\alpha_1, \alpha_i, \ldots, \alpha_{k-1}$. The last parameter is not needed because PROC CATMOD constrains the k parameters to sum to zero. In other words, PROC CATMOD uses a full-rank center-point parameterization to build design matrices. Here are two examples.

Data Levels	**Design Columns**	
A	A	
1	1	0
2	0	1
3	–1	–1

B	B
1	1
2	–1

For an effect with three levels, such as A, PROC CATMOD produces two parameter estimates for each response function. By default, the first (corresponding to the first row in the “Design Columns”) estimates the effect of level 1 of A compared to the average effect of the three levels of A. The second (corresponding to the second row in the “Design Columns”) estimates the effect of level 2 of A compared to the average effect of the three levels of A. The sum-to-zero constraint requires the effect of level 3 of A to be the negative of the sum of the level 1 and 2 effects (as shown by the third row in the “Design Columns”).

Crossed Effects (Interactions)

Crossed effects (such as A*B) are formed by the horizontal direct products of main effects, as illustrated in the following table.

Data Levels		**Design Matrix Columns**				
A	B	A		B	A*B	
1	1	1	0	1	1	0
1	2	1	0	–1	–1	0
2	1	0	1	1	0	1
2	2	0	1	–1	0	–1
3	1	–1	–1	1	–1	–1
3	2	–1	–1	–1	1	1

The number of degrees of freedom for a crossed effect (that is, the number of design matrix columns) is equal to the product of the numbers of degrees of freedom for the separate effects.

Nested Effects

The effect A(B) is read "A within B" and is the same as specifying an A main effect for every value of B. If n_a and n_b are the number of levels in A and B, respectively, then the number of columns for A(B) is $(n_a - 1)n_b$ when every combination of levels exists in the data. The following table gives an example.

Data Levels		**Design Matrix Columns**			
B	A	A(B)			
1	1	1	0	0	0
1	2	0	1	0	0
1	3	−1	−1	0	0
2	1	0	0	1	0
2	2	0	0	0	1
2	3	0	0	−1	−1

CAUTION: PROC CATMOD actually allocates a column for all possible combinations of values even though some combinations may not be present in the data. This may be of particular concern if the data are not balanced with respect to the nested levels.

Nested-by-value Effects

Instead of nesting an effect within all values of the main effect, you can nest an effect within specified values of the nested variable (A(B=1), for example). The four degrees of freedom for the A(B) effect shown in the preceding section can also be obtained by specifying the two separate nested effects with values.

Data Levels		**Design Matrix Columns**			
B	A	A(B=1)		A(B=2)	
1	1	1	0	0	0
1	2	0	1	0	0
1	3	−1	−1	0	0
2	1	0	0	1	0
2	2	0	0	0	1
2	3	0	0	−1	−1

Each effect has $n_a - 1$ degrees of freedom, assuming a complete combination. Thus, for the example, each effect has two degrees of freedom.

The procedure compares nested values to data values on the basis of formatted values. If a format is not specified for the variable, the procedure formats internal data values to BEST16, left-justified. The nested values specified in nested-by-value effects are also converted to a BEST16 formatted value, left-justified.

For example, if the numeric variable B has internal data values 1 and 2, then A(B=1), A(B=1.0), and A(B=1E0) are all valid nested-by-value effects. However, if the data value 1 is formatted as 'one', then A(B='one') is a valid effect, but A(B=1) is not since the formatted nested value (1) does not match the formatted data value (one).

To ensure correct nested-by-value effects, look at the tables of population and response profiles. These are displayed by default, and they contain the formatted data values. In addition, the population and response profiles are displayed when you specify the ONEWAY option in the MODEL statement.

Direct Effects

To request that the actual values of a variable be inserted into the design matrix, declare the variable in a DIRECT statement, and specify the effect by the variable name. For example, specifying the effects X1 and X2 in both the MODEL and DIRECT statements results in the following.

Data Levels		**Design Columns**	
X1	X2	X1	X2
1	1	1	1
2	4	2	4
3	9	3	9

Unless there is a POPULATION statement that excludes the direct variables, the direct variables help to define the sample populations. In general, the variables should not be continuous in the sense that every subject has a different value because this would induce a separate population for each subject (note, however, that such a strategy is used purposely for logistic regression).

If there is a POPULATION statement that omits mention of the direct variables, then the values of the direct variables must be identical for all subjects in a given population since there can only be one independent variable profile for each population.

Two or More Response Functions Per Population

When there is more than one response function per population, the structure of the design matrix depends on whether or not the model type is AVERAGED (see the AVERAGED option on page 842). The model type is AVERAGED if independent variable effects are averaged over the multiple responses within a population, rather than being nested in them.

The following subsections illustrate the effect of specifying (or not specifying) an AVERAGED model type. This section does not apply to log-linear models; for these models, see the "Log-Linear Model Design Matrices" section on page 884.

Model Type Not AVERAGED

Suppose the variable A has two levels, and you specify

```
proc catmod;
   model Y=A / design;
run;
```

If the variable Y has two levels, then there is only one response function per population, and the design matrix is as follows.

Sample	Design Matrix	
	Intercept	A
1	1	1
2	1	−1

But if the variable Y has three levels, then there are two response functions per population, and the preceding design matrix is assumed to hold for each of the two response functions. The response functions are always ordered so that the multiple response functions within a population are grouped together. For this example, the design matrix would be as follows.

Sample	Response Function Number	Design Matrix			
		Intercept		A	
1	1	1	0	1	0
1	2	0	1	0	1
2	1	1	0	−1	0
2	2	0	1	0	−1

Since the same submatrix applies to each of the multiple response functions, PROC CATMOD displays only the submatrix (that is, the one it would create if there were only one response function per population) rather than the entire design matrix. PROC CATMOD displays

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Ordering of Parameters

This grouping of multiple response functions within populations also has an effect in the table of parameter estimates displayed by PROC CATMOD. The following table shows some parameter estimates, where the four rows of the table correspond to the four columns in the preceding design matrix.

Effect	Parameter	Estimate
Intercept	1	1.4979
	2	0.8404
A	3	0.1116
	4	−0.3296

Notice that the intercept and the A effect each have two parameter estimates associated with them. The first estimate in each pair is associated with the first response function, and the second in each pair is associated with the second response function. Consequently, 0.1116 is the effect of the first level of A on the first response function. In any table of parameter estimates displayed by PROC CATMOD, as you read down the column of estimates, the response function level changes before levels of the variables making up the effect.

Model Type AVERAGED

When the model type is AVERAGED (for example, when the AVERAGED option is specified in the MODEL statement, when _RESPONSE_ is used in the MODEL statement, or when the design matrix is input directly in the MODEL statement), PROC CATMOD does not assume that the same submatrix applies to each of the q response functions per population. Rather, it averages any independent variable effects across the functions, and it enables you to study variation among the q functions. The first column of the design matrix is always a column of 1s corresponding to the intercept, unless the NOINT option is specified in the MODEL statement or the design matrix is input directly. Also, since the design matrix does not have any special submatrix structure, PROC CATMOD displays the entire matrix.

For example, suppose the dependent variable Y has three levels, the independent variable A has two levels, and you specify

```
proc catmod;
   response marginals;
   model y=a / averaged design;
run;
```

Then there are two response functions per population, and the response functions are always ordered so that the multiple response functions within a population are grouped together. For this example, the design matrix would be as follows.

	Response Function	Design Matrix	
Sample	Number	Intercept	A
1	1	1	1
1	2	1	1
2	1	1	−1
2	2	1	−1

Note that the model now has only two degrees of freedom. The remaining two degrees of freedom in the residual correspond to variation among the three levels of the dependent variable. Generally, that variation tends to be statistically significant and therefore should not be left out of the model. You can include it in the model by including the two effects, _RESPONSE_ and _RESPONSE_*A, but if the study is not a repeated measurement study, those sources of variation tend to be uninteresting. Thus, the usual solution for this type of study (one dependent variable) is to exclude the AVERAGED option from the MODEL statement.

An AVERAGED model type is automatically induced whenever you use the _RESPONSE_ keyword in the MODEL statement. The _RESPONSE_ effect models variation among the q response functions per population. If there is no REPEATED, FACTORS, or LOGLIN statement, then PROC CATMOD builds a main effect with $q - 1$ degrees of freedom. For example, three response functions would induce the following design columns.

Response Function Number	**Design Columns**	
	Response	
1	1	0
2	0	1
3	−1	−1

If there is more than one population, then the _RESPONSE_ effect is averaged over the populations. Also, the _RESPONSE_ effect can be crossed with any other effect, or it can be nested within an effect.

If there is a REPEATED statement that contains only one repeated measurement factor, then PROC CATMOD builds the design columns for _RESPONSE_ in the same way, except that the output labels the main effect with the factor name rather than with the word _RESPONSE_. For example, suppose an independent variable A has two levels, and the input statements are

```
proc catmod;
   response marginals;
   model Time1*Time2=A _response_ A*_response_ / design;
   repeated Time 2 / _response_=Time;
run;
```

If Time1 and Time2 each have two levels (so that they each have one independent marginal probability), then the RESPONSE statement causes PROC CATMOD to compute two response functions per population. Thus, the design matrix is as follows.

	Response	**Design Matrix**			
	Function				
Sample	**Number**	Intercept	A	Time	A*Time
1	1	1	1	1	1
1	2	1	1	−1	−1
2	1	1	−1	1	−1
2	2	1	−1	−1	1

However, if Time1 and Time2 each have three levels (so that they each have two independent marginal probabilities), then the RESPONSE statement causes PROC CATMOD to compute four response functions per population. In that case, since Time has two levels, PROC CATMOD groups the functions into sets of $2\ (= 4/2)$ and constructs the preceding submatrix for each function in the set. This results in the following design matrix, which is obtained from the previous one by multiplying each element by an identity matrix of order two.

	Response	**Design Matrix**							
Sample	**Function**	Intercept		A		Time		A*Time	
1	P(Time1=1)	1	0	1	0	1	0	1	0
1	P(Time1=2)	0	1	0	1	0	1	0	1
1	P(Time2=1)	1	0	1	0	−1	0	−1	0
1	P(Time2=2)	0	1	0	1	0	−1	0	−1
2	P(Time1=1)	1	0	−1	0	1	0	−1	0
2	P(Time1=2)	0	1	0	−1	0	1	0	−1
2	P(Time2=1)	1	0	−1	0	−1	0	1	0
2	P(Time2=2)	0	1	0	−1	0	−1	0	1

If there is a REPEATED statement that contains two or more repeated measurement factors, then PROC CATMOD builds the design columns for _RESPONSE_ according to the definition of _RESPONSE_ in the REPEATED statement. For example, suppose you specify

```
proc catmod;
   response marginals;
   model R11*R12*R21*R22=_response_ / design;
   repeated Time 2, Place 2 / _response_=Time Place;
run;
```

If each of the dependent variables has two levels, then PROC CATMOD builds four response functions. The _RESPONSE_ effect generates a main effects model with respect to Time and Place, and the design matrix is as follows.

Response Function Number	Variable	Time	Place	Design Matrix		
				Intercept	_Response_	
1	R11	1	1	1	1	1
2	R12	1	2	1	1	−1
3	R21	2	1	1	−1	1
4	R22	2	2	1	−1	−1

Log-Linear Model Design Matrices

When the response functions are the standard ones (generalized logits), then inclusion of the keyword _RESPONSE_ in every design effect induces a log-linear model. The design matrix for a log-linear model looks different from a standard design matrix because the standard one is transformed by the same linear transformation that converts the r response probabilities to $r-1$ generalized logits. For example, suppose the dependent variables X and Y each have two levels, and you specify a saturated log-linear model analysis:

```
proc catmod;
   model X*Y=_response_ / design;
   loglin X Y X*Y;
run;
```

Then the cross-classification of X and Y yields four response probabilities, p_{11}, p_{12}, p_{21}, and p_{22}, which are then reduced to three generalized logit response functions, $F_1 = \log(p_{11}/p_{22})$, $F_2 = \log(p_{12}/p_{22})$, and $F_3 = \log(p_{21}/p_{22})$.

Since the saturated log-linear model implies that

$$
\begin{aligned}
\begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix}
&= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \gamma - \lambda \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \boldsymbol{\beta} - \delta \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

where γ and $\boldsymbol{\beta}$ are parameter vectors, and λ and δ are normalizing constants required by the restriction that the probabilities sum to 1, it follows that the MODEL statement yields

$$
\begin{aligned}
\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \boldsymbol{\beta} \\
&= \begin{bmatrix} 2 & 2 & 0 \\ 2 & 0 & -2 \\ 0 & 2 & -2 \end{bmatrix} \boldsymbol{\beta}
\end{aligned}
$$

Thus, the design matrix is as follows.

Sample	**Response Function Number**	**Design Matrix** X	 Y	 X*Y
1	1	2	2	0
1	2	2	0	−2
1	3	0	2	−2

Design matrices for reduced models are constructed similarly. For example, suppose you request a main-effects log-linear model analysis of the factors X and Y:

```
proc catmod;
   model X*Y=_response_ / design;
   loglin X Y;
run;
```

Since the main-effects log-linear model implies that

$$
\begin{aligned}
\begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix} &= \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \gamma - \lambda \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{\beta} - \delta \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\end{aligned}
$$

it follows that the MODEL statement yields

$$
\begin{aligned}
\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{\beta} \\
&= \begin{bmatrix} 2 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \boldsymbol{\beta}
\end{aligned}
$$

Therefore, the corresponding design matrix is as follows.

Sample	Response Function Number	Design Matrix	
		X	Y
1	1	2	2
1	2	2	0
1	3	0	2

Since it is difficult to tell from the final design matrix whether PROC CATMOD used the parameterization that you intended, the procedure displays the untransformed _RESPONSE_ matrix for log-linear models. For example, the main-effects model in the preceding example induces the display of the following matrix.

Response Function Number	_Response_ Matrix	
	1	2
1	1	1
2	1	−1
3	−1	1
4	−1	−1

You can suppress the display of this matrix by specifying the NORESPONSE option in the MODEL statement.

Cautions

Effective Sample Size

Since the method depends on asymptotic approximations, you need to be careful that the sample sizes are sufficiently large to support the asymptotic normal distributions of the response functions. A general guideline is that you would like to have an effective sample size of at least 25 to 30 for each response function that is being analyzed. For example, if you have one dependent variable and $r = 4$ response levels, and you use the standard response functions to compute three generalized logits for each population, then you would like the sample size of each population to be at least 75. Moreover, the subjects should be dispersed throughout the table so that less than 20 percent of the response functions have an effective sample size less than 5. For example, if each population had less than 5 subjects in the first response category, then it would be wiser to pool this category with another category rather than to assume the asymptotic normality of the first response function. Or, if the dependent variable is ordinally scaled, an alternative is to request the mean score response function rather than three generalized logits.

If there is more than one dependent variable, and you specify RESPONSE MEANS, then the effective sample size for each response function is the same as the actual sample size. Thus, a sample size of 30 could be sufficient to support four response functions, provided that the functions are the means of four dependent variables.

A Singular Covariance Matrix

If there is a singular (noninvertible) covariance matrix for the response functions in any population, then PROC CATMOD writes an error message and stops processing. You have several options available to correct this situation:

- You can reduce the number of response functions according to how many can be supported by the populations with the smallest sample sizes.
- If there are three or more levels for any independent variable, you can pool the levels into a fewer number of categories, thereby reducing the number of populations. However, your interpretation of results must be done more cautiously since such pooling implies a different sampling scheme and masks any differences that existed among the pooled categories.
- If there are two or more independent variables, you can delete at least one of them from the model. However, this is just another form of pooling, and the same cautions that apply to the previous option also apply here.
- If there is one independent variable, then, in some situations, you might simply eliminate the populations that are causing the covariance matrices to be singular.
- You can use the ADDCELL= option in the MODEL statement to add a small amount (for example, 0.5) to every cell frequency, but this can seriously bias the results if the cell frequencies are small.

Zero Frequencies

There are two types of zero cells in a contingency table: structural and sampling. A structural zero cell has an expected value of zero, while a sampling zero cell may have nonzero expected value and may be estimable.

If you use the standard response functions and there are zero frequencies, you should use maximum likelihood estimation (the default is ML=NR) rather than weighted least-squares to analyze the data. For weighted least-squares analysis, the CATMOD procedure always computes the observed response functions and may need to take the logarithm of a zero proportion. In this case, PROC CATMOD issues a warning and then takes the log of a small value ($0.5/n_i$ for the probability) in order to continue, but this can produce invalid results if the cells contain too few observations. Maximum likelihood analysis, on the other hand, does not require computation of the observed response functions and therefore yields valid results for the parameter estimates and all of the predicted values.

For a log-linear model analysis using WLS or ML=NR, PROC CATMOD creates response profiles only for the observed profiles. Thus, for any log-linear model analysis with one population (the usual case), the contingency table will not contain zeros, which means that all zero frequencies are treated as structural zeros. If there is more than one population, then a zero in the body of the contingency table is treated as a sampling zero (as long as some population has a nonzero count for that profile). If you fit the log-linear model using ML=IPF, the contingency table is incomplete and the zeros are treated like structural zeros. If you want zero frequencies that PROC CATMOD would normally treat as structural zeros to be interpreted as sampling zeros, you may specify the ZERO=SAMPLING and MISSING=SAMPLING options in the MODEL statement. Alternatively, you can specify ZERO=1E−20 and MISSING=1E−20.

Refer to Bishop, Fienberg, and Holland (1975) for a discussion of the issues and Example 22.5 on page 919 for an illustration of a log-linear model analysis of data that contain both structural and sampling zeros.

If you perform a weighted least-squares analysis on a contingency table that contains zero cell frequencies, then avoid using the LOG transformation as the first transformation on the observed proportions. In general, it may be better to change the response functions or to pool some of the response categories than to settle for the 0.5 correction or to use the ADDCELL= option.

Testing the Wrong Hypothesis

If you use the keyword _RESPONSE_ in the MODEL statement, and you specify MARGINALS, LOGITS, ALOGITS, or CLOGITS in your RESPONSE statement, you may receive the following warning message:

```
Warning: The _RESPONSE_ effect may be testing the wrong
         hypothesis since the marginal levels of the
         dependent variables do not coincide. Consult the
         response profiles and the CATMOD documentation.
```

The following examples illustrate situations in which the _RESPONSE_ effect tests the wrong hypothesis.

Zeros in the Marginal Frequencies

Suppose you specify the following statements:

```
data A1;
   input Time1 Time2 @@;
   datalines;
1 2    2 3    1 3
;

proc catmod;
   response marginals;
   model Time1*Time2=_response_;
   repeated Time 2 / _response_=Time;
run;
```

One marginal probability is computed for each dependent variable, resulting in two response functions. The model is a saturated one: one degree of freedom for the intercept and one for the main effect of Time. Except for the warning message, PROC CATMOD produces an analysis with no apparent errors, but the “Response Profiles” table displayed by PROC CATMOD is as follows.

Response Profiles

Response	Time1	Time2
1	1	2
2	1	3
3	2	3

Since RESPONSE MARGINALS yields marginal probabilities for every level but the last, the two response functions being analyzed are Prob(Time1=1) and Prob(Time2=2). Thus, the Time effect is testing the hypothesis that Prob(Time1=1)=Prob(Time2=2). What it *should* be testing is the hypothesis that

```
Prob(Time1=1) = Prob(Time2=1)
Prob(Time1=2) = Prob(Time2=2)
Prob(Time1=3) = Prob(Time2=3)
```

but there are not enough data to support the test (assuming that none of the probabilities are structural zeros by the design of the study).

The ORDER=DATA Option

Suppose you specify

```
data a1;
   input Time1 Time2 @@;
   datalines;
2 1    2 2    1 1    1 2    2 1
;

proc catmod order=data;
   response marginals;
   model Time1*Time2=_response_;
   repeated Time 2 / _response_=Time;
run;
```

As in the preceding example, one marginal probability is computed for each dependent variable, resulting in two response functions. The model is also the same: one degree of freedom for the intercept and one for the main effect of Time. PROC CATMOD issues the warning message and displays the following “Response Profiles” table.

Response Profiles

Response	Time1	Time2
1	2	1
2	2	2
3	1	1
4	1	2

Although the marginal levels are the same for the two dependent variables, they are not in the same order because the ORDER=DATA option specified that they be ordered according to their appearance in the input stream. Since RESPONSE MARGINALS yields marginal probabilities for every level except the last, the two response functions being analyzed are Prob(Time1=2) and Prob(Time2=1). Thus, the Time effect is testing the hypothesis that Prob(Time1=2)=Prob(Time2=1). What it *should* be testing is the hypothesis that

```
Prob(Time1=1) = Prob(Time2=1)
Prob(Time1=2) = Prob(Time2=2)
```

Whenever the warning message appears, look at the “Response Profiles” table or the “One-Way Frequencies” table to determine what hypothesis is actually being tested. For the latter example, a correct analysis can be obtained by deleting the ORDER=DATA option or by reordering the data so that the (1,1) observation is first.

Computational Method

The notation used in PROC CATMOD differs slightly from that used in other literature. The following table provides a summary of the basic dimensions and the notation for a contingency table. See the "Computational Formulas" section, which follows, for a complete description.

Summary of Basic Dimensions

s = number of populations or samples (= number of rows in the underlying contingency table)

r = number of response categories (= number of columns in the underlying contingency table)

q = number of response functions computed for each population

d = number of parameters

Notation

$\mathbf{j}$ denotes a column vector of 1s.

$\mathbf{J}$ denotes a square matrix of 1s.

$\sum_k$ is the sum over all the possible values of k.

n_i denotes the row sum $\sum_j n_{ij}$.

$\mathbf{DIAG}_n(\mathbf{p})$ is the diagonal matrix formed from the first n elements of the vector $\mathbf{p}$.

$\mathbf{DIAG}_n^{-1}(\mathbf{p})$ is the inverse of $\mathbf{DIAG}_n(\mathbf{p})$.

$\mathbf{DIAG}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k)$ denotes a block diagonal matrix with the $\mathbf{A}$ matrices on the main diagonal.

Input data can be represented by a contingency table, as shown in Table 22.4.

Table 22.4. Input Data Represented by a Contingency Table

	Response				
Population	1	2	$\cdots$	r	**Total**
1	n_{11}	n_{12}	$\cdots$	n_{1r}	n_1
2	n_{21}	n_{22}	$\cdots$	n_{2r}	n_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s	n_{s1}	n_{s2}	$\cdots$	n_{sr}	n_s

Computational Formulas

The following formulas are shown for each population and for all populations combined.

Source	Formula	Dimension
Probability Estimates		
jth response	$p_{ij} = \frac{n_{ij}}{n_i}$	1×1
ith population	$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_{ir} \end{bmatrix}$	$r \times 1$
all populations	$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_s \end{bmatrix}$	$sr \times 1$
Variance of Probability Estimates		
ith population	$\mathbf{V}_i = \frac{1}{n_i}(\mathbf{DIAG}(\mathbf{p}_i) - \mathbf{p}_i\mathbf{p}_i{}')$	$r \times r$
all populations	$\mathbf{V} = \mathbf{DIAG}(\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_s)$	$sr \times sr$
Response Functions		
ith population	$\mathbf{F}_i = \mathbf{F}(\mathbf{p}_i)$	$q \times 1$
all populations	$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_s \end{bmatrix}$	$sq \times 1$
Derivative of Function with Respect to Probability Estimates		
ith population	$\mathbf{H}_i = \frac{\partial \mathbf{F}(\mathbf{p}_i)}{\partial \mathbf{p}_i}$	$q \times r$
all populations	$\mathbf{H} = \mathbf{DIAG}(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_s)$	$sq \times sr$
Variance of Functions		
ith population	$\mathbf{S}_i = \mathbf{H}_i\mathbf{V}_i\mathbf{H}_i{}'$	$q \times q$
all populations	$\mathbf{S} = \mathbf{DIAG}(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_s)$	$sq \times sq$
Inverse Variance of Functions		
ith population	$\mathbf{S}^i = (\mathbf{S}_i)^{-1}$	$q \times q$
all populations	$\mathbf{S}^{-1} = \mathbf{DIAG}(\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^s)$	$sq \times sq$

Derivative Table for Compound Functions: Y=F(G(p))

In the following table, let $\mathbf{G}(\mathbf{p})$ be a vector of functions of $\mathbf{p}$, and let $\mathbf{D}$ denote $\partial\mathbf{G}/\partial\mathbf{p}$, which is the first derivative matrix of $\mathbf{G}$ with respect to $\mathbf{p}$.

Function	$\mathbf{Y} = \mathbf{F}(\mathbf{G})$	**Derivative** $(\partial\mathbf{Y}/\partial\mathbf{p})$
Multiply matrix	$\mathbf{Y} = \mathbf{A} * \mathbf{G}$	$\mathbf{A} * \mathbf{D}$
Logarithm	$\mathbf{Y} = \mathbf{LOG}(\mathbf{G})$	$\mathbf{DIAG}^{-1}(\mathbf{G}) * \mathbf{D}$
Exponential	$\mathbf{Y} = \mathbf{EXP}(\mathbf{G})$	$\mathbf{DIAG}(\mathbf{Y}) * \mathbf{D}$
Add constant	$\mathbf{Y} = \mathbf{G} + \mathbf{A}$	$\mathbf{D}$

Default Response Functions: Generalized Logits

In the following table, subscripts i for the population are suppressed. Also denote $f_j = \log\left(\dfrac{p_j}{p_r}\right)$ for $j = 1, \ldots, r-1$ for each population $i = 1, \ldots, s$.

Inverse of Response Functions for a Population

$$\begin{aligned} p_j &= \frac{\exp(f_j)}{1 + \sum_k \exp(f_k)} \quad \text{for } j = 1, \ldots, r-1 \\ p_r &= \frac{1}{1 + \sum_k \exp(f_k)} \end{aligned}$$

Form of F and Derivative for a Population

$$\begin{aligned} \mathbf{F} &= \mathbf{KLOG}(\mathbf{p}) = (\mathbf{I}_{r-1}, -\mathbf{j})\ \mathbf{LOG}(\mathbf{p}) \\ \mathbf{H} &= \frac{\partial\mathbf{F}}{\partial\mathbf{p}} = \left(\mathbf{DIAG}_{r-1}^{-1}(\mathbf{p}), \frac{-1}{p_r}\mathbf{j}\right) \end{aligned}$$

Covariance Results for a Population

$$\begin{aligned} \mathbf{S} &= \mathbf{HVH}' \\ &= \frac{1}{n}\left(\mathbf{DIAG}_{r-1}^{-1}(\mathbf{p}) + \frac{1}{p_r}\mathbf{J}_{r-1}\right) \\ &\quad \text{where } \mathbf{V}, \mathbf{H}, \text{ and } \mathbf{J} \text{ are as previously defined.} \\ \mathbf{S}^{-1} &= n(\mathbf{DIAG}_{r-1}(\mathbf{p}) - \mathbf{qq}') \quad \text{where } \mathbf{q} = \mathbf{DIAG}_{r-1}(\mathbf{p})\ \mathbf{j} \\ \mathbf{S}^{-1}\mathbf{F} &= n\mathbf{DIAG}_{r-1}(\mathbf{p})\mathbf{F} - (n\sum_j p_j f_j)\ \mathbf{q} \\ \mathbf{F}'\mathbf{S}^{-1}\mathbf{F} &= n\sum_j p_j f_j^2 - n(\sum_j p_j f_j)^2 \end{aligned}$$

The following calculations are shown for each population and then for all populations combined.

Source	Formula	Dimension
Design Matrix		
ith population	$\mathbf{X}_i$	$q \times d$
all populations	$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_s \end{bmatrix}$	$sq \times d$
Crossproduct of Design Matrix		
ith population	$\mathbf{C}_i = \mathbf{X}_i{}'\mathbf{S}^i\mathbf{X}_i$	$d \times d$
all populations	$\mathbf{C} = \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} = \sum_i \mathbf{C}_i$	$d \times d$

In the following table, z_p is the 100pth percentile of the standard normal distribution.

Source	Formula	Dimension
Crossproduct of Design Matrix with Function		
	$\mathbf{R} = \mathbf{X}'\mathbf{S}^{-1}\mathbf{F} = \sum_i \mathbf{X}_i{}'\mathbf{S}^i\mathbf{F}_i$	$d \times 1$
Weighted Least-Squares Estimates		
	$\mathbf{b} = \mathbf{C}^{-1}\mathbf{R} = (\mathbf{X}'\mathbf{S}^{-1}\mathbf{X})^{-1}(\mathbf{X}'\mathbf{S}^{-1}\mathbf{F})$	$d \times 1$
Covariance of Weighted Least-Squares Estimates		
	$\mathbf{COV}(\mathbf{b}) = \mathbf{C}^{-1}$	$d \times d$
Wald Confidence Limits for Parameter Estimates		
	$b_k \pm z_{1-\alpha/2}\mathbf{C}_{kk}^{-1}$	$k = 1, \ldots, d$
Predicted Response Functions		
	$\hat{\mathbf{F}} = \mathbf{X}\mathbf{b}$	$sq \times 1$
Covariance of Predicted Response Functions		
	$\mathbf{V}_{\hat{\mathbf{F}}} = \mathbf{X}\mathbf{C}^{-1}\mathbf{X}'$	$sq \times sq$
Residual Chi-Square		
	$\text{RSS} = \mathbf{F}'\mathbf{S}^{-1}\mathbf{F} - \hat{\mathbf{F}}'\mathbf{S}^{-1}\hat{\mathbf{F}}$	1×1

Source	Formula	Dimension
Chi-Square for $H_0\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$		
	$\mathrm{Q} = (\mathbf{Lb})'(\mathbf{LC}^{-1}\mathbf{L}')^{-1}(\mathbf{Lb})$	1×1

Maximum Likelihood Method

Let $\mathbf{C}$ be the Hessian matrix and $\mathbf{G}$ be the gradient of the log-likelihood function (both functions of π and the parameters $\boldsymbol{\beta}$). Let $\mathbf{p}_i^*$ denote the vector containing the first $r-1$ sample proportions from population i, and let π_i^* denote the corresponding vector of probability estimates from the current iteration. Starting with the least-squares estimates $\mathbf{b}_0$ of $\boldsymbol{\beta}$ (if you use the ML and WLS options; with the ML option alone, the procedure starts with $\mathbf{0}$), the probabilities $\pi(\mathbf{b})$ are computed, and $\mathbf{b}$ is calculated iteratively by the Newton-Raphson method until it converges (see the EPSILON= option on page 842). The factor λ is a step-halving factor that equals one at the start of each iteration. For any iteration in which the likelihood decreases, PROC CATMOD uses a series of subiterations in which λ is iteratively divided by two. The subiterations continue until the likelihood is greater than that of the previous iteration. If the likelihood has not reached that point after ten subiterations, then convergence is assumed, and a warning message is displayed.

Sometimes, infinite parameters may be present in the model, either because of the presence of one or more zero frequencies or because of a poorly specified model with collinearity among the estimates. If an estimate is tending toward infinity, then PROC CATMOD flags the parameter as infinite and holds the estimate fixed in subsequent iterations. PROC CATMOD regards a parameter to be infinite when two conditions apply:

- The absolute value of its estimate exceeds five divided by the range of the corresponding variable.
- The standard error of its estimate is at least three times greater than the estimate itself.

The estimator of the asymptotic covariance matrix of the maximum likelihood predicted probabilities is given by Imrey, Koch, and Stokes (1981, eq. 2.18).

The following equations summarize the method:

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \lambda \mathbf{C}^{-1}\mathbf{G}$$

where

$$\begin{aligned}
\mathbf{C} &= \mathbf{X}'\mathbf{S}^{-1}(\pi)\mathbf{X} \\
\mathbf{N} &= \begin{bmatrix} n_1(\mathbf{p}_1^* - \pi_1^*) \\ \vdots \\ n_s(\mathbf{p}_s^* - \pi_s^*) \end{bmatrix} \\
\mathbf{G} &= \mathbf{X}'\mathbf{N}
\end{aligned}$$

Iterative Proportional Fitting

The algorithm used by PROC CATMOD for iterative proportional fitting is described in Bishop, Fienberg, and Holland (1975), Haberman (1972), and Agresti (2002). To illustrate the method, consider the observed three-dimensional table $\{n_{ijk}\}$ for the variables X, Y, and Z. The statements

```
model X*Y*Z = _response_ / ml=ipf;
loglin X|Y|Z@2;
```

request that PROC CATMOD use IPF to fit the hierarchical model

$$\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$$

Begin with a table of initial cell estimates $\{\hat{m}_{ijk}^{(0)}\}$; PROC CATMOD produces the initial estimates by setting the n_{sz} structural zero cells to 0 and all other cells to $n/(n_c - n_{sz})$, where n is the total weight of the table and n_c is the total number of cells in the table. Iteratively adjust the estimates at step $s-1$ to the observed marginal tables specified in the model by cycling through the following three-stage process to produce the estimates at step s.

$$\begin{aligned}
\hat{m}_{ijk}^{(s_1)} &= \hat{m}_{ijk}^{(s-1)} \frac{n_{ij\cdot}}{\hat{m}_{ij\cdot}^{(s-1)}} \\
\hat{m}_{ijk}^{(s_2)} &= \hat{m}_{ijk}^{(s_1)} \frac{n_{i\cdot k}}{\hat{m}_{i\cdot k}^{(s_1)}} \\
\hat{m}_{ijk}^{(s)} &= \hat{m}_{ijk}^{(s_2)} \frac{n_{\cdot jk}}{\hat{m}_{\cdot jk}^{(s_2)}}
\end{aligned}$$

The subscript "·" indicates summation over the missing subscript. The log-likelihood l_s is estimated at each step s by

$$l_s = \sum_{i,j,k} n_{ijk} \log\left(\frac{\hat{m}_{ijk}^{(s)}}{n}\right)$$

When the function $|(l_{s-1} - l_s)/l_{s-1}|$ is less than 10^{-8}, the iterations terminate. You can change the comparison value with the EPSILON= option, and you can change the convergence criterion with the CONV= option. The option CONV=CELL uses the maximum cell difference

$$\max_{i,j,k} |\hat{m}_{ijk}^{(s-1)} - \hat{m}_{ijk}^{(s)}|$$

as the criterion while the option CONV=MARGIN computes the maximum difference of the margins

$$\text{Maximum of} \left\{ \max_{i,j} |\hat{m}_{ij\cdot}^{(s-1)} - \hat{m}_{ij\cdot}^{(s)}|, \max_{i,k} |\hat{m}_{i\cdot k}^{(s-1)} - \hat{m}_{i\cdot k}^{(s)}|, \max_{j,k} |\hat{m}_{\cdot jk}^{(s-1)} - \hat{m}_{\cdot jk}^{(s)}| \right\}$$

Memory and Time Requirements

The memory and time required by PROC CATMOD are proportional to the number of parameters in the model.

Displayed Output

PROC CATMOD displays the following information in the “Data Summary” table:

- the Response effect
- the Weight Variable, if one is specified
- the Data Set name
- the number of Response Levels
- the number of samples or Populations
- the Total Frequency, which is the total sample size
- the number of Observations from the data set (the number of data records)
- the frequency of missing observations, labeled as “Frequency Missing”

Except for the analysis of variance table, all of the following items can be displayed or suppressed, depending on your specification of statements and options.

- The ONEWAY option produces the “One-Way Frequencies” table, which displays the frequencies of each variable value used in the analysis.
- The populations (or samples) are defined in a table labeled “Population Profiles.” The Sample Size and the values of the defining variables are displayed for each Sample. This table is suppressed if the NOPROFILE option is specified.
- The observed responses are defined in a table labeled “Response Profiles.” The values of the defining variables are displayed for each Response. This table is suppressed if the NOPROFILE option is specified.
- If the FREQ option is specified, then the “Response Frequencies” table is displayed, which shows the frequency of each response for each population.
- If the PROB option is specified, then the “Response Probabilities” table is produced. This table displays the probability of each response for each population.
- If the COV option is specified, the “Response Functions, Covariance Matrix” table, which shows the covariance matrix of the response functions for each Sample, is displayed.
- If the DESIGN option is specified, the Response Functions are displayed in the “Response Functions, Design Matrix” table. If the COV option is also specified, the Response Functions are displayed in the “Response Functions, Covariance Matrix” table.

- If the DESIGN option is specified, the design matrix is displayed in the “Response Functions, Design Matrix” table, and if a log-linear model is being fit, the _RESPONSE_ matrix is displayed in the “_Response_ Matrix” table. If the model type is AVERAGED, then the design matrix is displayed with $q * s$ rows, assuming q response functions for each of s populations. Otherwise, the design matrix is displayed with only s rows since the model is the same for each of the q response functions.
- The “X′*Inv(S)*X” matrix is displayed for weighted least-squares analyses if the XPX option is specified.
- The “Analysis of Variance” table for the weighted least-squares analysis reports the results of significance tests for each of the *design-effects* in the right-hand side of the MODEL statement. If _RESPONSE_ is a *design-effect* and is defined explicitly in the LOGLIN, FACTORS, or REPEATED statement, then the table contains test statistics for the individual effects constituting the _RESPONSE_ effect. If the design matrix is input directly, then the content of the displayed output depends on whether you specify any subsets of the parameters to be tested. If you specify one or more subsets, then the table contains one test for each subset. Otherwise, the table contains one test for the effect MODEL | MEAN. In every case, the table also contains the Residual goodness-of-fit test. Produced for each test of significance are the Source of variation, the number of degrees of freedom (DF), the Chi-Square value (which is a Wald statistic), and the significance probability (Pr > ChiSq).
- The “Analysis of Weighted Least-Squares Estimates” table lists, for each parameter in the model, the least-squares Estimate, the estimated Standard Error of the parameter estimate, the Chi-Square value (a Wald statistic, calculated as ((parameter estimate)/(standard error))^2) for testing that the parameter is zero, and the significance probability (Pr > ChiSq) of the test. If the CLPARM option is specified, then 95$\%$ Wald confidence intervals are displayed.

  Each row in the table is labeled with the Parameter (the model effect and the class levels) and the response Function Number; however, if the NOPREDVAR option or a REPEATED or FACTORS statement is specified or if the design matrix is directly input, the rows are labeled by the Effect in the model for which parameters are formed and the Parameter number.
- The “Covariance Matrix of the Parameter Estimates” table for the weighted least-squares analysis displays the estimated covariance matrix of the least-squares estimates of the parameters, provided the COVB option is specified.
- The “Correlation Matrix of the Parameter Estimates” table for the weighted least-squares analysis displays the estimated correlation matrix of the least-squares estimates of the parameters, provided that the CORRB option is specified.
- The “Maximum Likelihood Analysis” table is produced when the ML and ITPRINT options are specified for the standard response functions. It displays the Iteration number, the number of step-halving Sub-Iterations, -2 Log Likelihood for that iteration, the Convergence Criterion, and the Parameter Estimates for each iteration.

- The "Maximum Likelihood Analysis of Variance" table, displayed when the ML option is specified for the standard response functions, is similar to the table produced for the least-squares analysis. The Chi-Square test for each effect is a Wald test based on the information matrix from the likelihood calculations. The Likelihood Ratio statistic compares the specified model with the unrestricted (saturated) model and is an appropriate goodness-of-fit test for the model.
- The "Analysis of Maximum Likelihood Estimates" table, displayed when the ML option is specified for the standard response functions, is similar to the one produced for the least-squares analysis. The table includes the maximum likelihood estimates, the estimated Standard Errors based on the information matrix, and the Wald statistics (Chi-Square) based on estimated standard errors.
- The "Covariance Matrix of the Maximum Likelihood Estimates" table displays the estimated covariance matrix of the maximum likelihood estimates of the parameters, provided that the COVB and ML options are specified for the standard response functions.
- The "Correlation Matrix of the Maximum Likelihood Estimates" table displays the estimated correlation matrix of the maximum likelihood estimates of the parameters, provided that the CORRB and ML options are specified for the standard response functions.
- For each source of variation specified in a CONTRAST statement, the "Contrasts" table lists the label for the source (Contrast), the number of degrees of freedom (DF), the Chi-Square value (which is a Wald statistic), and the significance probability (Pr > ChiSq). If the ESTIMATE= option is specified, the "Analysis of Contrasts" table displays, for each row of the contrast, the label (Contrast), the Type (PARM or EXP), the Row of the contrast, the Estimate and its Standard Error, a Wald confidence interval, the Wald Chi-Square, and the p-value (Pr > ChiSq) for 1 degree of freedom.
- Specification of the PREDICT option in the MODEL statement has the following effect. Produced for each response function within each population are the Observed and Predicted Function values, their Standard Errors, and the Residual (Observed − Predicted). If the response functions are the default ones (generalized logits), additional information displayed for each response within each population includes the Observed and Predicted cell probabilities, their Standard Errors, and the Residual. However, specifying PRED=FREQ in the MODEL statement results in the display of the predicted cell frequencies, rather than the predicted cell probabilities. The displayed output includes the population profiles and, for the response function table, the Function Number, while the probability and frequency tables display the response profiles. If the NOPREDVAR option is specified in the MODEL statement, the population profiles are replaced with the Sample numbers, and the response profiles are replaced with the labels Pn for the nth cell probability, and Fn for the nth cell frequency.
- When there are multiple RESPONSE statements, the output for each statement starts on a new page. For each RESPONSE statement, the corresponding title, if specified, is displayed at the top of each page.

- If the ADDCELL= option is specified in the MODEL statement, and if there is a weighted least-squares analysis specified, the adjusted sample size for each population (with number added to each cell) is labeled Adjusted Sample Size in the "Population Profiles" table. Similarly, the adjusted response frequencies and probabilities are displayed in the "Adjusted Response Frequencies" and "Adjusted Response Probabilities" tables, respectively.
- If _RESPONSE_ is defined explicitly in the LOGLIN, FACTORS, or REPEATED statement, then the definition is displayed as a NOTE whenever _RESPONSE_ appears in the output.

ODS Table Names

PROC CATMOD assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 22.5. ODS Tables Produced in PROC CATMOD

ODS Table Name	Description	Statement	Option
ANOVA	Analysis of variance	MODEL	default
Contrasts	Contrasts	CONTRAST	default
ContrastEstimates	Analysis of Contrasts	CONTRAST	ESTIMATE=
ConvergenceStatus	Convergence status	MODEL	ML
CorrB	Correlation matrix of the estimates	MODEL	CORRB
CovB	Covariance matrix of the estimates	MODEL	COVB
DataSummary	Data summary	PROC	default
Estimates	Analysis of estimates	MODEL	default, unless NOPARM
MaxLikelihood	Maximum likelihood analysis	MODEL	ML and ITPRINT
OneWayFreqs	One-way frequencies	MODEL	ONEWAY
PopProfiles	Population profiles	MODEL	default, unless NOPROFILE
PredictedFreqs	Predicted frequencies	MODEL	PRED=FREQ
PredictedProbs	Predicted probabilities	MODEL	PREDICT or PRED=PROB
PredictedValues	Predicted values	MODEL	PREDICT or PRED=
ResponseCov	Response functions, covariance matrix	MODEL	COV
ResponseDesign	Response functions, design matrix	MODEL	DESIGN, unless NODESIGN
ResponseFreqs	Response frequencies	MODEL	FREQ
ResponseMatrix	_RESPONSE_ matrix	MODEL & LOGLIN	DESIGN, unless NORESPONSE
ResponseProbs	Response probabilities	MODEL	PROB
ResponseProfiles	Response profiles	MODEL	default, unless NOPROFILE
XPX	**X**'*Inv(**S**)***X** matrix	MODEL	XPX, for WLS*

* WLS estimation is the default for response functions other than the default (generalized logits).

Examples

Example 22.1. Linear Response Function, r=2 Responses

In an example from Ries and Smith (1963), the choice of detergent brand (Brand= M or X) is related to three other categorical variables: the softness of the laundry water (Softness= soft, medium, or hard), the temperature of the water (Temperature= high or low), and whether the subject was a previous user of Brand M (Previous= yes or no). The linear response function, which could also be specified as RESPONSE MARGINALS, yields one probability, Pr(brand preference=M), as the response function to be analyzed. Two models are fit in this example: the first model is a saturated one, containing all of the main effects and interactions, while the second is a reduced model containing only the main effects. The following statements produce Output 22.1.1 through Output 22.1.4:

```
data detergent;
   input Softness $ Brand $ Previous $ Temperature $ Count @@;
   datalines;
soft X yes high 19   soft X yes low 57
soft X no  high 29   soft X no  low 63
soft M yes high 29   soft M yes low 49
soft M no  high 27   soft M no  low 53
med  X yes high 23   med  X yes low 47
med  X no  high 33   med  X no  low 66
med  M yes high 47   med  M yes low 55
med  M no  high 23   med  M no  low 50
hard X yes high 24   hard X yes low 37
hard X no  high 42   hard X no  low 68
hard M yes high 43   hard M yes low 52
hard M no  high 30   hard M no  low 42
;
```

```
title 'Detergent Preference Study';
proc catmod data=detergent;
   response 1 0;
   weight Count;
   model Brand=Softness|Previous|Temperature / freq prob;
   title2 'Saturated Model';
run;
```

Output 22.1.1. Detergent Preference Study: Linear Model Analysis

```
                         Detergent Preference Study
                              Saturated Model

                            The CATMOD Procedure

                                Data Summary

       Response            Brand         Response Levels     2
       Weight Variable     Count         Populations        12
       Data Set            DETERGENT     Total Frequency  1008
       Frequency Missing   0             Observations       24
```

The “Data Summary” table (Output 22.1.1) indicates that you have two response levels and twelve populations.

Output 22.1.2. Population Profiles

```
                         Detergent Preference Study
                              Saturated Model

                            Population Profiles

    Sample    Softness    Previous    Temperature    Sample Size
    ------------------------------------------------------------
       1      hard        no          high                    72
       2      hard        no          low                    110
       3      hard        yes         high                    67
       4      hard        yes         low                     89
       5      med         no          high                    56
       6      med         no          low                    116
       7      med         yes         high                    70
       8      med         yes         low                    102
       9      soft        no          high                    56
      10      soft        no          low                    116
      11      soft        yes         high                    48
      12      soft        yes         low                    106
```

The “Population Profiles” table in Output 22.1.2 displays the ordering of independent variable levels as used in the table of parameter estimates.

Output 22.1.3. Response Profiles, Frequencies, and Probabilities

```
                    Detergent Preference Study
                         Saturated Model

                        Response Profiles

                        Response    Brand
                        -----------------
                            1       M
                            2       X


                       Response Frequencies

                              Response Number
                     Sample         1        2
                     ------------------------
                         1         30       42
                         2         42       68
                         3         43       24
                         4         52       37
                         5         23       33
                         6         50       66
                         7         47       23
                         8         55       47
                         9         27       29
                        10         53       63
                        11         29       19
                        12         49       57


                      Response Probabilities

                               Response Number
                   Sample           1          2
                   ----------------------------
                       1     0.41667    0.58333
                       2     0.38182    0.61818
                       3     0.64179    0.35821
                       4     0.58427    0.41573
                       5     0.41071    0.58929
                       6     0.43103    0.56897
                       7     0.67143    0.32857
                       8     0.53922    0.46078
                       9     0.48214    0.51786
                      10     0.45690    0.54310
                      11     0.60417    0.39583
                      12     0.46226    0.53774
```

Since Brand M is the first level in the “Response Profiles” table (Output 22.1.3), the RESPONSE statement causes Pr(Brand=M) to be the single response function modeled.

Output 22.1.4. Analysis of Variance and WLS Estimates

```
                           Detergent Preference Study
                                Saturated Model

                              Analysis of Variance

           Source                        DF   Chi-Square    Pr > ChiSq
           ----------------------------------------------------------
           Intercept                      1       983.13        <.0001
           Softness                       2         0.09        0.9575
           Previous                       1        22.68        <.0001
           Softness*Previous              2         3.85        0.1457
           Temperature                    1         3.67        0.0555
           Softness*Temperature           2         0.23        0.8914
           Previous*Temperature           1         2.26        0.1324
           Softnes*Previou*Temperat       2         0.76        0.6850

           Residual                       0          .            .


                 Analysis of Weighted Least Squares Estimates

                                                   Standard       Chi-
Parameter                             Estimate        Error     Square  Pr > ChiSq
----------------------------------------------------------------------------------
Intercept                               0.5069       0.0162     983.13      <.0001
Softness                 hard         -0.00073       0.0225       0.00      0.9740
                         med           0.00623       0.0226       0.08      0.7830
Previous                 no            -0.0770       0.0162      22.68      <.0001
Softness*Previous        hard no       -0.0299       0.0225       1.77      0.1831
                         med no        -0.0152       0.0226       0.45      0.5007
Temperature              high           0.0310       0.0162       3.67      0.0555
Softness*Temperature     hard high    -0.00786       0.0225       0.12      0.7265
                         med high     -0.00298       0.0226       0.02      0.8953
Previous*Temperature     no high       -0.0243       0.0162       2.26      0.1324
Softnes*Previou*Temperat hard no high   0.0187       0.0225       0.69      0.4064
                         med no high   -0.0138       0.0226       0.37      0.5415
```

The “Analysis of Variance” table in Output 22.1.4 shows that all of the interactions are nonsignificant. Therefore, a main-effects model is fit with the following statements:

```
      model Brand=Softness Previous Temperature
           / clparm noprofile design;
      title2 'Main-Effects Model';
   run;
   quit;
```

The PROC CATMOD statement is not required due to the interactive capability of the CATMOD procedure. The NOPROFILE option suppresses the redisplay of the “Response Profiles” table. The CLPARM option produces 95% confidence limits for the parameter estimates. Output 22.1.5 through Output 22.1.7 are produced.

Output 22.1.5. Main-Effects Design Matrix

```
                        Detergent Preference Study
                            Main-Effects Model

                          The CATMOD Procedure

                              Data Summary

          Response           Brand          Response Levels     2
          Weight Variable    Count          Populations        12
          Data Set           DETERGENT      Total Frequency  1008
          Frequency Missing  0              Observations       24


                   Response Functions and Design Matrix

                  Response                 Design Matrix
    Sample        Function        1        2        3        4        5
    -----------------------------------------------------------------
       1          0.41667         1        1        0        1        1
       2          0.38182         1        1        0        1       -1
       3          0.64179         1        1        0       -1        1
       4          0.58427         1        1        0       -1       -1
       5          0.41071         1        0        1        1        1
       6          0.43103         1        0        1        1       -1
       7          0.67143         1        0        1       -1        1
       8          0.53922         1        0        1       -1       -1
       9          0.48214         1       -1       -1        1        1
      10          0.45690         1       -1       -1        1       -1
      11          0.60417         1       -1       -1       -1        1
      12          0.46226         1       -1       -1       -1       -1
```

The design matrix in Output 22.1.5 displays the results of the factor effects modeling used in PROC CATMOD.

Output 22.1.6. ANOVA Table for the Main-Effects Model

```
                        Detergent Preference Study
                            Main-Effects Model

                          Analysis of Variance

              Source          DF   Chi-Square    Pr > ChiSq
              ---------------------------------------------
              Intercept        1      1004.93        <.0001
              Softness         2         0.24        0.8859
              Previous         1        20.96        <.0001
              Temperature      1         3.95        0.0468

              Residual         7         8.26        0.3100
```

The analysis of variance table in Output 22.1.6 shows that previous use of Brand M, together with the temperature of the laundry water, are significant factors in preferring Brand M laundry detergent. The table also shows that the additive model fits since the goodness-of-fit statistic (the Residual Chi-Square) is nonsignificant.

Output 22.1.7. WLS Estimates for the Main-Effects Model

```
                       Detergent Preference Study
                           Main-Effects Model

              Analysis of Weighted Least Squares Estimates

                            Standard      Chi-                 95% Confidence
Parameter        Estimate      Error    Square  Pr > ChiSq         Limits
-------------------------------------------------------------------------------
Intercept          0.5080     0.0160   1004.93      <.0001    0.4766    0.5394
Softness    hard -0.00256     0.0218      0.01      0.9066   -0.0454    0.0402
            med    0.0104     0.0218      0.23      0.6342   -0.0323    0.0530
Previous    no    -0.0711     0.0155     20.96      <.0001   -0.1015   -0.0407
Temperature high   0.0319     0.0161      3.95      0.0468  0.000446    0.0634
```

The chi-square test in Output 22.1.7 shows that the Softness parameters are not significantly different from zero; as expected, the Wald confidence limits for these two estimates contain zero. So softness of the water is not a factor in choosing Brand M.

The negative coefficient for Previous (-0.0711) indicates that the first level of Previous (which, from the table of population profiles, is 'no') is associated with a smaller probability of preferring Brand M than the second level of Previous (with coefficient constrained to be 0.0711 since the parameter estimates for a given effect must sum to zero). In other words, previous users of Brand M are much more likely to prefer it than those who have never used it before.

Similarly, the positive coefficient for Temperature indicates that the first level of Temperature (which, from the "Population Profiles" table, is 'high') has a larger probability of preferring Brand M than the second level of Temperature. In other words, those who do their laundry in hot water are more likely to prefer Brand M than those who do their laundry in cold water.

Example 22.2. Mean Score Response Function, r=3 Responses

Four surgical operations for duodenal ulcers are compared in a clinical trial at four hospitals. The operations performed are: Treatment=a, drainage and vagotomy; Treatment=b, 25%resection and vagotomy; Treatment=c, 50%resection and vagotomy; and Treatment=d, 75%resection. The response is severity of an undesirable complication called "dumping syndrome." The data are from Grizzle, Starmer, and Koch (1969, pp. 489–504).

```
data operate;
   input Hospital Treatment $ Severity $ wt @@;
   datalines;
1 a none 23    1 a slight  7    1 a moderate 2
1 b none 23    1 b slight 10    1 b moderate 5
1 c none 20    1 c slight 13    1 c moderate 5
1 d none 24    1 d slight 10    1 d moderate 6
```

```
   2 a none 18    2 a slight  6    2 a moderate 1
   2 b none 18    2 b slight  6    2 b moderate 2
   2 c none 13    2 c slight 13    2 c moderate 2
   2 d none  9    2 d slight 15    2 d moderate 2
   3 a none  8    3 a slight  6    3 a moderate 3
   3 b none 12    3 b slight  4    3 b moderate 4
   3 c none 11    3 c slight  6    3 c moderate 2
   3 d none  7    3 d slight  7    3 d moderate 4
   4 a none 12    4 a slight  9    4 a moderate 1
   4 b none 15    4 b slight  3    4 b moderate 2
   4 c none 14    4 c slight  8    4 c moderate 3
   4 d none 13    4 d slight  6    4 d moderate 4
   ;
```

The response variable (Severity) is ordinally scaled with three levels, so assignment of scores is appropriate (0=none, 0.5=slight, 1=moderate). For these scores, the response function yields the mean score. The following statements produce Output 22.2.1 through Output 22.2.6.

```
   title 'Dumping Syndrome Data';
   proc catmod data=operate order=data ;
      weight wt;
      response 0  0.5  1;
      model Severity=Treatment Hospital / freq oneway design;
      title2 'Main-Effects Model';
   quit;
```

The ORDER= option is specified so that the levels of the response variable remain in the correct order. A main effects model is fit. The FREQ option displays the frequency of each response within each sample (Output 22.2.3), and the ONEWAY option produces a table of the number of subjects within each variable level (Output 22.2.1).

Output 22.2.1. Surgical Data: Analysis of Mean Scores

```
                             Dumping Syndrome Data
                               Main-Effects Model

                              The CATMOD Procedure

                                  Data Summary

              Response           Severity     Response Levels    3
              Weight Variable    wt           Populations       16
              Data Set           OPERATE      Total Frequency  417
              Frequency Missing  0            Observations      48


                              One-Way Frequencies

                       Variable     Value      Frequency
                       ---------------------------------
                       Severity     none             240
                                    slight           129
                                    moderate          48

                       Treatment    a                 96
                                    b                104
                                    c                110
                                    d                107

                       Hospital            1         148
                                           2         105
                                           3          74
                                           4          90
```

Output 22.2.2. Population Sizes

```
                             Dumping Syndrome Data
                               Main-Effects Model

                              Population Profiles

                 Sample    Treatment    Hospital    Sample Size
                 ----------------------------------------------
                      1    a            1                    32
                      2    a            2                    25
                      3    a            3                    17
                      4    a            4                    22
                      5    b            1                    38
                      6    b            2                    26
                      7    b            3                    20
                      8    b            4                    20
                      9    c            1                    38
                     10    c            2                    28
                     11    c            3                    19
                     12    c            4                    25
                     13    d            1                    40
                     14    d            2                    26
                     15    d            3                    18
                     16    d            4                    23
```

Output 22.2.3. Response Frequencies

```
      Dumping Syndrome Data
        Main-Effects Model

        Response Profiles

       Response    Severity
       --------------------
           1       none
           2       slight
           3       moderate


      Response Frequencies

             Response Number
Sample        1        2        3
---------------------------------
    1        23        7        2
    2        18        6        1
    3         8        6        3
    4        12        9        1
    5        23       10        5
    6        18        6        2
    7        12        4        4
    8        15        3        2
    9        20       13        5
   10        13       13        2
   11        11        6        2
   12        14        8        3
   13        24       10        6
   14         9       15        2
   15         7        7        4
   16        13        6        4
```

You can use the oneway frequencies (Output 22.2.1) and the response profiles (Output 22.2.3) to verify that the response levels are in the desired order (none, slight, moderate) so that the response scores (0, 0.5, 1.0) are applied appropriately. If the ORDER=DATA option had not been used, the levels would have been in a different order.

Output 22.2.4. Design Matrix

```
                                 Dumping Syndrome Data
                                  Main-Effects Model

                         Response Functions and Design Matrix

              Response                          Design Matrix
Sample        Function       1       2       3       4       5       6       7
------------------------------------------------------------------------------
    1          0.17188       1       1       0       0       1       0       0
    2          0.16000       1       1       0       0       0       1       0
    3          0.35294       1       1       0       0       0       0       1
    4          0.25000       1       1       0       0      -1      -1      -1
    5          0.26316       1       0       1       0       1       0       0
    6          0.19231       1       0       1       0       0       1       0
    7          0.30000       1       0       1       0       0       0       1
    8          0.17500       1       0       1       0      -1      -1      -1
    9          0.30263       1       0       0       1       1       0       0
   10          0.30357       1       0       0       1       0       1       0
   11          0.26316       1       0       0       1       0       0       1
   12          0.28000       1       0       0       1      -1      -1      -1
   13          0.27500       1      -1      -1      -1       1       0       0
   14          0.36538       1      -1      -1      -1       0       1       0
   15          0.41667       1      -1      -1      -1       0       0       1
   16          0.30435       1      -1      -1      -1      -1      -1      -1
```

Output 22.2.5. ANOVA Table

```
                                 Dumping Syndrome Data
                                  Main-Effects Model

                                 Analysis of Variance

                   Source        DF   Chi-Square    Pr > ChiSq
                   -------------------------------------------
                   Intercept      1       248.77        <.0001
                   Treatment      3         8.90        0.0307
                   Hospital       3         2.33        0.5065

                   Residual       9         6.33        0.7069
```

The analysis of variance table (Output 22.2.5) shows that the additive model fits (since the Residual Chi-Square is not significant), that the Treatment effect is significant, and that the Hospital effect is not significant.

Output 22.2.6. Parameter Estimates

```
                         Dumping Syndrome Data
                           Main-Effects Model

              Analysis of Weighted Least Squares Estimates

                                    Standard        Chi-
Parameter        Estimate        Error        Square    Pr > ChiSq
------------------------------------------------------------
Intercept          0.2724       0.0173        248.77        <.0001
Treatment  a      -0.0552       0.0270          4.17        0.0411
           b      -0.0365       0.0289          1.59        0.2073
           c       0.0248       0.0280          0.78        0.3757
Hospital   1      -0.0204       0.0264          0.60        0.4388
           2      -0.0178       0.0268          0.44        0.5055
           3       0.0531       0.0352          2.28        0.1312
```

The coefficients of Treatment in Output 22.2.6 show that the first two treatments (with negative coefficients) have lower mean scores than the last two treatments (the fourth coefficient, not shown, must be positive since the four coefficients must sum to zero). In other words, the less severe treatments (the first two) cause significantly less severe dumping syndrome complications.

Example 22.3. Logistic Regression, Standard Response Function

In this data set, from Cox and Snell (1989), ingots are prepared with different heating and soaking times and tested for their readiness to be rolled. The response variable Y has value 1 for ingots that are not ready and value 0 otherwise. The explanatory variables are Heat and Soak.

```
data ingots;
   input Heat Soak nready ntotal @@;
   Count=nready;
   Y=1;
   output;
   Count=ntotal-nready;
   Y=0;
   output;
   drop nready ntotal;
   datalines;
7 1.0 0 10   14 1.0 0 31   27 1.0 1 56   51 1.0 3 13
7 1.7 0 17   14 1.7 0 43   27 1.7 4 44   51 1.7 0  1
7 2.2 0  7   14 2.2 2 33   27 2.2 0 21   51 2.2 0  1
7 2.8 0 12   14 2.8 0 31   27 2.8 1 22   51 4.0 0  1
7 4.0 0  9   14 4.0 0 19   27 4.0 1 16
;
```

Logistic regression analysis is often used to investigate the relationship between discrete response variables and continuous explanatory variables. For logistic regression, the continuous *design-effects* are declared in a DIRECT statement. The following statements produce Output 22.3.1 through Output 22.3.8.

```
title 'Maximum Likelihood Logistic Regression';
proc catmod data=ingots;
   weight Count;
   direct Heat Soak;
   model Y=Heat Soak / freq covb corrb itprint design;
quit;
```

Output 22.3.1. Maximum Likelihood Logistic Regression

```
                      Maximum Likelihood Logistic Regression

                               The CATMOD Procedure

                                   Data Summary

               Response            Y          Response Levels    2
               Weight Variable     Count      Populations       19
               Data Set            INGOTS     Total Frequency  387
               Frequency Missing   0          Observations      25


                               Population Profiles

                     Sample    Heat    Soak    Sample Size
                     -------------------------------------
                         1       7        1             10
                         2       7      1.7             17
                         3       7      2.2              7
                         4       7      2.8             12
                         5       7        4              9
                         6      14        1             31
                         7      14      1.7             43
                         8      14      2.2             33
                         9      14      2.8             31
                        10      14        4             19
                        11      27        1             56
                        12      27      1.7             44
                        13      27      2.2             21
                        14      27      2.8             22
                        15      27        4             16
                        16      51        1             13
                        17      51      1.7              1
                        18      51      2.2              1
                        19      51        4              1
```

You can verify that the populations are defined as you intended by looking at the "Population Profiles" table in Output 22.3.1.

Output 22.3.2. Response Summaries

```
Maximum Likelihood Logistic Regression

          Response Profiles

            Response    Y
            -------------
                1       0
                2       1


         Response Frequencies

                Response Number
       Sample        1        2
       ------------------------
           1        10        0
           2        17        0
           3         7        0
           4        12        0
           5         9        0
           6        31        0
           7        43        0
           8        31        2
           9        31        0
          10        19        0
          11        55        1
          12        40        4
          13        21        0
          14        21        1
          15        15        1
          16        10        3
          17         1        0
          18         1        0
          19         1        0
```

Since the “Response Profiles” table shows the response level ordering as 0, 1, the default response function, the logit, is defined as $\log\left(\frac{p_{Y=0}}{p_{Y=1}}\right)$.

Output 22.3.3. Design Matrix

```
                     Maximum Likelihood Logistic Regression

                     Response Functions and Design Matrix

                            Response        Design Matrix
                Sample      Function        1        2        3
                ----------------------------------------------
                    1        2.99573        1        7        1
                    2        3.52636        1        7      1.7
                    3        2.63906        1        7      2.2
                    4        3.17805        1        7      2.8
                    5        2.89037        1        7        4
                    6        4.12713        1       14        1
                    7        4.45435        1       14      1.7
                    8        2.74084        1       14      2.2
                    9        4.12713        1       14      2.8
                   10        3.63759        1       14        4
                   11        4.00733        1       27        1
                   12        2.30259        1       27      1.7
                   13        3.73767        1       27      2.2
                   14        3.04452        1       27      2.8
                   15        2.70805        1       27        4
                   16        1.20397        1       51        1
                   17        0.69315        1       51      1.7
                   18        0.69315        1       51      2.2
                   19        0.69315        1       51        4
```

The values of the continuous variable are inserted into the design matrix.

Output 22.3.4. Iteration History

```
                      Maximum Likelihood Logistic Regression

                           Maximum Likelihood Analysis

                 Sub        -2 Log  Convergence        Parameter Estimates
Iteration  Iteration    Likelihood    Criterion          1          2          3
------------------------------------------------------------------------------
     0          0       536.49592       1.0000          0          0          0
     1          0       152.58961       0.7156     2.1594    -0.0139  -0.003733
     2          0       106.76066       0.3003     3.5334    -0.0363    -0.0120
     3          0       96.692171       0.0943     4.7489    -0.0640    -0.0299
     4          0       95.383825       0.0135     5.4138    -0.0790    -0.0498
     5          0       95.345659     0.000400     5.5539    -0.0819    -0.0564
     6          0       95.345613    4.8289E-7     5.5592    -0.0820    -0.0568
     7          0       95.345613    7.731E-13     5.5592    -0.0820    -0.0568

                    Maximum likelihood computations converged.
```

Seven Newton-Raphson iterations are required to find the maximum likelihood estimates.

Output 22.3.5. Analysis of Variance Table

```
                      Maximum Likelihood Logistic Regression

                     Maximum Likelihood Analysis of Variance

                Source               DF   Chi-Square    Pr > ChiSq
                --------------------------------------------------
                Intercept             1        24.65        <.0001
                Heat                  1        11.95        0.0005
                Soak                  1         0.03        0.8639

                Likelihood Ratio     16        13.75        0.6171
```

The analysis of variance table (Output 22.3.5) shows that the model fits since the likelihood ratio goodness-of-fit test is nonsignificant. It also shows that the length of heating time is a significant factor with respect to readiness but that length of soaking time is not.

Output 22.3.6. Maximum Likelihood Estimates

```
                      Maximum Likelihood Logistic Regression

                     Analysis of Maximum Likelihood Estimates

                                    Standard        Chi-
            Parameter    Estimate      Error      Square    Pr > ChiSq
            ----------------------------------------------------------
            Intercept      5.5592     1.1197       24.65        <.0001
            Heat          -0.0820     0.0237       11.95        0.0005
            Soak          -0.0568     0.3312        0.03        0.8639
```

Output 22.3.7. Covariance Matrix

```
                      Maximum Likelihood Logistic Regression

               Covariance Matrix of the Maximum Likelihood Estimates

        Row      Parameter             Col1             Col2             Col3
        -------------------------------------------------------------------
          1      Intercept        1.2537133       -0.0215664       -0.2817648
          2      Heat            -0.0215664        0.0005633        0.0026243
          3      Soak            -0.2817648        0.0026243        0.1097020
```

Output 22.3.8. Correlation Matrix

```
                      Maximum Likelihood Logistic Regression

              Correlation Matrix of the Maximum Likelihood Estimates

       Row      Parameter             Col1             Col2             Col3
       -----------------------------------------------------------------------
         1      Intercept          1.00000         -0.81152         -0.75977
         2      Heat              -0.81152          1.00000          0.33383
         3      Soak              -0.75977          0.33383          1.00000
```

From the table of maximum likelihood estimates (Output 22.3.6), the fitted model is

$$\text{E}(\text{logit}(p)) = 5.559 - 0.082(\textsf{Heat}) - 0.057(\textsf{Soak})$$

For example, for Sample 1 with Heat $= 7$ and Soak $= 1$, the estimate is

$$\text{E}(\text{logit}(p)) = 5.559 - 0.082(7) - 0.057(1) = 4.9284$$

Predicted values of the logits, as well as the probabilities of readiness, could be obtained by specifying PRED=PROB in the MODEL statement. For the example of Sample 1 with Heat $= 7$ and Soak $= 1$, PRED=PROB would give an estimate of the probability of readiness equal to 0.9928 since

$$4.9284 = \log\left(\frac{\hat{p}}{1-\hat{p}}\right)$$

implies that

$$\hat{p} = \frac{e^{4.9284}}{1+e^{4.9284}} = 0.9928$$

As another consideration, since soaking time is nonsignificant, you could fit another model that deleted the variable Soak.

Example 22.4. Log-Linear Model, Three Dependent Variables

This analysis reproduces the predicted cell frequencies for Bartlett's data using a log-linear model of no three-variable interaction (Bishop, Fienberg, and Holland 1975, p. 89). Cuttings of two different lengths (Length=short or long) are planted at one of two time points (Time=now or spring), and their survival status (Status=dead or alive) is recorded.

As in the text, the variable levels are simply labeled 1 and 2. The following statements produce Output 22.4.1 through Output 22.4.5:

```
data bartlett;
   input Length Time Status wt @@;
   datalines;
1 1 1 156     1 1 2  84     1 2 1 84     1 2 2 156
2 1 1 107     2 1 2 133     2 2 1 31     2 2 2 209
;

title 'Bartlett''s Data';
proc catmod data=bartlett;
   weight wt;
   model Length*Time*Status=_response_
         / noparm pred=freq;
   loglin Length|Time|Status @ 2;
   title2 'Model with No 3-Variable Interaction';
quit;
```

Output 22.4.1. Analysis of Bartlett's Data: Log-Linear Model

```
                               Bartlett's Data
                    Model with No 3-Variable Interaction

                            The CATMOD Procedure

                                Data Summary

     Response           Length*Time*Status     Response Levels    8
     Weight Variable    wt                     Populations        1
     Data Set           BARTLETT               Total Frequency  960
     Frequency Missing  0                      Observations       8


                            Population Profiles

                           Sample    Sample Size
                           ---------------------
                               1            960
```

Output 22.4.2. Response Profiles

```
                               Bartlett's Data
                    Model with No 3-Variable Interaction

                             Response Profiles

                    Response    Length    Time    Status
                    ------------------------------------
                        1       1         1       1
                        2       1         1       2
                        3       1         2       1
                        4       1         2       2
                        5       2         1       1
                        6       2         1       2
                        7       2         2       1
                        8       2         2       2
```

Output 22.4.3. Analysis of Variance Table

```
                         Bartlett’s Data
                 Model with No 3-Variable Interaction

               Maximum Likelihood Analysis of Variance

          Source               DF   Chi-Square    Pr > ChiSq
          --------------------------------------------------
          Length                1         2.64        0.1041
          Time                  1         5.25        0.0220
          Length*Time           1         5.25        0.0220
          Status                1        48.94        <.0001
          Length*Status         1        48.94        <.0001
          Time*Status           1        95.01        <.0001

          Likelihood Ratio      1         2.29        0.1299
```

The analysis of variance table shows that the model fits since the likelihood ratio test for the three-variable interaction is nonsignificant. All of the two-variable interactions, however, are significant; this shows that there is mutual dependence among all three variables.

Output 22.4.4. Response Function Predicted Values

```
                              Bartlett’s Data
                    Model with No 3-Variable Interaction

                            The CATMOD Procedure

           Maximum Likelihood Predicted Values for Response Functions

                  ------Observed------    ------Predicted-----
     Function                 Standard                Standard
      Number      Function       Error    Function       Error    Residual
     ----------------------------------------------------------------------
        1         -0.29248    0.105806    -0.23565    0.098486    -0.05683
        2         -0.91152    0.129188    -0.94942    0.129948    0.037901
        3         -0.91152    0.129188    -0.94942    0.129948    0.037901
        4         -0.29248    0.105806    -0.23565    0.098486    -0.05683
        5         -0.66951    0.118872    -0.69362    0.120172    0.024113
        6         -0.45199    0.110921     -0.3897    0.102267    -0.06229
        7         -1.90835    0.192465    -1.73146    0.142969    -0.17688
```

The predicted values table (Output 22.4.4) displays observed and predicted values for the generalized logits. The predicted frequencies table (Output 22.4.5) displays observed and predicted cell frequencies, their standard errors, and residuals.

Output 22.4.5. Predicted Frequencies

```
                                  Bartlett's Data
                           Model with No 3-Variable Interaction

                      Maximum Likelihood Predicted Values for Frequencies

                                 -------Observed------    ------Predicted------
                                             Standard                 Standard
Length    Time    Status    Frequency       Error    Frequency       Error    Residual
--------------------------------------------------------------------------------------
1         1       1               156    11.43022     161.0961    11.07379    -5.09614
1         1       2                84    8.754999     78.90386    7.808613    5.096139
1         2       1                84    8.754999     78.90386    7.808613    5.096139
1         2       2               156    11.43022     161.0961    11.07379    -5.09614
2         1       1               107    9.750588     101.9039    8.924304    5.096139
2         1       2               133    10.70392     138.0961    10.33434    -5.09614
2         2       1                31     5.47713     36.09614    4.826315    -5.09614
2         2       2               209    12.78667     203.9039    12.21285     5.09614
```

Example 22.5. Log-Linear Model, Structural and Sampling Zeros

This example illustrates a log-linear model of independence, using data that contain structural zero frequencies as well as sampling (random) zero frequencies.

In a population of six squirrel monkeys, the joint distribution of genital display with respect to active or passive role was observed. The data are from Fienberg (1980, Table 8-2). Since a monkey cannot have both the active and passive roles in the same interaction, the diagonal cells of the table are structural zeros. See Agresti (2002) for more information on the quasi-independence model.

The DATA step replaces the structural zeros with missing values, and the MISSING=STRUCTURAL option is specified in the MODEL statement to remove these zeros from the analysis. The ZERO=SAMPLING option treats the off-diagonal zeros as sampling zeros. Also, the row for Monkey 't' is deleted since it contains all zeros; therefore, the cell frequencies predicted by a model of independence are also zero. In addition, the CONTRAST statement compares the behavior of the two monkeys labeled 'u' and 'v'. See the "Structural and Sampling Zeros with Raw Data" section on page 924 for information on how to perform this analysis when you have raw data. The following statements produce Output 22.5.1 through Output 22.5.8:

```
data Display;
   input Active $ Passive $ wt @@;
   if Active ne 't';
   if Active eq Passive then wt=.;
   datalines;
r r  0   r s  1   r t  5   r u  8   r v  9   r w  0
s r 29   s s  0   s t 14   s u 46   s v  4   s w  0
t r  0   t s  0   t t  0   t u  0   t v  0   t w  0
u r  2   u s  3   u t  1   u u  0   u v 38   u w  2
v r  0   v s  0   v t  0   v u  0   v v  0   v w  1
w r  9   w s 25   w t  4   w u  6   w v 13   w w  0
;
```

```
title 'Behavior of Squirrel Monkeys';
proc catmod data=Display;
   weight wt;
   model Active*Passive=_response_ /
         missing=structural zero=sampling
         freq pred=freq noparm oneway;
   loglin Active Passive;
   contrast 'Passive, U vs. V' Passive 0 0 0 1 -1;
   contrast 'Active,  U vs. V' Active  0 0 1 -1;
   title2 'Test Quasi-Independence for the Incomplete Table';
quit;
```

Output 22.5.1. Log-Linear Model Analysis with Zero Frequencies

```
                       Behavior of Squirrel Monkeys
              Test Quasi-Independence for the Incomplete Table

                          The CATMOD Procedure

                              Data Summary

     Response           Active*Passive     Response Levels   25
     Weight Variable    wt                 Populations        1
     Data Set           DISPLAY            Total Frequency  220
     Frequency Missing  0                  Observations      25
```

The results of the ONEWAY option are shown in Output 22.5.2. Monkey 't' does not show up as a value for the Active variable since that row was removed.

Output 22.5.2. Output from the ONEWAY option

```
                       Behavior of Squirrel Monkeys
              Test Quasi-Independence for the Incomplete Table

                          One-Way Frequencies

                     Variable    Value   Frequency
                     -----------------------------
                     Active      r              23
                                 s              93
                                 u              46
                                 v               1
                                 w              57

                     Passive     r              40
                                 s              29
                                 t              24
                                 u              60
                                 v              64
                                 w               3
```

Output 22.5.3. Profiles

```
                    Behavior of Squirrel Monkeys
            Test Quasi-Independence for the Incomplete Table

                          Population Profiles

                         Sample    Sample Size
                         ---------------------
                             1             220


                           Response Profiles

                     Response    Active    Passive
                     -----------------------------
                            1    r         s
                            2    r         t
                            3    r         u
                            4    r         v
                            5    r         w
                            6    s         r
                            7    s         t
                            8    s         u
                            9    s         v
                           10    s         w
                           11    u         r
                           12    u         s
                           13    u         t
                           14    u         v
                           15    u         w
                           16    v         r
                           17    v         s
                           18    v         t
                           19    v         u
                           20    v         w
                           21    w         r
                           22    w         s
                           23    w         t
                           24    w         u
                           25    w         v
```

Sampling zeros are displayed as 0 in Output 22.5.4. The Response Number corresponds to the value displayed in the Response Profiles in Output 22.5.3.

Output 22.5.4. Frequency of Response by Response Number

```
                     Behavior of Squirrel Monkeys
           Test Quasi-Independence for the Incomplete Table

                         Response Frequencies

                                Response Number
Sample        1        2        3        4        5        6        7        8
------------------------------------------------------------------------------
    1         1        5        8        9        0       29       14       46

                         Response Frequencies

                                Response Number
Sample        9       10       11       12       13       14       15       16
------------------------------------------------------------------------------
    1         4        0        2        3        1       38        2        0

                         Response Frequencies

                                Response Number
Sample       17       18       19       20       21       22       23       24
------------------------------------------------------------------------------
    1         0        0        0        1        9       25        4        6

                         Response Frequencies

                                  Response
                                   Number
                          Sample       25
                          ---------------
                              1        13
```

Output 22.5.5. Analysis of Variance Table

```
                     Behavior of Squirrel Monkeys
           Test Quasi-Independence for the Incomplete Table

               Maximum Likelihood Analysis of Variance

              Source               DF   Chi-Square    Pr > ChiSq
              --------------------------------------------------
              Active                4        56.58        <.0001
              Passive               5        47.94        <.0001

              Likelihood Ratio     15       135.17        <.0001
```

The analysis of variance table (Output 22.5.5) shows that the model of independence does not fit since the likelihood ratio test for the interaction is significant. In other words, active and passive behaviors of the squirrel monkeys are dependent behavior roles.

Output 22.5.6. Contrasts between Monkeys ‘u’ and ‘v’

```
                    Behavior of Squirrel Monkeys
          Test Quasi-Independence for the Incomplete Table

             Contrasts of Maximum Likelihood Estimates

         Contrast           DF    Chi-Square    Pr > ChiSq
         -------------------------------------------------
         Passive, U vs. V    1          1.31        0.2524
         Active,  U vs. V    1         14.87        0.0001
```

If the model fit these data, then the contrasts in Output 22.5.6 show that monkeys ‘u’ and ‘v’ appear to have similar passive behavior patterns but very different active behavior patterns.

Output 22.5.7. Response Function Predicted Values

```
                        Behavior of Squirrel Monkeys
              Test Quasi-Independence for the Incomplete Table

                            The CATMOD Procedure

        Maximum Likelihood Predicted Values for Response Functions

                 ------Observed------    ------Predicted-----
     Function                Standard                Standard
      Number     Function       Error    Function       Error    Residual
     --------------------------------------------------------------------
        1        -2.56495    1.037749    -0.97355    0.339019     -1.5914
        2        -0.95551    0.526235    -1.72504    0.345438    0.769529
        3        -0.48551    0.449359    -0.52751    0.309254    0.042007
        4        -0.36772    0.433629    -0.73927    0.249006    0.371543
        5               .           .    -3.56052    0.634104           .
        6        0.802346    0.333775    0.320589     0.26629    0.481758
        7        0.074108    0.385164    -0.29934    0.295634     0.37345
        8        1.263692    0.314105    0.898184    0.250857    0.365508
        9        -1.17865    0.571772    0.686431    0.173396    -1.86509
       10               .           .    -2.13482    0.608071           .
       11         -1.8718    0.759555     -0.2415    0.287218    -1.63031
       12        -1.46634    0.640513    -0.10994    0.303568     -1.3564
       13        -2.56495    1.037749    -0.86143    0.314794    -1.70352
       14        1.072637    0.321308    0.124346    0.204345     0.94829
       15         -1.8718    0.759555     -2.6969    0.617433      0.8251
       16               .           .    -4.14787    1.024508           .
       17               .           .    -4.01632    1.030062           .
       18               .           .    -4.76781    1.032457           .
       19               .           .    -3.57028    1.020794           .
       20        -2.56495    1.037749    -6.60328    1.161289    4.038332
       21        -0.36772    0.433629    -0.36584    0.202959    -0.00188
       22        0.653926     0.34194    -0.23429    0.232794    0.888212
       23        -1.17865    0.571772    -0.98577    0.239408    -0.19288
       24        -0.77319    0.493548    0.211754    0.185007    -0.98494
```

Output 22.5.8. Predicted Frequencies

```
                          Behavior of Squirrel Monkeys
                    Test Quasi-Independence for the Incomplete Table

                   Maximum Likelihood Predicted Values for Frequencies

                      -------Observed------    ------Predicted------
                                  Standard                 Standard
Active    Passive     Frequency      Error     Frequency      Error    Residual
-------------------------------------------------------------------------------
r         s                   1   0.997725      5.259508    1.36156   -4.25951
r         t                   5   2.210512      2.480726   0.691066   2.519274
r         u                   8   2.776525      8.215948   1.855146   -0.21595
r         v                   9   2.937996      6.648049    1.50932   2.351951
r         w                   0          0      0.395769   0.240268   -0.39577
s         r                  29   5.017696      19.18599   3.147915   9.814007
s         t                  14   3.620648      10.32172   2.169599   3.678284
s         u                  46   6.031734      34.18463   4.428706   11.81537
s         v                   4   1.981735      27.66096   3.722788    -23.661
s         w                   0          0        1.6467   0.952712    -1.6467
u         r                   2   1.407771       10.9364    2.12322    -8.9364
u         s                   3   1.720201      12.47407   2.554336   -9.47407
u         t                   1   0.997725      5.883583   1.380655   -4.88358
u         v                  38   5.606814       15.7673   2.684692    22.2327
u         w                   2   1.407771      0.938652   0.551645   1.061348
v         r                   0          0      0.219966   0.221779   -0.21997
v         s                   0          0      0.250893   0.253706   -0.25089
v         t                   0          0      0.118338   0.120314   -0.11834
v         u                   0          0      0.391924   0.393255   -0.39192
v         w                   1   0.997725      0.018879   0.021728   0.981121
w         r                   9   2.937996      9.657645   1.808656   -0.65765
w         s                  25   4.707344      11.01553   2.275019   13.98447
w         t                   4   1.981735      5.195638   1.184452   -1.19564
w         u                   6   2.415857       17.2075   2.772098   -11.2075
w         v                  13   3.497402      13.92369    2.24158   -0.92369
```

Output 22.5.7 displays the predicted response functions and Output 22.5.8 displays predicted cell frequencies (from the PRED=FREQ option), but since the model does not fit, these should be ignored. Note that, since the response function is the generalized logit with the twenty-fifth response as the baseline, the observed response functions for the sampling zeros are missing.

Structural and Sampling Zeros with Raw Data

The preceding PROC CATMOD step uses cell count data as input. Prior to invoking the CATMOD procedure, structural and sampling zeros are easily identified and manipulated in a single DATA step. For the situation where structural or sampling zeros (or both) may exist and the input data set is raw data, use the following steps:

1. Run PROC FREQ on the raw data. In the TABLES statement, list all dependent and independent variables separated by asterisks and use the SPARSE option and the OUT= option. This creates an output data set that contains all possible zero frequencies. Since the tabled output can be huge, you should also specify the NOPRINT option on the TABLES statement.
2. Use a DATA step to change the zero frequencies associated with either sampling zeros or structural zeros to missing.
3. Use the resulting data set as input to PROC CATMOD, specify the statement WEIGHT COUNT to use adjusted frequencies, and specify the ZERO= and MISSING= options to define your sampling and structural zeros.

For example, suppose the data set RawDisplay contains the raw data for the squirrel monkey data. The following statements show how to obtain the same analysis as shown previously:

```
proc freq data=RawDisplay;
   tables Active*Passive / sparse out=Combos noprint;
run;

data Combos2;
   set Combos;
   if Active ne ’t’;
   if Active eq Passive then count=.;
run;

proc catmod data=Combos2;
   weight count;
   model Active*Passive=_response_ /
         zero=sampling missing=structural
         freq pred=freq noparm noresponse;
   loglin Active Passive;
quit;
```

The first IF statement in the DATA step is needed only for this particular example; since observations for Monkey ‘t’ were deleted from the Display data set, they also need to be deleted from Combos2.

Example 22.6. Repeated Measures, 2 Response Levels, 3 Populations

In this multi-population repeated measures example, from Guthrie (1981), subjects from three groups have their responses (0 or 1) recorded in each of four trials. The analysis of the marginal probabilities is directed at assessing the main effects of the repeated measurement factor (Trial) and the independent variable (Group), as well as their interaction. Although the contingency table is incomplete (only thirteen of the sixteen possible responses are observed), this poses no problem in the computation of the marginal probabilities. The following statements produce Output 22.6.1 through Output 22.6.5:

```
data group;
   input a b c d Group wt @@;
   datalines;
1 1 1 1 2 2    0 0 0 0 2 2    0 0 1 0 1 2    0 0 1 0 2 2
0 0 0 1 1 4    0 0 0 1 2 1    0 0 0 1 3 3    1 0 0 1 2 1
0 0 1 1 1 1    0 0 1 1 2 2    0 0 1 1 3 5    0 1 0 0 1 4
0 1 0 0 2 1    0 1 0 1 2 1    0 1 0 1 3 2    0 1 1 0 3 1
1 0 0 0 1 3    1 0 0 0 2 1    0 1 1 1 2 1    0 1 1 1 3 2
1 0 1 0 1 1    1 0 1 1 2 1    1 0 1 1 3 2
;
```

```
title 'Multi-Population Repeated Measures';
proc catmod data=group;
   weight wt;
   response marginals;
   model a*b*c*d=Group _response_ Group*_response_
         / freq;
   repeated Trial 4;
   title2 'Saturated Model';
run;
```

Output 22.6.1. Analysis of Multiple-Population Repeated Measures

```
                         Multi-Population Repeated Measures
                                  Saturated Model

                                The CATMOD Procedure

                                    Data Summary

                Response           a*b*c*d     Response Levels  13
                Weight Variable    wt          Populations       3
                Data Set           GROUP       Total Frequency  45
                Frequency Missing  0           Observations     23


                                Population Profiles

                          Sample    Group    Sample Size
                          ------------------------------
                              1     1                 15
                              2     2                 15
                              3     3                 15
```

Output 22.6.2. Response Profiles

```
                         Multi-Population Repeated Measures
                                  Saturated Model

                                 Response Profiles

                           Response    a    b    c    d
                           ----------------------------
                               1       0    0    0    0
                               2       0    0    0    1
                               3       0    0    1    0
                               4       0    0    1    1
                               5       0    1    0    0
                               6       0    1    0    1
                               7       0    1    1    0
                               8       0    1    1    1
                               9       1    0    0    0
                              10       1    0    0    1
                              11       1    0    1    0
                              12       1    0    1    1
                              13       1    1    1    1
```

Output 22.6.3. Response Frequencies

```
                         Multi-Population Repeated Measures
                                  Saturated Model

                                Response Frequencies

                                       Response Number
Sample         1         2         3         4         5         6         7         8
-------------------------------------------------------------------------------------
    1          0         4         2         1         4         0         0         0
    2          2         1         2         2         1         1         0         1
    3          0         3         0         5         0         2         1         2

                               Response Frequencies

                                      Response Number
             Sample         9        10        11        12        13
             ---------------------------------------------------
                 1          3         0         1         0         0
                 2          1         1         0         1         2
                 3          0         0         0         2         0
```

Output 22.6.4. Analysis of Variance Table

```
                         Multi-Population Repeated Measures
                                  Saturated Model

                                Analysis of Variance

              Source               DF   Chi-Square    Pr > ChiSq
              --------------------------------------------------
              Intercept             1       354.88        <.0001
              Group                 2        24.79        <.0001
              Trial                 3        21.45        <.0001
              Group*Trial           6        18.71        0.0047

              Residual              0          .            .
```

Output 22.6.5. Parameter Estimates

```
                        Multi-Population Repeated Measures
                                 Saturated Model

                   Analysis of Weighted Least Squares Estimates

                                                Standard        Chi-
Effect                Parameter    Estimate        Error      Square    Pr > ChiSq
------------------------------------------------------------------------------
Intercept                     1      0.5833       0.0310      354.88        <.0001
Group                         2      0.1333       0.0335       15.88        <.0001
                              3     -0.0333       0.0551        0.37        0.5450
Trial                         4      0.1722       0.0557        9.57        0.0020
                              5      0.1056       0.0647        2.66        0.1028
                              6     -0.0722       0.0577        1.57        0.2107
Group*Trial                   7     -0.1556       0.0852        3.33        0.0679
                              8     -0.0556       0.0800        0.48        0.4877
                              9     -0.0889       0.0953        0.87        0.3511
                             10      0.0111       0.0866        0.02        0.8979
                             11      0.0889       0.0822        1.17        0.2793
                             12     -0.0111       0.0824        0.02        0.8927
```

The analysis of variance table in Output 22.6.4 shows that there is a significant interaction between the independent variable Group and the repeated measurement factor Trial. Thus, an intermediate model (not shown) is fit in which the effects Trial and Group* Trial are replaced by Trial(Group=1), Trial(Group=2), and Trial(Group=3). Of these three effects, only the last is significant, so it is retained in the final model. The following statements produce Output 22.6.6 and Output 22.6.7:

```
      model a*b*c*d=Group _response_(Group=3)
            / noprofile noparm design;
      title2 'Trial Nested within Group 3';
   quit;
```

Output 22.6.6. Final Model: Design Matrix

```
                       Multi-Population Repeated Measures
                          Trial Nested within Group 3

                             The CATMOD Procedure

                                 Data Summary

               Response           a*b*c*d     Response Levels  13
               Weight Variable    wt          Populations       3
               Data Set           GROUP       Total Frequency  45
               Frequency Missing  0           Observations     23


                     Response Functions and Design Matrix

         Function     Response                    Design Matrix
Sample    Number      Function       1       2       3       4       5       6
------------------------------------------------------------------------------
    1          1       0.73333       1       1       0       0       0       0
               2       0.73333       1       1       0       0       0       0
               3       0.73333       1       1       0       0       0       0
               4       0.66667       1       1       0       0       0       0

    2          1       0.66667       1       0       1       0       0       0
               2       0.66667       1       0       1       0       0       0
               3       0.46667       1       0       1       0       0       0
               4       0.40000       1       0       1       0       0       0

    3          1       0.86667       1      -1      -1       1       0       0
               2       0.66667       1      -1      -1       0       1       0
               3       0.33333       1      -1      -1       0       0       1
               4       0.06667       1      -1      -1      -1      -1      -1
```

Output 22.6.6 displays the design matrix resulting from retaining the nested effect.

Output 22.6.7. ANOVA Table

```
                       Multi-Population Repeated Measures
                          Trial Nested within Group 3

                            Analysis of Variance

             Source                   DF   Chi-Square    Pr > ChiSq
             ------------------------------------------------------
             Intercept                 1       386.94        <.0001
             Group                     2        25.42        <.0001
             Trial(Group=3)            3        75.07        <.0001

             Residual                  6         5.09        0.5319
```

The residual goodness-of-fit statistic tests the joint effect of Trial(Group=1) and Trial(Group=2). The analysis of variance table in Output 22.6.7 shows that the final model fits, that there is a significant Group effect, and that there is a significant Trial effect in Group 3.

Example 22.7. Repeated Measures, 4 Response Levels, 1 Population

This example illustrates a repeated measurement analysis in which there are more than two levels of response. In this study, from Grizzle, Starmer, and Koch (1969, p. 493), 7,477 women aged 30–39 are tested for vision in both right and left eyes. Since there are four response levels for each dependent variable, the RESPONSE statement computes three marginal probabilities for each dependent variable, resulting in six response functions for analysis. Since the model contains a repeated measurement factor (Side) with two levels (Right, Left), PROC CATMOD groups the functions into sets of three (=6/2). Therefore, the Side effect has three degrees of freedom (one for each marginal probability), and it is the appropriate test of marginal homogeneity. The following statements produce Output 22.7.1 through Output 22.7.6:

```
title 'Vision Symmetry';
data vision;
   input Right Left count @@;
   datalines;
1 1 1520    1 2  266    1 3  124    1 4  66
2 1  234    2 2 1512    2 3  432    2 4  78
3 1  117    3 2  362    3 3 1772    3 4 205
4 1   36    4 2   82    4 3  179    4 4 492
;


proc catmod data=vision;
   weight count;
   response marginals;
   model Right*Left=_response_ / freq design;
   repeated Side 2;
   title2 'Test of Marginal Homogeneity';
quit;
```

Output 22.7.1. Vision Study: Analysis of Marginal Homogeneity

```
                                 Vision Symmetry
                           Test of Marginal Homogeneity

                               The CATMOD Procedure

                                  Data Summary

             Response            Right*Left     Response Levels    16
             Weight Variable     count          Populations         1
             Data Set            VISION         Total Frequency  7477
             Frequency Missing   0              Observations       16


                               Population Profiles

                              Sample    Sample Size
                              ---------------------
                                  1           7477
```

Output 22.7.2. Response Profiles

```
                          Test of Marginal Homogeneity

                               Response Profiles

                          Response    Right    Left
                          -------------------------
                              1         1        1
                              2         1        2
                              3         1        3
                              4         1        4
                              5         2        1
                              6         2        2
                              7         2        3
                              8         2        4
                              9         3        1
                             10         3        2
                             11         3        3
                             12         3        4
                             13         4        1
                             14         4        2
                             15         4        3
                             16         4        4
```

Output 22.7.3. Response Frequencies

```
                          Test of Marginal Homogeneity

                             Response Frequencies

                                    Response Number
Sample        1         2         3         4         5         6         7         8
-------------------------------------------------------------------------------------
    1      1520       266       124        66       234      1512       432        78

                             Response Frequencies

                                    Response Number
Sample        9        10        11        12        13        14        15        16
-------------------------------------------------------------------------------------
    1       117       362      1772       205        36        82       179       492
```

Output 22.7.4. Design Matrix

```
                         Test of Marginal Homogeneity

                     Response Functions and Design Matrix

         Function     Response                   Design Matrix
Sample    Number      Function       1       2       3       4       5       6
------------------------------------------------------------------------------
    1         1        0.26428       1       0       0       1       0       0
              2        0.30173       0       1       0       0       1       0
              3        0.32847       0       0       1       0       0       1
              4        0.25505       1       0       0      -1       0       0
              5        0.29718       0       1       0       0      -1       0
              6        0.33529       0       0       1       0       0      -1
```

Output 22.7.5. ANOVA Table

```
                         Test of Marginal Homogeneity

                             Analysis of Variance

                 Source         DF   Chi-Square    Pr > ChiSq
                 --------------------------------------------
                 Intercept       3     78744.17        <.0001
                 Side            3        11.98        0.0075

                 Residual        0          .           .
```

Output 22.7.6. Parameter Estimates

```
                         Test of Marginal Homogeneity

                 Analysis of Weighted Least Squares Estimates

                                         Standard        Chi-
   Effect        Parameter    Estimate      Error      Square    Pr > ChiSq
   ------------------------------------------------------------------------
   Intercept             1      0.2597    0.00468     3073.03        <.0001
                         2      0.2995    0.00464     4160.17        <.0001
                         3      0.3319    0.00483     4725.25        <.0001
   Side                  4     0.00461    0.00194        5.65        0.0174
                         5     0.00227    0.00255        0.80        0.3726
                         6    -0.00341    0.00252        1.83        0.1757
```

The analysis of variance table in Output 22.7.5 shows that the Side effect is significant, so there is not marginal homogeneity between left-eye vision and right-eye vision. In other words, the distribution of the quality of right-eye vision differs significantly from the quality of left-eye vision in the same subjects. The test of the Side effect is equivalent to Bhapkar's test (Agresti 1990).

Example 22.8. Repeated Measures, Logistic Analysis of Growth Curve

The data, from a longitudinal study reported in Koch et al. (1977), are from patients in four populations (2 diagnostic groups × 2 treatments) who are measured at three times to assess their response (n=normal or a=abnormal) to treatment.

```
title 'Growth Curve Analysis';
data growth2;
   input Diagnosis $ Treatment $ week1 $ week2 $ week4 $ count @@;
   datalines;
mild std n n n 16    severe std n n n  2
mild std n n a 13    severe std n n a  2
mild std n a n  9    severe std n a n  8
mild std n a a  3    severe std n a a  9
mild std a n n 14    severe std a n n  9
mild std a n a  4    severe std a n a 15
mild std a a n 15    severe std a a n 27
mild std a a a  6    severe std a a a 28
mild new n n n 31    severe new n n n  7
mild new n n a  0    severe new n n a  2
mild new n a n  6    severe new n a n  5
mild new n a a  0    severe new n a a  2
mild new a n n 22    severe new a n n 31
mild new a n a  2    severe new a n a  5
mild new a a n  9    severe new a a n 32
mild new a a a  0    severe new a a a  6
;
```

The analysis is directed at assessing the effect of the repeated measurement factor, Time, as well as the independent variables, Diagnosis (mild or severe) and Treatment (std or new). The RESPONSE statement is used to compute the logits of the marginal probabilities. The times used in the design matrix (0, 1, 2) correspond to the logarithms (base 2) of the actual times (1, 2, 4). The following statements produce Output 22.8.1 through Output 22.8.7:

```
proc catmod data=growth2 order=data;
   title2 'Reduced Logistic Model';
   weight count;
   population Diagnosis Treatment;
   response logit;
   model week1*week2*week4=(1 0 0 0,  /* mild, std */
                            1 0 1 0,
                            1 0 2 0,

                            1 0 0 0,  /* mild, new */
                            1 0 0 1,
                            1 0 0 2,

                            0 1 0 0,  /* severe, std */
                            0 1 1 0,
                            0 1 2 0,
                                  0 1 0 0,  /* severe, new */
                                  0 1 0 1,
                                  0 1 0 2)
                    (1='Mild diagnosis, week 1',
                     2='Severe diagnosis, week 1',
                     3='Time effect for std trt',
                     4='Time effect for new trt')
                     / freq design;
      contrast 'Diagnosis effect, week 1' all_parms 1 -1 0 0;
      contrast 'Equal time effects' all_parms 0 0 1 -1;
   quit;
```

Output 22.8.1. Logistic Analysis of Growth Curve

```
                              Growth Curve Analysis
                              Reduced Logistic Model

                               The CATMOD Procedure

                                   Data Summary

          Response            week1*week2*week4     Response Levels    8
          Weight Variable     count                 Populations        4
          Data Set            GROWTH2               Total Frequency  340
          Frequency Missing   0                     Observations      29
```

Output 22.8.2. Population and Response Profiles

```
                              Growth Curve Analysis
                              Reduced Logistic Model

                               Population Profiles

                 Sample    Diagnosis    Treatment    Sample Size
                 -----------------------------------------------
                     1     mild         std                   80
                     2     mild         new                   70
                     3     severe       std                  100
                     4     severe       new                   90


                                Response Profiles

                       Response    week1    week2    week4
                       -----------------------------------
                           1       n        n        n
                           2       n        n        a
                           3       n        a        n
                           4       n        a        a
                           5       a        n        n
                           6       a        n        a
                           7       a        a        n
                           8       a        a        a
```

The samples and the response numbers are defined in Output 22.8.2, and Output 22.8.3 displays the frequency distribution of the response numbers within the samples. Output 22.8.4 displays the design matrix specified in the MODEL statement, and the observed logits of the marginal probabilities are displayed in the Response Function column.

Output 22.8.3. Response Frequencies

```
                             Growth Curve Analysis
                             Reduced Logistic Model

                              Response Frequencies

                                     Response Number
Sample          1        2        3        4        5        6        7        8
--------------------------------------------------------------------------------
     1         16       13        9        3       14        4       15        6
     2         31        0        6        0       22        2        9        0
     3          2        2        8        9        9       15       27       28
     4          7        2        5        2       31        5       32        6
```

Output 22.8.4. Design Matrix

```
                             Growth Curve Analysis
                             Reduced Logistic Model

                     Response Functions and Design Matrix

          Function      Response                  Design Matrix
     Sample    Number      Function          1        2        3        4
     ----------------------------------------------------------------------
         1         1       0.05001          1        0        0        0
                   2       0.35364          1        0        1        0
                   3       0.73089          1        0        2        0

         2         1       0.11441          1        0        0        0
                   2       1.29928          1        0        0        1
                   3       3.52636          1        0        0        2

         3         1      -1.32493          0        1        0        0
                   2      -0.94446          0        1        1        0
                   3      -0.16034          0        1        2        0

         4         1      -1.53148          0        1        0        0
                   2       0.00000          0        1        0        1
                   3       1.60944          0        1        0        2
```

Output 22.8.5. Analysis of Variance

```
                          Growth Curve Analysis
                          Reduced Logistic Model

                           Analysis of Variance

            Source                     DF   Chi-Square    Pr > ChiSq
            ----------------------------------------------------------
            Mild diagnosis, week 1      1         0.28        0.5955
            Severe diagnosis, week 1    1       100.48        <.0001
            Time effect for std trt     1        26.35        <.0001
            Time effect for new trt     1       125.09        <.0001

            Residual                    8         4.20        0.8387
```

The analysis of variance table (Output 22.8.5) shows that the data can be adequately modeled by two parameters that represent diagnosis effects at week 1 and two log-linear time effects (one for each treatment). Both of the time effects are significant.

Output 22.8.6. Parameter Estimates

```
                          Growth Curve Analysis
                          Reduced Logistic Model

               Analysis of Weighted Least Squares Estimates

                                          Standard       Chi-
       Effect    Parameter    Estimate       Error     Square    Pr > ChiSq
       --------------------------------------------------------------------
       Model             1     -0.0716      0.1348       0.28        0.5955
                         2     -1.3529      0.1350     100.48        <.0001
                         3      0.4944      0.0963      26.35        <.0001
                         4      1.4552      0.1301     125.09        <.0001
```

Output 22.8.7. Contrasts

```
                          Growth Curve Analysis
                          Reduced Logistic Model

                          Analysis of Contrasts

            Contrast                 DF    Chi-Square    Pr > ChiSq
            -------------------------------------------------------
            Diagnosis effect, week 1    1         77.02        <.0001
            Equal time effects          1         59.12        <.0001
```

The analysis of contrasts (Output 22.8.7) shows that the diagnosis effect at week 1 is highly significant. In Output 22.8.6, since the estimate of the logit for the severe diagnosis effect (parameter 2) is more negative than it is for the mild diagnosis effect (parameter 1), there is a smaller predicted probability of the first response (normal) for the severe diagnosis group. In other words, those subjects with a severe diagnosis have a significantly higher probability of abnormal response at week 1 than those subjects with a mild diagnosis.

The analysis of contrasts also shows that the time effect for the standard treatment is significantly different than the one for the new treatment. The table of parameter estimates (Output 22.8.6) shows that the time effect for the new treatment (parameter 4) is stronger than it is for the standard treatment (parameter 3).

Example 22.9. Repeated Measures, Two Repeated Measurement Factors

This example, from MacMillan et al. (1981), illustrates a repeated measurement analysis in which there are two repeated measurement factors. Two diagnostic procedures (standard and test) are performed on each subject, and the results of both are evaluated at each of two times as being positive or negative.

```
title 'Diagnostic Procedure Comparison';
data a;
   input std1 $ test1 $ std2 $ test2 $ wt @@;
   datalines;
neg neg neg neg 509  neg neg neg pos  4  neg neg pos neg  17
neg neg pos pos   3  neg pos neg neg 13  neg pos neg pos   8
neg pos pos pos   8  pos neg neg neg 14  pos neg neg pos   1
pos neg pos neg  17  pos neg pos pos  9  pos pos neg neg   7
pos pos neg pos   4  pos pos pos neg  9  pos pos pos pos 170
;
```

For the initial model, the response functions are marginal probabilities, and the repeated measurement factors are Time and Treatment. The model is a saturated one, containing effects for Time, Treatment, and Time*Treatment. The following statements produce Output 22.9.1 through Output 22.9.5:

```
proc catmod data=a;
   title2 'Marginal Symmetry, Saturated Model';
   weight wt;
   response marginals;
   model std1*test1*std2*test2=_response_ / freq design noparm;
   repeated Time 2, Treatment 2 / _response_=Time Treatment
            Time*Treatment;
run;
```

Output 22.9.1. Diagnosis Data: Two Repeated Measurement Factors

```
                        Diagnostic Procedure Comparison
                       Marginal Symmetry, Saturated Model

                              The CATMOD Procedure

                                 Data Summary

       Response            std1*test1*std2*test2     Response Levels   15
       Weight Variable     wt                        Populations        1
       Data Set            A                         Total Frequency  793
       Frequency Missing   0                         Observations      15


                              Population Profiles

                             Sample    Sample Size
                             ---------------------
                                 1             793
```

Output 22.9.2. Response Profiles

```
                        Diagnostic Procedure Comparison
                       Marginal Symmetry, Saturated Model

                               Response Profiles

                   Response    std1    test1    std2    test2
                   ------------------------------------------
                       1       neg     neg      neg     neg
                       2       neg     neg      neg     pos
                       3       neg     neg      pos     neg
                       4       neg     neg      pos     pos
                       5       neg     pos      neg     neg
                       6       neg     pos      neg     pos
                       7       neg     pos      pos     pos
                       8       pos     neg      neg     neg
                       9       pos     neg      neg     pos
                      10       pos     neg      pos     neg
                      11       pos     neg      pos     pos
                      12       pos     pos      neg     neg
                      13       pos     pos      neg     pos
                      14       pos     pos      pos     neg
                      15       pos     pos      pos     pos
```

Output 22.9.3. Response Frequencies

```
                      Diagnostic Procedure Comparison
                     Marginal Symmetry, Saturated Model

                            Response Frequencies

                                   Response Number
Sample        1         2         3         4         5         6         7         8
-------------------------------------------------------------------------------
    1       509         4        17         3        13         8         8        14

                           Response Frequencies

                                  Response Number
    Sample        9        10        11        12        13        14        15
    --------------------------------------------------------------------
        1         1        17         9         7         4         9       170
```

Output 22.9.4. Design Matrix

```
                      Diagnostic Procedure Comparison
                     Marginal Symmetry, Saturated Model

                   Response Functions and Design Matrix

               Function       Response                 Design Matrix
     Sample     Number        Function         1         2         3         4
     --------------------------------------------------------------------
         1           1         0.70870         1         1         1         1
                     2         0.72383         1         1        -1        -1
                     3         0.70618         1        -1         1        -1
                     4         0.73897         1        -1        -1         1
```

Output 22.9.5. ANOVA Table

```
                      Diagnostic Procedure Comparison
                     Marginal Symmetry, Saturated Model

                            Analysis of Variance

               Source             DF    Chi-Square    Pr > ChiSq
               ------------------------------------------------
               Intercept           1       2385.34        <.0001
               Time                1          0.85        0.3570
               Treatment           1          8.20        0.0042
               Time*Treatment      1          2.40        0.1215

               Residual            0           .             .
```

The analysis of variance table in Output 22.9.5 shows that there is no significant effect of Time, either by itself or in its interaction with Treatment. Thus, the second model includes only the Treatment effect. Again, the response functions are marginal probabilities, and the repeated measurement factors are Time and Treatment. A main effect model with respect to Treatment is fit. The following statements produce Output 22.9.6 through Output 22.9.10:

```
      title2 'Marginal Symmetry, Reduced Model';
      model std1*test1*std2*test2=_response_ / corrb design noprofile;
      repeated Time 2, Treatment 2 / _response_=Treatment;
   run;
```

Output 22.9.6. Diagnosis Data: Reduced Model

```
                         Diagnostic Procedure Comparison
                         Marginal Symmetry, Reduced Model

                               The CATMOD Procedure

                                  Data Summary

        Response           std1*test1*std2*test2     Response Levels   15
        Weight Variable    wt                        Populations        1
        Data Set           A                         Total Frequency  793
        Frequency Missing  0                         Observations      15
```

Output 22.9.7. Design Matrix

```
                         Diagnostic Procedure Comparison
                         Marginal Symmetry, Reduced Model

                       Response Functions and Design Matrix

                          Function      Response     Design Matrix
                Sample     Number       Function        1        2
                --------------------------------------------------
                    1          1         0.70870        1        1
                               2         0.72383        1       -1
                               3         0.70618        1        1
                               4         0.73897        1       -1
```

Output 22.9.8. ANOVA Table

```
                         Diagnostic Procedure Comparison
                         Marginal Symmetry, Reduced Model

                               Analysis of Variance

                   Source         DF   Chi-Square    Pr > ChiSq
                   --------------------------------------------
                   Intercept       1      2386.97        <.0001
                   Treatment       1         9.55        0.0020

                   Residual        2         3.51        0.1731
```

Output 22.9.9. Parameter Estimates

```
                        Diagnostic Procedure Comparison
                        Marginal Symmetry, Reduced Model

                  Analysis of Weighted Least Squares Estimates

                                             Standard        Chi-
Effect        Parameter     Estimate        Error        Square    Pr > ChiSq
-----------------------------------------------------------------------------
Intercept             1       0.7196       0.0147       2386.97        <.0001
Treatment             2      -0.0128      0.00416          9.55        0.0020
```

Output 22.9.10. Correlation Matrix

```
                        Diagnostic Procedure Comparison
                        Marginal Symmetry, Reduced Model

                 Correlation Matrix of the Parameter Estimates

                      Row             Col1             Col2
                      -----------------------------------
                        1          1.00000          0.04194
                        2          0.04194          1.00000
```

The analysis of variance table for the reduced model (Output 22.9.8) shows that the model fits (since the Residual is nonsignificant) and that the treatment effect is significant. The negative parameter estimate for Treatment in Output 22.9.9 shows that the first level of treatment (std) has a smaller probability of the first response level (neg) than the second level of treatment (test). In other words, the standard diagnostic procedure gives a significantly higher probability of a positive response than the test diagnostic procedure.

The next example illustrates a RESPONSE statement that, at each time, computes the sensitivity and specificity of the test diagnostic procedure with respect to the standard procedure. Since these are measures of the relative accuracy of the two diagnostic procedures, the repeated measurement factors in this case are labeled Time and Accuracy. Only fifteen of the sixteen possible responses are observed, so additional care must be taken in formulating the RESPONSE statement for computation of sensitivity and specificity.

The following statements produce Output 22.9.11 through Output 22.9.15:

```
   title2 'Sensitivity and Specificity Analysis, '
          'Main-Effects Model';
   model std1*test1*std2*test2=_response_ / covb design noprofile;
   repeated Time 2, Accuracy 2 / _response_=Time Accuracy;
   response exp  1 -1  0  0  0  0  0  0,
                 0  0  1 -1  0  0  0  0,
                 0  0  0  0  1 -1  0  0,
                 0  0  0  0  0  0  1 -1

             log 0 0 0 0   0 0 0   0 0 0 0   1 1 1 1,
                 0 0 0 0   0 0 0   1 1 1 1   1 1 1 1,
                 1 1 1 1   0 0 0   0 0 0 0   0 0 0 0,
                 1 1 1 1   1 1 1   0 0 0 0   0 0 0 0,
                 0 0 0 1   0 0 1   0 0 0 1   0 0 0 1,
                 0 0 1 1   0 0 1   0 0 1 1   0 0 1 1,
                 1 0 0 0   1 0 0   1 0 0 0   1 0 0 0,
                 1 1 0 0   1 1 0   1 1 0 0   1 1 0 0;
quit;
```

Output 22.9.11. Diagnosis Data: Sensitivity and Specificity Analysis

```
                 Diagnostic Procedure Comparison
     Sensitivity and Specificity Analysis, Main-Effects Model

                       The CATMOD Procedure

                          Data Summary

Response           std1*test1*std2*test2     Response Levels   15
Weight Variable    wt                        Populations        1
Data Set           A                         Total Frequency  793
Frequency Missing  0                         Observations      15
```

Output 22.9.12. Design Matrix

```
                 Diagnostic Procedure Comparison
     Sensitivity and Specificity Analysis, Main-Effects Model

              Response Functions and Design Matrix

             Function      Response         Design Matrix
   Sample     Number       Function        1        2        3
   ------------------------------------------------------------
       1          1         0.82251        1        1        1
                  2         0.94840        1        1       -1
                  3         0.81545        1       -1        1
                  4         0.96964        1       -1       -1
```

For the sensitivity and specificity analysis, the four response functions displayed next to the design matrix (Output 22.9.12) represent the following:

1. sensitivity, time 1
2. specificity, time 1
3. sensitivity, time 2
4. specificity, time 2

The sensitivities and specificities are for the test diagnostic procedure relative to the standard procedure.

Output 22.9.13. ANOVA Table

```
                         Diagnostic Procedure Comparison
             Sensitivity and Specificity Analysis, Main-Effects Model

                               Analysis of Variance

                   Source         DF   Chi-Square    Pr > ChiSq
                   --------------------------------------------
                   Intercept       1      6448.79        <.0001
                   Time            1         4.10        0.0428
                   Accuracy        1        38.81        <.0001

                   Residual        1         1.00        0.3178
```

The ANOVA table (Output 22.9.13) shows that an additive model fits, that there is a significant effect of time, and that the sensitivity is significantly different from the specificity.

Output 22.9.14. Parameter Estimates

```
                         Diagnostic Procedure Comparison
             Sensitivity and Specificity Analysis, Main-Effects Model

                   Analysis of Weighted Least Squares Estimates

                                             Standard        Chi-
    Effect        Parameter    Estimate        Error       Square    Pr > ChiSq
    ------------------------------------------------------------------------
    Intercept             1      0.8892       0.0111      6448.79        <.0001
    Time                  2    -0.00932      0.00460         4.10        0.0428
    Accuracy              3     -0.0702       0.0113        38.81        <.0001
```

Output 22.9.15. Covariance Matrix

```
                     Diagnostic Procedure Comparison
           Sensitivity and Specificity Analysis, Main-Effects Model

                 Covariance Matrix of the Parameter Estimates

              Row             Col1            Col2            Col3
              ---------------------------------------------------
                1       0.00012260      0.00000229      0.00010137
                2       0.00000229      0.00002116      -.00000587
                3       0.00010137      -.00000587      0.00012697
```

Output 22.9.14 shows that the predicted sensitivities and specificities are lower for time 1 (since parameter 2 is negative). It also shows that the sensitivity is significantly less than the specificity.

Example 22.10. Direct Input of Response Functions and Covariance Matrix

This example illustrates the ability of PROC CATMOD to operate on an existing vector of functions and the corresponding covariance matrix. The estimates under investigation are composite indices summarizing the responses to eighteen psychological questions pertaining to general well-being. These estimates are computed for domains corresponding to an age by sex cross-classification, and the covariance matrix is calculated via the method of balanced repeated replications. The analysis is directed at obtaining a description of the variation among these domain estimates. The data are from Koch and Stokes (1979).

```
data fbeing(type=est);
   input   b1-b5   _type_ $  _name_ $  b6-b10 #2;
   datalines;
 7.93726   7.92509   7.82815   7.73696   8.16791  parms    .
 7.24978   7.18991   7.35960   7.31937   7.55184
 0.00739   0.00019   0.00146  -0.00082   0.00076  cov      b1
 0.00189   0.00118   0.00140  -0.00140   0.00039
 0.00019   0.01172   0.00183   0.00029   0.00083  cov      b2
-0.00123  -0.00629  -0.00088  -0.00232   0.00034
 0.00146   0.00183   0.01050  -0.00173   0.00011  cov      b3
 0.00434  -0.00059  -0.00055   0.00023  -0.00013
-0.00082   0.00029  -0.00173   0.01335   0.00140  cov      b4
 0.00158   0.00212   0.00211   0.00066   0.00240
 0.00076   0.00083   0.00011   0.00140   0.01430  cov      b5
-0.00050  -0.00098   0.00239  -0.00010   0.00213
 0.00189  -0.00123   0.00434   0.00158  -0.00050  cov      b6
 0.01110   0.00101   0.00177  -0.00018  -0.00082
 0.00118  -0.00629  -0.00059   0.00212  -0.00098  cov      b7
 0.00101   0.02342   0.00144   0.00369   0.00253
 0.00140  -0.00088  -0.00055   0.00211   0.00239  cov      b8
 0.00177   0.00144   0.01060   0.00157   0.00226
-0.00140  -0.00232   0.00023   0.00066  -0.00010  cov      b9
-0.00018   0.00369   0.00157   0.02298   0.00918
 0.00039   0.00034  -0.00013   0.00240   0.00213  cov     b10
-0.00082   0.00253   0.00226   0.00918   0.01921
;
```

The following statements produce Output 22.10.1 through Output 22.10.3:

```
proc catmod data=fbeing;
   title 'Complex Sample Survey Analysis';
   response read b1-b10;
   factors sex $ 2, age $ 5 / _response_=sex age
                           profile=(male     '25-34',
                                    male     '35-44',
                                    male     '45-54',
                                    male     '55-64',
                                    male     '65-74',
                                    female   '25-34',
                                    female   '35-44',
                                    female   '45-54',
                                    female   '55-64',
                                    female   '65-74');
   model _f_=_response_
         / design title='Main Effects for Sex and Age';
run;
```

Output 22.10.1. Health Survey Data: Using Direct Input

```
                          Complex Sample Survey Analysis

                           Main Effects for Sex and Age

                              The CATMOD Procedure

                      Response Functions and Design Matrix

         Function     Response                     Design Matrix
Sample    Number      Function        1        2        3        4        5        6
-----------------------------------------------------------------------------------
    1          1       7.93726        1        1        1        0        0        0
               2       7.92509        1        1        0        1        0        0
               3       7.82815        1        1        0        0        1        0
               4       7.73696        1        1        0        0        0        1
               5       8.16791        1        1       -1       -1       -1       -1
               6       7.24978        1       -1        1        0        0        0
               7       7.18991        1       -1        0        1        0        0
               8       7.35960        1       -1        0        0        1        0
               9       7.31937        1       -1        0        0        0        1
              10       7.55184        1       -1       -1       -1       -1       -1
```

Output 22.10.2. ANOVA Table

```
                         Complex Sample Survey Analysis

                              Analysis of Variance

                   Source         DF    Chi-Square    Pr > ChiSq
                   --------------------------------------------
                   Intercept       1      28089.07        <.0001
                   sex             1         65.84        <.0001
                   age             4          9.21        0.0561

                   Residual        4          2.92        0.5713
```

Output 22.10.3. Parameter Estimates

```
                         Complex Sample Survey Analysis

                   Analysis of Weighted Least Squares Estimates

                                                Standard        Chi-
     Effect        Parameter     Estimate         Error       Square    Pr > ChiSq
     ----------------------------------------------------------------------------
     Intercept             1       7.6319        0.0455     28089.07        <.0001
     sex                   2       0.2900        0.0357        65.84        <.0001
     age                   3     -0.00780        0.0645         0.01        0.9037
                           4      -0.0465        0.0636         0.54        0.4642
                           5      -0.0343        0.0557         0.38        0.5387
                           6      -0.1098        0.0764         2.07        0.1506
```

The analysis of variance table (Output 22.10.2) shows that the additive model fits and that there is a significant effect of both sex and age. The following statements produce Output 22.10.4:

```
      contrast 'No Age Effect for Age<65' all_parms 0 0 1 0 0 -1,
                                          all_parms 0 0 0 1 0 -1,
                                          all_parms 0 0 0 0 1 -1;
   run;
```

Output 22.10.4. Age<65 Contrast

```
                     Complex Sample Survey Analysis

                       Main Effects for Sex and Age

                          The CATMOD Procedure

                         Analysis of Contrasts

            Contrast                   DF     Chi-Square    Pr > ChiSq
            ---------------------------------------------------------
            No Age Effect for Age<65    3           0.72        0.8678
```

The analysis of the contrast shows that there is no significant difference among the four age groups that are under age 65. Thus, the next model contains a binary age effect (less than 65 versus 65 and over). The following statements produce Output 22.10.5 through Output 22.10.7:

```
      model _f_=(1  1  1,
                 1  1  1,
                 1  1  1,
                 1  1  1,
                 1  1 -1,
                 1 -1  1,
                 1 -1  1,
                 1 -1  1,
                 1 -1  1,
                 1 -1 -1)
                          (1='Intercept' ,
                           2='Sex'       ,
                           3='Age (25-64 vs. 65-74)')
            / design title='Binary Age Effect (25-64 vs. 65-74)' ;
   run;
   quit;
```

Output 22.10.5. Design Matrix

```
                         Complex Sample Survey Analysis

                       Binary Age Effect (25-64 vs. 65-74)

                              The CATMOD Procedure

                      Response Functions and Design Matrix

                   Function      Response         Design Matrix
         Sample     Number       Function        1        2        3
         -----------------------------------------------------------
             1          1         7.93726        1        1        1
                        2         7.92509        1        1        1
                        3         7.82815        1        1        1
                        4         7.73696        1        1        1
                        5         8.16791        1        1       -1
                        6         7.24978        1       -1        1
                        7         7.18991        1       -1        1
                        8         7.35960        1       -1        1
                        9         7.31937        1       -1        1
                       10         7.55184        1       -1       -1
```

Output 22.10.6. ANOVA Table

```
                         Complex Sample Survey Analysis

                              Analysis of Variance

           Source                    DF   Chi-Square    Pr > ChiSq
           -------------------------------------------------------
           Intercept                  1     19087.16        <.0001
           Sex                        1        72.64        <.0001
           Age (25-64 vs. 65-74)      1         8.49        0.0036

           Residual                   7         3.64        0.8198
```

Output 22.10.7. Parameter Estimates

```
                         Complex Sample Survey Analysis

                  Analysis of Weighted Least Squares Estimates

                                           Standard        Chi-
     Effect     Parameter     Estimate        Error      Square    Pr > ChiSq
     ------------------------------------------------------------------------
     Model              1       7.7183       0.0559    19087.16        <.0001
                        2       0.2800       0.0329       72.64        <.0001
                        3      -0.1304       0.0448        8.49        0.0036
```

The analysis of variance table in Output 22.10.6 shows that the model fits (note that the goodness-of-fit statistic is the sum of the previous one (Output 22.10.2) plus the chi-square for the contrast matrix in Output 22.10.4). The age and sex effects are significant. Since the second parameter in the table of estimates is positive, males (the first level for the sex variable) have a higher predicted index of well-being than females. Since the third parameter estimate is negative, those younger than age 65 (the first level of age) have a lower predicted index of well-being than those 65 and older.

Example 22.11. Predicted Probabilities

Suppose you have collected marketing research data to examine the relationship between a prospect's likelihood of buying your product and their education and income. Specifically, the variables are as follows.

Variable	Levels	Interpretation
Education	high, low	prospect's education level
Income	high, low	prospect's income level
Purchase	yes, no	Did prospect purchase product?

The following statements first create a data set, loan, that contains the marketing research data, then they use the CATMOD procedure to fit a model, obtain the parameter estimates, and obtain the predicted probabilities of interest. These statements produce Output 22.11.1 through Output 22.11.5.

```
data loan;
   input Education $ Income $ Purchase $ wt;
   datalines;
high  high  yes   54
high  high  no    23
high  low   yes   41
high  low   no    12
low   high  yes   35
low   high  no    42
low   low   yes   19
low   low   no     8
;
ods output PredictedValues=Predicted (keep=Education Income PredFunction);

proc catmod data=loan order=data;
   weight wt;
   response marginals;
   model Purchase=Education Income / pred design;
run;
proc sort data=Predicted;
   by descending PredFunction;
run;

proc print data=Predicted;
run;
```

Notice that the preceding statements use the Output Delivery System (ODS) to output the parameter estimates instead of the OUT= option, though either can be used.

Output 22.11.1. Marketing Research Data: Obtaining Predicted Probabilities

```
                            The CATMOD Procedure

                                Data Summary

          Response            Purchase     Response Levels    2
          Weight Variable     wt           Populations        4
          Data Set            LOAN         Total Frequency  234
          Frequency Missing   0            Observations       8
```

Output 22.11.2. Profiles and Design Matrix

```
                            Population Profiles

              Sample    Education    Income    Sample Size
              --------------------------------------------
                 1      high         high               77
                 2      high         low                53
                 3      low          high               77
                 4      low          low                27


                             Response Profiles

                            Response    Purchase
                            --------------------
                                1       yes
                                2       no


                   Response Functions and Design Matrix

                          Response         Design Matrix
            Sample        Function         1        2        3
            ----------------------------------------------------
               1           0.70130         1        1        1
               2           0.77358         1        1       -1
               3           0.45455         1       -1        1
               4           0.70370         1       -1       -1
```

Output 22.11.3. ANOVA Table and Parameter Estimates

```
                            Analysis of Variance

                   Source         DF   Chi-Square    Pr > ChiSq
                   ------------------------------------------
                   Intercept       1       418.36        <.0001
                   Education       1         8.85        0.0029
                   Income          1         4.70        0.0302

                   Residual        1         1.84        0.1745


                  Analysis of Weighted Least Squares Estimates

                                         Standard        Chi-
         Parameter          Estimate       Error       Square    Pr > ChiSq
         ---------------------------------------------------------------
         Intercept            0.6481      0.0317       418.36        <.0001
         Education high       0.0924      0.0311         8.85        0.0029
         Income    high      -0.0675      0.0312         4.70        0.0302
```

Output 22.11.4. Predicted Values and Residuals

```
                          Predicted Values for Response Functions

                                          ------Observed------    ------Predicted-----
                               Function                Standard                Standard
Education      Income          Number      Function       Error     Function       Error     Residual
-------------------------------------------------------------------------------------------------------
high           high              1         0.701299    0.052158      0.67294    0.047794     0.028359
high           low               1         0.773585    0.057487     0.808034    0.051586     -0.03445
low            high              1         0.454545    0.056744      0.48811    0.051077     -0.03356
low            low               1         0.703704    0.087877     0.623204    0.064867     0.080499
```

Output 22.11.5. Predicted Probabilities Data Set

```
                                           Pred
                Obs    Education    Income    Function

                 1       high        low      0.808034
                 2       high        high      0.67294
                 3       low         low      0.623204
                 4       low         high      0.48811
```

You can use the predicted values (values of PredFunction in Output 22.11.5) as scores representing the likelihood that a randomly chosen subject from one of these populations will purchase the product. Notice that the Response Profiles in Output 22.11.2 show you that the first sorted level of Purchase is “yes,” indicating that the predicted probabilities are for Pr(Purchase=’yes’). For example, someone with high education and low income has an estimated probability of purchase of 0.808. As with any response function estimate given by PROC CATMOD, this estimate can be obtained by cross-multiplying the row from the design matrix corresponding to the sample (sample number 2 in this case) with the vector of parameter estimates $((1 * 0.6481) + (1 * 0.0924) + (-1 * (-0.0675)))$.

This ranking of scores can help in decision making (for example, with respect to allocation of advertising dollars, choice of advertising media, choice of print media, and so on).

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons, Inc.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Bishop, Y.M.M., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: The MIT Press.

Breslow, N. (1982), “Covariance Adjustment of Relative-Risk Estimates in Matched Studies,” *Biometrics*, 38, 661–672.

Christensen, R. (1990), *Log-Linear Models*, New York: Springer-Verlag.

Cox, D.R. and Snell, E.J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman and Hall.

Fienberg, S.E. (1980), *The Analysis of Cross-Classified Categorical Data*, Second Edition, Cambridge, MA: The MIT Press.

Forthofer, R.N. and Koch, G.G. (1973), “An Analysis of Compounded Functions of Categorical Data,” *Biometrics*, 29, 143–157.

Forthofer, R.N. and Lehnen R.G. (1981), *Public Program Analysis: A New Categorical Data Approach*, Belmont, CA: Wadsworth.

Freeman, D.H. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker Inc.

Grizzle, J.E., Starmer, C.F., and Koch, G.G. (1969), “Analysis of Categorical Data by Linear Models,” *Biometrics*, 25, 489–504.

Guthrie, D. (1981), “Analysis of Dichotomous Variables in Repeated Measures Experiments,” *Psychological Bulletin*, 90, 189–195.

Haberman, S.J. (1972), “Log-Linear Fit for Contingency Tables,” *Applied Statistics*, 21, 218–225.

Haslett, S. (1990), “Degrees of Freedom and Parameter Estimability in Hierarchical Models for Sparse Complete Contingency Tables,” *Computational Statistics and Data Analysis*, 9, 179–195.

Imrey, P.B., Koch, G.G., and Stokes, M.E. (1981), “Categorical Data Analysis: Some Reflections on the Log Linear Model and Logistic Regression. Part I: Historical and Methodological Overview,” *International Statistical Review*, 49, 265–283.

Koch, G.G., Landis, J.R., Freeman, J.L., Freeman, D.H., and Lehnen, R.G. (1977), “A General Methodology for the Analysis of Experiments with Repeated Measurement of Categorical Data,” *Biometrics*, 33, 133–158.

Koch, G.G. and Stokes, M.E. (1979), “Annotated Computer Applications of Weighted Least Squares Methods for Illustrative Analyses of Examples Involving Health Survey Data.” Technical Report prepared for the U.S. National Center for Health Statistics.

Landis, J.R., Stanish, W.M., Freeman, J.L., and Koch, G.G. (1976), “A Computer Program for the Generalized Chi-Square Analysis of Categorical Data Using Weighted Least Squares (GENCAT),” *Computer Programs in Biomedicine*, 6, 196–231.

MacMillan, J., Becker, C., Koch, G.G., Stokes, M., and Vandiviere, H.M. (1981), “An Application of Weighted Least Squares Methods to the Analysis of Measurement Process Components of Variability in an Observational Study,” *American Statistical Association Proceedings of Survey Research Methods*, 680–685.

Ries, P.N. and Smith, H. (1963), “The Use of Chi-Square for Preference Testing in Multidimensional Problems,” *Chemical Engineering Progress*, 59, 39–43.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Stanish, W.M. and Koch, G.G. (1984), “The Use of CATMOD for Repeated Measurement Analysis of Categorical Data,” *Proceedings of the Ninth Annual SAS Users Group International Conference*, 9, 761–770.

Stokes, M.E., Davis, C.S., and Koch, G.G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Wald, A. (1943), “Tests of Statistical Hypotheses Concerning General Parameters When the Number of Observations Is Large,” *Transactions of the American Mathematical Society*, 54, 426–482.

Chapter 23
The CLUSTER Procedure

Chapter Contents

Chapter 23
The CLUSTER Procedure

Overview

The CLUSTER procedure hierarchically clusters the observations in a SAS data set using one of eleven methods. The CLUSTER procedure finds hierarchical clusters of the observations in a SAS data set. The data can be coordinates or distances. If the data are coordinates, PROC CLUSTER computes (possibly squared) Euclidean distances. If you want to perform a cluster analysis on non-Euclidean distance data, it is possible to do so by using the DISTANCE procedure. This procedure can produce an appropriate distance data that can then be used set as input to PROC CLUSTER.

One situation where analyzing non-Euclidean distance data can be useful is when you have categorical data, where the distance data are calculated using an association measure. For more information, see Example 23.5 on page 1044.

The clustering methods available are average linkage, the centroid method, complete linkage, density linkage (including Wong's hybrid and kth-nearest-neighbor methods), maximum likelihood for mixtures of spherical multivariate normal distributions with equal variances but possibly unequal mixing proportions, the flexible-beta method, McQuitty's similarity analysis, the median method, single linkage, two-stage density linkage, and Ward's minimum-variance method.

All methods are based on the usual agglomerative hierarchical clustering procedure. Each observation begins in a cluster by itself. The two closest clusters are merged to form a new cluster that replaces the two old clusters. Merging of the two closest clusters is repeated until only one cluster is left. The various clustering methods differ in how the distance between two clusters is computed. Each method is described in the section "Clustering Methods" on page 975.

The CLUSTER procedure is not practical for very large data sets because, with most methods, the CPU time varies as the square or cube of the number of observations. The FASTCLUS procedure requires time proportional to the number of observations and can, therefore, be used with much larger data sets than PROC CLUSTER. If you want to cluster a very large data set hierarchically, you can use PROC FASTCLUS for a preliminary cluster analysis producing a large number of clusters and then use PROC CLUSTER to cluster the preliminary clusters hierarchically. This method is used to find clusters for the Fisher Iris data in Example 23.3, later in this chapter.

PROC CLUSTER displays a history of the clustering process, giving statistics useful for estimating the number of clusters in the population from which the data are sampled. PROC CLUSTER also creates an output data set that can be used by the TREE procedure to draw a tree diagram of the cluster hierarchy or to output the cluster membership at any desired level. For example, to obtain the six-cluster solution, you could first use PROC CLUSTER with the OUTTREE= option then use this output data set as the input data set to the TREE procedure. With PROC TREE, specify

NCLUSTERS=6 and the OUT= options to obtain the six-cluster solution and draw a tree diagram. For an example, see Example 76.1 on page 4758 in Chapter 76, "The TREE Procedure."

Before you perform a cluster analysis on coordinate data, it is necessary to consider scaling or transforming the variables since variables with large variances tend to have more effect on the resulting clusters than those with small variances. The ACECLUS procedure is useful for performing linear transformations of the variables. You can also use the PRINCOMP procedure with the STD option, although in some cases it tends to obscure clusters or magnify the effect of error in the data when all components are retained. The STD option in the CLUSTER procedure standardizes the variables to mean 0 and standard deviation 1. Standardization is not always appropriate. See Milligan and Cooper (1987) for a Monte Carlo study on various methods of variable standardization. You should remove outliers before using PROC PRINCOMP or before using PROC CLUSTER with the STD option unless you specify the TRIM= option.

Nonlinear transformations of the variables may change the number of population clusters and should, therefore, be approached with caution. For most applications, the variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required if raw data are used as input. Ordinal or ranked data are generally not appropriate.

Agglomerative hierarchical clustering is discussed in all standard references on cluster analysis, for example, Anderberg (1973), Sneath and Sokal (1973), Hartigan (1975), Everitt (1980), and Spath (1980). An especially good introduction is given by Massart and Kaufman (1983). Anyone considering doing a hierarchical cluster analysis should study the Monte Carlo results of Milligan (1980), Milligan and Cooper (1985), and Cooper and Milligan (1988). Other essential, though more advanced, references on hierarchical clustering include Hartigan (1977, pp. 60–68; 1981), Wong (1982), Wong and Schaack (1982), and Wong and Lane (1983). Refer to Blashfield and Aldenderfer (1978) for a discussion of the confusing terminology in hierarchical cluster analysis.

Getting Started

The following example demonstrates how you can use the CLUSTER procedure to compute hierarchical clusters of observations in a SAS data set.

Suppose you want to determine whether national figures for birth rates, death rates, and infant death rates can be used to determine certain types or categories of countries. You want to perform a cluster analysis to determine whether the observations can be formed into groups suggested by the data. Previous studies indicate that the clusters computed from this type of data can be elongated and elliptical. Thus, you need to perform some linear transformation on the raw data before the cluster analysis.

The following data[*] from Rouncefield (1995) are birth rates, death rates, and infant

[*] These data have been compiled from the United Nations Demographic Yearbook 1990 (United

death rates for 97 countries. The DATA step creates the SAS data set Poverty:

```
data Poverty;
   input Birth Death InfantDeath Country $20. @@;
   datalines;
24.7  5.7  30.8 Albania             12.5 11.9  14.4 Bulgaria
13.4 11.7  11.3 Czechoslovakia      12   12.4   7.6 Former_E._Germany
11.6 13.4  14.8 Hungary             14.3 10.2    16 Poland
13.6 10.7  26.9 Romania               14    9  20.2 Yugoslavia
17.7   10    23 USSR                15.2  9.5  13.1 Byelorussia_SSR
13.4 11.6    13 Ukrainian_SSR       20.7  8.4  25.7 Argentina
46.6   18   111 Bolivia             28.6  7.9    63 Brazil
23.4  5.8  17.1 Chile               27.4  6.1    40 Columbia
32.9  7.4    63 Ecuador             28.3  7.3    56 Guyana
34.8  6.6    42 Paraguay            32.9  8.3 109.9 Peru
  18  9.6  21.9 Uruguay             27.5  4.4  23.3 Venezuela
  29 23.2    43 Mexico                12 10.6   7.9 Belgium
13.2 10.1   5.8 Finland             12.4 11.9   7.5 Denmark
13.6  9.4   7.4 France              11.4 11.2   7.4 Germany
10.1  9.2    11 Greece              15.1  9.1   7.5 Ireland
 9.7  9.1   8.8 Italy               13.2  8.6   7.1 Netherlands
14.3 10.7   7.8 Norway              11.9  9.5  13.1 Portugal
10.7  8.2   8.1 Spain               14.5 11.1   5.6 Sweden
12.5  9.5   7.1 Switzerland         13.6 11.5   8.4 U.K.
14.9  7.4     8 Austria              9.9  6.7   4.5 Japan
14.5  7.3   7.2 Canada              16.7  8.1   9.1 U.S.A.
40.4 18.7 181.6 Afghanistan         28.4  3.8    16 Bahrain
42.5 11.5 108.1 Iran                42.6  7.8    69 Iraq
22.3  6.3   9.7 Israel              38.9  6.4    44 Jordan
26.8  2.2  15.6 Kuwait              31.7  8.7    48 Lebanon
45.6  7.8    40 Oman                42.1  7.6    71 Saudi_Arabia
29.2  8.4    76 Turkey              22.8  3.8    26 United_Arab_Emirates
42.2 15.5   119 Bangladesh          41.4 16.6   130 Cambodia
21.2  6.7    32 China               11.7  4.9   6.1 Hong_Kong
30.5 10.2    91 India               28.6  9.4    75 Indonesia
23.5 18.1    25 Korea               31.6  5.6    24 Malaysia
36.1  8.8    68 Mongolia            39.6 14.8   128 Nepal
30.3  8.1 107.7 Pakistan            33.2  7.7    45 Philippines
17.8  5.2   7.5 Singapore           21.3  6.2  19.4 Sri_Lanka
22.3  7.7    28 Thailand            31.8  9.5    64 Vietnam
35.5  8.3    74 Algeria             47.2 20.2   137 Angola
48.5 11.6    67 Botswana            46.1 14.6    73 Congo
38.8  9.5  49.4 Egypt               48.6 20.7   137 Ethiopia
39.4 16.8   103 Gabon               47.4 21.4   143 Gambia
44.4 13.1    90 Ghana                 47 11.3    72 Kenya
  44  9.4    82 Libya               48.3   25   130 Malawi
35.5  9.8    82 Morocco               45 18.5   141 Mozambique
  44 12.1   135 Namibia             48.5 15.6   105 Nigeria
48.2 23.4   154 Sierra_Leone        50.1 20.2   132 Somalia
32.1  9.9    72 South_Africa        44.6 15.8   108 Sudan
46.8 12.5   118 Swaziland           31.1  7.3    52 Tunisia
52.2 15.6   103 Uganda              50.5   14   106 Tanzania
45.6 14.2    83 Zaire               51.1 13.7    80 Zambia
41.7 10.3    66 Zimbabwe
```

The data set Poverty contains the character variable Country and the numeric vari-

Nations publications, Sales No. E/F.91.XII.1, copyright 1991, United Nations, New York) and are reproduced with the permission of the United Nations.

ables Birth, Death, and InfantDeath, which represent the birth rate per thousand, death rate per thousand, and infant death rate per thousand. The `$20.` in the INPUT statement specifies that the variable Country is a character variable with a length of 20. The double trailing at sign (@@) in the INPUT statement holds the input line for further iterations of the DATA step, specifying that observations are input from each line until all values are read.

Because the variables in the data set do not have equal variance, you must perform some form of scaling or transformation. One method is to standardize the variables to mean zero and variance one. However, when you suspect that the data contain elliptical clusters, you can use the ACECLUS procedure to transform the data such that the resulting within-cluster covariance matrix is spherical. The procedure obtains approximate estimates of the pooled within-cluster covariance matrix and then computes canonical variables to be used in subsequent analyses.

The following statements perform the ACECLUS transformation using the SAS data set Poverty. The OUT= option creates an output SAS data set called Ace to contain the canonical variable scores.

```
proc aceclus data=Poverty out=Ace p=.03 noprint;
   var Birth Death InfantDeath;
run;
```

The P= option specifies that approximately three percent of the pairs are included in the estimation of the within-cluster covariance matrix. The NOPRINT option suppresses the display of the output. The VAR statement specifies that the variables Birth, Death, and InfantDeath are used in computing the canonical variables.

The following statements invoke the CLUSTER procedure, using the SAS data set ACE created in the previous PROC ACECLUS run.

```
proc cluster data=Ace outtree=Tree method=ward
             ccc pseudo print=15;
   var can1 can2 can3 ;
   id Country;
run;
```

The OUTTREE= option creates an output SAS data set called Tree that can be used by the TREE procedure to draw a tree diagram. Ward's minimum-variance clustering method is specified by the METHOD= option. The CCC option displays the cubic clustering criterion, and the PSEUDO option displays pseudo F and t^2 statistics. Only the last 15 generations of the cluster history are displayed, as defined by the PRINT= option.

The VAR statement specifies that the canonical variables computed in the ACECLUS procedure are used in the cluster analysis. The ID statement specifies that the variable Country should be added to the Tree output data set.

The results of this analysis are displayed in the following figures.

PROC CLUSTER first displays the table of eigenvalues of the covariance matrix for the three canonical variables (Figure 23.1). The first two columns list each eigenvalue and the difference between the eigenvalue and its successor. The last two columns display the individual and cumulative proportion of variation associated with each eigenvalue.

```
                              The CLUSTER Procedure
                      Ward's Minimum Variance Cluster Analysis

                        Eigenvalues of the Covariance Matrix

                    Eigenvalue    Difference    Proportion    Cumulative

               1    64.5500051    54.7313223        0.8091        0.8091
               2     9.8186828     4.4038309        0.1231        0.9321
               3     5.4148519                      0.0679        1.0000


            Root-Mean-Square Total-Sample Standard Deviation = 5.156987
            Root-Mean-Square Distance Between Observations   = 12.63199
```

Figure 23.1. Table of Eigenvalues of the Covariance Matrix

As displayed in the last column, the first two canonical variables account for about 93% of the total variation. Figure 23.1 also displays the root mean square of the total sample standard deviation and the root mean square distance between observations.

Figure 23.2 displays the last 15 generations of the cluster history. First listed are the number of clusters and the names of the clusters joined. The observations are identified either by the ID value or by CLn, where n is the number of the cluster. Next, PROC CLUSTER displays the number of observations in the new cluster and the semipartial R^2. The latter value represents the decrease in the proportion of variance accounted for by joining the two clusters.

```
                                        The CLUSTER Procedure
                                 Ward's Minimum Variance Cluster Analysis

                         Root-Mean-Square Total-Sample Standard Deviation = 5.156987
                         Root-Mean-Square Distance Between Observations   = 12.63199


                                           Cluster History
                                                                                                       T
                                                                                                       i
 NCL    --------------Clusters Joined--------------      FREQ     SPRSQ     RSQ     ERSQ     CCC      PSF     PST2    e

  15    Oman                    CL37                        5    0.0039    .957     .933    6.03      132     12.1
  14    CL31                    CL22                       13    0.0040    .953     .928    5.81      131      9.7
  13    CL41                    CL17                       32    0.0041    .949     .922    5.70      131     13.1
  12    CL19                    CL21                       10    0.0045    .945     .916    5.65      132      6.4
  11    CL39                    CL15                        9    0.0052    .940     .909    5.60      134      6.3
  10    CL76                    CL27                        6    0.0075    .932     .900    5.25      133     18.1
   9    CL23                    CL11                       15    0.0130    .919     .890    4.20      125     12.4
   8    CL10                    Afghanistan                 7    0.0134    .906     .879    3.55      122      7.3
   7    CL9                     CL25                       17    0.0217    .884     .864    2.26      114     11.6
   6    CL8                     CL20                       14    0.0239    .860     .846    1.42      112     10.5
   5    CL14                    CL13                       45    0.0307    .829     .822    0.65      112     59.2
   4    CL16                    CL7                        28    0.0323    .797     .788    0.57      122     14.8
   3    CL12                    CL6                        24    0.0323    .765     .732    1.84      153     11.6
   2    CL3                     CL4                        52    0.1782    .587     .613    -.82      135     48.9
   1    CL5                     CL2                        97    0.5866    .000     .000    0.00        .      135
```

Figure 23.2. Cluster Generation History and R-Square Values

Next listed is the squared multiple correlation, R^2, which is the proportion of variance accounted for by the clusters. Figure 23.2 shows that, when the data are grouped into three clusters, the proportion of variance accounted for by the clusters (R^2) is about 77%. The approximate expected value of R^2 is given in the column labeled “ERSQ.”

The next three columns display the values of the cubic clustering criterion (CCC), pseudo F (PSF), and t^2 (PST2) statistics. These statistics are useful in determining the number of clusters in the data.

Values of the cubic clustering criterion greater than 2 or 3 indicate good clusters; values between 0 and 2 indicate potential clusters, but they should be considered with caution; large negative values can indicate outliers. In Figure 23.2, there is a local peak of the CCC when the number of clusters is 3. The CCC drops at 4 clusters and then steadily increases, levelling off at 11 clusters.

Another method of judging the number of clusters in a data set is to look at the pseudo F statistic (PSF). Relatively large values indicate a stopping point. Reading down the PSF column, you can see that this method indicates a possible stopping point at 11 clusters and another at 3 clusters.

A general rule for interpreting the values of the pseudo t^2 statistic is to move down the column until you find the first value markedly larger than the previous value and move back up the column by one cluster. Moving down the PST2 column, you can see possible clustering levels at 11 clusters, 6 clusters, 3 clusters, and 2 clusters.

The final column in Figure 23.2 lists ties for minimum distance; a blank value indicates the absence of a tie.

These statistics indicate that the data can be clustered into 11 clusters or 3 clusters.

The following statements examine the results of clustering the data into 3 clusters.

A graphical view of the clustering process can often be helpful in interpreting the clusters. The following statements use the TREE procedure to produce a tree diagram of the clusters:

```
goptions vsize=8in htext=1pct htitle=2.5pct;
axis1 order=(0 to 1 by 0.2);
proc tree data=Tree out=New nclusters=3
          graphics haxis=axis1 horizontal;
   height _rsq_;
   copy can1 can2 ;
   id country;
run;
```

The AXIS1 statement defines axis parameters that are used in the TREE procedure. The ORDER= option specifies the data values in the order in which they should appear on the axis.

The preceding statements use the SAS data set Tree as input. The OUT= option creates an output SAS data set named New to contain information on cluster membership. The NCLUSTERS= option specifies the number of clusters desired in the data set New.

The GRAPHICS option directs the procedure to use high resolution graphics. The HAXIS= option specifies AXIS1 to customize the appearance of the horizontal axis. Use this option only when the GRAPHICS option is in effect. The HORIZONTAL option orients the tree diagram horizontally. The HEIGHT statement specifies the variable _RSQ_ (R^2) as the height variable.

The COPY statement copies the canonical variables can1 and can2 (computed in the ACECLUS procedure) into the output SAS data set New. Thus, the SAS output data set New contains information for three clusters and the first two of the original canonical variables.

Figure 23.3 displays the tree diagram. The figure provides a graphical view of the information in Figure 23.2. As the number of branches grows to the left from the root, the R^2 approaches 1; the first three clusters (branches of the tree) account for over half of the variation (about 77%, from Figure 23.2). In other words, only three clusters are necessary to explain over three-fourths of the variation.

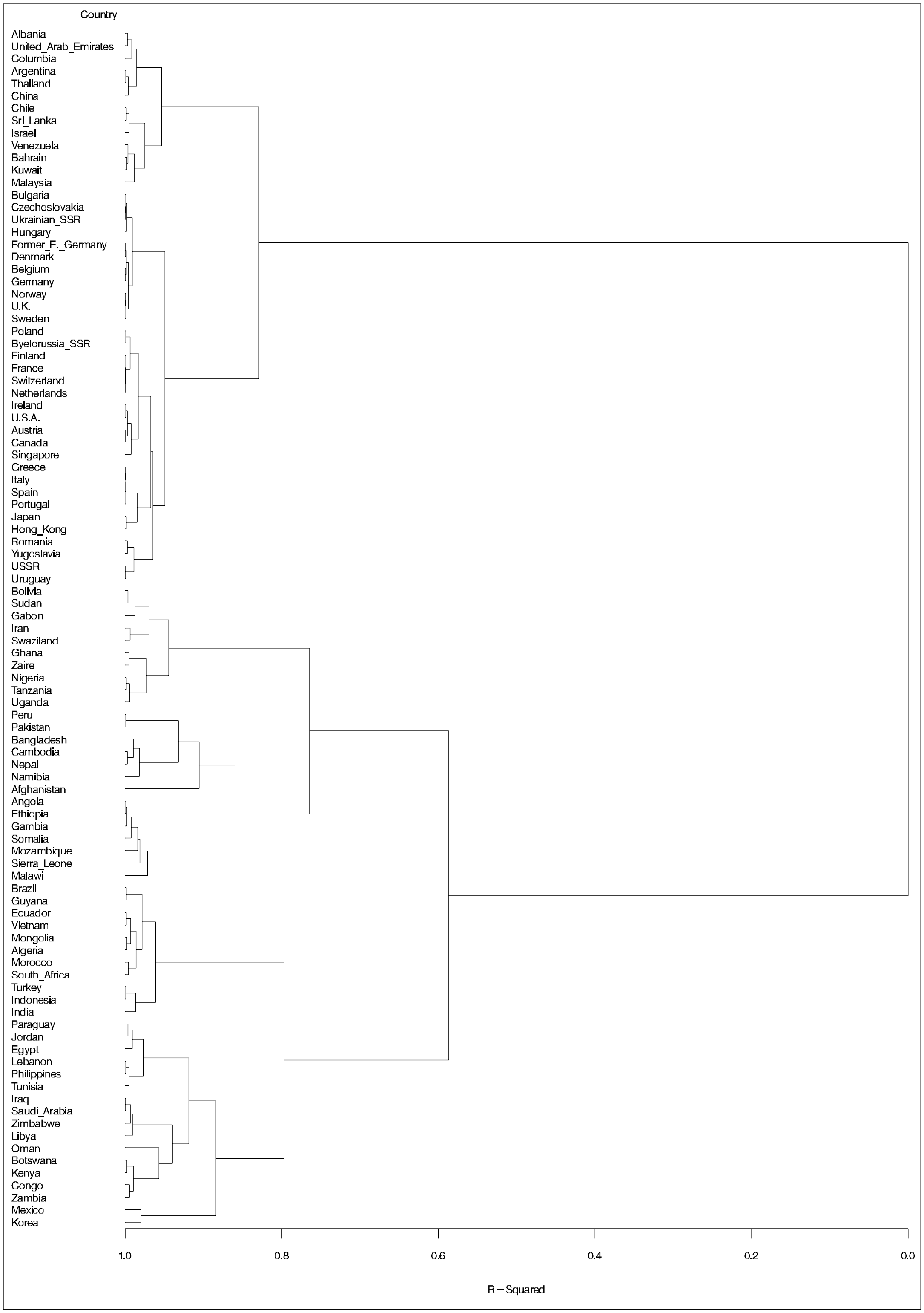


Figure 23.3. Tree Diagram of Clusters versus R-Square Values

The following statements invoke the GPLOT procedure on the SAS data set New.

```
legend1 frame cframe=ligr cborder=black
        position=center value=(justify=center);

axis1 label=(angle=90 rotate=0) minor=none order=(-10 to 20 by 5);
axis2 minor=none order=(-10 to 20 by 5);

proc gplot data=New ;
   plot can2*can1=cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

The PLOT statement requests a plot of the two canonical variables, using the value of the variable cluster as the identification variable.

Figure 23.4 displays the separation of the clusters when three clusters are calculated. The plotting symbol is the cluster number.

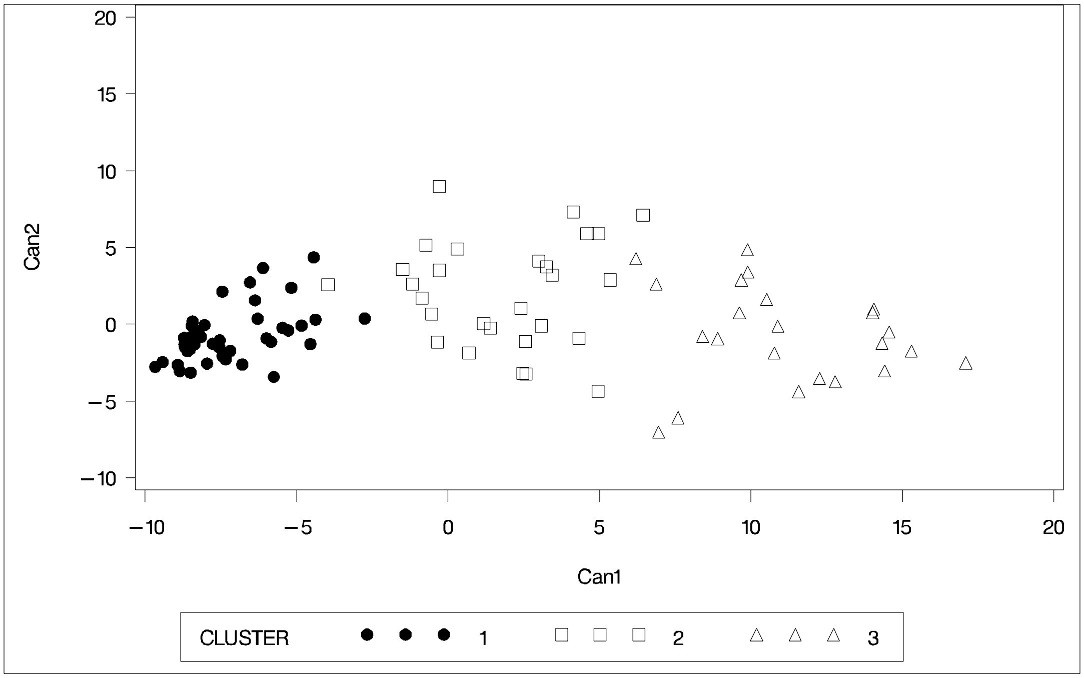


Figure 23.4. Plot of Canonical Variables and Cluster for Three Clusters

The statistics in Figure 23.2, the tree diagram in Figure 23.3, and the plot of the canonical variables assist in the determination of clusters in the data. There seems to be reasonable separation in the clusters. However, you must use this information, along with experience and knowledge of the field, to help in deciding the correct number of clusters.

Syntax

The following statements are available in the CLUSTER procedure.

PROC CLUSTER *METHOD = name < options >* **;**
  BY *variables* **;**
  COPY *variables* **;**
  FREQ *variable* **;**
  ID *variable* **;**
  RMSSTD *variable* **;**
  VAR *variables* **;**

Only the PROC CLUSTER statement is required, except that the FREQ statement is required when the RMSSTD statement is used; otherwise the FREQ statement is optional. Usually only the VAR statement and possibly the ID and COPY statements are needed in addition to the PROC CLUSTER statement. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC CLUSTER statement. The remaining statements are covered in alphabetical order.

PROC CLUSTER Statement

PROC CLUSTER *METHOD=name < options >* **;**

The PROC CLUSTER statement starts the CLUSTER procedure, identifies a clustering method, and optionally identifies details for clustering methods, data sets, data processing, and displayed output. The METHOD= specification determines the clustering method used by the procedure. Any one of the following 11 methods can be specified for *name*:

AVERAGE | AVE
: requests average linkage (group average, unweighted pair-group method using arithmetic averages, UPGMA). Distance data are squared unless you specify the NOSQUARE option.

CENTROID | CEN
: requests the centroid method (unweighted pair-group method using centroids, UPGMC, centroid sorting, weighted-group method). Distance data are squared unless you specify the NOSQUARE option.

COMPLETE | COM
: requests complete linkage (furthest neighbor, maximum method, diameter method, rank order typal analysis). To reduce distortion of clusters by outliers, the TRIM= option is recommended.

DENSITY | DEN
: requests density linkage, which is a class of clustering methods using nonparametric probability density estimation. You must also specify one of the K=, R=, or HYBRID options to indicate the type of density estimation to be

used. See also the MODE= and DIM= options in this section.

EML
requests maximum-likelihood hierarchical clustering for mixtures of spherical multivariate normal distributions with equal variances but possibly unequal mixing proportions. Use METHOD=EML only with coordinate data. See the PENALTY= option on page 971. The NONORM option does not affect the reported likelihood values but does affect other unrelated criteria. The EML method is much slower than the other methods in the CLUSTER procedure.

FLEXIBLE | FLE
requests the Lance-Williams flexible-beta method. See the BETA= option in this section.

MCQUITTY | MCQ
requests McQuitty's similarity analysis, which is weighted average linkage, weighted pair-group method using arithmetic averages (WPGMA).

MEDIAN | MED
requests Gower's median method, which is weighted pair-group method using centroids (WPGMC). Distance data are squared unless you specify the NOSQUARE option.

SINGLE | SIN
requests single linkage (nearest neighbor, minimum method, connectedness method, elementary linkage analysis, or dendritic method). To reduce chaining, you can use the TRIM= option with METHOD=SINGLE.

TWOSTAGE | TWO
requests two-stage density linkage. You must also specify the K=, R=, or HYBRID option to indicate the type of density estimation to be used. See also the MODE= and DIM= options in this section.

WARD | WAR
requests Ward's minimum-variance method (error sum of squares, trace W). Distance data are squared unless you specify the NOSQUARE option. To reduce distortion by outliers, the TRIM= option is recommended. See the NONORM option.

The following table summarizes the options in the PROC CLUSTER statement.

Tasks	Options
Specify input and output data sets	
specify input data set	DATA=
create output data set	OUTTREE=
Specify clustering methods	
specify clustering method	METHOD=
beta for flexible beta method	BETA=
minimum number of members for modal clusters	MODE=
penalty coefficient for maximum-likelihood	PENALTY=
Wong's hybrid clustering method	HYBRID
Control data processing prior to clustering	
suppress computation of eigenvalues	NOEIGEN
suppress normalizing of distances	NONORM
suppress squaring of distances	NOSQUARE
standardize variables	STANDARD
omit points with low probability densities	TRIM=
Control density estimation	
dimensionality for estimates	DIM=
number of neighbors for kth-nearest-neighbor	K=
radius of sphere of support for uniform-kernel	R=
Suppress checking for ties	NOTIE
Control display of the cluster history	
display cubic clustering criterion	CCC
suppress display of ID values	NOID
specify number of generations to display	PRINT=
display pseudo F and t^2 statistics	PSEUDO
display root-mean-square standard deviation	RMSSTD
display R^2 and semipartial R^2	RSQUARE
Control other aspects of output	
suppress display of all output	NOPRINT
display simple summary statistics	SIMPLE

The following list provides details on these options.

BETA=*n*

specifies the beta parameter for METHOD=FLEXIBLE. The value of n should be less than 1, usually between 0 and -1. By default, BETA=-0.25. Milligan (1987) suggests a somewhat smaller value, perhaps -0.5, for data with many outliers.

CCC

displays the cubic clustering criterion and approximate expected R^2 under the uniform null hypothesis (Sarle 1983). The statistics associated with the RSQUARE option, R^2 and semipartial R^2, are also displayed. The CCC option applies only to coordinate data. The CCC option is not appropriate with METHOD=SINGLE because of the method's tendency to chop off tails of distributions.

DATA=*SAS-data-set*

names the input data set containing observations to be clustered. By default, the procedure uses the most recently created SAS data set. If the data set is TYPE=DISTANCE, the data are interpreted as a distance matrix; the number of variables must equal the number of observations in the data set or in each BY group. The distances are assumed to be Euclidean, but the procedure accepts other types of distances or dissimilarities. If the data set is not TYPE=DISTANCE, the data are interpreted as coordinates in a Euclidean space, and Euclidean distances are computed. For more on TYPE=DISTANCE data sets, see Appendix A, "Special SAS Data Sets."

You cannot use a TYPE=CORR data set as input to PROC CLUSTER, since the procedure uses dissimilarity measures. Instead, you can use a DATA step or the IML procedure to extract the correlation matrix from a TYPE=CORR data set and transform the values to dissimilarities such as $1-r$ or $1-r^2$, where r is the correlation.

All methods produce the same results when used with coordinate data as when used with Euclidean distances computed from the coordinates. However, the DIM= option must be used with distance data if you specify METHOD=TWOSTAGE or METHOD=DENSITY or if you specify the TRIM= option.

Certain methods that are most naturally defined in terms of coordinates require *squared* Euclidean distances to be used in the combinatorial distance formulas (Lance and Williams 1967). For this reason, distance data are automatically squared when used with METHOD=AVERAGE, METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD. If you want the combinatorial formulas to be applied to the (unsquared) distances with these methods, use the NOSQUARE option.

DIM=*n*

specifies the dimensionality used when computing density estimates with the TRIM= option, METHOD=DENSITY, or METHOD=TWOSTAGE. The values of n must be greater than or equal to 1. The default is the number of variables if the data are coordinates; the default is 1 if the data are distances.

HYBRID

requests Wong's (1982) hybrid clustering method in which density estimates are computed from a preliminary cluster analysis using the k-means method. The DATA= data set must contain means, frequencies, and root-mean-square standard deviations of the preliminary clusters (see the FREQ and RMSSTD statements). To use HYBRID, you must use either a FREQ statement or a DATA= data set that contains a _FREQ_ variable, and you must also use either an RMSSTD statement or a DATA= data set that contains a _RMSSTD_ variable.

The MEAN= data set produced by the FASTCLUS procedure is suitable for input to the CLUSTER procedure for hybrid clustering. Since this data set contains _FREQ_ and _RMSSTD_ variables, you can use it as input and then omit the FREQ and RMSSTD statements.

You must specify either METHOD=DENSITY or METHOD=TWOSTAGE with the

HYBRID option. You cannot use this option in combination with the TRIM=, K=, or R= option.

K=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation (Silverman 1986, pp. 19–21 and 96–99). The number of neighbors (n) must be at least two but less than the number of observations. See the MODE= option, which follows.

If you request an analysis that requires density estimation (the TRIM= option, METHOD=DENSITY, or METHOD=TWOSTAGE), you must specify one of the K=, HYBRID, or R= options.

MODE=*n*

specifies that, when two clusters are joined, each must have at least n members for either cluster to be designated a modal cluster. If you specify MODE=1, each cluster must also have a maximum density greater than the fusion density for either cluster to be designated a modal cluster.

Use the MODE= option only with METHOD=DENSITY or METHOD=TWOSTAGE. With METHOD=TWOSTAGE, the MODE= option affects the number of modal clusters formed. With METHOD=DENSITY, the MODE= option does not affect the clustering process but does determine the number of modal clusters reported on the output and identified by the _MODE_ variable in the output data set.

If you specify the K= option, the default value of MODE= is the same as the value of K= because the use of kth-nearest-neighbor density estimation limits the resolution that can be obtained for clusters with fewer than k members. If you do not specify the K= option, the default is MODE=2.

If you specify MODE=0, the default value is used instead of 0.

If you specify a FREQ statement or if a _FREQ_ variable appears in the input data set, the MODE= value is compared with the number of actual observations in the clusters being joined, not with the sum of the frequencies in the clusters.

NOEIGEN

suppresses computation of eigenvalues for the cubic clustering criterion. Specifying the NOEIGEN option saves time if the number of variables is large, but it should be used only if the variables are nearly uncorrelated or if you are not interested in the cubic clustering criterion. If you specify the NOEIGEN option and the variables are highly correlated, the cubic clustering criterion may be very liberal. The NOEIGEN option applies only to coordinate data.

NOID

suppresses the display of ID values for the clusters joined at each generation of the cluster history.

NONORM

prevents the distances from being normalized to unit mean or unit root mean square with most methods. With METHOD=WARD, the NONORM option prevents the between-cluster sum of squares from being normalized by the total sum of squares to yield a squared semipartial correlation. The NONORM option does not affect the reported likelihood values with METHOD=EML, but it does affect other unrelated criteria, such as the _DIST_ variable.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

NOSQUARE

prevents input distances from being squared with METHOD=AVERAGE, METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD.

If you specify the NOSQUARE option with distance data, the data are assumed to be squared Euclidean distances for computing R-squared and related statistics defined in a Euclidean coordinate system.

If you specify the NOSQUARE option with coordinate data with METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD, then the combinatorial formula is applied to unsquared Euclidean distances. The resulting cluster distances do not have their usual Euclidean interpretation and are, therefore, labeled "False" in the output.

NOTIE

prevents PROC CLUSTER from checking for ties for minimum distance between clusters at each generation of the cluster history. If your data are measured with such sufficient precision that ties are unlikely, then you can specify the NOTIE option to reduce slightly the time and space required by the procedure. See the section "Ties" on page 987.

OUTTREE=*SAS-data-set*

creates an output data set that can be used by the TREE procedure to draw a tree diagram. You must give the data set a two-level name to save it. Refer to *SAS Language Reference: Concepts* for a discussion of permanent data sets. If you omit the OUTTREE= option, the data set is named using the DATAn convention and is not permanently saved. If you do not want to create an output data set, use OUTTREE=_NULL_.

PENALTY=*p*

specifies the penalty coefficient used with METHOD=EML. See the section "Clustering Methods" on page 975. Values for p must be greater than zero. By default, PENALTY=2.

PRINT=*n* **|** **P=***n*

specifies the number of generations of the cluster history to display. The P= option displays the latest n generations; for example, P=5 displays the cluster history from 1 cluster through 5 clusters. The value of P= must be a nonnegative integer. The default is to display all generations. Specify PRINT=0 to suppress the cluster history.

PSEUDO

displays pseudo F and t^2 statistics. This option is effective only when the data are coordinates or when METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD. See the section "Miscellaneous Formulas" on page 984. The PSEUDO option is not appropriate with METHOD=SINGLE because of the method's tendency to chop off tails of distributions.

R=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation (Silverman 1986, pp. 11–13 and 75–94). The value of R= must be greater than zero.

If you request an analysis that requires density estimation (the TRIM= option, METHOD=DENSITY, or METHOD=TWOSTAGE), you must specify one of the K=, HYBRID, or R= options.

RMSSTD

displays the root-mean-square standard deviation of each cluster. This option is effective only when the data are coordinates or when METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD. See the section "Miscellaneous Formulas" on page 984.

RSQUARE | RSQ

displays the R^2 and semipartial R^2. This option is effective only when the data are coordinates or when METHOD=AVERAGE or METHOD=CENTROID. The R^2 and semipartial R^2 statistics are always displayed with METHOD=WARD. See the section "Miscellaneous Formulas" on page 984.

SIMPLE | S

displays means, standard deviations, skewness, kurtosis, and a coefficient of bimodality. The SIMPLE option applies only to coordinate data. See the section "Miscellaneous Formulas" on page 984.

STANDARD | STD

standardizes the variables to mean 0 and standard deviation 1. The STANDARD option applies only to coordinate data.

TRIM=*p*

omits points with low estimated probability densities from the analysis. Valid values for the TRIM= option are $0 \leq p < 100$. If $p < 1$, then p is the proportion of observations omitted. If $p \geq 1$, then p is interpreted as a percentage. A specification of TRIM=10, which trims 10 percent of the points, is a reasonable value for many data sets. Densities are estimated by the kth-nearest-neighbor or uniform-kernel methods. Trimmed points are indicated by a negative value of the _FREQ_ variable in the OUTTREE= data set.

You must use either the K= or R= option when you use TRIM=. You cannot use the HYBRID option in combination with TRIM=, so you may want to use the DIM= option instead. If you specify the STANDARD option in combination with TRIM=, the variables are standardized both before and after trimming.

The TRIM= option is useful for removing outliers and reducing chaining. Trimming is highly recommended with METHOD=WARD or METHOD=COMPLETE because clusters from these methods can be severely distorted by outliers. Trimming is also valuable with METHOD=SINGLE since single linkage is the method most susceptible to chaining. Most other methods also benefit from trimming. However, trimming is unnecessary with METHOD=TWOSTAGE or METHOD=DENSITY when kth-nearest-neighbor density estimation is used.

Use of the TRIM= option may spuriously inflate the cubic clustering criterion and the pseudo F and t^2 statistics. Trimming only outliers improves the accuracy of the statistics, but trimming saddle regions between clusters yields excessively large values.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CLUSTER to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CLUSTER procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

COPY Statement

COPY *variables* **;**

The variables in the COPY statement are copied from the input data set to the OUTTREE= data set. Observations in the OUTTREE= data set that represent clusters of more than one observation from the input data set have missing values for the COPY variables.

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC CLUSTER then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value.

If you omit the FREQ statement but the DATA= data set contains a variable called _FREQ_, then frequencies are obtained from the _FREQ_ variable. If neither a FREQ statement nor a _FREQ_ variable is present, each observation is assumed to have a frequency of one.

If each observation in the DATA= data set represents a cluster (for example, clusters formed by PROC FASTCLUS), the variable specified in the FREQ statement should give the number of original observations in each cluster.

If you specify the RMSSTD statement, a FREQ statement is required. A FREQ statement or _FREQ_ variable is required when you specify the HYBRID option.

With most clustering methods, the same clusters are obtained from a data set with a FREQ variable as from a similar data set without a FREQ variable, if each observation is repeated as many times as the value of the FREQ variable in the first data set. The FLEXIBLE method can yield different results due to the nature of the combinatorial formula. The DENSITY and TWOSTAGE methods are also exceptions because two identical observations can be absorbed one at a time by a cluster with a higher density. If you are using a FREQ statement with either the DENSITY or TWOSTAGE method, see the MODE=option on page 970.

ID Statement

ID *variable* **;**

The values of the ID variable identify observations in the displayed cluster history and in the OUTTREE= data set. If the ID statement is omitted, each observation is denoted by OBn, where n is the observation number.

RMSSTD Statement

RMSSTD *variable* **;**

If the coordinates in the DATA= data set represent cluster means (for example, formed by the FASTCLUS procedure), you can obtain accurate statistics in the cluster histories for METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD if the data set contains

- a variable giving the number of original observations in each cluster (see the discussion of the FREQ statement earlier in this chapter)
- a variable giving the root-mean-square standard deviation of each cluster

Specify the name of the variable containing root-mean-square standard deviations in the RMSSTD statement. If you specify the RMSSTD statement, you must also specify a FREQ statement.

If you omit the RMSSTD statement but the DATA= data set contains a variable called _RMSSTD_, then root-mean-square standard deviations are obtained from the _RMSSTD_ variable.

An RMSSTD statement or _RMSSTD_ variable is required when you specify the HYBRID option.

A data set created by FASTCLUS using the MEAN= option contains _FREQ_ and _RMSSTD_ variables, so you do not have to use FREQ and RMSSTD statements when using such a data set as input to the CLUSTER procedure.

VAR Statement

VAR *variables* **;**

The VAR statement lists numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not listed in other statements are used.

Details

Clustering Methods

The following notation is used, with lowercase symbols generally pertaining to observations and uppercase symbols pertaining to clusters:

n	number of observations
v	number of variables if data are coordinates
G	number of clusters at any given level of the hierarchy
x_i or $\mathbf{x}_i$	ith observation (row vector if coordinate data)
C_K	Kth cluster, subset of $\{1, 2, \ldots, n\}$
N_K	number of observations in C_K
$\bar{\mathbf{x}}$	sample mean vector
$\bar{\mathbf{x}}_K$	mean vector for cluster C_K
$\|\mathbf{x}\|$	Euclidean length of the vector $\mathbf{x}$, that is, the square root of the sum of the squares of the elements of $\mathbf{x}$
T	$\sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2$
W_K	$\sum_{i \in C_k} \|\mathbf{x}_i - \bar{\mathbf{x}}_K\|^2$
P_G	$\sum W_J$, where summation is over the G clusters at the Gth level of the hierarchy
B_{KL}	$W_M - W_K - W_L$ if $C_M = C_K \cup C_L$

$d(\mathbf{x},\mathbf{y})$	any distance or dissimilarity measure between observations or vectors $\mathbf{x}$ and $\mathbf{y}$
D_{KL}	any distance or dissimilarity measure between clusters C_K and C_L

The distance between two clusters can be defined either directly or combinatorially (Lance and Williams 1967), that is, by an equation for updating a distance matrix when two clusters are joined. In all of the following combinatorial formulas, it is assumed that clusters C_K and C_L are merged to form C_M, and the formula gives the distance between the new cluster C_M and any other cluster C_J.

For an introduction to most of the methods used in the CLUSTER procedure, refer to Massart and Kaufman (1983).

Average Linkage

The following method is obtained by specifying METHOD=AVERAGE. The distance between two clusters is defined by

$$D_{KL} = \frac{1}{N_K N_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

If $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$, then

$$D_{KL} = \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2 + \frac{W_K}{N_K} + \frac{W_L}{N_L}$$

The combinatorial formula is

$$D_{JM} = \frac{N_K D_{JK} + N_L D_{JL}}{N_M}$$

In average linkage the distance between two clusters is the average distance between pairs of observations, one in each cluster. Average linkage tends to join clusters with small variances, and it is slightly biased toward producing clusters with the same variance.

Average linkage was originated by Sokal and Michener (1958).

Centroid Method

The following method is obtained by specifying METHOD=CENTROID. The distance between two clusters is defined by

$$D_{KL} = \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2$$

If $d(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{N_K D_{JK} + N_L D_{JL}}{N_M} - \frac{N_K N_L D_{KL}}{N_M^2}$$

In the centroid method, the distance between two clusters is defined as the (squared) Euclidean distance between their centroids or means. The centroid method is more robust to outliers than most other hierarchical methods but in other respects may not perform as well as Ward's method or average linkage (Milligan 1980).

The centroid method was originated by Sokal and Michener (1958).

Complete Linkage

The following method is obtained by specifying METHOD=COMPLETE. The distance between two clusters is defined by

$$D_{KL} = \max_{i \in C_K} \max_{j \in C_L} d(x_i, x_j)$$

The combinatorial formula is

$$D_{JM} = \max(D_{JK}, D_{JL})$$

In complete linkage, the distance between two clusters is the maximum distance between an observation in one cluster and an observation in the other cluster. Complete linkage is strongly biased toward producing clusters with roughly equal diameters, and it can be severely distorted by moderate outliers (Milligan 1980).

Complete linkage was originated by Sorensen (1948).

Density Linkage

The phrase *density linkage* is used here to refer to a class of clustering methods using nonparametric probability density estimates (for example, Hartigan 1975, pp. 205–212; Wong 1982; Wong and Lane 1983). Density linkage consists of two steps:

1. A new dissimilarity measure, d^*, based on density estimates and adjacencies is computed. If x_i and x_j are adjacent (the definition of *adjacency* depends on the method of density estimation), then $d^*(x_i, x_j)$ is the reciprocal of an estimate of the density midway between x_i and x_j; otherwise, $d^*(x_i, x_j)$ is infinite.
2. A single linkage cluster analysis is performed using d^*.

The CLUSTER procedure supports three types of density linkage: the kth-nearest-neighbor method, the uniform kernel method, and Wong's hybrid method. These are obtained by using METHOD=DENSITY and the K=, R=, and HYBRID options, respectively.

kth-Nearest Neighbor Method

The kth-nearest-neighbor method (Wong and Lane 1983) uses kth-nearest neighbor density estimates. Let $r_k(x)$ be the distance from point x to the kth-nearest observation, where k is the value specified for the K= option. Consider a closed sphere centered at x with radius $r_k(x)$. The estimated density at x, $f(x)$, is the proportion of observations within the sphere divided by the volume of the sphere. The new dissimilarity measure is computed as

$$d^*(x_i, x_j) = \begin{cases} \frac{1}{2}\left(\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right) & \text{if } d(x_i, x_j) \le \max(r_k(x_i), r_k(x_j)) \\ \infty & \text{otherwise} \end{cases}$$

Wong and Lane (1983) show that kth-nearest-neighbor density linkage is strongly set consistent for high-density (density-contour) clusters if k is chosen such that $k/n \to 0$ and $k/\ln(n) \to \infty$ as $n \to \infty$. Wong and Schaack (1982) discuss methods for estimating the number of population clusters using kth-nearest-neighbor clustering.

Uniform-Kernel Method

The uniform-kernel method uses uniform-kernel density estimates. Let r be the value specified for the R= option. Consider a closed sphere centered at point x with radius r. The estimated density at x, $f(x)$, is the proportion of observations within the sphere divided by the volume of the sphere. The new dissimilarity measure is computed as

$$d^*(x_i, x_j) = \begin{cases} \frac{1}{2}\left(\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right) & \text{if } d(x_i, x_j) \le r \\ \infty & \text{otherwise} \end{cases}$$

Wong's Hybrid Method

Wong's (1982) hybrid clustering method uses density estimates based on a preliminary cluster analysis by the k-means method. The preliminary clustering can be done by the FASTCLUS procedure, using the MEAN= option to create a data set containing cluster means, frequencies, and root-mean-square standard deviations. This data set is used as input to the CLUSTER procedure, and the HYBRID option is specified with METHOD=DENSITY to request the hybrid analysis. The hybrid method is appropriate for very large data sets but should not be used with small data sets, say fewer than 100 observations in the original data. The term *preliminary cluster* refers to an observation in the DATA= data set.

For preliminary cluster C_K, N_K and W_K are obtained from the input data set, as are the cluster means or the distances between the cluster means. Preliminary clusters C_K and C_L are considered adjacent if the midpoint between $\bar{x}_K$ and $\bar{\mathbf{x}}_L$ is closer to either $\bar{\mathbf{x}}_K$ or $\bar{\mathbf{x}}_L$ than to any other preliminary cluster mean or, equivalently, if $d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L) < d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_M) + d^2(\bar{\mathbf{x}}_L, \bar{\mathbf{x}}_M)$ for all other preliminary clusters C_M, $M \ne K$ or L. The new dissimilarity measure is computed as

$$d^*(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L) = \begin{cases} \frac{\left(W_K + W_L + \frac{1}{4}(N_K + N_L) d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L)\right)^{\frac{v}{2}}}{(N_K + N_L)^{1+\frac{v}{2}}} & \text{if } C_K \text{ and } C_L \text{ are adjacent} \\ \infty & \text{otherwise} \end{cases}$$

Using the K= and R= Options

The values of the K= and R= options are called *smoothing parameters*. Small values of K= or R= produce jagged density estimates and, as a consequence, many modes. Large values of K= or R= produce smoother density estimates and fewer modes. In the hybrid method, the smoothing parameter is the number of clusters in the preliminary cluster analysis. The number of modes in the final analysis tends to increase as the number of clusters in the preliminary analysis increases. Wong (1982) suggests using $n^{0.3}$ preliminary clusters, where n is the number of observations in the original data set. There is no general rule-of-thumb for selecting K= values. For all types of density linkage, you should repeat the analysis with several different values of the smoothing parameter (Wong and Schaack 1982).

There is no simple answer to the question of which smoothing parameter to use (Silverman 1986, pp. 43–61, 84–88, and 98–99). It is usually necessary to try several different smoothing parameters. A reasonable first guess for the R= option in many coordinate data sets is given by

$$\left[\frac{2^{v+2}(v+2)\Gamma(\frac{v}{2}+1)}{nv^2}\right]^{\frac{1}{v+4}} \sqrt{\sum_{l=1}^{v} s_l^2}$$

where s_l^2 is the standard deviation of the lth variable. The estimate for R= can be computed in a DATA step using the GAMMA function for Γ. This formula is derived under the assumption that the data are sampled from a multivariate normal distribution and tends, therefore, to be too large (oversmooth) if the true distribution is multimodal. Robust estimates of the standard deviations may be preferable if there are outliers. If the data are distances, the factor $\sum s_l^2$ can be replaced by an average (mean, trimmed mean, median, root-mean-square, and so on) distance divided by $\sqrt{2}$. To prevent outliers from appearing as separate clusters, you can also specify K=2, or more generally K=m, $m \geq 2$, which in most cases forces clusters to have at least m members.

If the variables all have unit variance (for example, if the STANDARD option is used), Table 23.1 can be used to obtain an initial guess for the R= option:

Table 23.1. Reasonable First Guess for the R= Option for Standardized Data

Number of	Number of Variables									
Observations	1	2	3	4	5	6	7	8	9	10
20	1.01	1.36	1.77	2.23	2.73	3.25	3.81	4.38	4.98	5.60
35	0.91	1.24	1.64	2.08	2.56	3.08	3.62	4.18	4.77	5.38
50	0.84	1.17	1.56	1.99	2.46	2.97	3.50	4.06	4.64	5.24
75	0.78	1.09	1.47	1.89	2.35	2.85	3.38	3.93	4.50	5.09
100	0.73	1.04	1.41	1.82	2.28	2.77	3.29	3.83	4.40	4.99
150	0.68	0.97	1.33	1.73	2.18	2.66	3.17	3.71	4.27	4.85
200	0.64	0.93	1.28	1.67	2.11	2.58	3.09	3.62	4.17	4.75
350	0.57	0.85	1.18	1.56	1.98	2.44	2.93	3.45	4.00	4.56
500	0.53	0.80	1.12	1.49	1.91	2.36	2.84	3.35	3.89	4.45
750	0.49	0.74	1.06	1.42	1.82	2.26	2.74	3.24	3.77	4.32
1000	0.46	0.71	1.01	1.37	1.77	2.20	2.67	3.16	3.69	4.23
1500	0.43	0.66	0.96	1.30	1.69	2.11	2.57	3.06	3.57	4.11
2000	0.40	0.63	0.92	1.25	1.63	2.05	2.50	2.99	3.49	4.03

Since infinite d^* values occur in density linkage, the final number of clusters can exceed one when there are wide gaps between the clusters or when the smoothing parameter results in little smoothing.

Density linkage applies no constraints to the shapes of the clusters and, unlike most other hierarchical clustering methods, is capable of recovering clusters with elongated or irregular shapes. Since density linkage employs less prior knowledge about the shape of the clusters than do methods restricted to compact clusters, density linkage is less effective at recovering compact clusters from small samples than are methods that always recover compact clusters, regardless of the data.

EML

The following method is obtained by specifying METHOD=EML. The distance between two clusters is given by

$$D_{KL} = nv \ln\left(1 + \frac{B_{KL}}{P_G}\right) - 2\left(N_M \ln(N_M) - N_K \ln(N_K) - N_L \ln(N_L)\right)$$

The EML method joins clusters to maximize the likelihood at each level of the hierarchy under the following assumptions.

- multivariate normal mixture
- equal spherical covariance matrices

- unequal sampling probabilities

The EML method is similar to Ward's minimum-variance method but removes the bias toward equal-sized clusters. Practical experience has indicated that EML is somewhat biased toward unequal-sized clusters. You can specify the PENALTY= option to adjust the degree of bias. If you specify PENALTY=p, the formula is modified to

$$D_{KL} = nv \ln\left(1 + \frac{B_{KL}}{P_G}\right) - p\left(N_M \ln(N_M) - N_K \ln(N_K) - N_L \ln(N_L)\right)$$

The EML method was derived by W.S. Sarle of SAS Institute Inc. from the maximum-likelihood formula obtained by Symons (1981, p. 37, equation 8) for disjoint clustering. There are currently no other published references on the EML method.

Flexible-Beta Method

The following method is obtained by specifying METHOD=FLEXIBLE. The combinatorial formula is

$$D_{JM} = (D_{JK} + D_{JL})\frac{1-b}{2} + D_{KL}b$$

where b is the value of the BETA= option, or -0.25 by default.

The flexible-beta method was developed by Lance and Williams (1967). See also Milligan (1987).

McQuitty's Similarity Analysis

The following method is obtained by specifying METHOD=MCQUITTY. The combinatorial formula is

$$D_{JM} = \frac{D_{JK} + D_{JL}}{2}$$

The method was independently developed by Sokal and Michener (1958) and McQuitty (1966).

Median Method

The following method is obtained by specifying METHOD=MEDIAN. If $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{D_{JK} + D_{JL}}{2} - \frac{D_{KL}}{4}$$

The median method was developed by Gower (1967).

Single Linkage

The following method is obtained by specifying METHOD=SINGLE. The distance between two clusters is defined by

$$D_{KL} = \min_{i \in C_K} \min_{j \in C_L} d(x_i, x_j)$$

The combinatorial formula is

$$D_{JM} = \min(D_{JK}, D_{JL})$$

In single linkage, the distance between two clusters is the minimum distance between an observation in one cluster and an observation in the other cluster. Single linkage has many desirable theoretical properties (Jardine and Sibson 1971; Fisher and Van Ness 1971; Hartigan 1981) but has fared poorly in Monte Carlo studies (for example, Milligan 1980). By imposing no constraints on the shape of clusters, single linkage sacrifices performance in the recovery of compact clusters in return for the ability to detect elongated and irregular clusters. You must also recognize that single linkage tends to chop off the tails of distributions before separating the main clusters (Hartigan 1981). The notorious chaining tendency of single linkage can be alleviated by specifying the TRIM= option (Wishart 1969, pp. 296–298).

Density linkage and two-stage density linkage retain most of the virtues of single linkage while performing better with compact clusters and possessing better asymptotic properties (Wong and Lane 1983).

Single linkage was originated by Florek et al. (1951a, 1951b) and later reinvented by McQuitty (1957) and Sneath (1957).

Two-Stage Density Linkage

If you specify METHOD=DENSITY, the modal clusters often merge before all the points in the tails have clustered. The option METHOD=TWOSTAGE is a modification of density linkage that ensures that all points are assigned to modal clusters before the modal clusters are allowed to join. The CLUSTER procedure supports the same three varieties of two-stage density linkage as of ordinary density linkage: kth-nearest neighbor, uniform kernel, and hybrid.

In the first stage, disjoint modal clusters are formed. The algorithm is the same as the single linkage algorithm ordinarily used with density linkage, with one exception: two clusters are joined only if at least one of the two clusters has fewer members than the number specified by the MODE= option. At the end of the first stage, each point belongs to one modal cluster.

In the second stage, the modal clusters are hierarchically joined by single linkage. The final number of clusters can exceed one when there are wide gaps between the clusters or when the smoothing parameter is small.

Each stage forms a tree that can be plotted by the TREE procedure. By default, the TREE procedure plots the tree from the first stage. To obtain the tree for the second stage, use the option HEIGHT=MODE in the PROC TREE statement. You can also produce a single tree diagram containing both stages, with the number of clusters as the height axis, by using the option HEIGHT=N in the PROC TREE statement. To produce an output data set from PROC TREE containing the modal clusters, use _HEIGHT_ for the HEIGHT variable (the default) and specify LEVEL=0.

Two-stage density linkage was developed by W.S. Sarle of SAS Institute Inc. There are currently no other published references on two-stage density linkage.

Ward's Minimum-Variance Method

The following method is obtained by specifying METHOD=WARD. The distance between two clusters is defined by

$$D_{KL} = B_{KL} = \frac{\|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

If $d(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{(N_J + N_K)D_{JK} + (N_J + N_L)D_{JL} - N_J D_{KL}}{N_J + N_M}$$

In Ward's minimum-variance method, the distance between two clusters is the *ANOVA* sum of squares between the two clusters added up over all the variables. At each generation, the within-cluster sum of squares is minimized over all partitions obtainable by merging two clusters from the previous generation. The sums of squares are easier to interpret when they are divided by the total sum of squares to give proportions of variance (squared semipartial correlations).

Ward's method joins clusters to maximize the likelihood at each level of the hierarchy under the following assumptions:

- multivariate normal mixture
- equal spherical covariance matrices
- equal sampling probabilities

Ward's method tends to join clusters with a small number of observations, and it is strongly biased toward producing clusters with roughly the same number of observations. It is also very sensitive to outliers (Milligan 1980).

Ward (1963) describes a class of hierarchical clustering methods including the minimum variance method.

Miscellaneous Formulas

The root-mean-square standard deviation of a cluster C_K is

$$\text{RMSSTD} = \sqrt{\frac{W_K}{v(N_K - 1)}}$$

The R^2 statistic for a given level of the hierarchy is

$$R^2 = 1 - \frac{P_G}{T}$$

The squared semipartial correlation for joining clusters C_K and C_L is

$$\text{semipartial } R^2 = \frac{B_{KL}}{T}$$

The bimodality coefficient is

$$b = \frac{m_3^2 + 1}{m_4 + \frac{3(n-1)^2}{(n-2)(n-3)}}$$

where m_3 is skewness and m_4 is kurtosis. Values of b greater than 0.555 (the value for a uniform population) may indicate bimodal or multimodal marginal distributions. The maximum of 1.0 (obtained for the Bernoulli distribution) is obtained for a population with only two distinct values. Very heavy-tailed distributions have small values of b regardless of the number of modes.

Formulas for the cubic-clustering criterion and approximate expected R^2 are given in Sarle (1983).

The pseudo F statistic for a given level is

$$\text{pseudo } F = \frac{\frac{T-P_G}{G-1}}{\frac{P_G}{n-G}}$$

The pseudo t^2 statistic for joining C_K and C_L is

$$\text{pseudo } t^2 = \frac{B_{KL}}{\frac{W_K+W_L}{N_K+N_L-2}}$$

The pseudo F and t^2 statistics may be useful indicators of the number of clusters, but they are *not* distributed as F and t^2 random variables. If the data are independently sampled from a multivariate normal distribution with a scalar covariance matrix and if the clustering method allocates observations to clusters randomly (which no clustering method actually does), then the pseudo F statistic is distributed as an F random variable with $v(G-1)$ and $v(n-G)$ degrees of freedom. Under the same assumptions, the pseudo t^2 statistic is distributed as an F random variable with v and $v(N_K + N_L - 2)$ degrees of freedom. The pseudo t^2 statistic differs computationally from Hotelling's T^2 in that the latter uses a general symmetric covariance matrix instead of a scalar covariance matrix. The pseudo F statistic was suggested by Calinski and Harabasz (1974). The pseudo t^2 statistic is related to the $J_e(2)/J_e(1)$ statistic of Duda and Hart (1973) by

$$\frac{J_e(2)}{J_e(1)} = \frac{W_K + W_L}{W_M} = \frac{1}{1 + \frac{t^2}{N_K + N_L - 2}}$$

See Milligan and Cooper (1985) and Cooper and Milligan (1988) regarding the performance of these statistics in estimating the number of population clusters. Conservative tests for the number of clusters using the pseudo F and t^2 statistics can be obtained by the Bonferroni approach (Hawkins, Muller, and ten Krooden 1982, pp. 337–340).

Ultrametrics

A dissimilarity measure $d(x, y)$ is called an *ultrametric* if it satisfies the following conditions:

- $d(x, x) = 0$ for all x
- $d(x, y) \geq 0$ for all x, y
- $d(x, y) = d(y, x)$ for all x, y
- $d(x, y) \leq \max\left(d(x, z), d(y, z)\right)$ for all x, y, and z

Any hierarchical clustering method induces a dissimilarity measure on the observations, say $h(x_i, x_j)$. Let C_M be the cluster with the fewest members that contains both x_i and x_j. Assume C_M was formed by joining C_K and C_L. Then define $h(x_i, x_j) = D_{KL}$.

If the fusion of C_K and C_L reduces the number of clusters from g to $g-1$, then define $D_{(g)} = D_{KL}$. Johnson (1967) shows that if

$$0 \leq D_{(n)} \leq D_{(n-1)} \leq \cdots \leq D_{(2)}$$

then $h(\cdot, \cdot)$ is an ultrametric. A method that always satisfies this condition is said to be a *monotonic* or *ultrametric clustering method*. All methods implemented in PROC CLUSTER except CENTROID, EML, and MEDIAN are ultrametric (Milligan 1979; Batagelj 1981).

Algorithms

Anderberg (1973) describes three algorithms for implementing agglomerative hierarchical clustering: stored data, stored distance, and sorted distance. The algorithms used by PROC CLUSTER for each method are indicated in Table 23.2. For METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, either the stored data or the stored distance algorithm can be used. For these methods, if the data are distances or if you specify the NOSQUARE option, the stored distance algorithm is used; otherwise, the stored data algorithm is used.

Table 23.2. Three Algorithms for Implementing Agglomerative Hierarchical Clustering

	Algorithm		
Stored Method	**Stored Data**	**Stored Distance**	**Sorted Distance**
AVERAGE	x	x	
CENTROID	x	x	
COMPLETE		x	
DENSITY			x
EML	x		
FLEXIBLE		x	
MCQUITTY		x	
MEDIAN		x	
SINGLE		x	
TWOSTAGE			x
WARD	x	x	

Computational Resources

The CLUSTER procedure stores the data (including the COPY and ID variables) in memory or, if necessary, on disk. If eigenvalues are computed, the covariance matrix is stored in memory. If the stored distance or sorted distance algorithm is used, the distances are stored in memory or, if necessary, on disk.

With coordinate data, the increase in CPU time is roughly proportional to the number of variables. The VAR statement should list the variables in order of decreasing variance for greatest efficiency.

For both coordinate and distance data, the dominant factor determining CPU time is the number of observations. For density methods with coordinate data, the asymptotic time requirements are somewhere between $n \ln(n)$ and n^2, depending on how the smoothing parameter increases. For other methods except EML, time is roughly proportional to n^2. For the EML method, time is roughly proportional to n^3.

PROC CLUSTER runs much faster if the data can be stored in memory and, if the stored distance algorithm is used, the distance matrix can be stored in memory as well. To estimate the bytes of memory needed for the data, use the following equation and round up to the nearest multiple of d.

$$\begin{array}{llll} n(vd & + & 8d\ +\ i & \\ & + & i & \text{if density estimation or the} \\ & & & \text{sorted distance algorithm used} \\ & + & 3d & \text{if stored data algorithm used} \\ & + & 3d & \text{if density estimation used} \\ & + & \text{max(8, length of ID variable)} & \text{if ID variable used} \\ & + & \text{length of ID variable} & \text{if ID variable used} \\ & + & \text{sum of lengths of COPY variables)} & \text{if COPY variables used} \end{array}$$

where

n is the number of observations

v is the number of variables

d is the size of a C variable of type *double*. For most computers, $d = 8$.

i is the size of a C variable of type *int*. For most computers, $i = 4$.

The number of bytes needed for the distance matrix is $dn(n + 1)/2$.

Missing Values

If the data are coordinates, observations with missing values are excluded from the analysis. If the data are distances, missing values are not allowed in the lower triangle of the distance matrix. The upper triangle is ignored. For more on TYPE=DISTANCE data sets, see Appendix A, “Special SAS Data Sets.”

Ties

At each level of the clustering algorithm, PROC CLUSTER must identify the pair of clusters with the minimum distance. Sometimes, usually when the data are discrete, there may be two or more pairs with the same minimum distance. In such cases the tie must be broken in some arbitrary way. If there are ties, then the results of the cluster analysis depend on the order of the observations in the data set. The presence of ties is reported in the SAS log and in the column of the cluster history labeled “Tie” unless the NOTIE option is specified.

PROC CLUSTER breaks ties as follows. Each cluster is identified by the smallest observation number among its members. For each pair of clusters, there is a smaller identification number and a larger identification number. If two or more pairs of clusters are tied for minimum distance between clusters, the pair that has the minimum larger identification number is merged. If there is a tie for minimum larger identification number, the pair that has the minimum smaller identification number is merged.

This method for breaking ties is different from that used in Version 5. The change in the algorithm may produce changes in the resulting clusters.

A tie means that the level in the cluster history at which the tie occurred and possibly some of the subsequent levels are not uniquely determined. Ties that occur early in the cluster history usually have little effect on the later stages. Ties that occur in the middle part of the cluster history are cause for further investigation. Ties late in the cluster history indicate important indeterminacies.

The importance of ties can be assessed by repeating the cluster analysis for several different random permutations of the observations. The discrepancies at a given level can be examined by crosstabulating the clusters obtained at that level for all of the permutations. See Example 23.4 on page 1027 for details.

Size, Shape, and Correlation

In some biological applications, the organisms that are being clustered may be at different stages of growth. Unless it is the growth process itself that is being studied, differences in size among such organisms are not of interest. Therefore, distances among organisms should be computed in such a way as to control for differences in size while retaining information about differences in shape.

If coordinate data are measured on an interval scale, you can control for size by subtracting a measure of the overall size of each observation from each datum. For example, if no other direct measure of size is available, you could subtract the mean of each row of the data matrix, producing a row-centered coordinate matrix. An easy way to subtract the mean of each row is to use PROC STANDARD on the transposed coordinate matrix:

```
proc transpose data= coordinate-datatype ;
proc standard m=0;
proc transpose out=row-centered-coordinate-data;
```

Another way to remove size effects from interval-scale coordinate data is to do a principal component analysis and discard the first component (Blackith and Reyment 1971).

If the data are measured on a ratio scale, you can control for size by dividing each datum by a measure of overall size; in this case, the geometric mean is a more natural measure of size than the arithmetic mean. However, it is often more meaningful to analyze the logarithms of ratio-scaled data, in which case you can subtract the arithmetic mean after taking logarithms. You must also consider the dimensions of measurement. For example, if you have measures of both length and weight, you may need to cube the measures of length or take the cube root of the weights. Various other complications may also arise in real applications, such as different growth rates for different parts of the body (Sneath and Sokal 1973).

Issues of size and shape are pertinent to many areas besides biology (for example, Hamer and Cunningham 1981). Suppose you have data consisting of subjective ratings made by several different raters. Some raters may tend to give higher overall ratings than other raters. Some raters may also tend to spread out their ratings over more of the scale than do other raters. If it is impossible for you to adjust directly for rater differences, then distances should be computed in such a way as to control for both differences in size and variability. For example, if the data are considered to be measured on an interval scale, you can subtract the mean of each observation and divide by the standard deviation, producing a row-standardized coordinate matrix. With some clustering methods, analyzing squared Euclidean distances from a row-standardized coordinate matrix is equivalent to analyzing the matrix of correlations among rows, since squared Euclidean distance is an affine transformation of the correlation (Hartigan 1975, p. 64).

If you do an analysis of row-centered or row-standardized data, you need to consider whether the columns (variables) should be standardized before centering or standardizing the rows, after centering or standardizing the rows, or both before and after. If you standardize the columns after standardizing the rows, then strictly speaking you are not analyzing shape because the profiles are distorted by standardizing the columns; however, this type of double standardization may be necessary in practice to get reasonable results. It is not clear whether iterating the standardization of rows and columns may be of any benefit.

The choice of distance or correlation measure should depend on the meaning of the data and the purpose of the analysis. Simulation studies that compare distance and correlation measures are useless unless the data are generated to mimic data from your field of application; conclusions drawn from artificial data cannot be generalized because it is possible to generate data such that distances that include size effects work better or such that correlations work better.

You can standardize the rows of a data set by using a DATA step or by using the TRANSPOSE and STANDARD procedures. You can also use PROC TRANSPOSE and then have PROC CORR create a TYPE=CORR data set containing a correlation matrix. If you want to analyze a TYPE=CORR data set with PROC CLUSTER, you must use a DATA step to perform the following steps:

1. Set the data set TYPE= to DISTANCE.
2. Convert the correlations to dissimilarities by computing $1 - r$, $\sqrt{1 - r}$, $1 - r^2$, or some other decreasing function.
3. Delete observations for which the variable _TYPE_ does not have the value ’CORR’.

See Example 23.6 on page 1044 for an analysis of a data set in which size information is detrimental to the classification.

Output Data Set

The OUTTREE= data set contains one observation for each observation in the input data set, plus one observation for each cluster of two or more observations (that is, one observation for each node of the cluster tree). The total number of output observations is usually $2n - 1$, where n is the number of input observations. The density methods may produce fewer output observations when the number of clusters cannot be reduced to one.

The label of the OUTTREE= data set identifies the type of cluster analysis performed and is automatically displayed when the TREE procedure is invoked.

The variables in the OUTTREE= data set are as follows:

- the BY variables, if you use a BY statement
- the ID variable, if you use an ID statement
- the COPY variables, if you use a COPY statement
- _NAME_, a character variable giving the name of the node. If the node is a cluster, the name is CLn, where n is the number of the cluster. If the node is an observation, the name is OBn, where n is the observation number. If the node is an observation and the ID statement is used, the name is the formatted value of the ID variable.
- _PARENT_, a character variable giving the value of _NAME_ of the parent of the node
- _NCL_, the number of clusters
- _FREQ_, the number of observations in the current cluster
- _HEIGHT_, the distance or similarity between the last clusters joined, as defined in the section "Clustering Methods" on page 975. The variable _HEIGHT_ is used by the TREE procedure as the default height axis. The label of the _HEIGHT_ variable identifies the between-cluster distance measure. For METHOD=TWOSTAGE, the _HEIGHT_ variable contains the densities at which clusters joined in the first stage; for clusters formed in the second stage, _HEIGHT_ is a very small negative number.

If the input data set contains coordinates, the following variables appear in the output data set:

- the variables containing the coordinates used in the cluster analysis. For output observations that correspond to input observations, the values of the coordinates are the same in both data sets except for some slight numeric error possibly introduced by standardizing and unstandardizing if the STANDARD option is used. For output observations that correspond to clusters of more than one input observation, the values of the coordinates are the cluster means.
- _ERSQ_, the approximate expected value of R^2 under the uniform null hypothesis

- _RATIO_, equal to $\frac{1-\text{\_ERSQ\_}}{1-\text{\_RSQ\_}}$
- _LOGR_, natural logarithm of _RATIO_
- _CCC_, the cubic clustering criterion

The variables _ERSQ_, _RATIO_, _LOGR_, and _CCC_ have missing values when the number of clusters is greater than one-fifth the number of observations.

If the input data set contains coordinates and METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then the following variables appear in the output data set.

- _DIST_, the Euclidean distance between the means of the last clusters joined
- _AVLINK_, the average distance between the last clusters joined

If the input data set contains coordinates or METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then the following variables appear in the output data set:

- _RMSSTD_, the root-mean-square standard deviation of the current cluster
- _SPRSQ_, the semipartial squared multiple correlation or the decrease in the proportion of variance accounted for due to joining two clusters to form the current cluster
- _RSQ_, the squared multiple correlation
- _PSF_, the pseudo F statistic
- _PST2_, the pseudo t^2 statistic

If METHOD=EML, then the following variable appears in the output data set:

- _LNLR_, the log-likelihood ratio

If METHOD=TWOSTAGE or METHOD=DENSITY, the following variable appears in the output data set:

- _MODE_, pertaining to the modal clusters. With METHOD=DENSITY, the _MODE_ variable indicates the number of modal clusters contained by the current cluster. With METHOD=TWOSTAGE, the _MODE_ variable gives the maximum density in each modal cluster and the fusion density, d^*, for clusters containing two or more modal clusters; for clusters containing no modal clusters, _MODE_ is missing.

If nonparametric density estimates are requested (when METHOD=DENSITY or METHOD=TWOSTAGE and the HYBRID option is not used; or when the TRIM= option is used), the output data set contains

- _DENS_, the maximum density in the current cluster

Displayed Output

If you specify the SIMPLE option and the data are coordinates, PROC CLUSTER produces simple descriptive statistics for each variable:

- the Mean
- the standard deviation, Std Dev
- the Skewness
- the Kurtosis
- a coefficient of Bimodality

If the data are coordinates and you do not specify the NOEIGEN option, PROC CLUSTER displays

- the Eigenvalues of the Correlation or Covariance Matrix
- the Difference between successive eigenvalues
- the Proportion of variance explained by each eigenvalue
- the Cumulative proportion of variance explained

If the data are coordinates, PROC CLUSTER displays the Root-Mean-Square Total-Sample Standard Deviation of the variables

If the distances are normalized, PROC CLUSTER displays one of the following, depending on whether squared or unsquared distances are used:

- the Root-Mean-Square Distance Between Observations
- the Mean Distance Between Observations

For the generations in the clustering process specified by the PRINT= option, PROC CLUSTER displays

- the Number of Clusters or NCL
- the names of the Clusters Joined. The observations are identified by the formatted value of the ID variable, if any; otherwise, the observations are identified by OBn, where n is the observation number. The CLUSTER procedure displays the entire value of the ID variable in the cluster history instead of truncating at 16 characters. Long ID values may be flowed onto several lines. Clusters of two or more observations are identified as CLn, where n is the number of clusters existing after the cluster in question is formed.
- the number of observations in the new cluster, Frequency of New Cluster or FREQ

If you specify the RMSSTD option and if the data are coordinates or if you specify METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then PROC CLUSTER displays the root-mean-square standard deviation of the new cluster, RMS Std of New Cluster or RMS Std.

PROC CLUSTER displays the following items if you specify METHOD=WARD. It also displays them if you specify the RSQUARE option and either the data are coordinates or you specify METHOD=AVERAGE or METHOD=CENTROID:

- the decrease in the proportion of variance accounted for resulting from joining the two clusters, Semipartial R-Squared or SPRSQ. This equals the between-cluster sum of squares divided by the corrected total sum of squares.
- the squared multiple correlation, R-Squared or RSQ. R^2 is the proportion of variance accounted for by the clusters.

If you specify the CCC option and the data are coordinates, PROC CLUSTER displays

- Approximate Expected R-Squared or ERSQ, the approximate expected value of R^2 under the uniform null hypothesis
- the Cubic Clustering Criterion or CCC. The cubic clustering criterion and approximate expected R^2 are given missing values when the number of clusters is greater than one-fifth the number of observations.

If you specify the PSEUDO option and if the data are coordinates or METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then PROC CLUSTER displays

- Pseudo F or PSF, the pseudo F statistic measuring the separation among all the clusters at the current level
- Pseudo t^2 or PST2, the pseudo t^2 statistic measuring the separation between the two clusters most recently joined

If you specify the NOSQUARE option and METHOD=AVERAGE, PROC CLUSTER displays the (Normalized) Average Distance or (Norm) Aver Dist, the average distance between pairs of objects in the two clusters joined with one object from each cluster.

If you do not specify the NOSQUARE option and METHOD=AVERAGE, PROC CLUSTER displays the (Normalized) RMS Distance or (Norm) RMS Dist, the root-mean-square distance between pairs of objects in the two clusters joined with one object from each cluster.

If METHOD=CENTROID, PROC CLUSTER displays the (Normalized) Centroid Distance or (Norm) Cent Dist, the distance between the two cluster centroids.

If METHOD=COMPLETE, PROC CLUSTER displays the (Normalized) Maximum Distance or (Norm) Max Dist, the maximum distance between the two clusters.

If METHOD=DENSITY or METHOD=TWOSTAGE, PROC CLUSTER displays

- Normalized Fusion Density or Normalized Fusion Dens, the value of d^* as defined in the section "Clustering Methods" on page 975
- the Normalized Maximum Density in Each Cluster joined, including the Lesser or Min, and the Greater or Max, of the two maximum density values

If METHOD=EML, PROC CLUSTER displays

- Log Likelihood Ratio or LNLR
- Log Likelihood or LNLIKE

If METHOD=FLEXIBLE, PROC CLUSTER displays the (Normalized) Flexible Distance or (Norm) Flex Dist, the distance between the two clusters based on the Lance-Williams flexible formula.

If METHOD=MEDIAN, PROC CLUSTER displays the (Normalized) Median Distance or (Norm) Med Dist, the distance between the two clusters based on the median method.

If METHOD=MCQUITTY, PROC CLUSTER displays the (Normalized) McQuitty's Similarity or (Norm) MCQ, the distance between the two clusters based on McQuitty's similarity method.

If METHOD=SINGLE, PROC CLUSTER displays the (Normalized) Minimum Distance or (Norm) Min Dist, the minimum distance between the two clusters.

If you specify the NONORM option and METHOD=WARD, PROC CLUSTER displays the Between-Cluster Sum of Squares or BSS, the *ANOVA* sum of squares between the two clusters joined.

If you specify neither the NOTIE option nor METHOD=TWOSTAGE or METHOD=DENSITY, PROC CLUSTER displays Tie, where a T in the column indicates a tie for minimum distance and a blank indicates the absence of a tie.

After the cluster history, if METHOD=TWOSTAGE or METHOD=DENSITY, PROC CLUSTER displays the number of modal clusters.

ODS Table Names

PROC CLUSTER assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 23.3. ODS Tables Produced in PROC CLUSTER

ODS Table Name	Description	Statement	Option
ClusterHistory	Obs or clusters joined, frequencies and other cluster statistics	PROC	default
SimpleStatistics	Simple statistics, before or after trimming	PROC	SIMPLE
EigenvalueTable	Eigenvalues of the CORR or COV matrix	PROC	default

Examples

Example 23.1. Cluster Analysis of Flying Mileages between Ten American Cities

This first example clusters ten American cities based on the flying mileages between them. Six clustering methods are shown with corresponding tree diagrams produced by the TREE procedure. The EML method cannot be used because it requires coordinate data. The other omitted methods produce the same clusters, although not the same distances between clusters, as one of the illustrated methods: complete linkage and the flexible-beta method yield the same clusters as Ward's method, McQuitty's similarity analysis produces the same clusters as average linkage, and the median method corresponds to the centroid method.

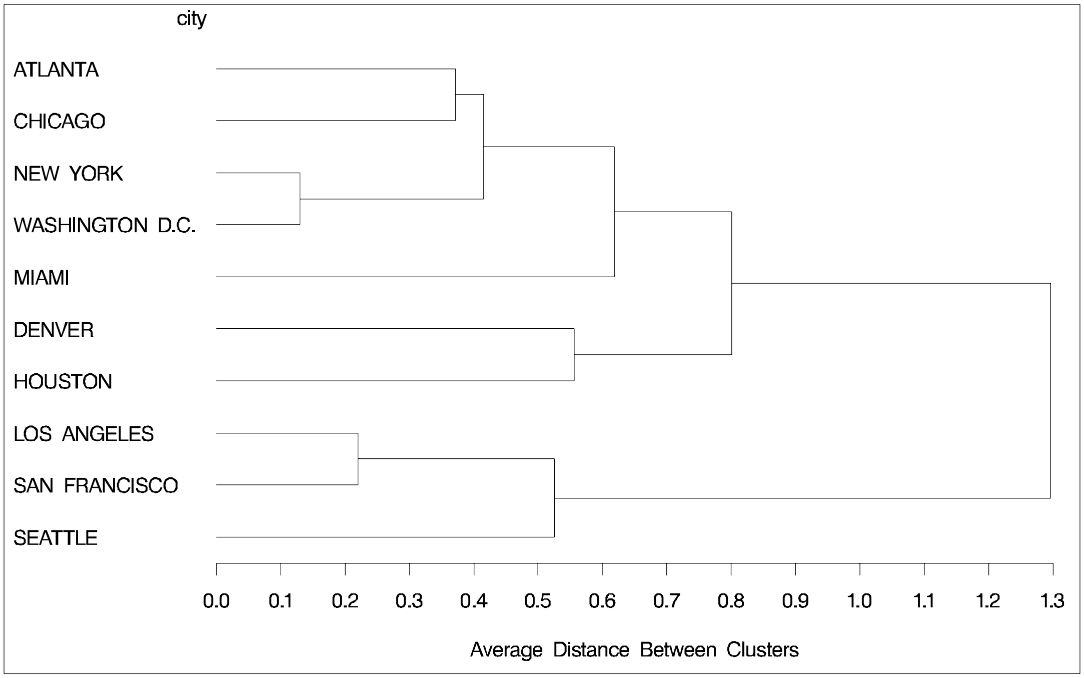
All of the methods suggest a division of the cities into two clusters along the east-west dimension. There is disagreement, however, about which cluster Denver should belong to. Some of the methods indicate a possible third cluster containing Denver and Houston. The following statements produce Output 23.1.1:

```
title 'Cluster Analysis of Flying Mileages Between 10 American Cities';
data mileages(type=distance);
   input (atlanta chicago denver houston losangeles
         miami newyork sanfran seattle washdc) (5.)
         @55 city $15.;
   datalines;
    0                                                 ATLANTA
  587    0                                            CHICAGO
 1212  920    0                                       DENVER
  701  940  879    0                                  HOUSTON
 1936 1745  831 1374    0                             LOS ANGELES
  604 1188 1726  968 2339    0                        MIAMI
  748  713 1631 1420 2451 1092    0                   NEW YORK
 2139 1858  949 1645  347 2594 2571    0              SAN FRANCISCO
 2182 1737 1021 1891  959 2734 2408  678    0         SEATTLE
  543  597 1494 1220 2300  923  205 2442 2329    0    WASHINGTON D.C.
;


/*---------------------- Average linkage --------------------*/
```

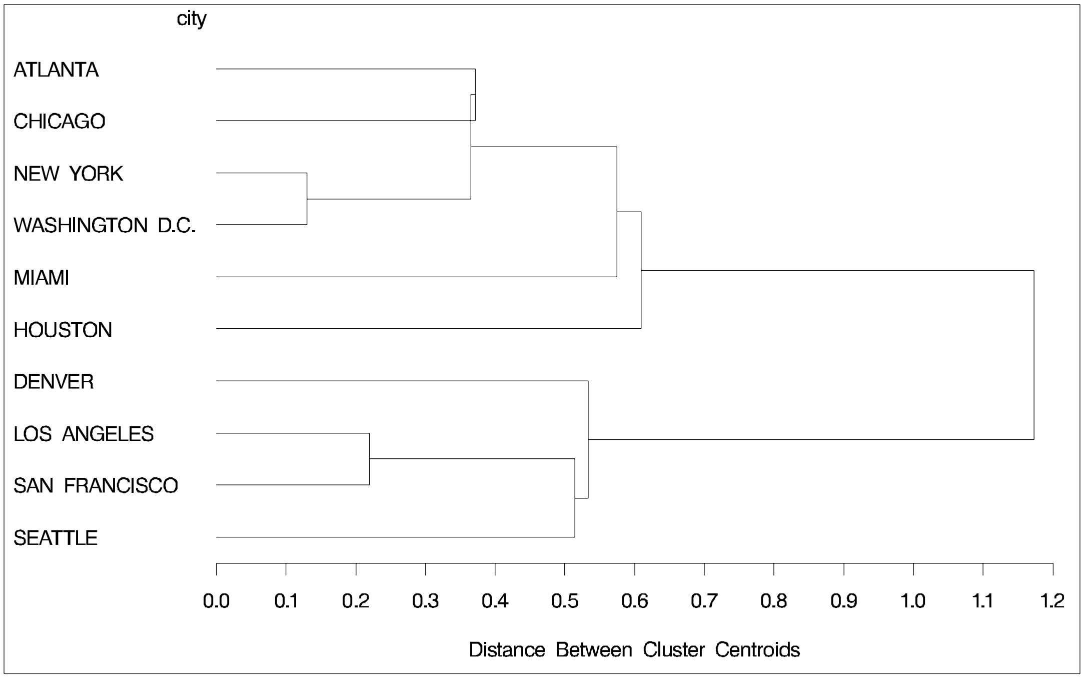
```
 proc cluster data=mileages method=average pseudo;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;

/*---------------------- Centroid method -------------------*/
proc cluster data=mileages method=centroid pseudo;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;

/*-------- Density linkage with 3rd-nearest-neighbor --------*/
proc cluster data=mileages method=density k=3;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;

/*--------------------- Single linkage ---------------------*/
proc cluster data=mileages method=single;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;

/*--- Two-stage density linkage with 3rd-nearest-neighbor ---*/
proc cluster data=mileages method=twostage k=3;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;

/* Ward’s minimum variance with pseudo $F$ and $t^2$ statistics */
proc cluster data=mileages method=ward pseudo;
   id city;
run;

proc tree horizontal spaces=2;
   id city;
run;
```

Output 23.1.1. Statistics and Tree Diagrams for Six Different Clustering Methods

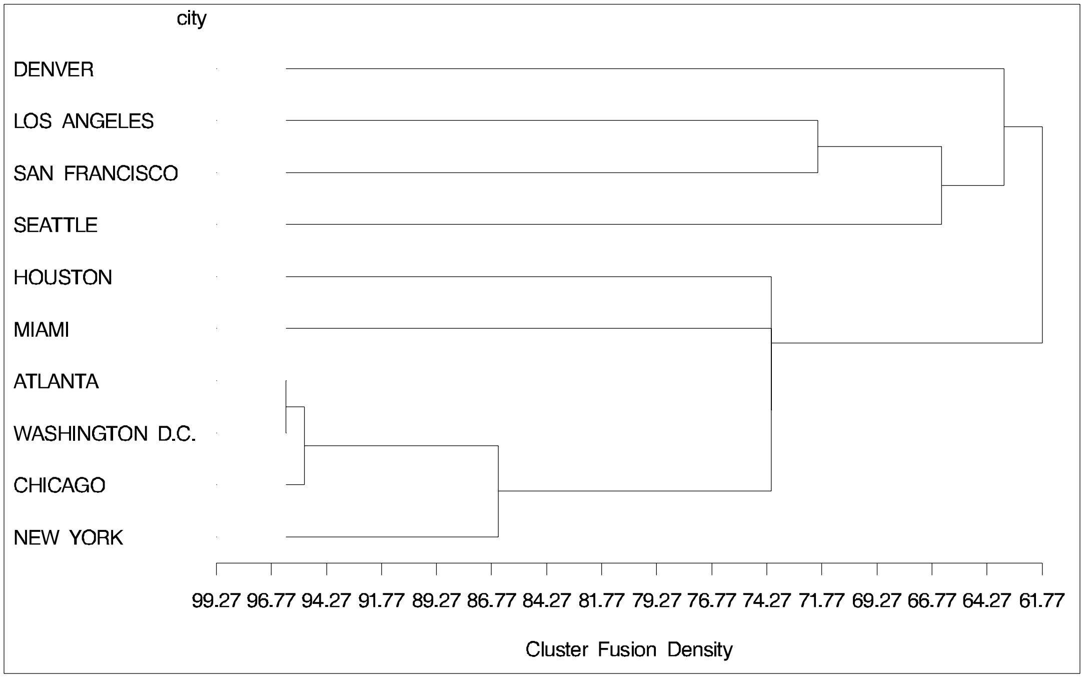
```
                   Cluster Analysis of Flying Mileages Between 10 American Cities

                                     The CLUSTER Procedure
                                 Average Linkage Cluster Analysis

                    Root-Mean-Square Distance Between Observations   = 1580.242


                                          Cluster History
                                                                             Norm    T
                                                                              RMS    i
          NCL    ---------Clusters Joined----------      FREQ     PSF    PST2    Dist    e

            9    NEW YORK           WASHINGTON D.C.         2    66.7       .    0.1297
            8    LOS ANGELES        SAN FRANCISCO           2    39.2       .    0.2196
            7    ATLANTA            CHICAGO                 2    21.7       .    0.3715
            6    CL7                CL9                     4    14.5     3.4    0.4149
            5    CL8                SEATTLE                 3    12.4     7.3    0.5255
            4    DENVER             HOUSTON                 2    13.9       .    0.5562
            3    CL6                MIAMI                   5    15.5     3.8    0.6185
            2    CL3                CL4                     7    16.0     5.3    0.8005
            1    CL2                CL5                    10       .    16.0    1.2967
```



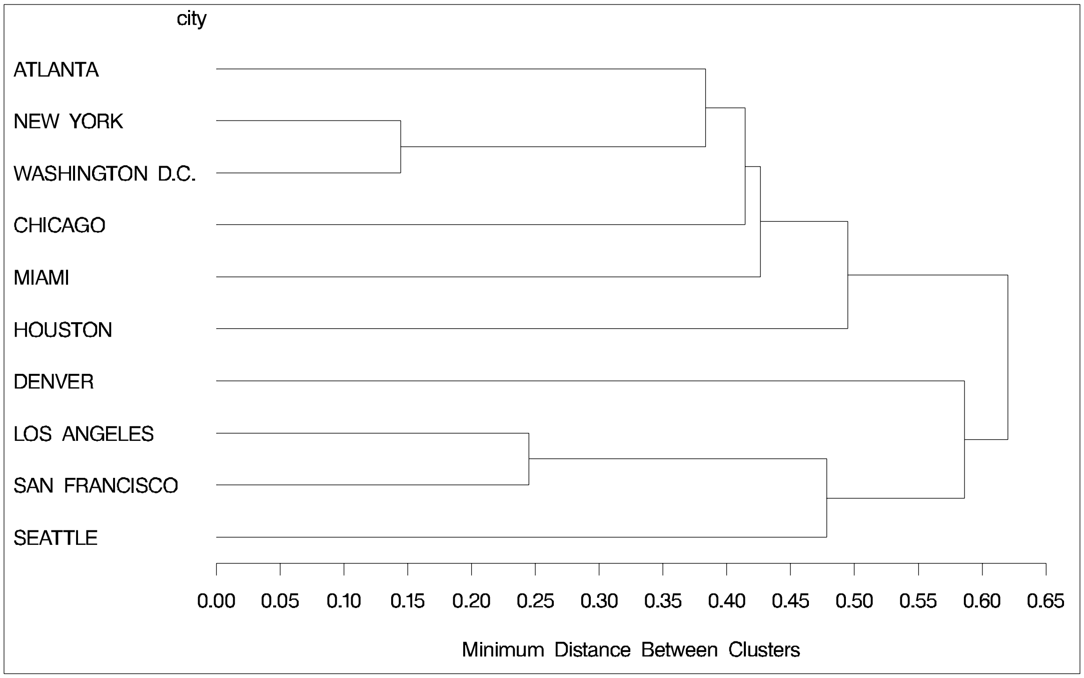
```
                    Cluster Analysis of Flying Mileages Between 10 American Cities

                                      The CLUSTER Procedure
                                Centroid Hierarchical Cluster Analysis

                     Root-Mean-Square Distance Between Observations   = 1580.242


                                          Cluster History
                                                                             Norm    T
                                                                             Cent    i
          NCL    ---------Clusters Joined----------      FREQ     PSF    PST2      Dist    e

            9    NEW YORK           WASHINGTON D.C.         2    66.7       .    0.1297
            8    LOS ANGELES        SAN FRANCISCO           2    39.2       .    0.2196
            7    ATLANTA            CHICAGO                 2    21.7       .    0.3715
            6    CL7                CL9                     4    14.5     3.4    0.3652
            5    CL8                SEATTLE                 3    12.4     7.3    0.5139
            4    DENVER             CL5                     4    12.4     2.1    0.5337
            3    CL6                MIAMI                   5    14.2     3.8    0.5743
            2    CL3                HOUSTON                 6    22.1     2.6    0.6091
            1    CL2                CL4                    10       .    22.1     1.173
```



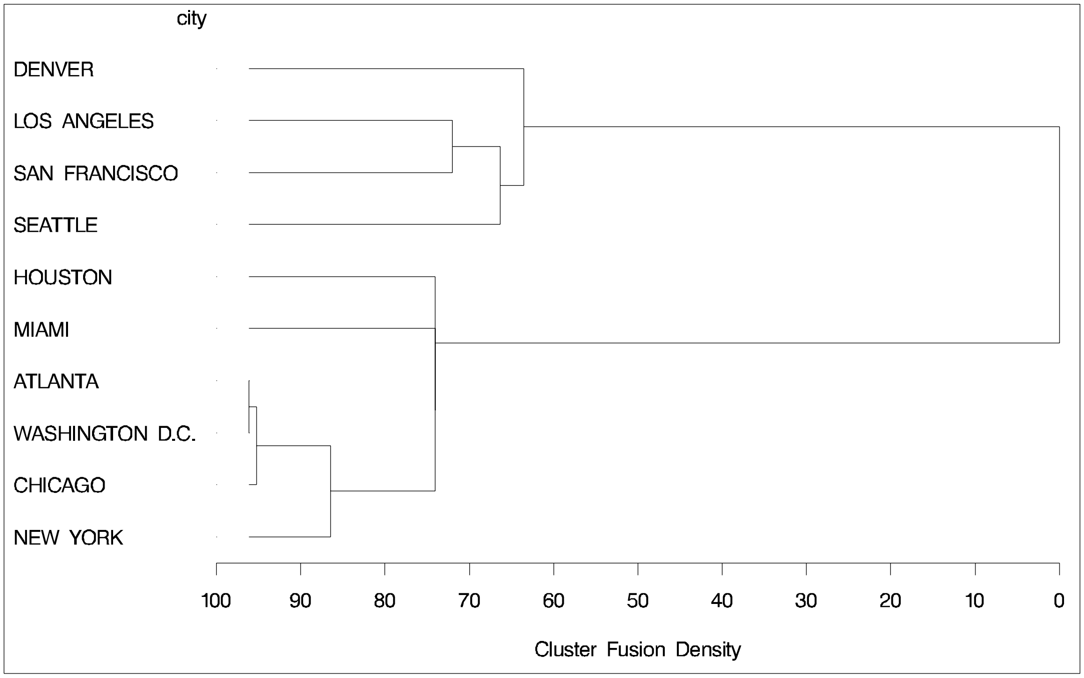
```
                 Cluster Analysis of Flying Mileages Between 10 American Cities

                                    The CLUSTER Procedure
                                Density Linkage Cluster Analysis

                                            K = 3


                                       Cluster History
                                                  Normalized           Maximum Density     T
                                                      Fusion           in Each Cluster     i
 NCL    ---------Clusters Joined----------      FREQ     Density           Lesser     Greater    e

   9    ATLANTA            WASHINGTON D.C.         2      96.106          92.5043       100.0
   8    CL9                CHICAGO                 3      95.263          90.9548       100.0
   7    CL8                NEW YORK                4      86.465          76.1571       100.0
   6    CL7                HOUSTON                 5      74.079          61.7747       100.0    T
   5    CL6                MIAMI                   6      74.079          58.8299       100.0
   4    LOS ANGELES        SAN FRANCISCO           2      71.968          65.3430     80.0885
   3    CL4                SEATTLE                 3      66.341          56.6215     80.0885
   2    CL3                DENVER                  4      63.509          61.7747     80.0885
   1    CL5                CL2                    10      61.775    *     80.0885       100.0

                         * indicates fusion of two modal or multimodal clusters
                                   2 modal clusters have been formed.
```



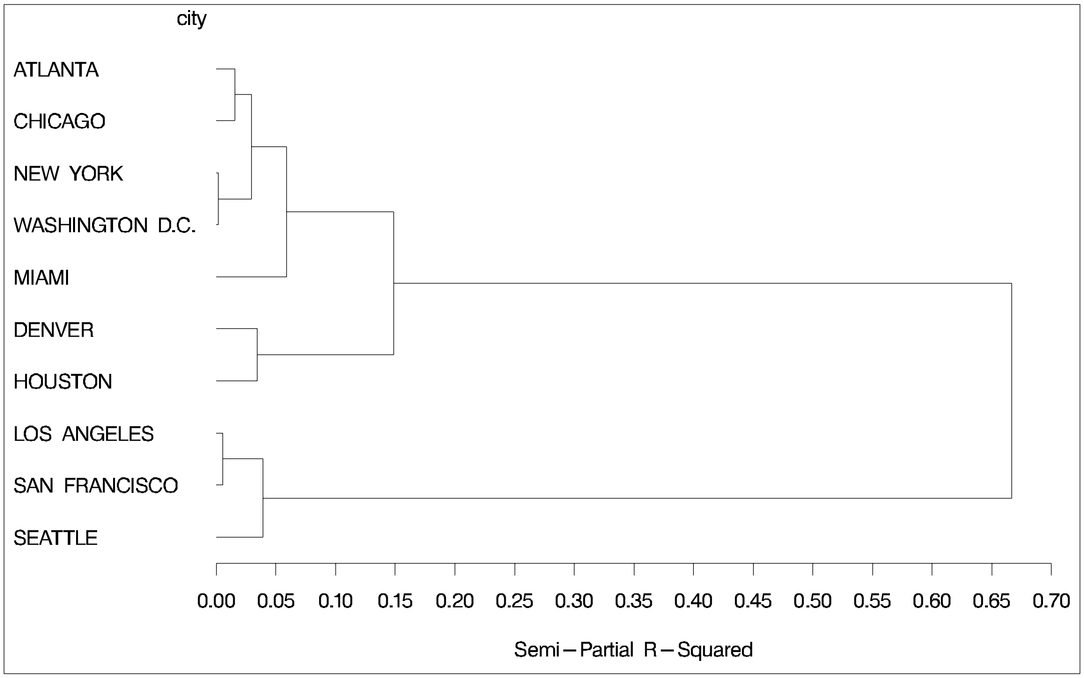
```
 Cluster Analysis of Flying Mileages Between 10 American Cities

                     The CLUSTER Procedure
                Single Linkage Cluster Analysis

  Mean Distance Between Observations               = 1417.133


                        Cluster History
                                                         Norm    T
                                                          Min    i
NCL    ---------Clusters Joined----------      FREQ      Dist    e

  9    NEW YORK           WASHINGTON D.C.         2    0.1447
  8    LOS ANGELES        SAN FRANCISCO           2    0.2449
  7    ATLANTA            CL9                     3    0.3832
  6    CL7                CHICAGO                 4    0.4142
  5    CL6                MIAMI                   5    0.4262
  4    CL8                SEATTLE                 3    0.4784
  3    CL5                HOUSTON                 6    0.4947
  2    DENVER             CL4                     4    0.5864
  1    CL3                CL2                    10    0.6203
```



```
                   Cluster Analysis of Flying Mileages Between 10 American Cities

                                     The CLUSTER Procedure
                               Two-Stage Density Linkage Clustering

                                             K = 3


                                        Cluster History
                                                           Normalized      Maximum Density      T
                                                             Fusion        in Each Cluster      i
          NCL    ---------Clusters Joined----------      FREQ    Density      Lesser     Greater    e

            9    ATLANTA            WASHINGTON D.C.         2     96.106     92.5043       100.0
            8    CL9                CHICAGO                 3     95.263     90.9548       100.0
            7    CL8                NEW YORK                4     86.465     76.1571       100.0
            6    CL7                HOUSTON                 5     74.079     61.7747       100.0    T
            5    CL6                MIAMI                   6     74.079     58.8299       100.0
            4    LOS ANGELES        SAN FRANCISCO           2     71.968     65.3430     80.0885
            3    CL4                SEATTLE                 3     66.341     56.6215     80.0885
            2    CL3                DENVER                  4     63.509     61.7747     80.0885
            1    CL5                CL2                    10     61.775     80.0885       100.0
                                  2 modal clusters have been formed.
```



```
                     Cluster Analysis of Flying Mileages Between 10 American Cities

                                        The CLUSTER Procedure
                                 Ward's Minimum Variance Cluster Analysis

                      Root-Mean-Square Distance Between Observations   = 1580.242


                                           Cluster History
                                                                                           T
                                                                                           i
         NCL    ---------Clusters Joined----------      FREQ     SPRSQ     RSQ     PSF    PST2    e

           9    NEW YORK           WASHINGTON D.C.         2    0.0019    .998    66.7      .
           8    LOS ANGELES        SAN FRANCISCO           2    0.0054    .993    39.2      .
           7    ATLANTA            CHICAGO                 2    0.0153    .977    21.7      .
           6    CL7                CL9                     4    0.0296    .948    14.5     3.4
           5    DENVER             HOUSTON                 2    0.0344    .913    13.2      .
           4    CL8                SEATTLE                 3    0.0391    .874    13.9     7.3
           3    CL6                MIAMI                   5    0.0586    .816    15.5     3.8
           2    CL3                CL5                     7    0.1488    .667    16.0     5.3
           1    CL2                CL4                    10    0.6669    .000      .     16.0
```



Example 23.2. Crude Birth and Death Rates

The following example uses the SAS data set Poverty created in the “Getting Started” section beginning on page 958. The data, from Rouncefield (1995), are birth rates, death rates, and infant death rates for 97 countries. Six cluster analyses are performed with eight methods. Scatter plots showing cluster membership at selected levels are produced instead of tree diagrams.

Each cluster analysis is performed by a macro called ANALYZE. The macro takes two arguments. The first, &METHOD, specifies the value of the METHOD= option to be used in the PROC CLUSTER statement. The second, &NCL, must be specified as a list of integers, separated by blanks, indicating the number of clusters desired in each scatter plot. For example, the first invocation of ANALYZE specifies the AVERAGE method and requests plots of 3 and 8 clusters. When two-stage density linkage is used, the K= and R= options are specified as part of the first argument.

The ANALYZE macro first invokes the CLUSTER procedure with METHOD=&METHOD, where &METHOD represents the value of the first argument to ANALYZE. This part of the macro produces the PROC CLUSTER output shown.

The %DO loop processes &NCL, the list of numbers of clusters to plot. The macro variable &K is a counter that indexes the numbers within &NCL. The %SCAN function picks out the &Kth number in &NCL, which is then assigned to the macro variable &N. When &K exceeds the number of numbers in &NCL, %SCAN returns a null string. Thus, the %DO loop executes while &N is not equal to a null string. In the %WHILE condition, a null string is indicated by the absence of any nonblank characters between the comparison operator (NE) and the right parenthesis that terminates the condition.

Within the %DO loop, the TREE procedure creates an output data set containing &N clusters. The GPLOT procedure then produces a scatter plot in which each observation is identified by the number of the cluster to which it belongs. The TITLE2 statement uses double quotes so that &N and &METHOD can be used within the title. At the end of the loop, &K is incremented by 1, and the next number is extracted from &NCL by %SCAN.

For this example, plots are obtained only for average linkage. To generate plots for other methods, follow the example shown in the first macro call. The following statements produce Output 23.2.1 through Output 23.2.7.

```
title ’Cluster Analysis of Birth and Death Rates’;

%macro analyze(method,ncl);
proc cluster data=poverty outtree=tree method=&method p=15 ccc pseudo;
   var birth death;
   title2;
run;
%let k=1;
%let n=%scan(&ncl,&k);
%do %while(&n NE);
   proc tree data=tree noprint out=out ncl=&n;
      copy birth death;
   run;
   legend1 frame cframe=ligr cborder=black
           position=center value=(justify=center);
   axis1 label=(angle=90 rotate=0) minor=none;
   axis2 minor=none;
   proc gplot;
      plot death*birth=cluster /
      frame cframe=ligr legend=legend1 vaxis=axis1 haxis=axis2;
      title2 "Plot of &n Clusters from METHOD=&METHOD";
   run;
   %let k=%eval(&k+1);
   %let n=%scan(&ncl,&k);
```

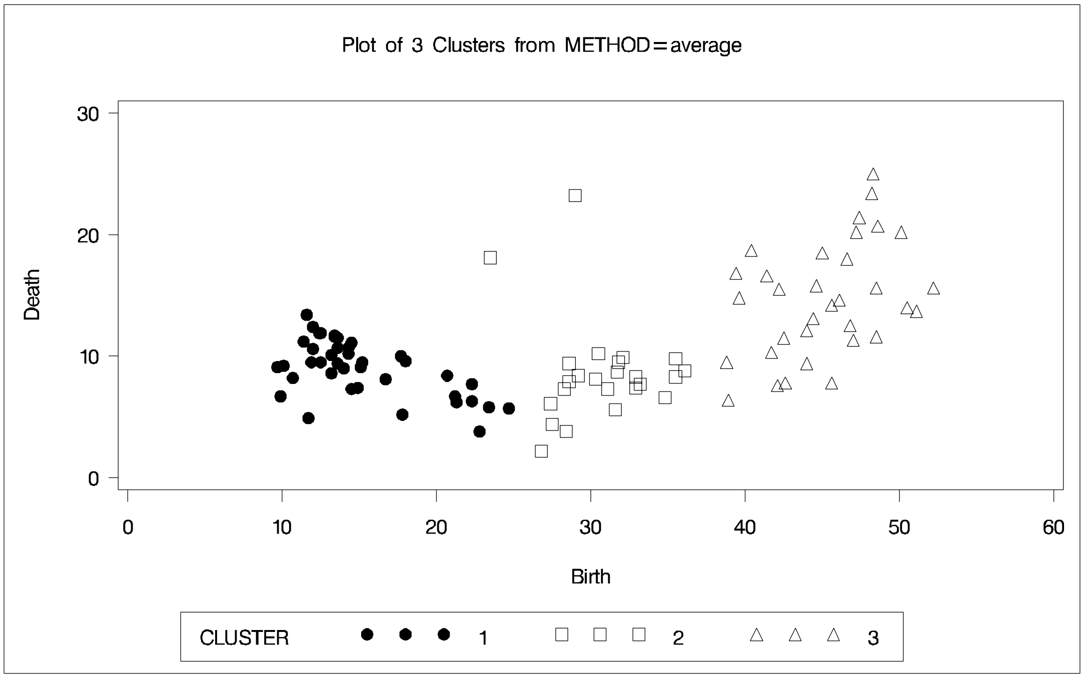
```
%end;
%mend;

%analyze(average,3 8)
%analyze(complete,3)
%analyze(single,7 10)
%analyze(two k=10,3)
%analyze(two k=18,2)
```

For average linkage, the CCC has peaks at 3, 8, 10, and 12 clusters, but the 3-cluster peak is lower than the 8-cluster peak. The pseudo F statistic has peaks at 3, 8, and 12 clusters. The pseudo t^2 statistic drops sharply at 3 clusters, continues to fall at 4 clusters, and has a particularly low value at 12 clusters. However, there are not enough data to seriously consider as many as 12 clusters. Scatter plots are given for 3 and 8 clusters. The results are shown in Output 23.2.1 through Output 23.2.2. In Output 23.2.2, the eighth cluster consists of the two outlying observations, Mexico and Korea.

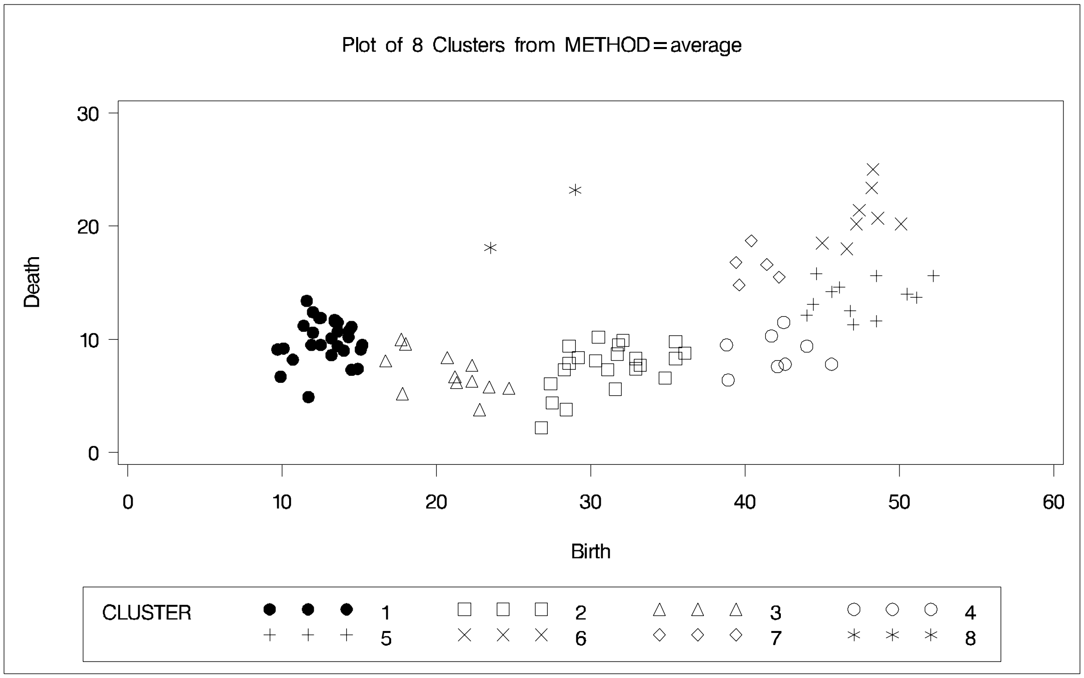
Output 23.2.1. Clusters for Birth and Death Rates: METHOD=AVERAGE

```
                             Cluster Analysis of Birth and Death Rates

                                       The CLUSTER Procedure
                                  Average Linkage Cluster Analysis

                                Eigenvalues of the Covariance Matrix

                            Eigenvalue    Difference    Proportion    Cumulative

                       1    189.106588    173.101020        0.9220        0.9220
                       2     16.005568                      0.0780        1.0000


                    Root-Mean-Square Total-Sample Standard Deviation =   10.127
                    Root-Mean-Square Distance Between Observations   = 20.25399


                                          Cluster History
                                                                                             Norm    T
                                                                                              RMS    i
NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2      Dist    e

 15    CL27        CL20            18    0.0035    .980    .975    2.61     292    18.6    0.2325
 14    CL23        CL17            28    0.0034    .977    .972    1.97     271    17.7    0.2358
 13    CL18        CL54             8    0.0015    .975    .969    2.35     279     7.1    0.2432
 12    CL21        CL26             8    0.0015    .974    .966    2.85     290     6.1    0.2493
 11    CL19        CL24            12    0.0033    .971    .962    2.78     285    14.8    0.2767
 10    CL22        CL16            12    0.0036    .967    .957    2.84     284    17.4    0.2858
  9    CL15        CL28            22    0.0061    .961    .951    2.45     271    17.5    0.3353
  8    OB23        OB61             2    0.0014    .960    .943    3.59     302      .     0.3703
  7    CL25        CL11            17    0.0098    .950    .933    3.01     284    23.3    0.4033
  6    CL7         CL12            25    0.0122    .938    .920    2.63     273    14.8    0.4132
  5    CL10        CL14            40    0.0303    .907    .902    0.59     225    82.7    0.4584
  4    CL13        CL6             33    0.0244    .883    .875    0.77     234    22.2    0.5194
  3    CL9         CL8             24    0.0182    .865    .827    2.13     300    27.7     0.735
  2    CL5         CL3             64    0.1836    .681    .697    -.55     203     148    0.8402
  1    CL2         CL4             97    0.6810    .000    .000    0.00      .      203    1.3348
```

Output 23.2.2. Plot of Three Clusters, METHOD=AVERAGE



Output 23.2.3. Plot of Eight Clusters, METHOD=AVERAGE

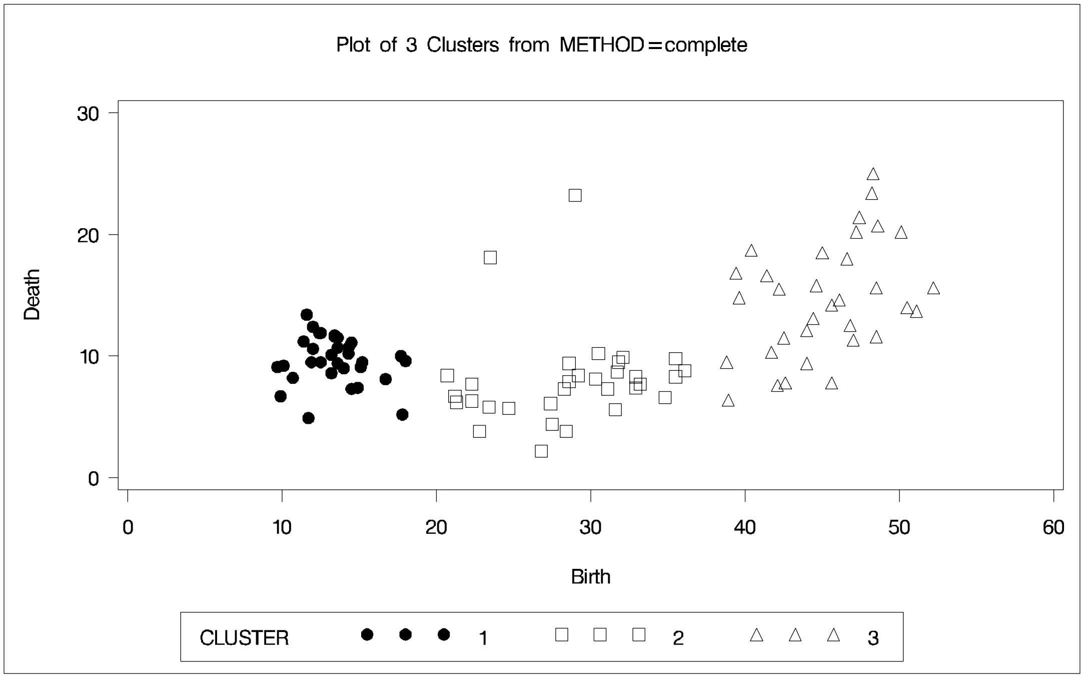


Complete linkage shows CCC peaks at 3, 8 and 12 clusters. The pseudo F statistic peaks at 3 and 12 clusters. The pseudo t^2 statistic indicates 3 clusters.

The scatter plot for 3 clusters is shown. The results are shown in Output 23.2.4.

Output 23.2.4. Clusters for Birth and Death Rates: METHOD=COMPLETE

```
                         Cluster Analysis of Birth and Death Rates

                                  The CLUSTER Procedure
                              Complete Linkage Cluster Analysis

                             Eigenvalues of the Covariance Matrix

                         Eigenvalue    Difference    Proportion    Cumulative

                    1    189.106588    173.101020        0.9220        0.9220
                    2     16.005568                      0.0780        1.0000

                 Root-Mean-Square Total-Sample Standard Deviation =   10.127
                 Mean Distance Between Observations               = 17.13099


                                       Cluster History
                                                                                    Norm    T
                                                                                     Max    i
NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2     Dist    e

 15    CL22        CL33             8    0.0015    .983    .975    3.80     329     6.1   0.4092
 14    CL56        CL18             8    0.0014    .981    .972    3.97     331     6.6   0.4255
 13    CL30        CL44             8    0.0019    .979    .969    4.04     330    19.0   0.4332
 12    OB23        OB61             2    0.0014    .978    .966    4.45     340     .     0.4378
 11    CL19        CL24            24    0.0034    .974    .962    4.17     327    24.1   0.4962
 10    CL17        CL28            12    0.0033    .971    .957    4.18     325    14.8   0.5204
  9    CL20        CL13            16    0.0067    .964    .951    3.38     297    25.2   0.5236
  8    CL11        CL21            32    0.0054    .959    .943    3.44     297    19.7   0.6001
  7    CL26        CL15            13    0.0096    .949    .933    2.93     282    28.9   0.7233
  6    CL14        CL10            20    0.0128    .937    .920    2.46     269    27.7   0.8033
  5    CL9         CL16            30    0.0237    .913    .902    1.29     241    47.1   0.8993
  4    CL6         CL7             33    0.0240    .889    .875    1.38     248    21.7   1.2165
  3    CL5         CL12            32    0.0178    .871    .827    2.56     317    13.6   1.2326
  2    CL3         CL8             64    0.1900    .681    .697    -.55     203     167   1.5412
  1    CL2         CL4             97    0.6810    .000    .000    0.00      .      203   2.5233
```

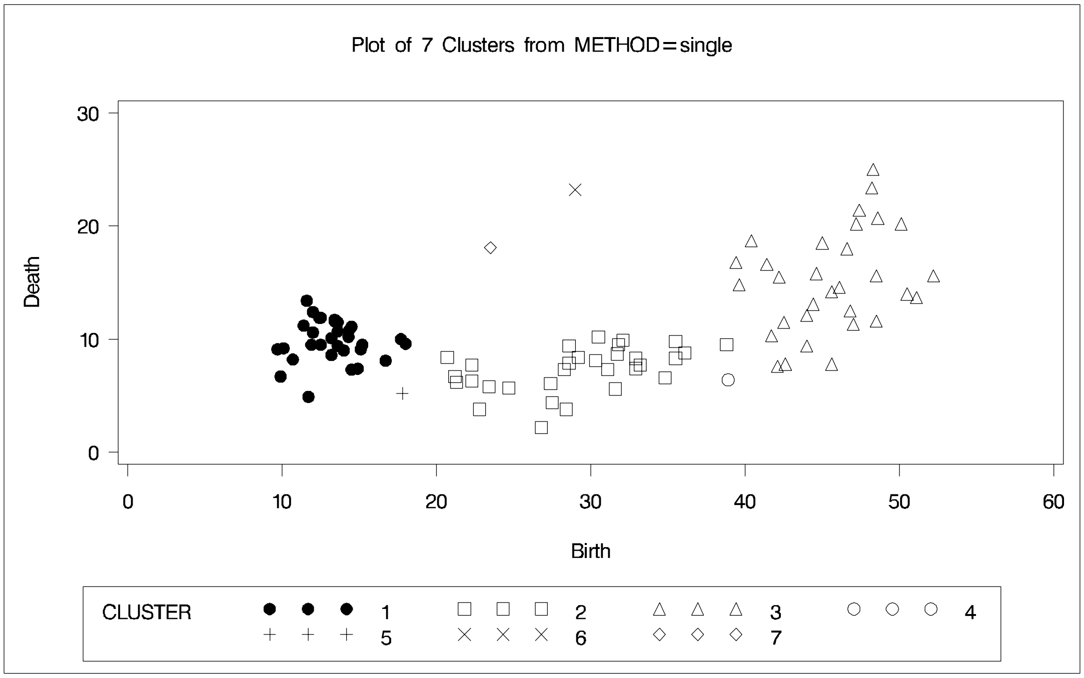


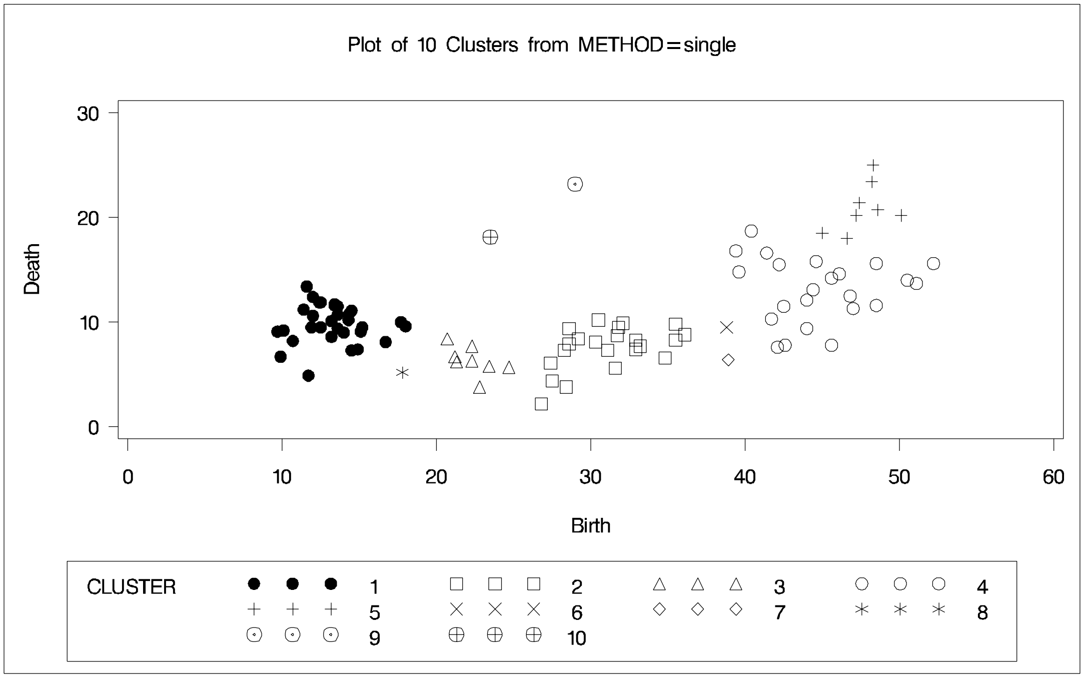
The CCC and pseudo F statistics are not appropriate for use with single linkage because of the method's tendency to chop off tails of distributions. The pseudo t^2 statistic can be used by looking for *large* values and taking the number of clusters to be one greater than the level at which the large pseudo t^2 value is displayed. For these data, there are large values at levels 6 and 9, suggesting 7 or 10 clusters.

The scatter plots for 7 and 10 clusters are shown. The results are shown in Output 23.2.5.

Output 23.2.5. Clusters for Birth and Death Rates: METHOD=SINGLE

```
                              Cluster Analysis of Birth and Death Rates

                                        The CLUSTER Procedure
                                    Single Linkage Cluster Analysis

                                   Eigenvalues of the Covariance Matrix

                              Eigenvalue    Difference    Proportion    Cumulative

                         1    189.106588    173.101020        0.9220        0.9220
                         2     16.005568                      0.0780        1.0000

                        Root-Mean-Square Total-Sample Standard Deviation =   10.127
                        Mean Distance Between Observations              = 17.13099


                                            Cluster History
                                                                                        Norm    T
                                                                                         Min    i
 NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2    Dist    e

  15    CL37        CL19             8    0.0014    .968    .975    -2.3     178     6.6  0.1331
  14    CL20        CL23            15    0.0059    .962    .972    -3.1     162    18.7  0.1412
  13    CL14        CL16            19    0.0054    .957    .969    -3.4     155     8.8  0.1442
  12    CL26        OB58            31    0.0014    .955    .966    -2.7     165     4.0  0.1486
  11    OB86        CL18             4    0.0003    .955    .962    -1.6     183     3.8  0.1495
  10    CL13        CL11            23    0.0088    .946    .957    -2.3     170    11.3  0.1518
   9    CL15        CL10            31    0.0210    .925    .951    -4.4     136    21.8  0.1593    T
   8    CL22        CL17            30    0.0235    .902    .943    -5.8     117    45.7  0.1593
   7    CL8         OB75            31    0.0052    .897    .933    -4.7     130     4.0  0.1628
   6    CL7         CL12            62    0.2023    .694    .920     -15    41.3     223  0.1725
   5    CL6         CL9             93    0.6681    .026    .902     -26     0.6     199  0.1756
   4    CL5         OB48            94    0.0056    .021    .875     -24     0.7     0.5  0.1811    T
   3    CL4         OB67            95    0.0083    .012    .827     -15     0.6     0.8  0.1811
   2    OB23        OB61             2    0.0014    .011    .697     -13     1.0       .  0.4378
   1    CL3         CL2             97    0.0109    .000    .000    0.00       .     1.0  0.5815
```





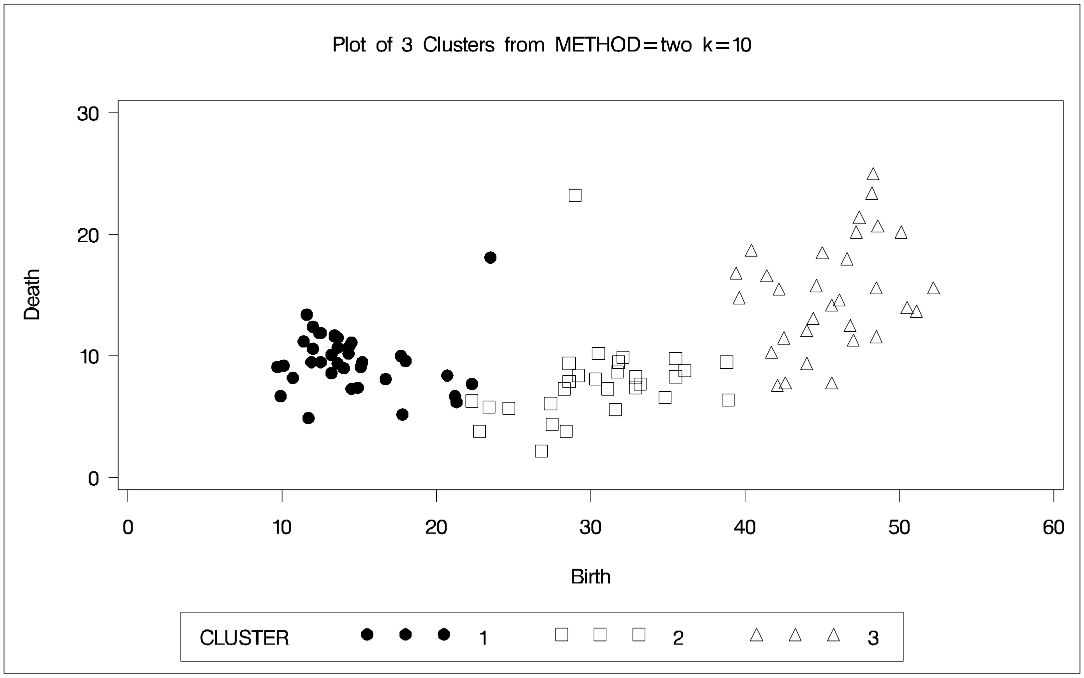
For kth-nearest-neighbor density linkage, the number of modes as a function of k is as follows (not all of these analyses are shown):

k	**modes**
3	13
4	6
5-7	4
8-15	3
16-21	2
22+	1

Thus, there is strong evidence of 3 modes and an indication of the possibility of 2 modes. Uniform-kernel density linkage gives similar results. For K=10 (10th-nearest-neighbor density linkage), the scatter plot for 3 clusters is shown; and for K=18, the scatter plot for 2 clusters is shown. The results are shown in Output 23.2.6.

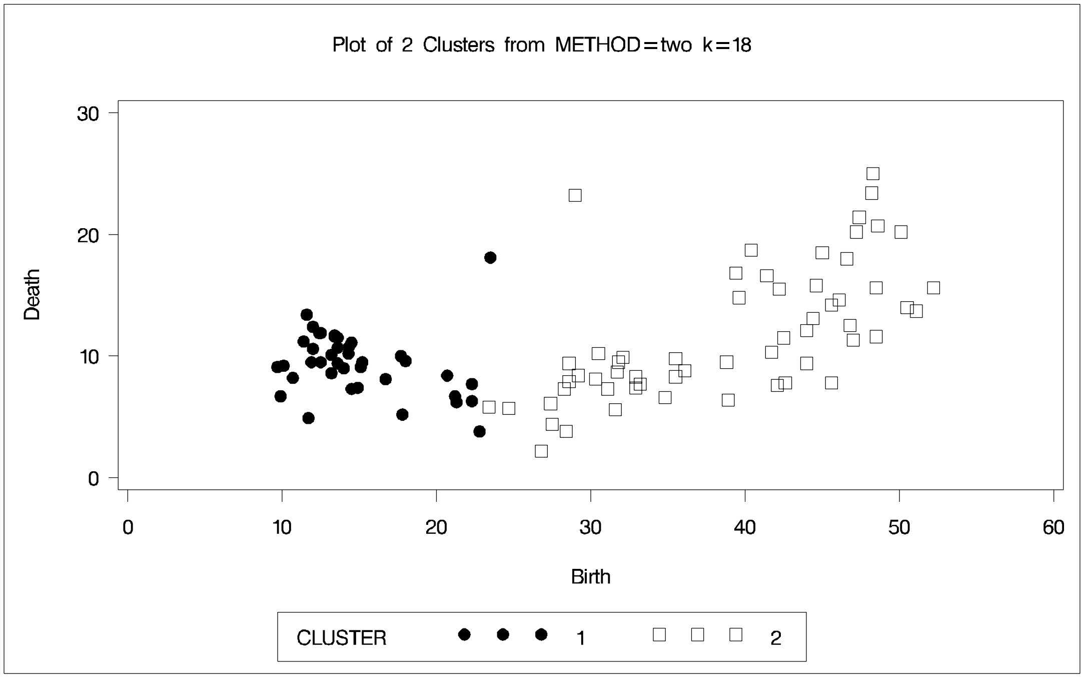
Output 23.2.6. Clusters for Birth and Death Rates: METHOD=TWOSTAGE, K=10

```
                                    Cluster Analysis of Birth and Death Rates

                                              The CLUSTER Procedure
                                       Two-Stage Density Linkage Clustering

                                       Eigenvalues of the Covariance Matrix

                                   Eigenvalue    Difference    Proportion    Cumulative

                              1    189.106588    173.101020        0.9220        0.9220
                              2     16.005568                      0.0780        1.0000
                                                      K = 10
                           Root-Mean-Square Total-Sample Standard Deviation =   10.127


                                                  Cluster History
                                                                                 Normalized     Maximum Density     T
                                                                                     Fusion     in Each Cluster     i
NCL   --Clusters Joined--     FREQ    SPRSQ    RSQ   ERSQ    CCC    PSF   PST2      Density    Lesser     Greater   e

 15   CL16       OB94           22   0.0015   .921   .975    -11   68.4    1.4       9.2234    6.7927     15.3069
 14   CL19       OB49           28   0.0021   .919   .972    -11   72.4    1.8       8.7369    5.9334     33.4385
 13   CL15       OB52           23   0.0024   .917   .969    -10   76.9    2.3       8.5847    5.9651     15.3069
 12   CL13       OB96           24   0.0018   .915   .966   -9.3   83.0    1.6       7.9252    5.4724     15.3069
 11   CL12       OB93           25   0.0025   .912   .962   -8.5   89.5    2.2       7.8913    5.4401     15.3069
 10   CL11       OB78           26   0.0031   .909   .957   -7.7   96.9    2.5        7.787    5.4082     15.3069
  9   CL10       OB76           27   0.0026   .907   .951   -6.7    107    2.1       7.7133    5.4401     15.3069
  8   CL9        OB77           28   0.0023   .904   .943   -5.5    120    1.7       7.4256    4.9017     15.3069
  7   CL8        OB43           29   0.0022   .902   .933   -4.1    138    1.6        6.927    4.4764     15.3069
  6   CL7        OB87           30   0.0043   .898   .920   -2.7    160    3.1        4.932    2.9977     15.3069
  5   CL6        OB82           31   0.0055   .892   .902   -1.1    191    3.7       3.7331    2.1560     15.3069
  4   CL22       OB61           37   0.0079   .884   .875   0.93    237   10.6       3.1713    1.6308       100.0
  3   CL14       OB23           29   0.0126   .872   .827   2.60    320   10.4       2.0654    1.0744     33.4385
  2   CL4        CL3            66   0.2129   .659   .697   -1.3    183    172       12.409   33.4385       100.0
  1   CL2        CL5            97   0.6588   .000   .000   0.00     .     183       10.071   15.3069       100.0
                                        3 modal clusters have been formed.
```



Output 23.2.7. Clusters for Birth and Death Rates: METHOD=TWOSTAGE, K=18

```
                              Cluster Analysis of Birth and Death Rates

                                       The CLUSTER Procedure
                                 Two-Stage Density Linkage Clustering

                                 Eigenvalues of the Covariance Matrix

                             Eigenvalue    Difference    Proportion    Cumulative

                        1    189.106588    173.101020        0.9220        0.9220
                        2     16.005568                      0.0780        1.0000
                                              K = 18
                     Root-Mean-Square Total-Sample Standard Deviation =   10.127


                                          Cluster History
                                                                                 Normalized     Maximum Density     T
                                                                                     Fusion     in Each Cluster     i
 NCL    --Clusters Joined--      FREQ    SPRSQ     RSQ    ERSQ     CCC     PSF    PST2    Density     Lesser     Greater     e

  15    CL16       OB72            46   0.0107    .799    .975     -21    23.3     3.0      10.118     7.7445     23.4457
  14    CL15       OB94            47   0.0098    .789    .972     -21    23.9     2.7       9.676     7.1257     23.4457
  13    CL14       OB51            48   0.0037    .786    .969     -20    25.6     1.0       9.409     6.8398     23.4457    T
  12    CL13       OB96            49   0.0099    .776    .966     -19    26.7     2.6       9.409     6.8398     23.4457
  11    CL12       OB76            50   0.0114    .764    .962     -19    27.9     2.9      8.8136     6.3138     23.4457
  10    CL11       OB77            51   0.0021    .762    .957     -18    31.0     0.5      8.6593     6.0751     23.4457
   9    CL10       OB78            52   0.0103    .752    .951     -17    33.3     2.5      8.6007     6.0976     23.4457
   8    CL9        OB43            53   0.0034    .748    .943     -16    37.8     0.8      8.4964     5.9160     23.4457
   7    CL8        OB93            54   0.0109    .737    .933     -15    42.1     2.6       8.367     5.7913     23.4457
   6    CL7        OB88            55   0.0110    .726    .920     -13    48.3     2.6       7.916     5.3679     23.4457
   5    CL6        OB87            56   0.0120    .714    .902     -12    57.5     2.7      6.6917     4.3415     23.4457
   4    CL20       OB61            39   0.0077    .707    .875    -9.8    74.7     8.3      6.2578     3.2882       100.0
   3    CL5        OB82            57   0.0138    .693    .827    -5.0     106     3.0      5.3605     3.2834     23.4457
   2    CL3        OB23            58   0.0117    .681    .697    -.54     203     2.5      3.2687     1.7568     23.4457
   1    CL2        CL4             97   0.6812    .000    .000    0.00       .     203      13.764    23.4457       100.0
                                  2 modal clusters have been formed.
```



In summary, most of the clustering methods indicate 3 or 8 clusters. Most methods agree at the 3-cluster level, but at the other levels, there is considerable disagreement about the composition of the clusters. The presence of numerous ties also complicates the analysis; see Example 23.4 on page 1027.

Example 23.3. Cluster Analysis of Fisher Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analyses of the iris data.

This example analyzes the iris data by Ward’s method and two-stage density linkage and then illustrates how the FASTCLUS procedure can be used in combination with PROC CLUSTER to analyze large data sets.

```
title ’Cluster Analysis of Fisher (1936) Iris Data’;
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
```

```
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
```

The following macro, SHOW, is used in the subsequent analyses to display cluster results. It invokes the FREQ procedure to crosstabulate clusters and species. The CANDISC procedure computes canonical variables for discriminating among the clusters, and the first two canonical variables are plotted to show cluster membership. See Chapter 21, "The CANDISC Procedure," for a canonical discriminant analysis of the iris species.

```
%macro show;
```

```
proc freq;
   tables cluster*species;
run;
proc candisc noprint out=can;
   class cluster;
   var petal: sepal:;
run;
legend1 frame cframe=ligr cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot;
   plot can2*can1=cluster /
      frame cframe=ligr legend=legend1 vaxis=axis1 haxis=axis2;
run;
%mend;
```

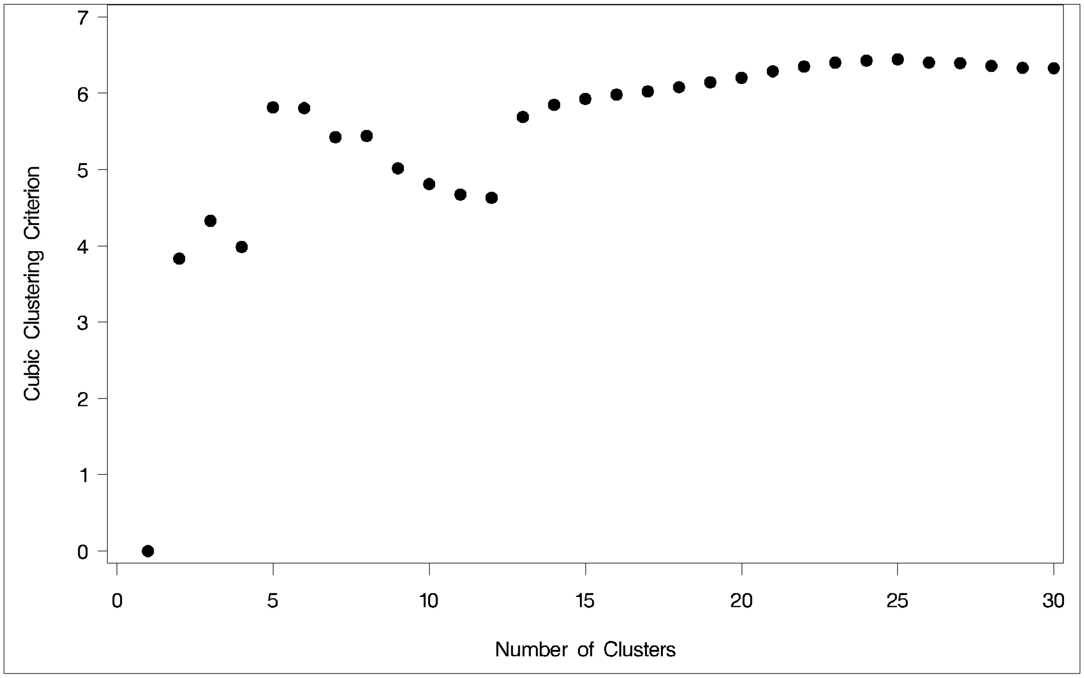
The first analysis clusters the iris data by Ward's method and plots the CCC and pseudo F and t^2 statistics. The CCC has a local peak at 3 clusters but a higher peak at 5 clusters. The pseudo F statistic indicates 3 clusters, while the pseudo t^2 statistic suggests 3 or 6 clusters. For large numbers of clusters, Version 6 of the SAS System produces somewhat different results than previous versions of PROC CLUSTER. This is due to changes in the treatment of ties. Results are identical for 5 or fewer clusters.

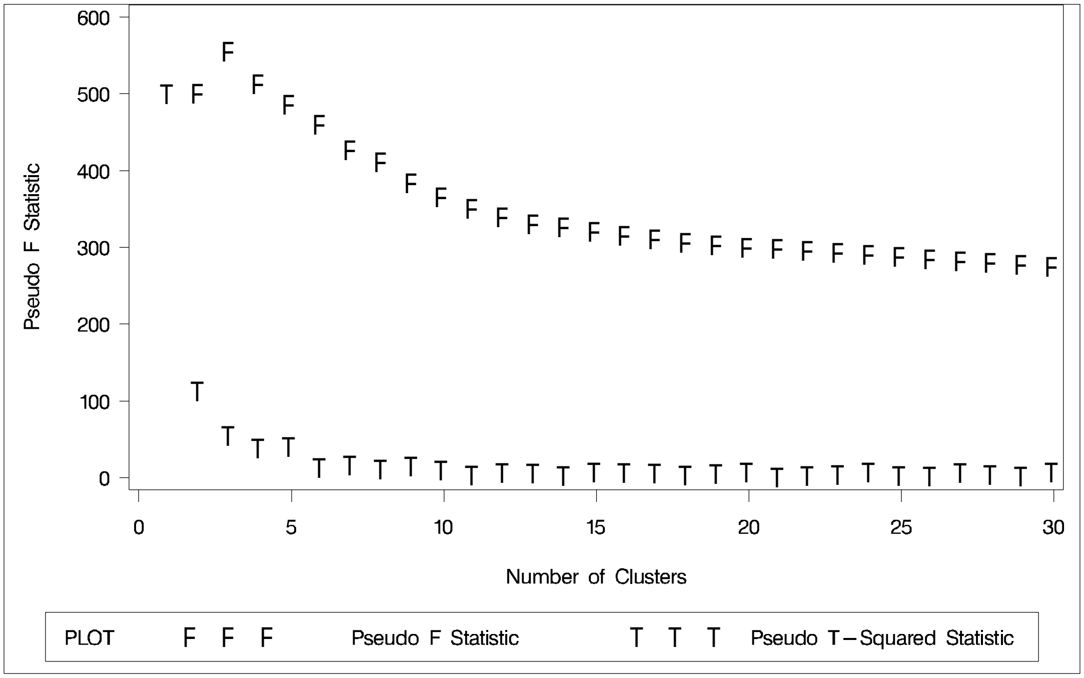
The TREE procedure creates an output data set containing the 3-cluster partition for use by the SHOW macro. The FREQ procedure reveals 16 misclassifications. The results are shown in Output 23.3.1.

```
title2 'By Ward''s Method';
proc cluster data=iris method=ward print=15 ccc pseudo;
   var petal: sepal:;
   copy species;
run;
legend1 frame cframe=ligr cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none order=(0 to 600 by 100);
axis2 minor=none order=(1 to 30 by 1);
axis3 label=(angle=90 rotate=0) minor=none order=(0 to 7 by 1);
proc gplot;
   plot _ccc_*_ncl_  /
      frame cframe=ligr legend=legend1 vaxis=axis3 haxis=axis2;
   plot _psf_*_ncl_  _pst2_*_ncl_  /overlay
      frame cframe=ligr legend=legend1 vaxis=axis1 haxis=axis2;
run;

proc tree noprint ncl=3 out=out;
   copy petal: sepal: species;
run;

%show;
```

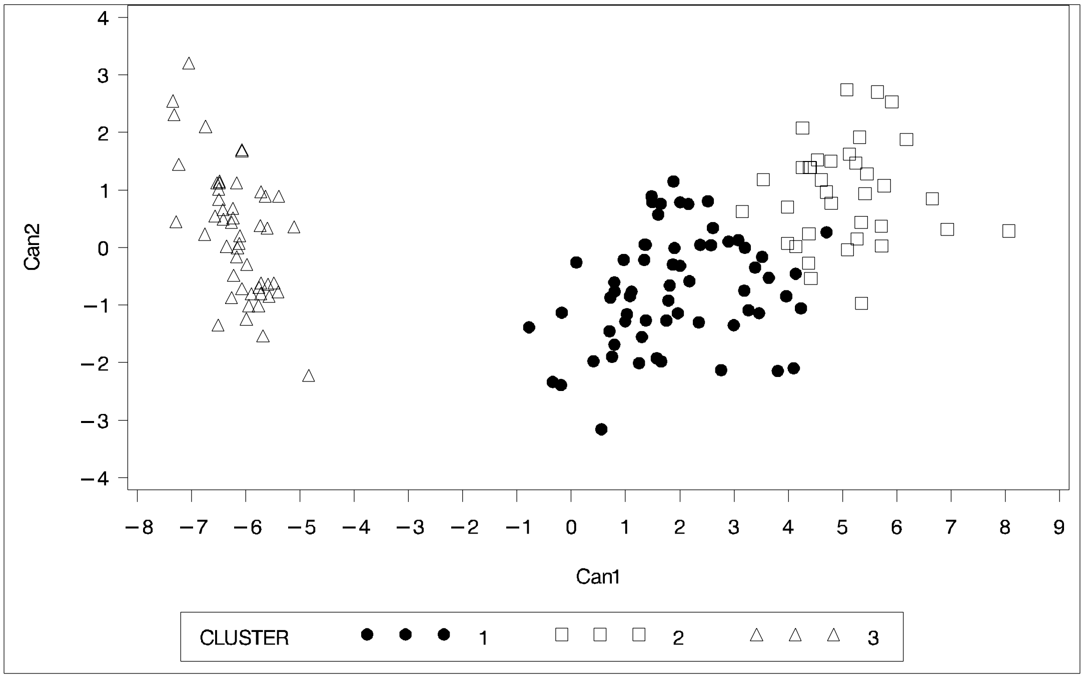
Output 23.3.1. Cluster Analysis of Fisher Iris Data:CLUSTER with METHOD=WARD

```
                         Cluster Analysis of Fisher (1936) Iris Data
                                      By Ward's Method

                                   The CLUSTER Procedure
                          Ward's Minimum Variance Cluster Analysis

                             Eigenvalues of the Covariance Matrix

                         Eigenvalue    Difference    Proportion    Cumulative

                    1    422.824171    398.557096        0.9246        0.9246
                    2     24.267075     16.446125        0.0531        0.9777
                    3      7.820950      5.437441        0.0171        0.9948
                    4      2.383509                      0.0052        1.0000


                Root-Mean-Square Total-Sample Standard Deviation = 10.69224
                Root-Mean-Square Distance Between Observations   = 30.24221


                                      Cluster History
                                                                                   T
                                                                                   i
        NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2    e

         15    CL24        CL28            15    0.0016    .971    .958    5.93     324     9.8
         14    CL21        CL53             7    0.0019    .969    .955    5.85     329     5.1
         13    CL18        CL48            15    0.0023    .967    .953    5.69     334     8.9
         12    CL16        CL23            24    0.0023    .965    .950    4.63     342     9.6
         11    CL14        CL43            12    0.0025    .962    .946    4.67     353     5.8
         10    CL26        CL20            22    0.0027    .959    .942    4.81     368    12.9
          9    CL27        CL17            31    0.0031    .956    .936    5.02     387    17.8
          8    CL35        CL15            23    0.0031    .953    .930    5.44     414    13.8
          7    CL10        CL47            26    0.0058    .947    .921    5.43     430    19.1
          6    CL8         CL13            38    0.0060    .941    .911    5.81     463    16.3
          5    CL9         CL19            50    0.0105    .931    .895    5.82     488    43.2
          4    CL12        CL11            36    0.0172    .914    .872    3.99     515    41.0
          3    CL6         CL7             64    0.0301    .884    .827    4.33     558    57.2
          2    CL4         CL3            100    0.1110    .773    .697    3.83     503     116
          1    CL5         CL2            150    0.7726    .000    .000    0.00       .     503
```


Cubic Clustering Criterion
0
1
2
3
4
5
6
7
Number of Clusters
0
5
10
15
20
25
30


Pseudo F Statistic
0
100
200
300
400
500
600
Number of Clusters
0
5
10
15
20
25
30
PLOT F F F Pseudo F Statistic T T T Pseudo T−Squared Statistic

```
                Cluster Analysis of Fisher (1936) Iris Data

                             The FREQ Procedure

                        Table of CLUSTER by Species

                CLUSTER     Species

                Frequency|
                Percent  |
                Row Pct  |
                Col Pct  |Setosa  |Versicol|Virginic|  Total
                         |        |or      |a       |
                ---------+--------+--------+--------+
                       1 |      0 |     49 |     15 |     64
                         |   0.00 |  32.67 |  10.00 |  42.67
                         |   0.00 |  76.56 |  23.44 |
                         |   0.00 |  98.00 |  30.00 |
                ---------+--------+--------+--------+
                       2 |      0 |      1 |     35 |     36
                         |   0.00 |   0.67 |  23.33 |  24.00
                         |   0.00 |   2.78 |  97.22 |
                         |   0.00 |   2.00 |  70.00 |
                ---------+--------+--------+--------+
                       3 |     50 |      0 |      0 |     50
                         |  33.33 |   0.00 |   0.00 |  33.33
                         | 100.00 |   0.00 |   0.00 |
                         | 100.00 |   0.00 |   0.00 |
                ---------+--------+--------+--------+
                Total          50       50       50      150
                            33.33    33.33    33.33   100.00
```



The second analysis uses two-stage density linkage. The raw data suggest 2 or 6 modes instead of 3:

k	**modes**
3	12
4-6	6
7	4
8	3
9-50	2
51+	1

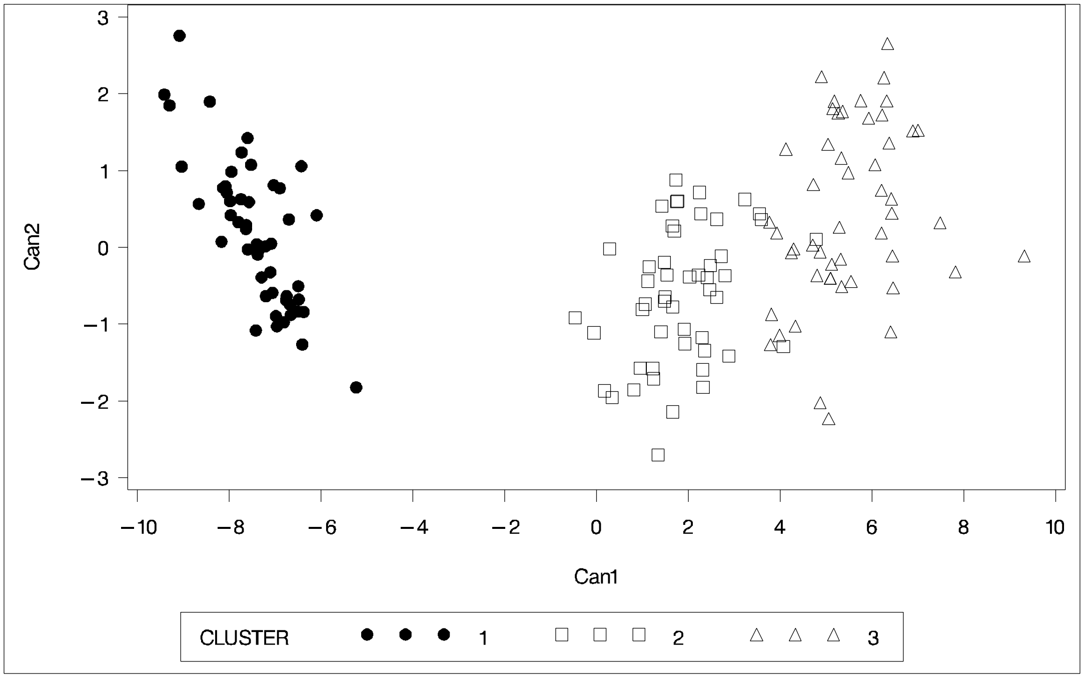
However, the ACECLUS procedure can be used to reveal 3 modes. This analysis uses K=8 to produce 3 clusters for comparison with other analyses. There are only 6 misclassifications. The results are shown in Output 23.3.2.

```
title2 'By Two-Stage Density Linkage';
proc cluster data=iris method=twostage k=8 print=15 ccc pseudo;
   var petal: sepal:;
   copy species;
run;

proc tree noprint ncl=3 out=out;
   copy petal: sepal: species;
run;

%show;
```

Output 23.3.2. Cluster Analysis of Fisher Iris Data: CLUSTER with METHOD=TWOSTAGE

```
                                      Cluster Analysis of Fisher (1936) Iris Data
                                              By Two-Stage Density Linkage

                                                 The CLUSTER Procedure
                                          Two-Stage Density Linkage Clustering

                                          Eigenvalues of the Covariance Matrix

                                      Eigenvalue    Difference    Proportion    Cumulative

                                 1    422.824171    398.557096        0.9246        0.9246
                                 2     24.267075     16.446125        0.0531        0.9777
                                 3      7.820950      5.437441        0.0171        0.9948
                                 4      2.383509                      0.0052        1.0000
                                                         K = 8
                              Root-Mean-Square Total-Sample Standard Deviation = 10.69224


                                                     Cluster History
                                                                                    Normalized     Maximum Density     T
                                                                                        Fusion     in Each Cluster     i
   NCL   --Clusters Joined--     FREQ    SPRSQ    RSQ   ERSQ    CCC    PSF   PST2      Density     Lesser     Greater   e

    15   CL17       OB127         44   0.0025   .916   .958    -11    105    3.4       0.3903     0.2066     3.5156
    14   CL16       OB137         50   0.0023   .913   .955    -11    110    5.6       0.3637     0.1837      100.0
    13   CL15       OB74          45   0.0029   .910   .953    -10    116    3.7       0.3553     0.2130     3.5156
    12   CL28       OB49          46   0.0036   .907   .950   -8.0    122    5.2       0.3223     0.1736     8.3678   T
    11   CL12       OB85          47   0.0036   .903   .946   -7.6    130    4.8       0.3223     0.1736     8.3678
    10   CL11       OB98          48   0.0033   .900   .942   -7.1    140    4.1       0.2879     0.1479     8.3678
     9   CL13       OB24          46   0.0037   .896   .936   -6.5    152    4.4       0.2802     0.2005     3.5156
     8   CL10       OB25          49   0.0019   .894   .930   -5.5    171    2.2       0.2699     0.1372     8.3678
     7   CL8        OB121         50   0.0035   .891   .921   -4.5    194    4.0       0.2586     0.1372     8.3678
     6   CL9        OB45          47   0.0042   .886   .911   -3.3    225    4.6       0.1412     0.0832     3.5156
     5   CL6        OB39          48   0.0049   .882   .895   -1.7    270    5.0        0.107     0.0605     3.5156
     4   CL5        OB21          49   0.0049   .877   .872   0.35    346    4.7       0.0969     0.0541     3.5156
     3   CL4        OB90          50   0.0047   .872   .827   3.28    500    4.1       0.0715     0.0370     3.5156
     2   CL3        CL7          100   0.0993   .773   .697   3.83    503   91.9       2.6277     3.5156     8.3678
                                           3 modal clusters have been formed.
```

```
                        Cluster Analysis of Fisher (1936) Iris Data

                                   The FREQ Procedure

                                Table of CLUSTER by Species

                        CLUSTER     Species

                        Frequency|
                        Percent  |
                        Row Pct  |
                        Col Pct  |Setosa  |Versicol|Virginic|  Total
                                 |        |or      |a       |
                        ---------+--------+--------+--------+
                               1 |     50 |      0 |      0 |     50
                                 |  33.33 |   0.00 |   0.00 |  33.33
                                 | 100.00 |   0.00 |   0.00 |
                                 | 100.00 |   0.00 |   0.00 |
                        ---------+--------+--------+--------+
                               2 |      0 |     47 |      3 |     50
                                 |   0.00 |  31.33 |   2.00 |  33.33
                                 |   0.00 |  94.00 |   6.00 |
                                 |   0.00 |  94.00 |   6.00 |
                        ---------+--------+--------+--------+
                               3 |      0 |      3 |     47 |     50
                                 |   0.00 |   2.00 |  31.33 |  33.33
                                 |   0.00 |   6.00 |  94.00 |
                                 |   0.00 |   6.00 |  94.00 |
                        ---------+--------+--------+--------+
                        Total          50       50       50      150
                                    33.33    33.33    33.33   100.00
```



The CLUSTER procedure is not practical for very large data sets because, with most methods, the CPU time varies as the square or cube of the number of observations. The FASTCLUS procedure requires time proportional to the number of observations and can, therefore, be used with much larger data sets than PROC CLUSTER. If you want to hierarchically cluster a very large data set, you can use PROC FASTCLUS for a preliminary cluster analysis producing a large number of clusters and then use PROC CLUSTER to hierarchically cluster the preliminary clusters.

FASTCLUS automatically creates variables _FREQ_ and _RMSSTD_ in the MEAN= output data set. These variables are then automatically used by PROC CLUSTER in the computation of various statistics.

The iris data are used to illustrate the process of clustering clusters. In the preliminary analysis, PROC FASTCLUS produces ten clusters, which are then crosstabulated with species. The data set containing the preliminary clusters is sorted in preparation for later merges. The results are shown in Output 23.3.3.

```
title2 'Preliminary Analysis by FASTCLUS';
proc fastclus data=iris summary maxc=10 maxiter=99 converge=0
              mean=mean out=prelim cluster=preclus;
   var petal: sepal:;
run;

proc freq;
   tables preclus*species;
run;

proc sort data=prelim;
   by preclus;
run;
```

Output 23.3.3. Preliminary Analysis of Fisher Iris Data

```
                          Cluster Analysis of Fisher (1936) Iris Data
                                Preliminary Analysis by FASTCLUS

                                    The FASTCLUS Procedure
                   Replace=FULL  Radius=0  Maxclusters=10 Maxiter=99  Converge=0

                                       Cluster Summary

                                       Maximum Distance
                            RMS Std           from Seed     Radius     Nearest     Distance Between
Cluster     Frequency     Deviation      to Observation     Exceeded    Cluster    Cluster Centroids
-------------------------------------------------------------------------------------------------
   1               9         2.7067              8.2027                      5               8.7362
   2              19         2.2001              7.7340                      4               6.2243
   3              18         2.1496              6.2173                      8               7.5049
   4               4         2.5249              5.3268                      2               6.2243
   5               3         2.7234              5.8214                      1               8.7362
   6               7         2.2939              5.1508                      2               9.3318
   7              17         2.0274              6.9576                     10               7.9503
   8              18         2.2628              7.1135                      3               7.5049
   9              22         2.2666              7.5029                      8               9.0090
  10              33         2.0594             10.0033                      7               7.9503


                               Pseudo F Statistic =   370.58


                           Observed Over-All R-Squared =  0.95971


                    Approximate Expected Over-All R-Squared =   0.82928


                           Cubic Clustering Criterion =   27.077

                WARNING: The two values above are invalid for correlated variables.
```

```
Cluster Analysis of Fisher (1936) Iris Data
      Preliminary Analysis by FASTCLUS

             The FREQ Procedure

         Table of preclus by Species

preclus(Cluster)     Species

Frequency|
Percent  |
Row Pct  |
Col Pct  |Setosa  |Versicol|Virginic|  Total
         |        |or      |a       |
---------+--------+--------+--------+
       1 |      0 |      0 |      9 |      9
         |   0.00 |   0.00 |   6.00 |   6.00
         |   0.00 |   0.00 | 100.00 |
         |   0.00 |   0.00 |  18.00 |
---------+--------+--------+--------+
       2 |      0 |     19 |      0 |     19
         |   0.00 |  12.67 |   0.00 |  12.67
         |   0.00 | 100.00 |   0.00 |
         |   0.00 |  38.00 |   0.00 |
---------+--------+--------+--------+
       3 |      0 |     18 |      0 |     18
         |   0.00 |  12.00 |   0.00 |  12.00
         |   0.00 | 100.00 |   0.00 |
         |   0.00 |  36.00 |   0.00 |
---------+--------+--------+--------+
       4 |      0 |      3 |      1 |      4
         |   0.00 |   2.00 |   0.67 |   2.67
         |   0.00 |  75.00 |  25.00 |
         |   0.00 |   6.00 |   2.00 |
---------+--------+--------+--------+
       5 |      0 |      0 |      3 |      3
         |   0.00 |   0.00 |   2.00 |   2.00
         |   0.00 |   0.00 | 100.00 |
         |   0.00 |   0.00 |   6.00 |
---------+--------+--------+--------+
       6 |      0 |      7 |      0 |      7
         |   0.00 |   4.67 |   0.00 |   4.67
         |   0.00 | 100.00 |   0.00 |
         |   0.00 |  14.00 |   0.00 |
---------+--------+--------+--------+
       7 |     17 |      0 |      0 |     17
         |  11.33 |   0.00 |   0.00 |  11.33
         | 100.00 |   0.00 |   0.00 |
         |  34.00 |   0.00 |   0.00 |
---------+--------+--------+--------+
       8 |      0 |      3 |     15 |     18
         |   0.00 |   2.00 |  10.00 |  12.00
         |   0.00 |  16.67 |  83.33 |
         |   0.00 |   6.00 |  30.00 |
---------+--------+--------+--------+
       9 |      0 |      0 |     22 |     22
         |   0.00 |   0.00 |  14.67 |  14.67
         |   0.00 |   0.00 | 100.00 |
         |   0.00 |   0.00 |  44.00 |
---------+--------+--------+--------+
      10 |     33 |      0 |      0 |     33
         |  22.00 |   0.00 |   0.00 |  22.00
         | 100.00 |   0.00 |   0.00 |
         |  66.00 |   0.00 |   0.00 |
---------+--------+--------+--------+
Total          50       50       50      150
            33.33    33.33    33.33   100.00
```

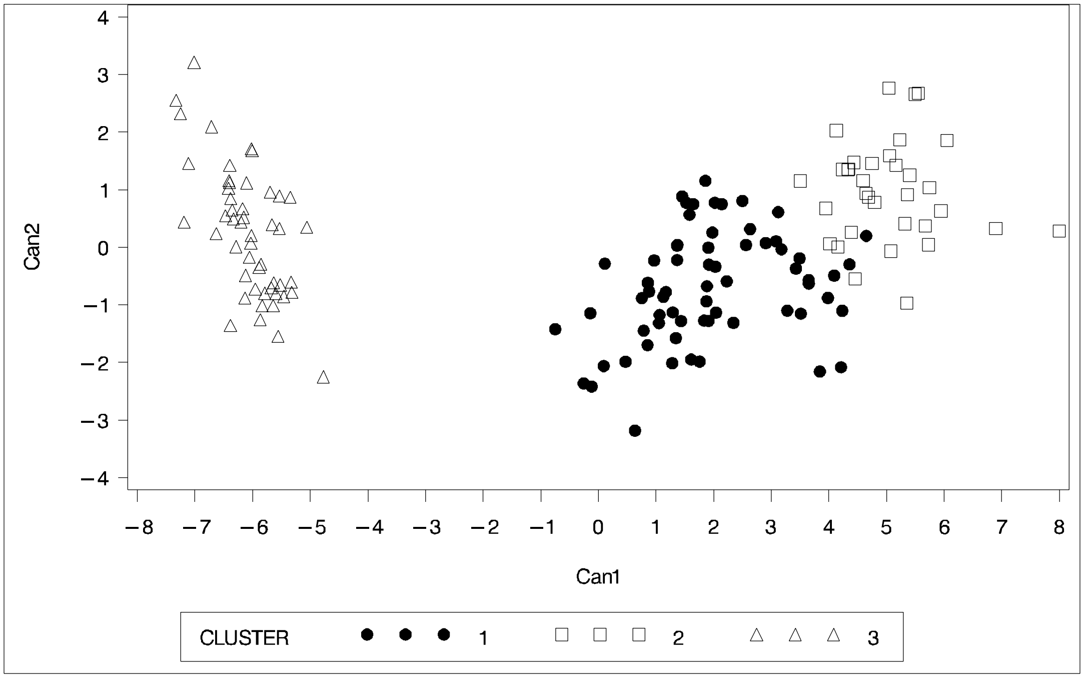
The following macro, CLUS, clusters the preliminary clusters. There is one argument to choose the METHOD= specification to be used by PROC CLUSTER. The TREE procedure creates an output data set containing the 3-cluster partition, which is sorted and merged with the OUT= data set from PROC FASTCLUS to determine to which cluster each of the original 150 observations belongs. The SHOW macro is then used to display the results. In this example, the CLUS macro is invoked using Ward's method, which produces 16 misclassifications, and Wong’s hybrid method, which produces 22 misclassifications. The results are shown in Output 23.3.4 and Output 23.3.5.

```
%macro clus(method);
proc cluster data=mean method=&method ccc pseudo;
   var petal: sepal:;
   copy preclus;
run;
proc tree noprint ncl=3 out=out;
   copy petal: sepal: preclus;
run;
proc sort data=out;
   by preclus;
run;
data clus;
   merge out prelim;
   by preclus;
run;
%show;
%mend;

title2 ’Clustering Clusters by Ward’’s Method’;
%clus(ward);

title2 ’Clustering Clusters by Wong’’s Hybrid Method’;
%clus(twostage hybrid);
```

Output 23.3.4. Clustering Clusters: with Ward's Method

```
                          Cluster Analysis of Fisher (1936) Iris Data
                              Clustering Clusters by Ward's Method

                                     The CLUSTER Procedure
                            Ward's Minimum Variance Cluster Analysis

                              Eigenvalues of the Covariance Matrix

                          Eigenvalue    Difference    Proportion    Cumulative

                     1    416.976349    398.666421        0.9501        0.9501
                     2     18.309928     14.952922        0.0417        0.9918
                     3      3.357006      3.126943        0.0076        0.9995
                     4      0.230063                      0.0005        1.0000


                  Root-Mean-Square Total-Sample Standard Deviation = 10.69224
                  Root-Mean-Square Distance Between Observations   = 30.24221


                                        Cluster History
                                                                                    T
                                                                                    i
  NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2    e

    9    OB2         OB4             23    0.0019    .958    .932    6.26     400     6.3
    8    OB1         OB5             12    0.0025    .955    .926    6.75     434     5.8
    7    CL9         OB6             30    0.0069    .948    .918    6.28     438    19.5
    6    OB3         OB8             36    0.0074    .941    .907    6.21     459    26.0
    5    OB7         OB10            50    0.0104    .931    .892    6.15     485    42.2
    4    CL8         OB9             34    0.0162    .914    .870    4.28     519    39.3
    3    CL7         CL6             66    0.0318    .883    .824    4.39     552    59.7
    2    CL4         CL3            100    0.1099    .773    .695    3.94     503     113
    1    CL2         CL5            150    0.7726    .000    .000    0.00       .     503
```

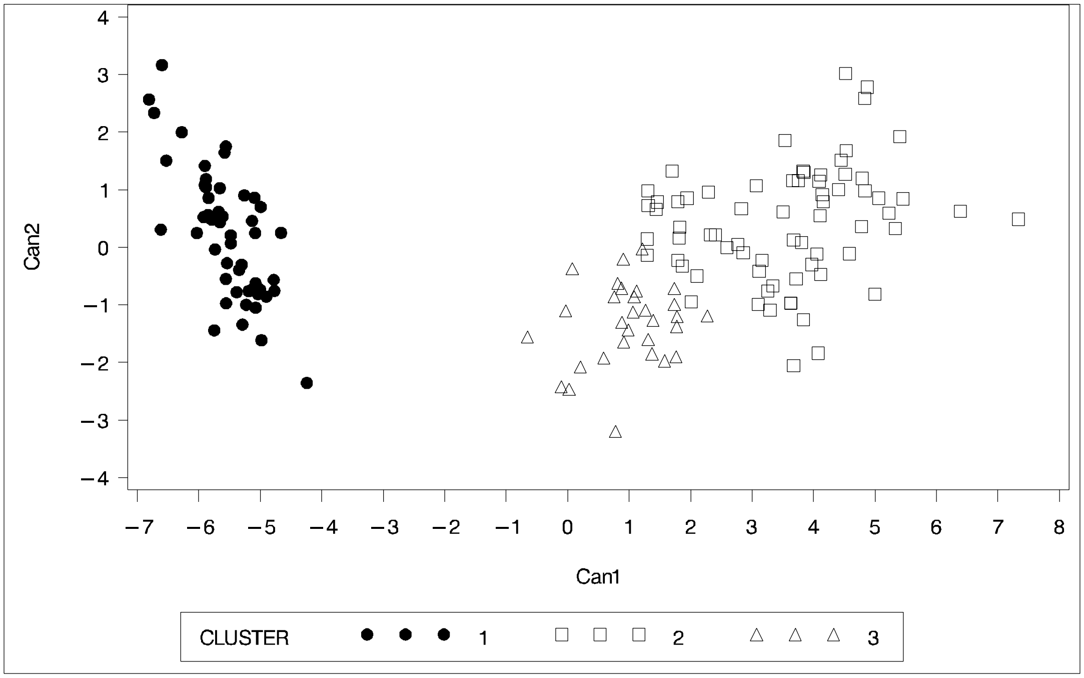
```
                          Cluster Analysis of Fisher (1936) Iris Data

                                      The FREQ Procedure

                                  Table of CLUSTER by Species

                          CLUSTER     Species

                          Frequency|
                          Percent  |
                          Row Pct  |
                          Col Pct  |Setosa  |Versicol|Virginic|  Total
                                   |        |or      |a       |
                          ---------+--------+--------+--------+
                                 1 |      0 |     50 |     16 |     66
                                   |   0.00 |  33.33 |  10.67 |  44.00
                                   |   0.00 |  75.76 |  24.24 |
                                   |   0.00 | 100.00 |  32.00 |
                          ---------+--------+--------+--------+
                                 2 |      0 |      0 |     34 |     34
                                   |   0.00 |   0.00 |  22.67 |  22.67
                                   |   0.00 |   0.00 | 100.00 |
                                   |   0.00 |   0.00 |  68.00 |
                          ---------+--------+--------+--------+
                                 3 |     50 |      0 |      0 |     50
                                   |  33.33 |   0.00 |   0.00 |  33.33
                                   | 100.00 |   0.00 |   0.00 |
                                   | 100.00 |   0.00 |   0.00 |
                          ---------+--------+--------+--------+
                          Total          50       50       50      150
                                      33.33    33.33    33.33   100.00
```



Output 23.3.5. Clustering Clusters: PROC CLUSTER with Wong's Hybrid Method

```
                          Cluster Analysis of Fisher (1936) Iris Data
                          Clustering Clusters by Wong's Hybrid Method

                                     The CLUSTER Procedure
                              Two-Stage Density Linkage Clustering

                              Eigenvalues of the Covariance Matrix

                          Eigenvalue    Difference    Proportion    Cumulative

                     1    416.976349    398.666421        0.9501        0.9501
                     2     18.309928     14.952922        0.0417        0.9918
                     3      3.357006      3.126943        0.0076        0.9995
                     4      0.230063                      0.0005        1.0000
                  Root-Mean-Square Total-Sample Standard Deviation = 10.69224


                                        Cluster History
                                                                         Normalized     Maximum Density     T
                                                                           Fusion     in Each Cluster     i
  NCL   --Clusters Joined--     FREQ    SPRSQ    RSQ   ERSQ    CCC    PSF   PST2     Density     Lesser    Greater   e

    9   OB10       OB7           50   0.0104   .949   .932   3.81   330   42.2       40.24    58.2179      100.0
    8   OB3        OB8           36   0.0074   .942   .926   3.22   329   26.0      27.981    39.4511    48.4350
    7   OB2        OB4           23   0.0019   .940   .918   4.24   373    6.3      23.775     8.9675    46.3026
    6   CL8        OB9           58   0.0194   .921   .907   2.13   334   46.3      20.724    46.8846    48.4350
    5   CL7        OB6           30   0.0069   .914   .892   3.09   383   19.5      13.303    17.6360    46.3026
    4   CL6        OB1           67   0.0292   .884   .870   1.21   372   41.0      8.4137    10.8758    48.4350
    3   CL4        OB5           70   0.0138   .871   .824   3.33   494   12.3      5.1855     6.2890    48.4350
    2   CL3        CL5          100   0.0979   .773   .695   3.94   503   89.5      19.513    46.3026    48.4350
    1   CL2        CL9          150   0.7726   .000   .000   0.00     .    503      1.3337    48.4350      100.0
                                   3 modal clusters have been formed.
```

```
                    Cluster Analysis of Fisher (1936) Iris Data

                                The FREQ Procedure

                           Table of CLUSTER by Species

                    CLUSTER     Species

                    Frequency|
                    Percent  |
                    Row Pct  |
                    Col Pct  |Setosa  |Versicol|Virginic|  Total
                             |        |or      |a       |
                    ---------+--------+--------+--------+
                           1 |     50 |      0 |      0 |     50
                             |  33.33 |   0.00 |   0.00 |  33.33
                             | 100.00 |   0.00 |   0.00 |
                             | 100.00 |   0.00 |   0.00 |
                    ---------+--------+--------+--------+
                           2 |      0 |     21 |     49 |     70
                             |   0.00 |  14.00 |  32.67 |  46.67
                             |   0.00 |  30.00 |  70.00 |
                             |   0.00 |  42.00 |  98.00 |
                    ---------+--------+--------+--------+
                           3 |      0 |     29 |      1 |     30
                             |   0.00 |  19.33 |   0.67 |  20.00
                             |   0.00 |  96.67 |   3.33 |
                             |   0.00 |  58.00 |   2.00 |
                    ---------+--------+--------+--------+
                    Total          50       50       50      150
                                33.33    33.33    33.33   100.00
```



Example 23.4. Evaluating the Effects of Ties

If, at some level of the cluster history, there is a tie for minimum distance between clusters, then one or more levels of the sample cluster tree are not uniquely determined. This example shows how the degree of indeterminacy can be assessed.

Mammals have four kinds of teeth: incisors, canines, premolars, and molars. The following data set gives the number of teeth of each kind on one side of the top and bottom jaws for 32 mammals.

Since all eight variables are measured in the same units, it is not strictly necessary to rescale the data. However, the canines have much less variance than the other kinds of teeth and, therefore, have little effect on the analysis if the variables are not standardized. An average linkage cluster analysis is run with and without standardization to allow comparison of the results. The results are shown in Output 23.4.1 and Output 23.4.2.

```
title ’Hierarchical Cluster Analysis of Mammals’’ Teeth Data’;
title2 ’Evaluating the Effects of Ties’;
data teeth;
   input mammal $ 1-16
         @21 (v1-v8) (1.);
   label v1=’Top incisors’
         v2=’Bottom incisors’
         v3=’Top canines’
         v4=’Bottom canines’
         v5=’Top premolars’
         v6=’Bottom premolars’
         v7=’Top molars’
         v8=’Bottom molars’;
   datalines;
BROWN BAT           23113333
MOLE                32103333
SILVER HAIR BAT     23112333
PIGMY BAT           23112233
HOUSE BAT           23111233
RED BAT             13112233
PIKA                21002233
RABBIT              21003233
BEAVER              11002133
GROUNDHOG           11002133
GRAY SQUIRREL       11001133
HOUSE MOUSE         11000033
PORCUPINE           11001133
WOLF                33114423
BEAR                33114423
RACCOON             33114432
MARTEN              33114412
WEASEL              33113312
WOLVERINE           33114412
BADGER              33113312
RIVER OTTER         33114312
SEA OTTER           32113312
JAGUAR              33113211
COUGAR              33113211
FUR SEAL            32114411
SEA LION            32114411
GREY SEAL           32113322
ELEPHANT SEAL       21114411
```

```
REINDEER            04103333
ELK                 04103333
DEER                04003333
MOOSE               04003333
;

proc cluster data=teeth method=average nonorm
             outtree=_null_;
   var v1-v8;
   id mammal;
   title3 'Raw Data';
run;

proc cluster data=teeth std method=average nonorm
             outtree=_null_;
   var v1-v8;
   id mammal;
   title3 'Standardized Data';
run;
```

Output 23.4.1. Average Linkage Analysis of Mammals' Teeth Data: Raw Data

```
                           Hierarchical Cluster Analysis of Mammals' Teeth Data
                                      Evaluating the Effects of Ties
                                                 Raw Data

                                           The CLUSTER Procedure
                                      Average Linkage Cluster Analysis

                                    Eigenvalues of the Covariance Matrix

                              Eigenvalue    Difference    Proportion    Cumulative

                         1    3.76799365    2.33557185        0.5840        0.5840
                         2    1.43242180    0.91781899        0.2220        0.8061
                         3    0.51460281    0.08414950        0.0798        0.8858
                         4    0.43045331    0.30021485        0.0667        0.9525
                         5    0.13023846    0.03814626        0.0202        0.9727
                         6    0.09209220    0.04216914        0.0143        0.9870
                         7    0.04992305    0.01603541        0.0077        0.9947
                         8    0.03388764                      0.0053        1.0000


                     Root-Mean-Square Total-Sample Standard Deviation = 0.898027


                                              Cluster History
                                                                                       T
                                                                                RMS    i
                  NCL    ----------Clusters Joined-----------      FREQ      Dist    e

                   31    BEAVER              GROUNDHOG                 2         0    T
                   30    GRAY SQUIRREL       PORCUPINE                 2         0    T
                   29    WOLF                BEAR                      2         0    T
                   28    MARTEN              WOLVERINE                 2         0    T
                   27    WEASEL              BADGER                    2         0    T
                   26    JAGUAR              COUGAR                    2         0    T
                   25    FUR SEAL            SEA LION                  2         0    T
                   24    REINDEER            ELK                       2         0    T
                   23    DEER                MOOSE                     2         0
                   22    BROWN BAT           SILVER HAIR BAT           2         1    T
                   21    PIGMY BAT           HOUSE BAT                 2         1    T
                   20    PIKA                RABBIT                    2         1    T
                   19    CL31                CL30                      4         1    T
                   18    CL28                RIVER OTTER               3         1    T
                   17    CL27                SEA OTTER                 3         1    T
                   16    CL24                CL23                      4         1
                   15    CL21                RED BAT                   3    1.2247
                   14    CL17                GREY SEAL                 4     1.291
                   13    CL29                RACCOON                   3    1.4142    T
                   12    CL25                ELEPHANT SEAL             3    1.4142
                   11    CL18                CL14                      7    1.5546
                   10    CL22                CL15                      5    1.5811
                    9    CL20                CL19                      6    1.8708    T
                    8    CL11                CL26                      9    1.9272
                    7    CL8                 CL12                     12    2.2278
                    6    MOLE                CL13                      4    2.2361
                    5    CL9                 HOUSE MOUSE               7    2.4833
                    4    CL6                 CL7                      16    2.5658
                    3    CL10                CL16                      9    2.8107
                    2    CL3                 CL5                      16    3.7054
                    1    CL2                 CL4                      32    4.2939
```

Output 23.4.2. Average Linkage Analysis of Mammals’ Teeth Data: Standardized Data

```
                 Hierarchical Cluster Analysis of Mammals’ Teeth Data
                              Evaluating the Effects of Ties
                                   Standardized Data

                                 The CLUSTER Procedure
                             Average Linkage Cluster Analysis

                           Eigenvalues of the Correlation Matrix

                   Eigenvalue    Difference    Proportion    Cumulative

               1    4.74153902    3.27458808        0.5927        0.5927
               2    1.46695094    0.70824118        0.1834        0.7761
               3    0.75870977    0.25146252        0.0948        0.8709
               4    0.50724724    0.30264737        0.0634        0.9343
               5    0.20459987    0.05925818        0.0256        0.9599
               6    0.14534169    0.03450100        0.0182        0.9780
               7    0.11084070    0.04606994        0.0139        0.9919
               8    0.06477076                      0.0081        1.0000


           The data have been standardized to mean 0 and variance 1
           Root-Mean-Square Total-Sample Standard Deviation =        1


                                   Cluster History
                                                                        T
                                                                RMS     i
       NCL    ----------Clusters Joined-----------      FREQ      Dist    e

        31    BEAVER              GROUNDHOG                2         0    T
        30    GRAY SQUIRREL       PORCUPINE                2         0    T
        29    WOLF                BEAR                     2         0    T
        28    MARTEN              WOLVERINE                2         0    T
        27    WEASEL              BADGER                   2         0    T
        26    JAGUAR              COUGAR                   2         0    T
        25    FUR SEAL            SEA LION                 2         0    T
        24    REINDEER            ELK                      2         0    T
        23    DEER                MOOSE                    2         0
        22    PIGMY BAT           RED BAT                  2    0.9157
        21    CL28                RIVER OTTER              3    0.9169
        20    CL31                CL30                     4    0.9428    T
        19    BROWN BAT           SILVER HAIR BAT          2    0.9428    T
        18    PIKA                RABBIT                   2    0.9428
        17    CL27                SEA OTTER                3    0.9847
        16    CL22                HOUSE BAT                3    1.1437
        15    CL21                CL17                     6    1.3314
        14    CL25                ELEPHANT SEAL            3    1.3447
        13    CL19                CL16                     5    1.4688
        12    CL15                GREY SEAL                7    1.6314
        11    CL29                RACCOON                  3     1.692
        10    CL18                CL20                     6    1.7357
         9    CL12                CL26                     9    2.0285
         8    CL24                CL23                     4    2.1891
         7    CL9                 CL14                    12    2.2674
         6    CL10                HOUSE MOUSE              7     2.317
         5    CL11                CL7                     15    2.6484
         4    CL13                MOLE                     6    2.8624
         3    CL4                 CL8                     10    3.5194
         2    CL3                 CL6                     17    4.1265
         1    CL2                 CL5                     32    4.7753
```

There are ties at 16 levels for the raw data but at only 10 levels for the standardized data. There are more ties for the raw data because the increments between successive values are the same for all of the raw variables but different for the standardized variables.

One way to assess the importance of the ties in the analysis is to repeat the analysis on several random permutations of the observations and then to see to what extent the results are consistent at the interesting levels of the cluster history. Three macros are presented to facilitate this process.

```
/* --------------------------------------------------------- */
/*                                                           */
/* The macro CLUSPERM randomly permutes observations and     */
/* does a cluster analysis for each permutation.             */
/* The arguments are as follows:                             */
/*                                                           */
/*    data    data set name                                  */
/*    var     list of variables to cluster                   */
/*    id      id variable for proc cluster                   */
/*    method  clustering method (and possibly other options) */
/*    nperm   number of random permutations.                 */
/*                                                           */
/* --------------------------------------------------------- */
%macro CLUSPERM(data,var,id,method,nperm);
/* ------CREATE TEMPORARY DATA SET WITH RANDOM NUMBERS------ */
data _temp_;
   set &data;
   array _random_ _ran_1-_ran_&nperm;
   do over _random_;
     _random_=ranuni(835297461);
   end;
run;
/* ------PERMUTE AND CLUSTER THE DATA----------------------- */
%do n=1 %to &nperm;
    proc sort data=_temp_(keep=_ran_&n &var &id) out=_perm_;
       by _ran_&n;
    run;
    proc cluster method=&method noprint outtree=_tree_&n;
       var &var;
       id &id;
    run;
%end;
%mend;

/* --------------------------------------------------------- */
/*                                                           */
/* The macro PLOTPERM plots various cluster statistics       */
/* against the number of clusters for each permutation.      */
/* The arguments are as follows:                             */
/*                                                           */
/*    stats   names of variables from tree data set          */
/*    nclus   maximum number of clusters to be plotted       */
/*    nperm   number of random permutations.                 */
/*                                                           */
/* --------------------------------------------------------- */
%macro PLOTPERM(stat,nclus,nperm);
/* ---CONCATENATE TREE DATA SETS FOR 20 OR FEWER CLUSTERS--- */
data _plot_;
   set %do n=1 %to &nperm; _tree_&n(in=_in_&n) %end; ;
   if _ncl_<=&nclus;
   %do n=1 %to &nperm;
```

```
      if _in_&n then _perm_=&n;
   %end;
   label _perm_='permutation number';
   keep _ncl_ &stat _perm_;
run;
/* ---PLOT THE REQUESTED STATISTICS BY NUMBER OF CLUSTERS--- */

proc plot;
   plot (&stat)*_ncl_=_perm_ /vpos=26;
title2 'Symbol is value of _PERM_';
run;
%mend;

/* -------------------------------------------------------- */
/*                                                          */
/* The macro TREEPERM generates cluster-membership variables */
/* for a specified number of clusters for each permutation.  */
/* PROC PRINT lists the objects in each cluster-combination, */
/* and PROC TABULATE gives the frequencies and means.  The   */
/* arguments are as follows:                                 */
/*                                                           */
/*    var     list of variables to cluster                   */
/*            (no "-" or ":" allowed)                        */
/*    id      id variable for proc cluster                   */
/*    meanfmt format for printing means in PROC TABULATE     */
/*    nclus   number of clusters desired                     */
/*    nperm   number of random permutations.                 */
/*                                                           */
/* -------------------------------------------------------- */
%macro TREEPERM(var,id,meanfmt,nclus,nperm);
/* ------CREATE DATA SETS GIVING CLUSTER MEMBERSHIP--------- */
%do n=1 %to &nperm;
   proc tree data=_tree_&n noprint n=&nclus
             out=_out_&n(drop=clusname
                           rename=(cluster=_clus_&n));
      copy &var;
      id &id;
   run;
   proc sort;
      by &id &var;
   run;
%end;
/* ------MERGE THE CLUSTER VARIABLES----------------------- */
data _merge_;
   merge
      %do n=1 %to &nperm;
         _out_&n
      %end; ;
   by &id &var;
   length all_clus $ %eval(3*&nperm);
   %do n=1 %to &nperm;
      substr( all_clus, %eval(1+(&n-1)*3), 3) =
         put( _clus_&n, 3.);
   %end;
```

```
run;


/* ------PRINT AND TABULATE CLUSTER COMBINATIONS----------- */
proc sort;
   by _clus_:;
run;
proc print;
   var &var;
   id &id;
   by all_clus notsorted;
run;
proc tabulate order=data formchar='           ';
   class all_clus;
   var &var;
   table all_clus, n='FREQ'*f=5. mean*f=&meanfmt*(&var) /
      rts=%eval(&nperm*3+1);
run;
%mend;
```

To use these, it is first convenient to define a macro, VLIST, listing the teeth variables, since the forms V1-V8 or V: cannot be used with the TABULATE procedure in the TREEPERM macro:

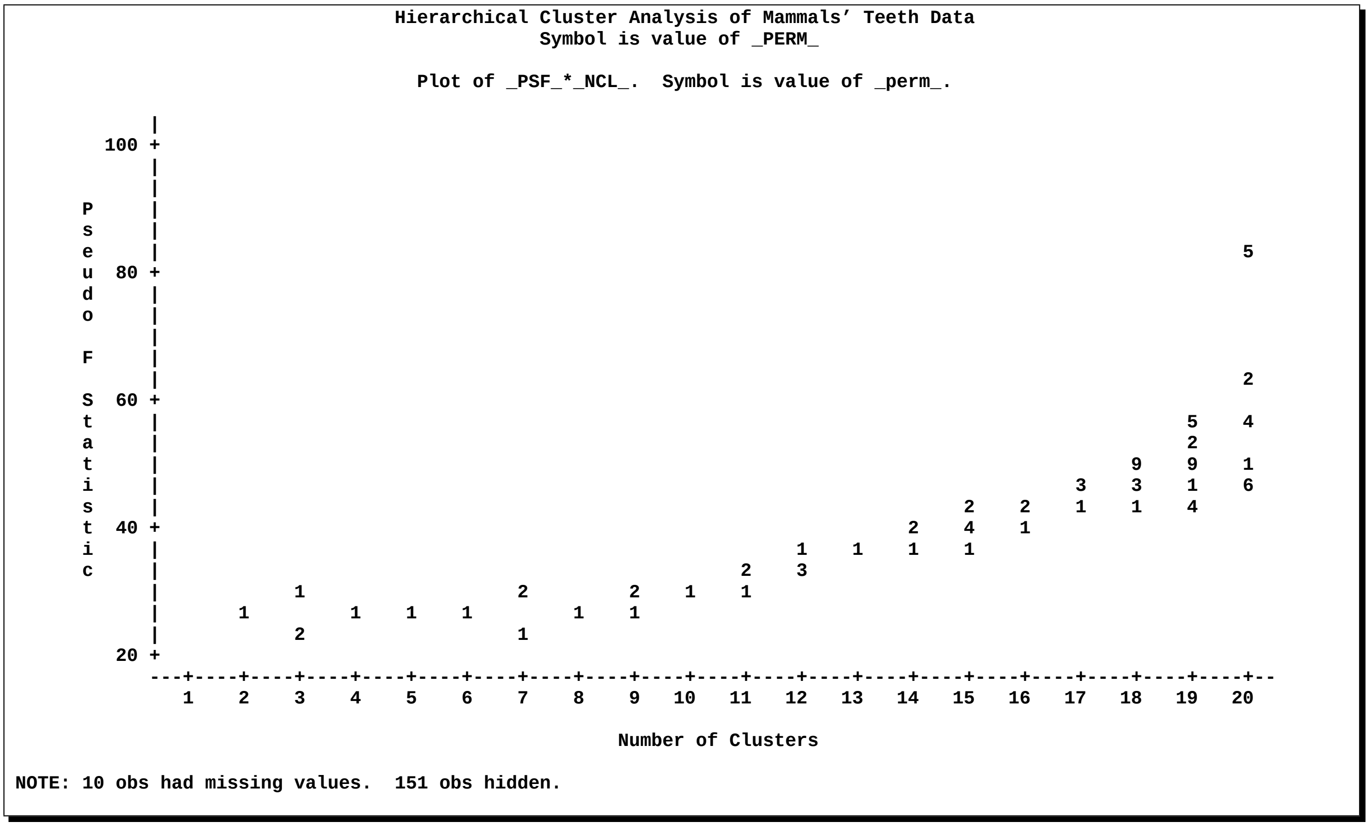
```
/* -TABULATE does not accept hyphens or colons in VAR lists- */
%let vlist=v1 v2 v3 v4 v5 v6 v7 v8;
```

The CLUSPERM macro is then called to analyze ten random permutations. The PLOTPERM macro plots the pseudo F and t^2 statistics and the cubic clustering criterion. Since the data are discrete, the pseudo F statistic and the cubic clustering criterion can be expected to increase as the number of clusters increases, so local maxima or large jumps in these statistics are more relevant than the global maximum in determining the number of clusters. For the raw data, only the pseudo t^2 statistic indicates the possible presence of clusters, with the 4-cluster level being suggested. Hence, the TREEPERM macro is used to analyze the results at the 4-cluster level:

```
title3 'Raw Data';

/* ------CLUSTER RAW DATA WITH AVERAGE LINKAGE------------- */
%clusperm( teeth, &vlist, mammal, average, 10);

/* -----PLOT STATISTICS FOR THE LAST 20 LEVELS------------- */
%plotperm( _psf_ _pst2_ _ccc_, 20, 10);

/* ------ANALYZE THE 4-CLUSTER LEVEL----------------------- */
%treeperm( &vlist, mammal, 9.1, 4, 10);
```

The results are shown in Output 23.4.3.

Output 23.4.3. Analysis of Ten Random Permutations of Raw Mammals' Teeth Data: Indeterminacy at the 4-Cluster Level



```
                                  Hierarchical Cluster Analysis of Mammals' Teeth Data
                                               Symbol is value of _PERM_

                                    Plot of _PSF_*_NCL_.  Symbol is value of _perm_.

            |
        100 +
            |
            |
      P     |
      s     |
      e     |                                                                                                 5
      u  80 +
      d     |
      o     |
            |
      F     |
            |                                                                                                 2
      S  60 +
      t     |                                                                                            5    4
      a     |                                                                                            2
      t     |                                                                                       9    9    1
      i     |                                                                                  3    3    1    6
      s     |                                                                        2    2    1    1    4
      t  40 +                                                                   2    4    1
      i     |                                                         1    1    1    1
      c     |                                                    2    3
            |            1                   2         2    1    1
            |       1         1    1    1         1    1
            |            2                   1
         20 +
            ---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
               1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20

                                                      Number of Clusters

NOTE: 10 obs had missing values.  151 obs hidden.
```

```
                                  Hierarchical Cluster Analysis of Mammals' Teeth Data
                                               Symbol is value of _PERM_

                                   Plot of _PST2_*_NCL_.  Symbol is value of _perm_.

       P    |
       s 30 +
       e    |
       u    |  1
       d    |
       o 25 +
            |
       T    |
       -    |            1
       S 20 +
       q    |       1
       u    |
       a    |
       r 15 +
       e    |
       d    |            2
            |       2                                  1
       S 10 +
       t    |
       a    |                 2                        2                        3
       t    |                 1    2    2    1    1              1
       i  5 +                           1                                       2    5
       s    |                      1                        2    3              4    1
       t    |                                               1
       i    |                                                                   1    2    1
       c  0 +
            ---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
               1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20

                                                      Number of Clusters

NOTE: 69 obs had missing values.  104 obs hidden.
```

```
                                  Hierarchical Cluster Analysis of Mammals’ Teeth Data
                                               Symbol is value of _PERM_

                                    Plot of _CCC_*_NCL_.  Symbol is value of _perm_.

C   |
u 4 +
b   |
i   |
c   |
    |                              2
C   |
l 3 +
u   |                        1     1
s   |
t   |
e   |
r   |                        2
i 2 +
n   |                  1
g   |            1     2
    |
C   |
r   |
i 1 +
t   |
e   |
r   |            2
i   |
o   |      1
n 0 +1
    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20

                                                     Number of Clusters

NOTE: 140 obs had missing values.  50 obs hidden.
```

```
------------------------------------- all_clus='  1  3  1  1  1  3  3  3  2  3' -------------------------------------

                                  mammal      v1    v2    v3    v4    v5    v6    v7    v8

                                  DEER         0     4     0     0     3     3     3     3
                                  ELK          0     4     1     0     3     3     3     3
                                  MOOSE        0     4     0     0     3     3     3     3
                                  REINDEER     0     4     1     0     3     3     3     3


------------------------------------- all_clus='  2  2  2  2  2  2  1  2  1  1' -------------------------------------

                               mammal           v1    v2    v3    v4    v5    v6    v7    v8

                               BADGER            3     3     1     1     3     3     1     2
                               BEAR              3     3     1     1     4     4     2     3
                               COUGAR            3     3     1     1     3     2     1     1
                               ELEPHANT SEAL     2     1     1     1     4     4     1     1
                               FUR SEAL          3     2     1     1     4     4     1     1
                               GREY SEAL         3     2     1     1     3     3     2     2
                               JAGUAR            3     3     1     1     3     2     1     1
                               MARTEN            3     3     1     1     4     4     1     2
                               RACCOON           3     3     1     1     4     4     3     2
                               RIVER OTTER       3     3     1     1     4     3     1     2
                               SEA LION          3     2     1     1     4     4     1     1
                               SEA OTTER         3     2     1     1     3     3     1     2
                               WEASEL            3     3     1     1     3     3     1     2
                               WOLF              3     3     1     1     4     4     2     3
                               WOLVERINE         3     3     1     1     4     4     1     2


------------------------------------- all_clus='  2  4  2  2  4  2  1  2  1  1' -------------------------------------

                                   mammal    v1    v2    v3    v4    v5    v6    v7    v8

                                    MOLE      3     2     1     0     3     3     3     3


------------------------------------- all_clus='  3  1  3  3  3  1  2  1  3  2' -------------------------------------

                               mammal           v1    v2    v3    v4    v5    v6    v7    v8

                               BEAVER            1     1     0     0     2     1     3     3
                               GRAY SQUIRREL     1     1     0     0     1     1     3     3
                               GROUNDHOG         1     1     0     0     2     1     3     3
                               HOUSE MOUSE       1     1     0     0     0     0     3     3
                               PORCUPINE         1     1     0     0     1     1     3     3


------------------------------------- all_clus='  3  4  3  3  4  1  2  1  3  2' -------------------------------------

                                   mammal    v1    v2    v3    v4    v5    v6    v7    v8

                                   PIKA       2     1     0     0     2     2     3     3
                                   RABBIT     2     1     0     0     3     2     3     3


------------------------------------- all_clus='  4  4  4  4  4  4  4  4  4  4' -------------------------------------

                              mammal            v1    v2    v3    v4    v5    v6    v7    v8

                              BROWN BAT          2     3     1     1     3     3     3     3
                              HOUSE BAT          2     3     1     1     1     2     3     3
                              PIGMY BAT          2     3     1     1     2     2     3     3
                              RED BAT            1     3     1     1     2     2     3     3
                              SILVER HAIR BAT    2     3     1     1     2     3     3     3
```

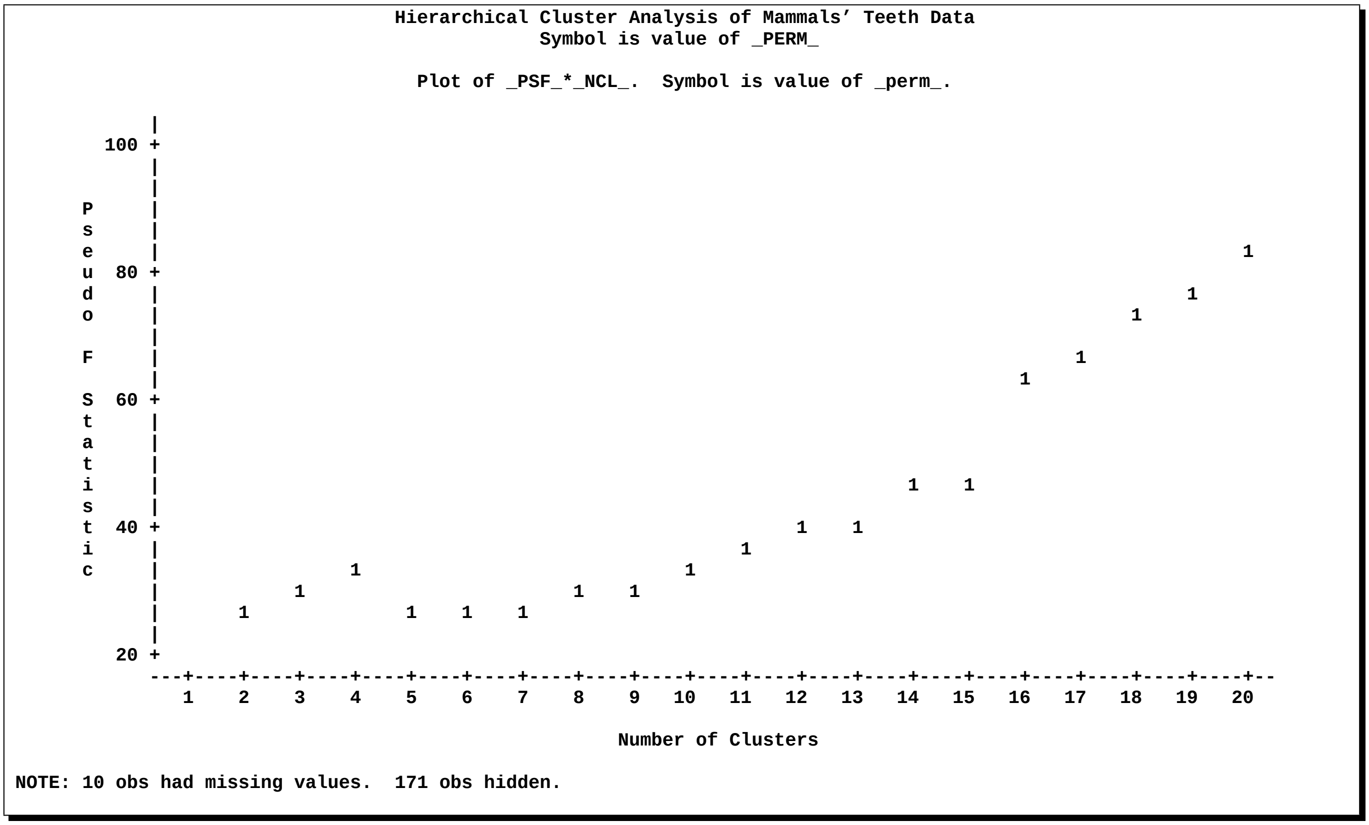
```
                                                                Mean

                                            Top     Bottom      Top    Bottom      Top    Bottom      Top    Bottom
                                   FREQ  incisors  incisors   canines  canines  premolars premolars  molars    molars

all_clus

1   3   1   1   1   3   3   3   2   3     4       0.0       4.0       0.5       0.0       3.0       3.0       3.0       3.0

2   2   2   2   2   2   1   2   1   1    15       2.9       2.6       1.0       1.0       3.6       3.4       1.3       1.8

2   4   2   2   4   2   1   2   1   1     1       3.0       2.0       1.0       0.0       3.0       3.0       3.0       3.0

3   1   3   3   3   1   2   1   3   2     5       1.0       1.0       0.0       0.0       1.2       0.8       3.0       3.0

3   4   3   3   4   1   2   1   3   2     2       2.0       1.0       0.0       0.0       2.5       2.0       3.0       3.0

4   4   4   4   4   4   4   4   4   4     5       1.8       3.0       1.0       1.0       2.0       2.4       3.0       3.0
```

From the TABULATE and PRINT output, you can see that two types of clustering are obtained. In one case, the mole is grouped with the carnivores, while the pika and rabbit are grouped with the rodents. In the other case, both the mole and the lagomorphs are grouped with the bats.

Next, the analysis is repeated with the standardized data. The pseudo F and t^2 statistics indicate 3 or 4 clusters, while the cubic clustering criterion shows a sharp rise up to 4 clusters and then levels off up to 6 clusters. So the TREEPERM macro is used again at the 4-cluster level. In this case, there is no indeterminacy, as the same four clusters are obtained with every permutation, although in different orders. It must be emphasized, however, that lack of indeterminacy in no way indicates validity. The results are shown in Output 23.4.4.

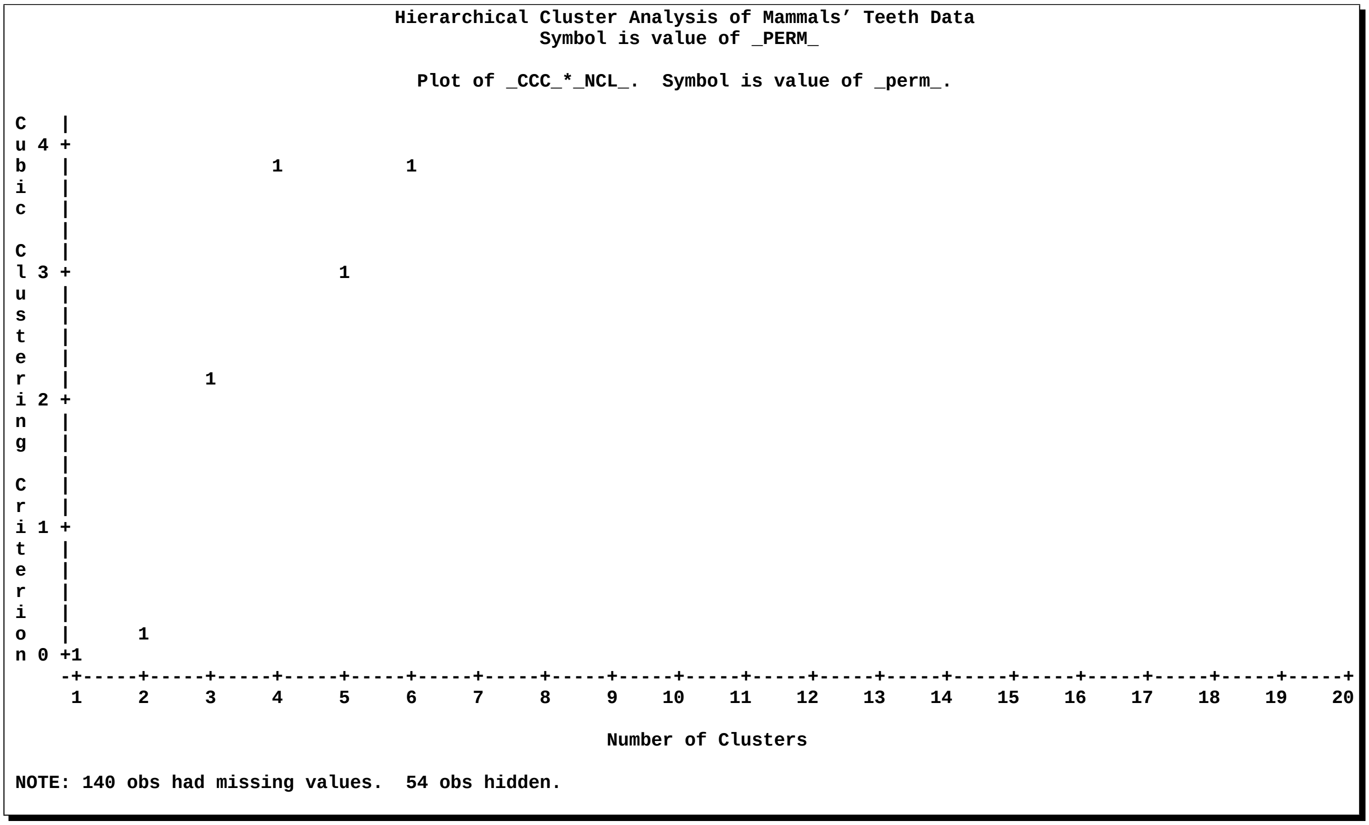
```
title3 'Standardized Data';

/*------CLUSTER STANDARDIZED DATA WITH AVERAGE LINKAGE------*/
%clusperm( teeth, &vlist, mammal, average std, 10);

/*------PLOT STATISTICS FOR THE LAST 20 LEVELS--------------*/
%plotperm( _psf_ _pst2_ _ccc_, 20, 10);

/*------ANALYZE THE 4-CLUSTER LEVEL-------------------------*/
%treeperm( &vlist, mammal, 9.1, 4, 10);
```

Output 23.4.4. Analysis of Ten Random Permutations of Standardized Mammals’ Teeth Data: No Indeterminacy at the 4-Cluster Level



```
                                  Hierarchical Cluster Analysis of Mammals’ Teeth Data
                                               Symbol is value of _PERM_

                                    Plot of _PSF_*_NCL_.  Symbol is value of _perm_.

            |
        100 +
            |
            |
      P     |
      s     |
      e     |                                                                                                 1
      u  80 +
      d     |                                                                                            1
      o     |                                                                                       1
            |
      F     |                                                                                  1
            |                                                                             1
      S  60 +
      t     |
      a     |
      t     |
      i     |                                                                   1    1
      s     |
      t  40 +                                                         1    1
      i     |                                                    1
      c     |                 1                             1
            |            1                        1    1
            |       1              1    1    1
            |
         20 +
            ---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
               1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20

                                                      Number of Clusters

NOTE: 10 obs had missing values.  171 obs hidden.
```

```
                                  Hierarchical Cluster Analysis of Mammals’ Teeth Data
                                               Symbol is value of _PERM_

                                   Plot of _PST2_*_NCL_.  Symbol is value of _perm_.

       P    |
       s    |
       e    |
       u    |
       d    |
       o    |
         30 +
       T    |  1
       -    |
       S    |
       q    |
       u    |
       a 20 +
       r    |
       e    |       1
       d    |
            |            1
       S    |
       t 10 +                                               1
       a    |
       t    |                 1    1         1         1                             1
       i    |
       s    |                           1                             1    1
       t    |                                                                             1
       i  0 +
       c    |
            ---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
               1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20

                                                      Number of Clusters

NOTE: 70 obs had missing values.  117 obs hidden.
```



```
                                  Hierarchical Cluster Analysis of Mammals’ Teeth Data
                                               Symbol is value of _PERM_

                                    Plot of _CCC_*_NCL_.  Symbol is value of _perm_.

C   |
u 4 +
b   |                  1           1
i   |
c   |
    |
C   |
l 3 +                        1
u   |
s   |
t   |
e   |
r   |            1
i 2 +
n   |
g   |
    |
C   |
r   |
i 1 +
t   |
e   |
r   |
i   |
o   |      1
n 0 +1
    -+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16    17    18    19    20

                                                     Number of Clusters

NOTE: 140 obs had missing values.  54 obs hidden.
```

```
------------------------------------- all_clus=’  1  3  1  1  1  3  3  3  2  3’ -------------------------------------

                                mammal      v1    v2    v3    v4    v5    v6    v7    v8

                                DEER         0     4     0     0     3     3     3     3
                                ELK          0     4     1     0     3     3     3     3
                                MOOSE        0     4     0     0     3     3     3     3
                                REINDEER     0     4     1     0     3     3     3     3


------------------------------------- all_clus=’  2  2  2  2  2  2  1  2  1  1’ -------------------------------------

                             mammal           v1    v2    v3    v4    v5    v6    v7    v8

                             BADGER            3     3     1     1     3     3     1     2
                             BEAR              3     3     1     1     4     4     2     3
                             COUGAR            3     3     1     1     3     2     1     1
                             ELEPHANT SEAL     2     1     1     1     4     4     1     1
                             FUR SEAL          3     2     1     1     4     4     1     1
                             GREY SEAL         3     2     1     1     3     3     2     2
                             JAGUAR            3     3     1     1     3     2     1     1
                             MARTEN            3     3     1     1     4     4     1     2
                             RACCOON           3     3     1     1     4     4     3     2
                             RIVER OTTER       3     3     1     1     4     3     1     2
                             SEA LION          3     2     1     1     4     4     1     1
                             SEA OTTER         3     2     1     1     3     3     1     2
                             WEASEL            3     3     1     1     3     3     1     2
                             WOLF              3     3     1     1     4     4     2     3
                             WOLVERINE         3     3     1     1     4     4     1     2


------------------------------------- all_clus=’  3  1  3  3  3  1  2  1  3  2’ -------------------------------------

                             mammal           v1    v2    v3    v4    v5    v6    v7    v8

                             BEAVER            1     1     0     0     2     1     3     3
                             GRAY SQUIRREL     1     1     0     0     1     1     3     3
                             GROUNDHOG         1     1     0     0     2     1     3     3
                             HOUSE MOUSE       1     1     0     0     0     0     3     3
                             PIKA              2     1     0     0     2     2     3     3
                             PORCUPINE         1     1     0     0     1     1     3     3
                             RABBIT            2     1     0     0     3     2     3     3


------------------------------------- all_clus=’  4  4  4  4  4  4  4  4  4  4’ -------------------------------------

                            mammal            v1    v2    v3    v4    v5    v6    v7    v8

                            BROWN BAT          2     3     1     1     3     3     3     3
                            HOUSE BAT          2     3     1     1     1     2     3     3
                            MOLE               3     2     1     0     3     3     3     3
                            PIGMY BAT          2     3     1     1     2     2     3     3
                            RED BAT            1     3     1     1     2     2     3     3
                            SILVER HAIR BAT    2     3     1     1     2     3     3     3



                                                                                     Mean

                                            Top      Bottom      Top      Bottom      Top    Bottom      Top     Bottom
                                 FREQ  incisors  incisors   canines   canines  premolars premolars  molars    molars

  all_clus

  1  3  1  1  1  3  3  3  2  3      4       0.0       4.0       0.5       0.0       3.0       3.0       3.0       3.0

  2  2  2  2  2  2  1  2  1  1     15       2.9       2.6       1.0       1.0       3.6       3.4       1.3       1.8

  3  1  3  3  3  1  2  1  3  2      7       1.3       1.0       0.0       0.0       1.6       1.1       3.0       3.0

  4  4  4  4  4  4  4  4  4  4      6       2.0       2.8       1.0       0.8       2.2       2.5       3.0       3.0
```

Example 23.5. Computing a Distance Matrix

An example of the use of distance and similarity measures in cluster analysis is given in Example 26.1 in the PROC DISTANCE chapter.

Example 23.6. Size, Shape, and Correlation

The following example shows the analysis of a data set in which size information is detrimental to the classification. Imagine that an archaeologist of the future is excavating a 20th century grocery store. The archaeologist has discovered a large number of boxes of various sizes, shapes, and colors and wants to do a preliminary classification based on simple external measurements: height, width, depth, weight, and the predominant color of the box. It is known that a given product may have been sold in packages of different size, so the archaeologist wants to remove the effect of size from the classification. It is not known whether color is relevant to the use of the products, so the analysis should be done both with and without color information.

Unknown to the archaeologist, the boxes actually fall into six general categories according to the use of the product: breakfast cereals, crackers, laundry detergents, Little Debbie snacks, tea, and toothpaste. These categories are shown in the analysis so that you can evaluate the effectiveness of the classification.

Since there is no reason for the archaeologist to assume that the true categories have equal sample sizes or variances, the centroid method is used to avoid undue bias. Each analysis is done with Euclidean distances after suitable transformations of the data. Color is coded as five dummy variables with values of 0 or 1. The DATA step is as follows:

```
options ls=120;
title ’Cluster Analysis of Grocery Boxes’;
data grocery2;
   length name $35   /* name of product */
          class $16  /* category of product */
          unit $1    /* unit of measurement for weights:
                           g=gram
                           o=ounce
                           l=lb
                        all weights are converted to grams */
          color $8   /* predominant color of box */
          height 8   /* height of box in cm. */
          width 8    /* width of box in cm. */
          depth 8    /* depth of box (front to back) in cm. */
          weight 8   /* weight of box in grams */
          c_white c_yellow c_red c_green c_blue 4;
                     /* dummy variables */
   retain class;
   drop unit;

   /*--- read name with possible embedded blanks ---*/
   input name & @;

   /*--- if name starts with "---",              ---*/
   /*--- it’s really a category value            ---*/
   if substr(name,1,3) = ’---’ then do;
      class = substr(name,4,index(substr(name,4),’-’)-1);
      delete;
      return;
   end;

   /*--- read the rest of the variables ---*/
   input height width depth weight unit color;

   /*--- convert weights to grams ---*/
   select (unit);
      when (’l’) weight = weight * 454;
      when (’o’) weight = weight * 28.3;
      when (’g’) ;
      otherwise put ’Invalid unit ’ unit;
   end;

   /*--- use 0/1 coding for dummy variables for colors ---*/
   c_white  = (color = ’w’);
   c_yellow = (color = ’y’);
   c_red    = (color = ’r’);
   c_green  = (color = ’g’);
   c_blue   = (color = ’b’);

datalines;

---Breakfast cereals---

Cheerios                            32.5 22.4  8.4  567 g y
Cheerios                            30.3 20.4  7.2  425 g y
Cheerios                            27.5 19    6.2  283 g y
Cheerios                            24.1 17.2  5.3  198 g y
Special K                           30.1 20.5  8.5   18 o w
Special K                           29.6 19.2  6.7   12 o w
Special K                           23.4 16.6  5.7    7 o w
Corn Flakes                         33.7 25.4  8     24 o w
Corn Flakes                         30.2 20.6  8.4   18 o w
Corn Flakes                         30   19.1  6.6   12 o w
Grape Nuts                          21.7 16.3  4.9  680 g w
Shredded Wheat                      19.7 19.9  7.5  283 g y
Shredded Wheat, Spoon Size          26.6 19.6  5.6  510 g r
All-Bran                            21.1 14.3  5.2 13.8 o y
Froot Loops                         30.2 20.8  8.5 19.7 o r
Froot Loops                         25   17.7  6.4   11 o r

---Crackers---

Wheatsworth                         11.1 25.2  5.5  326 g w
Ritz                                23.1 16    5.3  340 g r
Ritz                                23.1 20.7  5.2  454 g r
Premium Saltines                    11   25   10.7  454 g w
Waverly Wafers                      14.4 22.5  6.2  454 g g
```

```
---Detergent---

Arm & Hammer Detergent              38.8 30   16.9   25 l y
Arm & Hammer Detergent              39.5 25.8 11   14.2 l y
Arm & Hammer Detergent              33.7 22.8  7      7 l y
Arm & Hammer Detergent              27.8 19.4  6.3    4 l y
Tide                                39.4 24.8 11.3  9.2 l r
Tide                                32.5 23.2  7.3  4.5 l r
Tide                                26.5 19.9  6.3   42 o r
Tide                                19.3 14.6  4.7   17 o r

---Little Debbie---

Figaroos                            13.5 18.6  3.7   12 o y
Swiss Cake Rolls                    10.1 21.8  5.8   13 o w
Fudge Brownies                      11   30.8  2.5   12 o w
Marshmallow Supremes                 9.4 32    7     10 o w
Apple Delights                      11.2 30.1  4.9   15 o w
Snack Cakes                         13.4 32    3.4   13 o b
Nutty Bar                           13.2 18.5  4.2   12 o y
Lemon Stix                          13.2 18.5  4.2    9 o w
Fudge Rounds                         8.1 28.3  5.4  9.5 o w

---Tea---

Celestial Saesonings Mint Magic      7.8 13.8  6.3   49 g b
Celestial Saesonings Cranberry Cove  7.8 13.8  6.3   46 g r
Celestial Saesonings Sleepy Time     7.8 13.8  6.3   37 g g
Celestial Saesonings Lemon Zinger    7.8 13.8  6.3   56 g y
Bigelow Lemon Lift                   7.7 13.4  6.9   40 g y
Bigelow Plantation Mint              7.7 13.4  6.9   35 g g
Bigelow Earl Grey                    7.7 13.4  6.9   35 g b
Luzianne                             8.9 22.8  6.4    6 o r
Luzianne                            18.4 20.2  6.9    8 o r
Luzianne Decaffeinated               8.9 22.8  6.4 5.25 o g
Lipton Tea Bags                     17.1 20    6.7    8 o r
Lipton Tea Bags                     11.5 14.4  6.6 3.75 o r
Lipton Tea Bags                      6.7 10    5.7 1.25 o r
Lipton Family Size Tea Bags         13.7 24    9     12 o r
Lipton Family Size Tea Bags          8.7 20.8  8.2    6 o r
Lipton Family Size Tea Bags          8.9 11.1  8.2    3 o r
Lipton Loose Tea                    12.7 10.9  5.4    8 o r

---Paste, Tooth---

Colgate                              4.4 22    3.5    7 o r
Colgate                              3.6 15.6  3.3    3 o r
Colgate                              4.2 18.3  3.5    5 o r
Crest                                4.3 21.7  3.7  6.4 o w
Crest                                4.3 17.4  3.6  4.6 o w
Crest                                3.5 15.2  3.2  2.7 o w
Crest                                3.0 10.9  2.8  .85 o w
Arm & Hammer                         4.4 17    3.7    5 o w
;
```

```
data grocery;
   length name $16;
   set grocery2;
```

The FORMAT procedure is used to define to formats to make the output easier to read. The STARS. format is used for graphical crosstabulations in the TABULATE procedure. The $COLOR format displays the names of the colors instead of just the first letter.

```
    /*------ formats and macros for displaying ------*/
    /*------ cluster results                   ------*/
proc format; value stars
      0=’               ’
      1=’              #’
      2=’             ##’
      3=’            ###’
      4=’           ####’
      5=’          #####’
      6=’         ######’
      7=’        #######’
      8=’       ########’
      9=’      #########’
     10=’     ##########’
     11=’    ###########’
     12=’   ############’
     13=’  #############’
     14=’ ##############’
15-high=’>##############’;
run;

proc format; value $color
   ’w’=’White’
   ’y’=’Yellow’
   ’r’=’Red’
   ’g’=’Green’
   ’b’=’Blue’;
run;
```

Since a full display of the results of each cluster analysis would be very long, a macro is used with five macro variables to select parts of the output. The macro variables are set to select only the PROC CLUSTER output and the crosstabulation of clusters and true categories for the first two analyses. The example could be run with different settings of the macro variables to show the full output or other selected parts.

```
%let cluster=1;   /* 1=show CLUSTER output, 0=don’t */
%let tree=0;      /* 1=print TREE diagram, 0=don’t */
%let list=0;      /* 1=list clusters, 0=don’t */
%let crosstab=1;  /* 1=crosstabulate clusters and classes,
                     0=don’t                              */
```

```
%let crosscol=0;  /* 1=crosstabulate clusters and colors,
                     0=don’t                               */

   /*--- define macro with options for TREE ---*/
%macro treeopt;
   %if &tree %then h page=1;
   %else noprint;
%mend;

   /*--- define macro with options for CLUSTER ---*/
%macro clusopt;
   %if &cluster %then pseudo ccc p=20;
   %else noprint;
%mend;

   /*------ macro for showing cluster results ------*/
%macro show(n); /* n=number of clusters
                   to show results for */

proc tree data=tree %treeopt n=&n out=out;
   id name;
   copy class height width depth weight color;
run;

%if &list %then %do;
   proc sort;
      by cluster;
   run;

   proc print;
      var class name height width depth weight color;
      by cluster clusname;
   run;
%end;

%if &crosstab %then %do;
   proc tabulate noseps /* formchar=’           ’ */;
        class class cluster;
        table cluster, class*n=’
              ’*f=stars./rts=10 misstext=’ ’;
run;
%end;

%if &crosscol %then %do;
   proc tabulate noseps /* formchar=’           ’ */;
      class color cluster;
      table cluster, color*n=’
            ’*f=stars./rts=10 misstext=’ ’;
      format color $color.;
run;
%end;
%mend;
```

The first analysis uses the variables height, width, depth, and weight in standardized form to show the effect of including size information. The CCC, pseudo F, and pseudo t^2 statistics indicate 10 clusters. Most of the clusters do not correspond closely to the true categories, and four of the clusters have only one or two observations.

```
/************************************************************/
/*                                                          */
/*        Analysis 1: standardized box measurements         */
/*                                                          */
/************************************************************/
title2 'Analysis 1: Standardized data';
proc cluster data=grocery m=cen std %clusopt outtree=tree;
   var height width depth weight;
   id name;
   copy class color;
run;

%show(10);
```

Output 23.6.1. Analysis of Standardized Data

```
                          Cluster Analysis of Grocery Boxes
                            Analysis 1: Standardized data

                                The CLUSTER Procedure
                        Centroid Hierarchical Cluster Analysis

                         Eigenvalues of the Correlation Matrix

                  Eigenvalue    Difference    Proportion    Cumulative

             1    2.44512438    1.64456210        0.6113        0.6113
             2    0.80056228    0.33149770        0.2001        0.8114
             3    0.46906458    0.18381582        0.1173        0.9287
             4    0.28524876                      0.0713        1.0000


           The data have been standardized to mean 0 and variance 1
           Root-Mean-Square Total-Sample Standard Deviation =        1
           Root-Mean-Square Distance Between Observations   = 2.828427
```

```
                                    Cluster Analysis of Grocery Boxes
                                      Analysis 1: Standardized data

                                          The CLUSTER Procedure
                                  Centroid Hierarchical Cluster Analysis

                       The data have been standardized to mean 0 and variance 1
                       Root-Mean-Square Total-Sample Standard Deviation =        1
                       Root-Mean-Square Distance Between Observations   = 2.828427


                                             Cluster History
                                                                                                Norm    T
                                                                                                Cent    i
   NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2      Dist    e

    20    CL22        OB54            11    0.0028    .974    .        .      85.4     4.5    0.3073
    19    CL36        OB8              5    0.0026    .972    .        .      83.7    15.3    0.3146
    18    CL24        CL41            12    0.0080    .964    .        .      70.2    10.0    0.3316
    17    CL18        CL30            18    0.0144    .949    .        .      53.8    12.7    0.3343
    16    OB33        CL29             3    0.0024    .947    .        .      55.8     4.7    0.3363
    15    CL50        CL33             7    0.0055    .941    .        .      55.0    24.4     0.346
    14    CL46        CL15            10    0.0069    .934    .        .      53.7     8.1    0.3192
    13    CL27        OB53             6    0.0035    .931    .        .      56.1     6.3     0.362
    12    CL31        CL16             5    0.0075    .923    .861    8.03    55.8     6.6    0.4416
    11    CL19        CL23             7    0.0102    .913    .848    7.59    54.6    12.7    0.4713
    10    OB23        OB26             2    0.0037    .909    .835    8.36    59.1      .     0.4781
     9    CL11        CL17            25    0.0393    .870    .819    4.72    45.2    19.3    0.4918
     8    CL13        CL14            16    0.0329    .837    .801    2.95    40.4    23.7    0.5215
     7    CL8         CL20            27    0.0629    .774    .779    -.31    32.0    25.9    0.5467
     6    CL7         OB62            28    0.0112    .763    .752    0.61    36.7     2.4    0.6003
     5    CL9         CL6             53    0.1879    .575    .718    -5.9    19.6    43.4    0.6641
     4    CL5         CL21            55    0.0345    .541    .672    -5.2    23.2     4.5     0.745
     3    CL4         CL12            60    0.1137    .427    .602    -5.3    22.4    14.5    0.8769
     2    CL3         CL10            62    0.1511    .276    .471    -4.3    23.2    15.8    1.5559
     1    CL2         OB22            63    0.2759    .000    .000    0.00      .     23.2     2.948
```

```
----------------------------------------------------------------------------------------------------------
|        |                                             class                                             |
|        |-----------------------------------------------------------------------------------------------|
|        |   Breakfast   |               |               |               |               |               |
|        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
|--------+---------------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |               |
|1       |               |               |               |               |               |    ###########|
|2       |               |             ##|               |              #|               |            ###|
|3       |          #####|               |             ##|               |               |               |
|4       |               |               |               |            ###|        #######|               |
|5       |    ###########|             ##|            ###|               |               |             ##|
|6       |               |               |               |          #####|               |               |
|7       |               |              #|               |               |               |              #|
|8       |               |               |             ##|               |               |               |
|9       |               |               |               |               |              #|               |
|10      |               |               |              #|               |               |               |
----------------------------------------------------------------------------------------------------------
```

The second analysis uses logarithms of height, width, depth, and the cube root of weight; the cube root is used for consistency with the linear measures. The rows are then centered to remove size information. Finally, the columns are standardized to have a standard deviation of 1. There is no compelling a priori reason to standardize the columns, but if they are not standardized, height dominates the analysis because of its large variance. The STANDARD procedure is used instead of the STD option in PROC CLUSTER so that a subsequent analysis can separately standardize the dummy variables for color.

```
/**********************************************************/
```

```
/*                                                         */
/*    Analysis 2: standardized row-centered logarithms    */
/*                                                         */
/**********************************************************/

title2 'Row-centered logarithms';
data shape;
   set grocery;
   array x height width depth weight;
   array l l_height l_width l_depth l_weight;
                          /* logarithms */
   weight=weight**(1/3);  /* take cube root to conform with
                             the other linear measurements */
   do over l;             /* take logarithms */
      l=log(x);
   end;
   mean=mean( of l(*));   /* find row mean of logarithms */
   do over l;
      l=l-mean;           /* center row */
   end;
run;

title2 'Analysis 2: Standardized row-centered logarithms';
proc standard data=shape out=shapstan m=0 s=1;
   var l_height l_width l_depth l_weight;
run;


proc cluster data=shapstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight;
   id name;
   copy class height width depth weight color;
run;

%show(8);
```

The results of the second analysis are shown for eight clusters. Clusters 1 through 4 correspond fairly well to tea, toothpaste, breakfast cereals, and detergents. Crackers and Little Debbie products are scattered among several clusters.

Output 23.6.2. Analysis of Standardized Row-Centered Logarithms

```
                                    Cluster Analysis of Grocery Boxes
                             Analysis 2: Standardized row-centered logarithms

                                          The CLUSTER Procedure
                                  Centroid Hierarchical Cluster Analysis

                                   Eigenvalues of the Covariance Matrix

                               Eigenvalue    Difference    Proportion    Cumulative

                          1    1.94931049    0.34845395        0.4873        0.4873
                          2    1.60085654    1.15102358        0.4002        0.8875
                          3    0.44983296    0.44983296        0.1125        1.0000
                          4    -.00000000                     -0.0000        1.0000


                       Root-Mean-Square Total-Sample Standard Deviation =        1
                       Root-Mean-Square Distance Between Observations   = 2.828427


                                             Cluster History
                                                                                                Norm    T
                                                                                                Cent    i
   NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2      Dist    e

    20    CL29        OB14             4    0.0017    .977    .        .      94.7     2.9    0.2658
    19    CL26        CL27             8    0.0045    .972    .        .      85.4     8.4    0.3047
    18    OB38        OB62             2    0.0016    .971    .        .      87.2      .     0.3193
    17    OB32        OB35             2    0.0018    .969    .        .      89.1      .     0.3331
    16    OB22        OB55             2    0.0019    .967    .        .      91.3      .     0.3434
    15    CL23        CL18             5    0.0050    .962    .        .      86.5     4.8    0.3587
    14    CL37        CL21             5    0.0051    .957    .        .      83.5    10.4    0.3613
    13    CL30        CL24             9    0.0068    .950    .        .      79.2    12.9    0.3682
    12    CL32        CL20            16    0.0142    .936    .892    5.75    67.6    29.3    0.3826
    11    CL22        OB34             4    0.0037    .932    .881    6.31    71.4     3.2    0.3901
    10    CL11        CL31             7    0.0090    .923    .869    6.17    70.8     6.3    0.4032
     9    CL33        CL13            11    0.0092    .914    .853    6.25    71.7     7.6    0.4181
     8    CL19        CL16            10    0.0131    .901    .835    6.12    71.4    10.9     0.503
     7    CL14        CL9             16    0.0297    .871    .813    4.63    63.1    15.6    0.5173
     6    CL10        CL15            12    0.0329    .838    .785    3.69    59.1    13.6    0.5916
     5    CL6         CL28            19    0.0557    .783    .748    2.01    52.2    15.8    0.6252
     4    CL12        CL8             26    0.0885    .694    .697    -.16    44.6    48.8    0.6679
     3    CL5         CL17            21    0.0459    .648    .617    1.21    55.3     7.4    0.8863
     2    CL4         CL7             42    0.2841    .364    .384    -.56    34.9    60.3    0.9429
     1    CL2         CL3             63    0.3640    .000    .000    0.00      .     34.9    0.8978
```

```
----------------------------------------------------------------------------------------------------------
|        |                                             class                                             |
|        |-----------------------------------------------------------------------------------------------|
|        |   Breakfast   |               |               |               |               |               |
|        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
|--------+---------------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |               |
|1       |               |              #|               |               |               |     ##########|
|2       |               |               |               |               |        #######|               |
|3       | ##############|             ##|               |               |               |               |
|4       |              #|               |       ########|               |               |              #|
|5       |               |               |               |             ##|              #|             ##|
|6       |              #|               |               |               |               |           ####|
|7       |               |             ##|               |          #####|               |               |
|8       |               |               |               |             ##|               |               |
----------------------------------------------------------------------------------------------------------
```

The third analysis is similar to the second analysis except that the rows are standardized rather than just centered. There is a clear indication of seven clusters from the CCC, pseudo F, and pseudo t^2 statistics. The clusters are listed as well as crosstabulated with the true categories and colors.

```
/************************************************************/
/*                                                          */
/*  Analysis 3: standardized row-standardized logarithms  */
/*                                                          */
/************************************************************/

%let list=1;
%let crosscol=1;

title2 'Row-standardized logarithms';
data std;
   set grocery;
   array x height width depth weight;
   array l l_height l_width l_depth l_weight;
                      /* logarithms */
   weight=weight**(1/3); /* take cube root to conform with
                            the other linear measurements */
   do over l;
      l=log(x);          /* take logarithms */
   end;
   mean=mean( of l(*));  /* find row mean of logarithms */
   std=std( of l(*));    /* find row standard deviation */
   do over l;
      l=(l-mean)/std;    /* standardize row */
   end;
run;

title2 'Analysis 3: Standardized row-standardized logarithms';
proc standard data=std out=stdstan m=0 s=1;
   var l_height l_width l_depth l_weight;
run;

proc cluster data=stdstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight;
   id name;
   copy class height width depth weight color;
run;

%show(7);
```

The output from the third analysis shows that cluster 1 contains 9 of the 17 teas. Cluster 2 contains all of the detergents plus Grape Nuts, a very heavy cereal. Cluster 3 includes all of the toothpastes and one Little Debbie product that is of very similar shape, although roughly twice as large. Cluster 4 has most of the cereals, Ritz crackers (which come in a box very similar to most of the cereal boxes), and Lipton Loose Tea (all the other teas in the sample come in tea bags). Clusters 5 and 6 each contain several Luzianne and Lipton teas and one or two miscellaneous items. Cluster 7 includes most of the Little Debbie products and two types of crackers. Thus, the crackers are not identified and the teas are broken up into three clusters, but the other categories correspond to single clusters. This analysis classifies toothpaste and Little Debbie products slightly better than the second analysis,

Output 23.6.3. Analysis of Standardized Row-Standardized Logarithms

```
                                  Cluster Analysis of Grocery Boxes
                             Analysis 3: Standardized row-standardized logarithms

                                         The CLUSTER Procedure
                                  Centroid Hierarchical Cluster Analysis

                                  Eigenvalues of the Covariance Matrix

                              Eigenvalue    Difference    Proportion    Cumulative

                         1    2.42684848    0.94583675        0.6067        0.6067
                         2    1.48101173    1.38887193        0.3703        0.9770
                         3    0.09213980    0.09213980        0.0230        1.0000
                         4    -.00000000                     -0.0000        1.0000


                    Root-Mean-Square Total-Sample Standard Deviation =        1
                    Root-Mean-Square Distance Between Observations   = 2.828427


                                            Cluster History
                                                                                              Norm    T
                                                                                              Cent    i
     NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC      PSF    PST2     Dist    e

      20    CL35        CL33             8    0.0024    .990     .        .       229    32.0   0.1923
      19    CL22        OB19             5    0.0010    .989     .        .       224     2.9   0.2014
      18    CL44        CL27             6    0.0018    .987     .        .       206    20.5   0.2073
      17    CL18        CL26             9    0.0025    .985     .        .       187     6.4   0.1956
      16    OB38        OB62             2    0.0009    .984     .        .       192      .      0.24
      15    CL24        CL23             5    0.0029    .981     .        .       177     7.8   0.2753
      14    CL25        OB21             4    0.0021    .979     .        .       175     7.7   0.2917
      13    CL30        CL19            17    0.0101    .969     .        .       130    41.0   0.2974
      12    CL16        CL31             9    0.0049    .964    .932    5.49     124    20.5   0.3121
      11    CL21        OB52             4    0.0029    .961    .924    5.81     129     8.2   0.3445
      10    CL41        CL11             6    0.0045    .957    .915    5.94     130     5.0    0.323
       9    CL29        OB50             4    0.0031    .953    .904    6.52     138    20.3   0.3603
       8    CL14        CL15             9    0.0101    .943    .890    6.08     131    10.7   0.3761
       7    CL20        OB54             9    0.0047    .939    .872    6.89     143    11.7   0.4063
       6    CL13        CL9             21    0.0272    .911    .848    5.23     117    30.0   0.5101
       5    CL6         CL17            30    0.0746    .837    .814    1.30    74.3    42.2    0.606
       4    CL10        CL7             15    0.0440    .793    .764    1.40    75.3    36.4   0.6152
       3    CL8         CL12            18    0.0642    .729    .681    2.02    80.6    44.0   0.6648
       2    CL3         CL4             33    0.2580    .471    .470    0.01    54.2    54.4   0.9887
       1    CL5         CL2             63    0.4707    .000    .000    0.00      .     54.2   0.9636
```

```
------------------------------------------------ CLUSTER=1 CLUSNAME=CL7 ------------------------------------------------

                    Obs    class          name          height    width    depth     weight    color

                      1     Tea     Bigelow Plantati      7.7      13.4     6.9     3.27107      g
                      2     Tea     Bigelow Earl Gre      7.7      13.4     6.9     3.27107      b
                      3     Tea     Celestial Saeson      7.8      13.8     6.3     3.65931      b
                      4     Tea     Celestial Saeson      7.8      13.8     6.3     3.58305      r
                      5     Tea     Bigelow Lemon Li      7.7      13.4     6.9     3.41995      y
                      6     Tea     Celestial Saeson      7.8      13.8     6.3     3.82586      y
                      7     Tea     Celestial Saeson      7.8      13.8     6.3     3.33222      g
                      8     Tea     Lipton Tea Bags       6.7      10.0     5.7     3.28271      r
                      9     Tea     Lipton Family Si      8.9      11.1     8.2     4.39510      r


----------------------------------------------- CLUSTER=2 CLUSNAME=CL17 ------------------------------------------------

              Obs    class               name                height    width    depth     weight    color

               10    Detergent           Tide                 26.5      19.9      6.3    10.5928      r
               11    Detergent           Tide                 19.3      14.6      4.7     7.8357      r
               12    Detergent           Tide                 32.5      23.2      7.3    12.6889      r
               13    Breakfast cereal    Grape Nuts           21.7      16.3      4.9     8.7937      w
               14    Detergent           Arm & Hammer Det     33.7      22.8      7.0    14.7023      y
               15    Detergent           Arm & Hammer Det     27.8      19.4      6.3    12.2003      y
               16    Detergent           Arm & Hammer Det     38.8      30.0     16.9    22.4732      y
               17    Detergent           Tide                 39.4      24.8     11.3    16.1045      r
               18    Detergent           Arm & Hammer Det     39.5      25.8     11.0    18.6115      y


----------------------------------------------- CLUSTER=3 CLUSNAME=CL12 ------------------------------------------------

                  Obs        class        name            height    width    depth     weight    color

                   19    Paste, Tooth     Colgate           3.6      15.6     3.3     4.39510      r
                   20    Paste, Tooth     Crest             3.5      15.2     3.2     4.24343      w
                   21    Paste, Tooth     Crest             4.3      17.4     3.6     5.06813      w
                   22    Paste, Tooth     Arm & Hammer      4.4      17.0     3.7     5.21097      w
                   23    Paste, Tooth     Colgate           4.2      18.3     3.5     5.21097      r
                   24    Paste, Tooth     Crest             4.3      21.7     3.7     5.65790      w
                   25    Paste, Tooth     Colgate           4.4      22.0     3.5     5.82946      r
                   26    Little Debbie    Fudge Rounds      8.1      28.3     5.4     6.45411      w
                   27    Paste, Tooth     Crest             3.0      10.9     2.8     2.88670      w
```

```
----------------------------------------------- CLUSTER=4 CLUSNAME=CL13 -----------------------------------------------

              Obs    class               name                height    width    depth     weight    color

               28    Breakfast cereal    Cheerios             27.5      19.0     6.2     6.56541      y
               29    Breakfast cereal    Froot Loops          25.0      17.7     6.4     6.77735      r
               30    Breakfast cereal    Special K            30.1      20.5     8.5     7.98644      w
               31    Breakfast cereal    Corn Flakes          30.2      20.6     8.4     7.98644      w
               32    Breakfast cereal    Special K            29.6      19.2     6.7     6.97679      w
               33    Breakfast cereal    Corn Flakes          30.0      19.1     6.6     6.97679      w
               34    Breakfast cereal    Froot Loops          30.2      20.8     8.5     8.23034      r
               35    Breakfast cereal    Cheerios             30.3      20.4     7.2     7.51847      y
               36    Breakfast cereal    Cheerios             24.1      17.2     5.3     5.82848      y
               37    Breakfast cereal    Corn Flakes          33.7      25.4     8.0     8.79021      w
               38    Breakfast cereal    Special K            23.4      16.6     5.7     5.82946      w
               39    Breakfast cereal    Cheerios             32.5      22.4     8.4     8.27677      y
               40    Breakfast cereal    Shredded Wheat,      26.6      19.6     5.6     7.98957      r
               41    Crackers            Ritz                 23.1      16.0     5.3     6.97953      r
               42    Breakfast cereal    All-Bran             21.1      14.3     5.2     7.30951      y
               43    Tea                 Lipton Loose Tea     12.7      10.9     5.4     6.09479      r
               44    Crackers            Ritz                 23.1      20.7     5.2     7.68573      r


----------------------------------------------- CLUSTER=5 CLUSNAME=CL10 -----------------------------------------------

                Obs    class             name                height    width    depth     weight    color

                 45    Tea               Luzianne              8.9      22.8      6.4    5.53748      r
                 46    Tea               Luzianne Decaffe      8.9      22.8      6.4    5.29641      g
                 47    Crackers          Premium Saltines     11.0      25.0     10.7    7.68573      w
                 48    Tea               Lipton Family Si      8.7      20.8      8.2    5.53748      r
                 49    Little Debbie     Marshmallow Supr      9.4      32.0      7.0    6.56541      w
                 50    Tea               Lipton Family Si     13.7      24.0      9.0    6.97679      r
```

```
------------------------------------------------ CLUSTER=6 CLUSNAME=CL9 -----------------------------------------------

               Obs    class               name               height    width    depth     weight    color

                51    Tea                 Luzianne            18.4      20.2     6.9     6.09479      r
                52    Tea                 Lipton Tea Bags     17.1      20.0     6.7     6.09479      r
                53    Breakfast cereal    Shredded Wheat      19.7      19.9     7.5     6.56541      y
                54    Tea                 Lipton Tea Bags     11.5      14.4     6.6     4.73448      r


------------------------------------------------ CLUSTER=7 CLUSNAME=CL8 -----------------------------------------------

                Obs    class             name                height    width    depth     weight    color

                 55    Crackers          Wheatsworth          11.1      25.2     5.5    6.88239      w
                 56    Little Debbie     Swiss Cake Rolls     10.1      21.8     5.8    7.16545      w
                 57    Little Debbie     Figaroos             13.5      18.6     3.7    6.97679      y
                 58    Little Debbie     Nutty Bar            13.2      18.5     4.2    6.97679      y
                 59    Little Debbie     Apple Delights       11.2      30.1     4.9    7.51552      w
                 60    Little Debbie     Lemon Stix           13.2      18.5     4.2    6.33884      w
                 61    Little Debbie     Fudge Brownies       11.0      30.8     2.5    6.97679      w
                 62    Little Debbie     Snack Cakes          13.4      32.0     3.4    7.16545      b
                 63    Crackers          Waverly Wafers       14.4      22.5     6.2    7.68573      g
```

```
--------------------------------------------------------------------------------------------------------
|        |                                         class                                               |
|        |-----------------------------------------------------------------------------------------------|
|        |   Breakfast   |               |               |               |               |               |
|        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
|--------+---------------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |               |
|1       |               |               |               |               |               |      #########|
|2       |              #|               |       ########|               |               |               |
|3       |               |               |               |              #|       ########|               |
|4       | ##############|             ##|               |               |               |              #|
|5       |               |              #|               |              #|               |           ####|
|6       |              #|               |               |               |               |            ###|
|7       |               |             ##|               |        #######|               |               |
--------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------------------
|        |                                     color                                     |
|        |-------------------------------------------------------------------------------|
|        |     Blue      |     Green     |      Red      |     White     |    Yellow     |
|--------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |
|1       |             ##|             ##|            ###|               |             ##|
|2       |               |               |           ####|              #|           ####|
|3       |               |               |            ###|         ######|               |
|4       |               |               |         ######|         ######|          #####|
|5       |               |              #|            ###|             ##|               |
|6       |               |               |            ###|               |              #|
|7       |              #|              #|               |          #####|             ##|
------------------------------------------------------------------------------------------
```

The last several analyses include color. Obviously, the dummy variables must not be included in calculations to standardize the rows. If the five dummy variables are simply standardized to variance 1.0 and included with the other variables, color dominates the analysis. The dummy variables should be scaled to a smaller variance, which must be determined by trial and error. Four analyses are done using PROC STANDARD to scale the dummy variables to a standard deviation of 0.2, 0.3, 0.4, or 0.8. The cluster listings are suppressed.

Since dummy variables drastically violate the normality assumption on which the CCC depends, the CCC tends to indicate an excessively large number of clusters.

```
/************************************************************/
/*                                                          */
/* Analyses 4-7: standardized row-standardized logs & color */
/*                                                          */
/************************************************************/
%let list=0;
%let crosscol=1;

title2
  'Analysis 4: Standardized row-standardized
               logarithms and color (s=.2)';
proc standard data=stdstan out=stdstan m=0 s=.2;
   var c_:;
run;

proc cluster data=stdstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight c_:;
   id name;
   copy class height width depth weight color;
run;

%show(7);

title2
  'Analysis 5: Standardized row-standardized
               logarithms and color (s=.3)';
proc standard data=stdstan out=stdstan m=0 s=.3;
   var c_:;
run;
```

```
proc cluster data=stdstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight c_:;
   id name;
   copy class height width depth weight color;
run;

%show(6);

title2
  ’Analysis 6: Standardized row-standardized
               logarithms and color (s=.4)’;
proc standard data=stdstan out=stdstan m=0 s=.4;
   var c_:;
run;

proc cluster data=stdstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight c_:;
   id name;
   copy class height width depth weight color;
run;

%show(3);

title2
  ’Analysis 7: Standardized row-standardized
               logarithms and color (s=.8)’;
proc standard data=stdstan out=stdstan m=0 s=.8;
   var c_:;
run;

proc cluster data=stdstan m=cen %clusopt outtree=tree;
   var l_height l_width l_depth l_weight c_:;
   id name;
   copy class height width depth weight color;
run;

%show(10);
```

Using PROC STANDARD on the dummy variables with S=0.2 causes four of the Little Debbie products to join the toothpastes. Using S=0.3 causes one of the tea clusters to merge with the breakfast cereals while three cereals defect to the detergents. Using S=0.4 produces three clusters consisting of (1) cereals and detergents, (2) Little Debbie products and toothpaste, and (3) teas, with crackers divided among all three clusters and a few other misclassifications. With S=0.8, ten clusters are indicated, each entirely monochrome. So, S=0.2 or S=0.3 degrades the classification, S=0.4 yields a good but perhaps excessively coarse classification, and higher values of the S= option produce clusters that are determined mainly by color.

Output 23.6.4. Analysis of Standardized Row-Standardized Logarithms and Color

```
                                  Cluster Analysis of Grocery Boxes
                Analysis 4: Standardized row-standardized logarithms and color (s=.2)

                                        The CLUSTER Procedure
                                Centroid Hierarchical Cluster Analysis

                                 Eigenvalues of the Covariance Matrix

                             Eigenvalue    Difference    Proportion    Cumulative

                        1    2.43584975    0.94791932        0.5800        0.5800
                        2    1.48793042    1.39363531        0.3543        0.9342
                        3    0.09429511    0.03686218        0.0225        0.9567
                        4    0.05743293    0.01036136        0.0137        0.9704
                        5    0.04707157    0.00489503        0.0112        0.9816
                        6    0.04217654    0.00693298        0.0100        0.9916
                        7    0.03524355    0.03524355        0.0084        1.0000
                        8    0.00000000    0.00000000        0.0000        1.0000
                        9    -.00000000                     -0.0000        1.0000


                     Root-Mean-Square Total-Sample Standard Deviation =  0.68313
                     Root-Mean-Square Distance Between Observations   = 2.898275


                                           Cluster History
                                                                                              Norm    T
                                                                                              Cent    i
 NCL    --Clusters Joined---       FREQ    SPRSQ     RSQ    ERSQ     CCC     PSF    PST2      Dist    e

  20    CL46        OB37              3   0.0016    .968    .        .      67.5    11.9    0.2706
  19    OB46        OB52              2   0.0014    .966    .        .      69.7      .     0.2995
  18    CL25        CL37              6   0.0041    .962    .        .      67.1     5.0    0.3081
  17    CL33        CL35             16   0.0099    .952    .        .      57.2    16.7    0.3196
  16    CL19        OB48              3   0.0024    .950    .        .      59.2     1.7    0.3357
  15    CL30        CL16              5   0.0042    .946    .        .      59.5     2.7    0.3299
  14    CL27        CL18              8   0.0057    .940    .        .      58.9     4.2    0.3429
  13    CL20        OB32              4   0.0031    .937    .        .      61.7     3.6    0.3564
  12    CL24        OB50              4   0.0031    .934    .905    3.23    65.2     4.7     0.359
  11    CL39        CL28              6   0.0068    .927    .896    3.17    65.9    12.1    0.3743
  10    CL13        OB35              5   0.0036    .923    .886    3.62    70.8     2.3    0.3755
   9    CL11        CL32             13   0.0176    .906    .874    2.70    64.8    16.0    0.4107
   8    CL14        OB54              9   0.0052    .900    .859    3.29    71.0     2.6    0.4265
   7    OB21        CL10              6   0.0052    .895    .841    4.09    79.8     2.4    0.4378
   6    CL17        CL12             20   0.0248    .870    .817    3.52    76.6    19.7    0.4898
   5    CL15        CL8              14   0.0326    .838    .783    3.08    75.0    14.0    0.5607
   4    CL6         CL21             30   0.0743    .764    .734    1.35    63.5    35.6    0.5877
   3    CL9         CL7              19   0.0579    .706    .653    2.17    72.0    22.8    0.6611
   2    CL4         CL3              49   0.3632    .343    .450    -2.6    31.8    73.0    0.9838
   1    CL2         CL5              63   0.3426    .000    .000    0.00      .     31.8    0.9876
```

```
        ----------------------------------------------------------------------------------------------------------
        |        |                                             class                                             |
        |        |-----------------------------------------------------------------------------------------------|
        |        |   Breakfast   |               |               |               |               |               |
        |        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
        |--------+---------------+---------------+---------------+---------------+---------------+---------------|
        |CLUSTER |               |               |               |               |               |               |
        |1       |             ##|               |       ########|               |               |               |
        |2       |               |              #|               |           ####|       ########|               |
        |3       |  #############|             ##|               |               |               |              #|
        |4       |              #|               |               |               |               |            ###|
        |5       |               |              #|               |          #####|               |               |
        |6       |               |               |               |               |               |      #########|
        |7       |               |              #|               |               |               |           ####|
        ----------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------------------
|        |                                  color                                        |
|        |-------------------------------------------------------------------------------|
|        |     Blue      |     Green     |      Red      |     White     |    Yellow     |
|--------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |
|1       |               |               |           ####|              #|          #####|
|2       |               |               |            ###|     ##########|               |
|3       |               |               |         ######|         ######|           ####|
|4       |               |               |            ###|               |              #|
|5       |              #|              #|               |             ##|             ##|
|6       |             ##|             ##|            ###|               |             ##|
|7       |               |              #|            ###|              #|               |
------------------------------------------------------------------------------------------
```

```
                                  Cluster Analysis of Grocery Boxes
                    Analysis 5: Standardized row-standardized logarithms and color (s=.3)

                                       The CLUSTER Procedure
                                Centroid Hierarchical Cluster Analysis

                                 Eigenvalues of the Covariance Matrix

                             Eigenvalue    Difference    Proportion    Cumulative

                        1    2.44752302    0.95026671        0.5500        0.5500
                        2    1.49725632    1.36701945        0.3365        0.8865
                        3    0.13023687    0.02135049        0.0293        0.9157
                        4    0.10888637    0.00867367        0.0245        0.9402
                        5    0.10021271    0.00628821        0.0225        0.9627
                        6    0.09392449    0.02196469        0.0211        0.9838
                        7    0.07195981    0.07195981        0.0162        1.0000
                        8    0.00000000    0.00000000        0.0000        1.0000
                        9    -.00000000                     -0.0000        1.0000


                     Root-Mean-Square Total-Sample Standard Deviation = 0.703167
                     Root-Mean-Square Distance Between Observations   = 2.983287


                                          Cluster History
                                                                                          Norm    T
                                                                                          Cent    i
    NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2      Dist    e

     20    CL24        CL28             4    0.0038    .953    .        .      45.7     2.7    0.3448
     19    OB11        CL23             6    0.0033    .950    .        .      46.0     3.5    0.3477
     18    CL46        OB37             3    0.0027    .947    .        .      47.1    21.9    0.3558
     17    CL21        OB50             4    0.0031    .944    .        .      48.2     2.5    0.3577
     16    CL39        CL33             6    0.0064    .937    .        .      46.9    12.1    0.3637
     15    CL19        CL29            14    0.0152    .922    .        .      40.6    12.4    0.3707
     14    CL18        OB32             4    0.0035    .919    .        .      42.5     2.5    0.3813
     13    CL16        CL25            13    0.0175    .901    .        .      38.0    13.7    0.4103
     12    CL22        OB54             5    0.0049    .896    .875    1.76    40.0     3.2    0.4353
     11    CL12        CL37             7    0.0089    .887    .865    1.71    40.9     4.6    0.4397
     10    CL20        OB48             5    0.0056    .882    .854    2.02    43.9     2.5    0.4669
      9    CL26        CL17            16    0.0222    .859    .841    1.20    41.3    16.6     0.479
      8    CL32        CL11             9    0.0125    .847    .826    1.31    43.5     4.5    0.4988
      7    CL14        OB35             5    0.0070    .840    .806    1.95    49.0     3.3     0.519
      6    OB21        CL7              6    0.0077    .832    .782    2.79    56.6     2.3    0.5366
      5    CL9         CL15            30    0.0716    .761    .749    0.54    46.1    28.3    0.5452
      4    CL10        CL8             14    0.0318    .729    .700    1.21    52.9     8.6    0.5542
      3    CL5         CL6             36    0.0685    .660    .622    1.50    58.3    14.2    0.6516
      2    CL13        CL4             27    0.2008    .460    .427    0.90    51.9    46.6    0.9611
      1    CL3         CL2             63    0.4595    .000    .000    0.00      .     51.9    0.9609
```

```
----------------------------------------------------------------------------------------------------------
|        |                                             class                                             |
|        |-----------------------------------------------------------------------------------------------|
|        |   Breakfast   |               |               |               |               |               |
|        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
|--------+---------------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |               |
|1       |            ###|             ##|       ########|               |               |              #|
|2       |               |              #|               |           ####|       ########|               |
|3       |  #############|               |               |               |               |            ###|
|4       |               |              #|               |          #####|               |               |
|5       |               |               |               |               |               |      #########|
|6       |               |              #|               |               |               |           ####|
----------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------------------
|        |                                     color                                     |
|        |-------------------------------------------------------------------------------|
|        |     Blue      |     Green     |      Red      |     White     |    Yellow     |
|--------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |
|1       |               |               |       ########|              #|          #####|
|2       |               |               |            ###|     ##########|               |
|3       |               |               |          #####|         ######|          #####|
|4       |              #|              #|               |             ##|             ##|
|5       |             ##|             ##|            ###|               |             ##|
|6       |               |              #|            ###|              #|               |
------------------------------------------------------------------------------------------
```

```
                             Cluster Analysis of Grocery Boxes
               Analysis 6: Standardized row-standardized logarithms and color (s=.4)

                                  The CLUSTER Procedure
                            Centroid Hierarchical Cluster Analysis

                             Eigenvalues of the Covariance Matrix

                           Eigenvalue    Difference    Proportion    Cumulative

                      1    2.46469435    0.95296119        0.5135        0.5135
                      2    1.51173316    1.28149311        0.3149        0.8284
                      3    0.23024005    0.04306536        0.0480        0.8764
                      4    0.18717469    0.01766446        0.0390        0.9154
                      5    0.16951023    0.01827481        0.0353        0.9507
                      6    0.15123542    0.06582379        0.0315        0.9822
                      7    0.08541162    0.08541162        0.0178        1.0000
                      8    0.00000000    0.00000000        0.0000        1.0000
                      9    -.00000000                     -0.0000        1.0000


                   Root-Mean-Square Total-Sample Standard Deviation = 0.730297
                   Root-Mean-Square Distance Between Observations   = 3.098387


                                       Cluster History
                                                                                    Norm    T
                                                                                    Cent    i
 NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC      PSF    PST2      Dist    e

  20    CL29        CL44            10    0.0074    .955     .        .      47.7     8.2    0.3789
  19    CL38        OB54             3    0.0031    .952     .        .      48.1     9.3    0.3792
  18    CL25        CL41            11    0.0155    .936     .        .      38.8    36.7    0.4192
  17    CL23        CL43            10    0.0120    .924     .        .      35.0    11.6    0.4208
  16    OB11        CL26             6    0.0050    .919     .        .      35.6     5.8    0.4321
  15    CL19        CL31             5    0.0074    .912     .        .      35.4     5.3    0.4362
  14    OB20        CL27             4    0.0046    .907     .        .      36.8     2.9    0.4374
  13    CL18        CL20            21    0.0352    .872     .        .      28.4    19.7    0.4562
  12    CL13        CL16            27    0.0372    .835    .839    -.37     23.4    12.0    0.4968
  11    CL21        CL17            15    0.0289    .806    .828    -1.5     21.6    13.6    0.5183
  10    CL14        CL15             9    0.0200    .786    .815    -1.8     21.6     7.2    0.5281
   9    OB21        OB48             2    0.0047    .781    .801    -1.2     24.1      .     0.5425
   8    CL10        CL24            12    0.0243    .757    .785    -1.3     24.5     5.8    0.5783
   7    CL12        CL46            29    0.0224    .735    .765    -1.3     25.8     5.3    0.6105
   6    CL8         CL37            14    0.0220    .712    .740    -1.1     28.3     4.0    0.6313
   5    CL6         CL32            16    0.0251    .687    .707    -.78     31.9     3.9    0.6664
   4    CL11        CL9             17    0.0287    .659    .660    -.04     38.0     7.0    0.7098
   3    CL4         OB35            18    0.0180    .641    .584    2.21     53.5     3.2    0.7678
   2    CL3         CL5             34    0.2175    .423    .400    0.67     44.8    31.4    0.8923
   1    CL7         CL2             63    0.4232    .000    .000    0.00       .     44.8    0.9156
```

```
      ------------------------------------------------------------------------------------------------------
      |        |                                        class                                              |
      |        |-------------------------------------------------------------------------------------------|
      |        |   Breakfast    |               |               |               |               |               |
      |        |    cereal      |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
      |--------+---------------+---------------+---------------+---------------+---------------+---------------|
      |CLUSTER |               |               |               |               |               |               |
      |1       |>##############|             ##|       ########|             ##|               |              #|
      |2       |               |             ##|               |        #######|       ########|              #|
      |3       |               |              #|               |               |               |>##############|
      ------------------------------------------------------------------------------------------------------
```

```
              -------------------------------------------------------------------------------------
              |        |                                  color                                   |
              |        |--------------------------------------------------------------------------|
              |        |     Blue      |     Green     |      Red      |     White     |    Yellow     |
              |--------+---------------+---------------+---------------+---------------+---------------|
              |CLUSTER |               |               |               |               |               |
              |1       |               |               |     ##########|        #######|   ############|
              |2       |              #|             ##|            ###|   ############|               |
              |3       |             ##|             ##|      #########|              #|             ##|
              -------------------------------------------------------------------------------------
```

```
                         Cluster Analysis of Grocery Boxes
         Analysis 7: Standardized row-standardized logarithms and color (s=.8)

                               The CLUSTER Procedure
                        Centroid Hierarchical Cluster Analysis

                          Eigenvalues of the Covariance Matrix

                         Eigenvalue    Difference    Proportion    Cumulative

                    1    2.61400794    0.93268930        0.3631        0.3631
                    2    1.68131864    0.77645948        0.2335        0.5966
                    3    0.90485916    0.22547234        0.1257        0.7222
                    4    0.67938683    0.00292216        0.0944        0.8166
                    5    0.67646466    0.12119211        0.0940        0.9106
                    6    0.55527255    0.46658428        0.0771        0.9877
                    7    0.08868827    0.08868827        0.0123        1.0000
                    8    0.00000000    0.00000000        0.0000        1.0000
                    9    -.00000000                     -0.0000        1.0000


              Root-Mean-Square Total-Sample Standard Deviation = 0.894427
              Root-Mean-Square Distance Between Observations   = 3.794733


                                    Cluster History
                                                                             Norm    T
                                                                             Cent    i
  NCL    --Clusters Joined---      FREQ     SPRSQ     RSQ    ERSQ     CCC     PSF    PST2    Dist    e

   20    CL29        CL44            10    0.0049    .970      .       .    72.7     8.2    0.3094
   19    CL38        OB54             3    0.0021    .968      .       .    73.3     9.3    0.3096
   18    CL21        CL23            12    0.0153    .952      .       .    53.0    15.0    0.4029
   17    OB21        OB48             2    0.0032    .949      .       .    53.8      .      0.443
   16    CL27        CL24             6    0.0095    .940      .       .    48.9    10.4     0.444
   15    CL19        CL16             9    0.0136    .926      .       .    43.0     6.1    0.4587
   14    CL41        OB11             7    0.0058    .920      .       .    43.6    51.2    0.4591
   13    CL26        CL46             7    0.0105    .910      .       .    42.1    22.0    0.4769
   12    CL25        CL13            12    0.0205    .889    .743    16.5    37.3    13.8     0.467
   11    CL18        OB20            13    0.0093    .880    .726    16.7    38.2     4.0    0.5586
   10    CL17        CL37             4    0.0134    .867    .706    16.5    38.3     7.9    0.6454
    9    CL14        CL20            17    0.0567    .810    .684    11.0    28.8    52.6    0.6534
    8    CL12        CL9             29    0.0828    .727    .659    5.03    20.9    20.7     0.604
    7    CL11        CL43            16    0.0359    .691    .631    4.25    20.9    14.4    0.6758
    6    CL15        CL31            11    0.0263    .665    .598    4.24    22.6     8.0    0.7065
    5    CL7         CL6             27    0.1430    .522    .557    -1.7    15.8    28.2    0.8247
    4    CL8         CL5             56    0.2692    .253    .507    -9.1     6.6    31.5    0.7726
    3    OB35        CL32             3    0.0216    .231    .435    -6.6     9.0    46.0    1.0027
    2    CL4         CL10            60    0.1228    .108    .289    -5.6     7.4     9.5    1.0096
    1    CL2         CL3             63    0.1083    .000    .000    0.00      .      7.4    1.0839
```

```
----------------------------------------------------------------------------------------------------------
|        |                                             class                                             |
|        |-----------------------------------------------------------------------------------------------|
|        |   Breakfast   |               |               |               |               |               |
|        |    cereal     |   Crackers    |   Detergent   | Little Debbie | Paste, Tooth  |      Tea      |
|--------+---------------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |               |
|1       |            ###|             ##|           ####|               |               |              #|
|2       |               |             ##|               |         ######|          #####|               |
|3       |        #######|               |               |               |               |               |
|4       |         ######|               |           ####|             ##|               |               |
|5       |               |               |               |               |            ###|               |
|6       |               |               |               |               |               |      #########|
|7       |               |              #|               |               |               |            ###|
|8       |               |               |               |               |               |             ##|
|9       |               |               |               |               |               |             ##|
|10      |               |               |               |              #|               |               |
----------------------------------------------------------------------------------------------------------
```

```
------------------------------------------------------------------------------------------
|        |                                     color                                     |
|        |-------------------------------------------------------------------------------|
|        |     Blue      |     Green     |      Red      |     White     |    Yellow     |
|--------+---------------+---------------+---------------+---------------+---------------|
|CLUSTER |               |               |               |               |               |
|1       |               |               |     ##########|               |               |
|2       |               |               |               |  #############|               |
|3       |               |               |               |        #######|               |
|4       |               |               |               |               |   ############|
|5       |               |               |            ###|               |               |
|6       |               |               |      #########|               |               |
|7       |               |           ####|               |               |               |
|8       |             ##|               |               |               |               |
|9       |               |               |               |               |             ##|
|10      |              #|               |               |               |               |
------------------------------------------------------------------------------------------
```

References

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, New York: Academic Press, Inc.

Batagelj, V. (1981), “Note on Ultrametric Hierarchical Clustering Algorithms,” *Psychometrika*, 46, 351–352.

Blackith, R.E. and Reyment, R.A. (1971), *Multivariate Morphometrics*, London: Academic Press.

Blashfield, R.K. and Aldenderfer, M.S. (1978), “The Literature on Cluster Analysis,” *Multivariate Behavioral Research*, 13, 271–295.

Calinski, T. and Harabasz, J. (1974), “A Dendrite Method for Cluster Analysis,” *Communications in Statistics*, 3, 1–27.

Cooper, M.C. and Milligan, G.W. (1988), “The Effect of Error on Determining the Number of Clusters,” *Proceedings of the International Workship on Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing and Related Areas of Research*, 319–328.

Duda, R.O. and Hart, P.E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons, Inc.

Everitt, B.S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books Ltd.

Fisher, L. and Van Ness, J.W. (1971), “Admissible Clustering Procedures,” *Biometrika*, 58, 91–104.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Florek, K., Lukaszewicz, J., Perkal, J., and Zubrzycki, S. (1951a), “Sur la Liaison et la Division des Points d’un Ensemble Fini,” *Colloquium Mathematicae*, 2, 282–285.

Florek, K., Lukaszewicz, J., Perkal, J., and Zubrzycki, S. (1951b), “Taksonomia Wroclawska,” *Przeglad Antropol.*, 17, 193–211.

Gower, J.C. (1967), “A Comparison of Some Methods of Cluster Analysis,” *Biometrics*, 23, 623–637.

Hamer, R.M. and Cunningham, J.W. (1981), “Cluster analyzing profile data with interrater differences: A comparison of profile association measures,” *Applied Psychological Measurement*, 5, 63–72.

Hartigan, J.A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons, Inc.

Hartigan, J.A. (1977), “Distribution Problems in Clustering,” in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press, Inc.

Hartigan, J.A. (1981), “Consistency of Single Linkage for High-density Clusters,” *Journal of the American Statistical Association*, 76, 388–394.

Hawkins, D.M., Muller, M.W., and ten Krooden, J.A. (1982), “Cluster Analysis,” in *Topics in Applied Multivariate Analysis*, ed. D.M. Hawkins, Cambridge: Cambridge University Press.

Jardine, N. and Sibson, R. (1971), *Mathematical Taxonomy*, New York: John Wiley & Sons, Inc.

Johnson, S.C. (1967), “Hierarchical Clustering Schemes,” *Psychometrika*, 32, 241–254.

Lance, G.N. and Williams, W.T. (1967), “A General Theory of Classificatory Sorting Strategies. I. Hierarchical Systems,” *Computer Journal*, 9, 373–380.

Massart, D.L. and Kaufman, L. (1983), *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*, New York: John Wiley & Sons, Inc.

McQuitty, L.L. (1957), “Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies,” *Educational and Psychological Measurement*, 17, 207–229.

McQuitty, L.L. (1966), “Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data,” *Educational and Psychological Measurement*, 26, 825–831.

Mezzich, J.E and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press, Inc.

Milligan, G.W. (1979), “Ultrametric Hierarchical Clustering Algorithms,” *Psychometrika*, 44, 343–346.

Milligan, G.W. (1980), “An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms,” *Psychometrika*, 45, 325–342.

Milligan, G.W. (1987), “A Study of the Beta-Flexible Clustering Method,” *College of Administrative Science Working Paper Series*, 87–61 Columbus, OH: The Ohio State University.

Milligan, G.W. and Cooper, M.C. (1985), “An Examination of Procedures for Determining the Number of Clusters in a Data Set,” *Psychometrika*, 50,159–179.

Milligan, G.W. and Cooper, M.C. (1987), “A Study of Variable Standardization,” *College of Administrative Science Working Paper Series*, 87–63, Columbus, OH: The Ohio State University.

Rouncefield, M. (1995), "The Statistics of Poverty and Inequality," *Journal of Statistics Education*, 3(2). [Online]: [http://www.stat.ncsu.edu/info/jse], accessed Dec. 19, 1997.

Sarle, W.S. (1983), *Cubic Clustering Criterion*, SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Silverman, B.W. (1986), *Density Estimation*, New York: Chapman and Hall.

Sneath, P.H.A. (1957), "The Application of Computers to Taxonomy," *Journal of General Microbiology*, 17, 201–226.

Sneath, P.H.A. and Sokal, R.R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

Sokal, R.R. and Michener, C.D. (1958), "A Statistical Method for Evaluating Systematic Relationships," *University of Kansas Science Bulletin*, 38, 1409–1438.

Sorensen, T. (1948), "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons," *Biologiske Skrifter*, 5, 1–34.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, England: Ellis Horwood.

Symons, M.J. (1981), "Clustering Criteria and Multivariate Normal Mixtures," *Biometrics*, 37, 35–43.

Ward, J.H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244.

Wishart, D. (1969), "Mode Analysis: A Generalisation of Nearest Neighbour Which Reduces Chaining Effects," in *Numerical Taxonomy*, ed. A.J. Cole, London: Academic Press.

Wong, M.A. (1982), "A Hybrid Clustering Method for Identifying High-Density Clusters," *Journal of the American Statistical Association*, 77, 841–847.

Wong, M.A. and Lane, T. (1983), "A kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M.A. and Schaack, C. (1982), "Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 24
The CORRESP Procedure

Chapter Contents

Chapter 24
The CORRESP Procedure

Overview

The CORRESP procedure performs simple and multiple correspondence analysis. You can use correspondence analysis to find a low-dimensional graphical representation of the rows and columns of a crosstabulation or contingency table. Each row and column is represented by a point in a plot determined from the cell frequencies. PROC CORRESP can also compute coordinates for supplementary rows and columns. Experimental graphics are now available with the CORRESP procedure. For more information, see the "ODS Graphics" section on page 1109.

PROC CORRESP can read two kinds of input: raw categorical responses on two or more classification variables, and a two-way contingency table. The correspondence analysis results can be output and displayed with the %PLOTIT macro.

Background

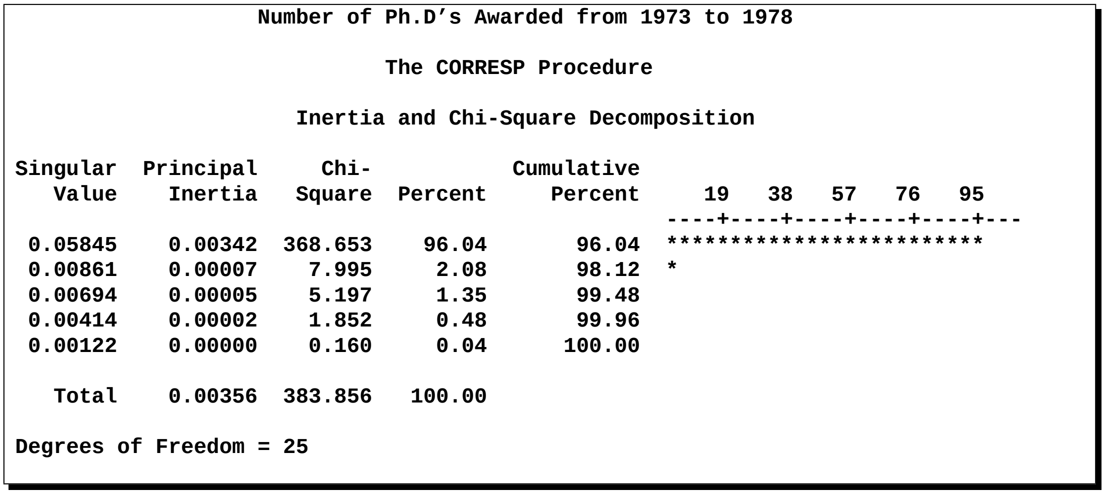
Correspondence analysis is a popular data analysis method in France and Japan. In France, correspondence analysis was developed under the strong influence of Jean-Paul Benzécri; in Japan, it was developed under Chikio Hayashi. The name *correspondence analysis* is a translation of the French *analyse des correspondances*. The technique apparently has many independent beginnings (for example, Richardson and Kuder 1933; Hirshfeld 1935; Horst 1935; Fisher 1940; Guttman 1941; Burt 1950; Hayashi 1950). It has had many other names, including optimal scaling, reciprocal averaging, optimal scoring, and appropriate scoring in the United States; quantification method in Japan; homogeneity analysis in the Netherlands; dual scaling in Canada; and scalogram analysis in Israel.

Correspondence analysis is described in more detail in French in Benzécri (1973) and Lebart, Morineau, and Tabard (1977). In Japanese, the subject is described in Komazawa (1982), Nishisato (1982), and Kobayashi (1981). In English, correspondence analysis is described in Lebart, Morineau, and Warwick (1984), Greenacre (1984), Nishisato (1980), Tenenhaus and Young (1985); Gifi (1990); Greenacre and Hastie (1987); and many other sources. Hoffman and Franke (1986) offer a short, introductory treatment using examples from the field of market research.

Getting Started

Data are available containing the numbers of Ph.Ds awarded in the United States during the years 1973 through 1978 (U.S. Bureau of the Census 1979). The table has six rows, one for each of six academic disciplines, and six columns for the six years. The following DATA step reads the complete table into a SAS data set, and PROC CORRESP displays correspondence analysis results including the inertia decomposition and coordinates. The concept of *inertia* in correspondence analysis is analogous to the concept of variance in principal component analysis, and it is proportional to the chi-square information. The %PLOTIT macro creates a graphical scatterplot of the results. See Appendix B, “Using the %PLOTIT Macro,” for more information on the %PLOTIT macro.

```
title "Number of Ph.D’s Awarded from 1973 to 1978";
data PhD;
   input Science $ 1-19 y1973-y1978;
   label y1973 = ’1973’
         y1974 = ’1974’
         y1975 = ’1975’
         y1976 = ’1976’
         y1977 = ’1977’
         y1978 = ’1978’;
   datalines;
Life Sciences       4489 4303 4402 4350 4266 4361
Physical Sciences   4101 3800 3749 3572 3410 3234
Social Sciences     3354 3286 3344 3278 3137 3008
Behavioral Sciences 2444 2587 2749 2878 2960 3049
Engineering         3338 3144 2959 2791 2641 2432
Mathematics         1222 1196 1149 1003  959  959
;

proc corresp data=PhD out=Results short;
   var y1973-y1978;
   id Science;
run;

%plotit(data=Results, datatype=corresp, plotvars=Dim1 Dim2)
```



```
                      Number of Ph.D’s Awarded from 1973 to 1978

                                The CORRESP Procedure

                         Inertia and Chi-Square Decomposition

Singular  Principal     Chi-             Cumulative
   Value    Inertia   Square  Percent       Percent    19   38   57   76   95
                                                     ----+----+----+----+----+---
 0.05845    0.00342  368.653    96.04         96.04  *************************
 0.00861    0.00007    7.995     2.08         98.12  *
 0.00694    0.00005    5.197     1.35         99.48
 0.00414    0.00002    1.852     0.48         99.96
 0.00122    0.00000    0.160     0.04        100.00

   Total    0.00356  383.856   100.00

Degrees of Freedom = 25
```

Figure 24.1. Inertia and Chi-Square Decomposition

The total chi-square statistic, which is a measure of the association between the rows and columns in the full five dimensions of the (centered) table, is 383.856. The maximum number of dimensions (or axes) is the minimum of the number of rows and columns, minus one. Over 96% of the total chi-square and inertia is explained by the first dimension, indicating that the association between the row and column categories is essentially one dimensional. The plot shows how the number of doctorates in the different areas changes over time. The plot shows that the number of doctorates in the behavioral sciences is associated with later years, and the number of doctorates in mathematics and engineering is associated with earlier years. This is consistent with the data which shows that number of doctorates in the behavioral sciences is increasing, the number of doctorates in every other discipline is decreasing, and the rate of decrease is greatest for mathematics and engineering.

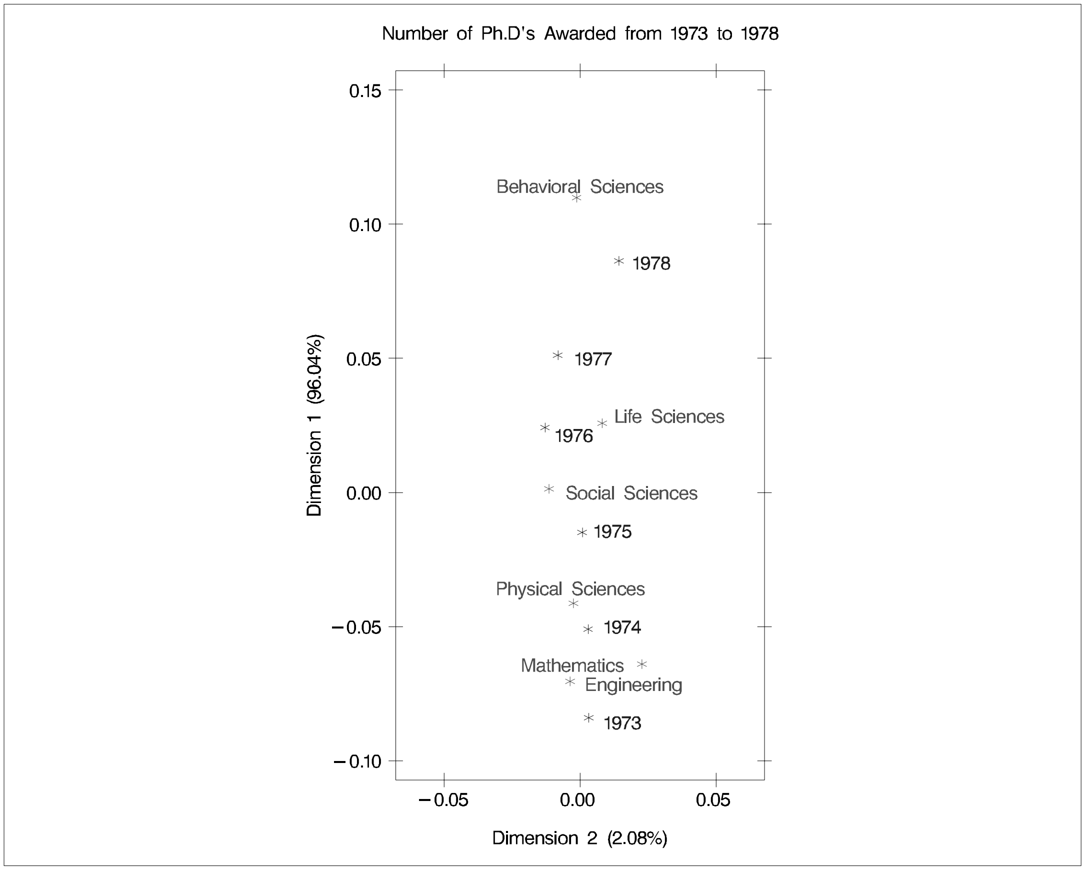


Figure 24.2. Plot of Dimension 1 versus Dimension 2 for Ph.D. Data

Syntax

The following statements are available in the CORRESP procedure.

PROC CORRESP < *options* > **;**
  TABLES < *row-variables,* > *column-variables* **;**
  VAR *variables* **;**
  BY *variables* **;**
  ID *variable* **;**
  SUPPLEMENTARY *variables* **;**
  WEIGHT *variable* **;**

There are two separate forms of input to PROC CORRESP. One form is specified in the TABLES statement, the other in the VAR statement. You must specify either the TABLES or the VAR statement, but not both, each time you run PROC CORRESP.

Specify the TABLES statement if you are using raw, categorical data, the levels of which define the rows and columns of a table.

Specify the VAR statement if your data are already in tabular form. PROC CORRESP is generally more efficient with VAR statement input than with TABLES statement input.

The other statements are optional. Each of the statements is explained in alphabetical order following the PROC CORRESP statement. All of the options in PROC CORRESP can be abbreviated to their first three letters, except for the OUTF= option. This is a special feature of PROC CORRESP and is not generally true of SAS/STAT procedures.

PROC CORRESP Statement

PROC CORRESP < *options* > **;**

The PROC CORRESP statement invokes the procedure. You can specify the following options in the PROC CORRESP statement. These options are described following Table 24.1.

Table 24.1. Summary of PROC CORRESP Statement Options

Task	Options
Specify data sets	
specify input SAS data set	DATA=
specify output coordinate SAS data set	OUTC=
specify output frequency SAS data set	OUTF=
Compute row and column coordinates	
specify the number of dimensions or axes	DIMENS=
perform multiple correspondence analysis	MCA
standardize the row and column coordinates	PROFILE=
Construct tables	
specify binary table	BINARY
specify cross levels of TABLES variables	CROSS=
specify input data in PROC FREQ output	FREQOUT
include observations with missing values	MISSING
Display output	
display all output	ALL
display inertias adjusted by Benzécri's method	BENZECRI
display cell contributions to chi-square	CELLCHI2
display column profile matrix	CP
display observed minus expected values	DEVIATION
display chi-square expected values	EXPECTED
display inertias adjusted by Greenacre's method	GREENACRE
suppress the display of column coordinates	NOCOLUMN=
suppress the display of all output	NOPRINT
suppress the display of row coordinates	NOROW=
display contingency table of observed frequencies	OBSERVED
display percentages or frequencies	PRINT=
display row profile matrix	RP
suppress all point and coordinate statistics	SHORT

Task	**Options**
display unadjusted inertias	UNADJUSTED
Other tasks	
specify rarely used column coordinate standardizations	COLUMN=
specify minimum inertia	MININERTIA=
specify number of classification variables	NVARS=
specify rarely used row coordinate standardizations	ROW=
specify effective zero	SINGULAR=
include level source in the OUTC= data set	SOURCE

The display options control the amount of displayed output. The CELLCHI2, EXPECTED, and DEVIATION options display additional chi-square information. See the "Details" section on page 1082 for more information. The unit of the matrices displayed by the CELLCHI2, CP, DEVIATION, EXPECTED, OBSERVED, and RP options depends on the value of the PRINT= option. The table construction options control the construction of the contingency table; these options are valid only when you also specify a TABLES statement.

You can specify the following options in the PROC CORRESP statement. They are described in alphabetical order.

ALL

is equivalent to specifying the OBSERVED, RP, CP, CELLCHI2, EXPECTED, and DEVIATION options. Specifying the ALL option does not affect the PRINT= option. Therefore, only frequencies (not percentages) for these options are displayed unless you specify otherwise with the PRINT= option.

BENZECRI | BEN

displays adjusted inertias when performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, adjusted inertias using a method proposed by Benzécri (1979) and described by Greenacre (1984, p. 145) can be displayed by specifying the BENZECRI option. Specify the UNADJUSTED option to output the usual table of unadjusted inertias as well. See the section "MCA Adjusted Inertias" on page 1102 for more information.

BINARY

enables you to create binary tables easily. When you specify the BINARY option, specify only column variables in the TABLES statement. Each input data set observation forms a single row in the constructed table.

CELLCHI2 | CEL

displays the contribution to the total chi-square test statistic for each cell. See also the descriptions of the DEVIATION, EXPECTED, and OBSERVED options.

COLUMN=B | BD | DB | DBD | DBD1/2 | DBID1/2
COL=B | BD | DB | DBD | DBD1/2 | DBID1/2

provides other standardizations of the column coordinates. The COLUMN= option is rarely needed. Typically, you should use the PROFILE= option instead (see the section "The PROFILE=, ROW=, and COLUMN= Options" on page 1099). By default, COLUMN=DBD.

CP

displays the column profile matrix. Column profiles contain the observed conditional probabilities of row membership given column membership. See also the RP option.

CROSS=BOTH | COLUMN | NONE | ROW
CRO=BOT | COL | NON | ROW

specifies the method of crossing (factorially combining) the levels of the TABLES variables. The default is CROSS=NONE.

- CROSS=NONE causes each level of every row variable to become a row label and each level of every column variable to become a column label.
- CROSS=ROW causes each combination of levels for all row variables to become a row label, whereas each level of every column variable becomes a column label.
- CROSS=COLUMN causes each combination of levels for all column variables to become a column label, whereas each level of every row variable becomes a row label.
- CROSS=BOTH causes each combination of levels for all row variables to become a row label and each combination of levels for all column variables to become a column label.

The "TABLES Statement" section on page 1081 provides a more detailed description of this option.

DATA=*SAS-data-set*

specifies the SAS data set to be used by PROC CORRESP. If you do not specify the DATA= option, PROC CORRESP uses the most recently created SAS data set.

DEVIATION | DEV

displays the matrix of deviations between the observed frequency matrix and the product of its row marginals and column marginals divided by its grand frequency. For ordinary two-way contingency tables, these are the observed minus expected frequencies under the hypothesis of row and column independence and are components of the chi-square test statistic. See also the CELLCHI2, EXPECTED, and OBSERVED options.

DIMENS=*n*
DIM=*n*

specifies the number of dimensions or axes to use. The default is DIMENS=2. The maximum value of the DIMENS= option in an $(n_r \times n_c)$ table is $n_r - 1$ or $n_c - 1$, whichever is smaller. For example, in a table with 4 rows and 5 columns, the maximum specification is DIMENS=3. If your table has 2 rows or 2 columns, specify DIMENS=1.

EXPECTED | EXP

displays the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence and are components of the chi-square test statistic. In other situations, this interpretation is not strictly valid. See also the CELLCHI2, DEVIATION, and OBSERVED options.

FREQOUT | FRE

indicates that the PROC CORRESP input data set has the same form as an output data set from the FREQ procedure, even if it was not directly produced by PROC FREQ. The FREQOUT option enables PROC CORRESP to take shortcuts in constructing the contingency table.

When you specify the FREQOUT option, you must also specify a WEIGHT statement. The cell frequencies in a PROC FREQ output data set are contained in a variable called COUNT, so specify COUNT in a WEIGHT statement with PROC CORRESP. The FREQOUT option may produce unexpected results if the DATA= data set is structured incorrectly. Each of the two variable lists specified in the TABLES statement must consist of a single variable, and observations must be grouped by the levels of the row variable and then by the levels of the column variable. It is not required that the observations be sorted by the row variable and column variable, but they must be grouped consistently. There must be as many observations in the input data set (or BY group) as there are cells in the completed contingency table. Zero cells must be specified with zero weights. When you use PROC FREQ to create the PROC CORRESP input data set, you must specify the SPARSE option in the FREQ procedure's TABLES statement so that the zero cells are written to the output data set.

GREENACRE | GRE

displays adjusted inertias when performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, adjusted inertias using a method proposed by Greenacre (1994, p. 156) can be displayed by specifying the GREENACRE option. Specify the UNADJUSTED option to output the usual table of unadjusted inertias as well. See the section "MCA Adjusted Inertias" on page 1102 for more information.

MCA

requests a multiple correspondence analysis. This option requires that the input table be a Burt table, which is a symmetric matrix of crosstabulations among several categorical variables. If you specify the MCA option and a VAR statement, you must also specify the NVARS= option, which gives the number of categorical variables that were used to create the table. With raw categorical data, if you want results for the individuals as well as the categories, use the BINARY option instead.

MININERTIA=*n*
MIN=*n*

specifies the minimum inertia $(0 \leq n \leq 1)$ used to create the "best" tables—the indicator of which points best explain the inertia of each dimension. By default, MININERTIA=0.8. See the "Algorithm and Notation" section on page 1097 for more information.

MISSING | MIS

specifies that observations with missing values for the TABLES statement variables are included in the analysis. Missing values are treated as a distinct level of each categorical variable. By default, observations with missing values are excluded from the analysis.

NOCOLUMN $<$ = BOTH | DATA | PRINT $>$
NOC $<$ = BOT | DAT | PRI $>$

suppresses the display of the column coordinates and statistics and omits them from the output coordinate data set.

BOTH	suppresses all column information from both the SAS listing and the output data set. The NOCOLUMN option is equivalent to the option NOCOLUMN=BOTH.
DATA	suppresses all column information from the output data set.
PRINT	suppresses all column information from the SAS listing.

NOPRINT | NOP

suppresses the display of all output. This option is useful when you need only an output data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

NOROW $<$ = BOTH | DATA | PRINT $>$
NOR $<$ = BOT | DAT | PRI $>$

suppresses the display of the row coordinates and statistics and omits them from the output coordinate data set.

BOTH	suppresses all row information from both the SAS listing and the output data set. The NOROW option is equivalent to the option NOROW=BOTH.
DATA	suppresses all row information from the output data set.
PRINT	suppresses all row information from the SAS listing.

The NOROW option can be useful when the rows of the contingency table are replications.

NVARS=*n*
NVA=*n*

specifies the number of classification variables that were used to create the Burt table. For example, if the Burt table was originally created with the statement

```
tables a b c;
```

you must specify NVARS=3 to read the table with a VAR statement.

The NVARS= option is required when you specify both the MCA option and a VAR statement. (See the section "VAR Statement" on page 1081 for an example.)

OBSERVED | OBS

displays the contingency table of observed frequencies and its row, column, and grand totals. If you do not specify the OBSERVED or ALL option, the contingency table is not displayed.

OUTC=*SAS-data-set*
OUT=*SAS-data-set*

creates an output coordinate SAS data set to contain the row, column, supplementary observation, and supplementary variable coordinates. This data set also contains the masses, squared cosines, quality of each point's representation in the DIMENS=n dimensional display, relative inertias, partial contributions to inertia, and best indicators.

OUTF=*SAS-data-set*

creates an output frequency SAS data set to contain the contingency table, row, and column profiles, the expected values, and the observed minus expected values and contributions to the chi-square statistic.

PRINT=BOTH | FREQ | PERCENT
PRI=BOT | FRE | PER

affects the OBSERVED, RP, CP, CELLCHI2, EXPECTED, and DEVIATION options. The default is PRINT=FREQ.

- The PRINT=FREQ option displays output in the appropriate raw or natural units. (That is, PROC CORRESP displays raw frequencies for the OBSERVED option, relative frequencies with row marginals of 1.0 for the RP option, and so on.)
- The PRINT=PERCENT option scales results to percentages for the display of the output. (All elements in the OBSERVED matrix sum to 100.0, the row marginals are 100.0 for the RP option, and so on.)
- The PRINT=BOTH option displays both percentages and frequencies.

PROFILE=BOTH | COLUMN | NONE | ROW
PRO=BOT | COL | NON | ROW

specifies the standardization for the row and column coordinates. The default is PROFILE=BOTH.

PROFILE=BOTH specifies a standard correspondence analysis, which jointly displays the principal row and column coordinates. Row coordinates are computed from the row profile matrix, and column coordinates are computed from the column profile matrix.

PROFILE=ROW specifies a correspondence analysis of the row profile matrix. The row coordinates are weighted centroids of the column coordinates.

PROFILE=COLUMN specifies a correspondence analysis of the column profile matrix. The column coordinates are weighted centroids of the row coordinates.

PROFILE=NONE is rarely needed. Row and column coordinates are the generalized singular vectors, without the customary standardizations.

ROW=A | AD | DA | DAD | DAD1/2 | DAID1/2

provides other standardizations of the row coordinates. The ROW= option is rarely needed. Typically, you should use the PROFILE= option instead (see the section "The PROFILE=, ROW=, and COLUMN= Options" on page 1099). By default, ROW=DAD.

RP

displays the row profile matrix. Row profiles contain the observed conditional probabilities of column membership given row membership. See also the CP option.

SHORT | SHO

suppresses the display of all point and coordinate statistics except the coordinates. The following information is suppressed: each point's mass, relative contribution to the total inertia, and quality of representation in the DIMENS=n dimensional display; the squared cosines of the angles between each axis and a vector from the origin to the point; the partial contributions of each point to the inertia of each dimension; and the best indicators.

SINGULAR=*n*
SIN=*n*

specifies the largest value that is considered to be within rounding error of zero. The default value is 1E−8. This parameter is used when checking for zero rows and columns, when checking Burt table diagonal sums for equality, when checking denominators before dividing, and so on. Typically, you should not assign a value outside the range 1E−6 to 1E−12.

SOURCE | SOU

adds the variable _VAR_, which contains the name or label of the variable corresponding to the current level, to the OUTC= and OUTF= data sets.

UNADJUSTED | UNA

displays unadjusted inertias when performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, if adjusted inertias are requested by either the GREENACRE option or the BENZECRI option, then the unadjusted inertia table is not displayed unless the UNADJUSTED option is specified. See the section "MCA Adjusted Inertias" on page 1102 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CORRESP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CORRESP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

ID Statement

ID *variable* **;**

You specify the ID statement only in conjunction with the VAR statement. You cannot specify the ID statement when you use the TABLES statement or the MCA option. When you specify an ID variable, PROC CORRESP labels the rows of the tables with the ID values and places the ID variable in the output data set.

SUPPLEMENTARY Statement

SUPPLEMENTARY *variables* **;**

SUP *variables* **;**

The SUPPLEMENTARY statement specifies variables that are to be represented as points in the joint row and column space but that are not used when determining the locations of the other, active row and column points of the contingency table. Supplementary observations on supplementary variables are ignored in simple correspondence analysis but are needed to compute the squared cosines for multiple correspondence analysis. Variables that are specified in the SUPPLEMENTARY statement must also be specified in the TABLES or VAR statement.

When you specify a VAR statement, each SUPPLEMENTARY variable indicates one supplementary column of the table. Supplementary variables must be numeric with VAR statement input.

When you specify a TABLES statement, each SUPPLEMENTARY variable indicates a set of rows or columns of the table that is supplementary. Supplementary variables can be either character or numeric with TABLES statement input.

TABLES Statement

TABLES < *row-variables,* > *column-variables* **;**

The TABLES statement instructs PROC CORRESP to create a contingency table, Burt table, or binary table from the values of two or more categorical variables. The TABLES statement specifies classification variables that are used to construct the rows and columns of the contingency table. The variables can be either numeric or character. The variable lists in the TABLES statement and the CROSS= option together determine the row and column labels of the contingency table.

You can specify both row variables and column variables separated by a comma, or you can specify only column variables and no comma. If you do not specify row variables (that is, if you list variables but do not use the comma as a delimiter), then you should specify either the MCA or the BINARY option. With the MCA option, PROC CORRESP creates a Burt table, which is a crosstabulation of each variable with itself and every other variable. The Burt table is symmetric. With the BINARY option, PROC CORRESP creates a binary table, which consists of one row for each input data set observation and one column for each category of each TABLES statement variable. If the binary matrix is $\mathbf{Z}$, then the Burt table is $\mathbf{Z}'\mathbf{Z}$. Specifying the BINARY option with the NOROWS option produces the same results as specifying the MCA option (except for the chi-square statistics).

See Figure 24.3 for an example or see the section "The MCA Option" on page 1101 for a detailed description of Burt tables.

You can use the WEIGHT statement with the TABLES statement to read category frequencies. Specify the SUPPLEMENTARY statement to name variables with categories that are supplementary rows or columns. You cannot specify the ID or VAR statement with the TABLES statement.

See the section "Using the TABLES Statement" on page 1088 for an example.

VAR Statement

VAR *variables* **;**

You should specify the VAR statement when your data are in tabular form. The VAR variables must be numeric. The VAR statement instructs PROC CORRESP to read an existing contingency table, binary indicator matrix, fuzzy-coded indicator matrix, or Burt table, rather than raw data. See the "Algorithm and Notation" section on page 1097 for a description of a binary indicator matrix and a fuzzy-coded indicator matrix.

You can specify the WEIGHT statement with the VAR statement to read category frequencies and designate supplementary rows. Specify the SUPPLEMENTARY statement to name supplementary variables. You cannot specify the TABLES statement with the VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies weights for each observation and indicates supplementary observations for simple correspondence analyses with VAR statement input. You can include only one WEIGHT statement, and the weight variable must be numeric.

If you omit the WEIGHT statement, each observation contributes a value of 1 to the frequency count for its category. That is, each observation represents one subject. When you specify a WEIGHT statement, each observation contributes the value of the weighting variable for that observation. For example, a weight of 3 means that the observation represents 3 subjects. Weight values are not required to be integers.

You can specify the WEIGHT statement with a TABLES statement to indicate category frequencies, as in the following example:

```
proc freq;
   tables a*b / out=outfreq sparse;
run;

proc corresp freqout;
   tables a, b;
   weight count;
run;
```

If you specify a VAR statement, you can specify the WEIGHT statement to indicate supplementary observations and to weight some rows of the table more heavily than others. When the value of the WEIGHT variable is negative, the observation is treated as supplementary, and the absolute value of the weight is used as the weighting value.

You cannot specify a WEIGHT statement with a VAR statement and the MCA option, because the table must be symmetric. Supplementary variables are indicated with the SUPPLEMENTARY statement, so differential weighting of rows is inappropriate.

Details

Input Data Set

PROC CORRESP can read two kinds of input:

- raw category responses on two or more classification variables with the TABLES statement
- a two-way contingency table with the VAR statement

You can use output from PROC FREQ as input for PROC CORRESP.

The classification variables referred to by the TABLES statement can be either numeric or character variables. Normally, all observations for a given variable that have the same formatted value are placed in the same level, and observations with different values are placed in different levels.

The variables in the VAR statement must be numeric. The values of the observations specify the cell frequencies. These values are not required to be integers, but only those observations with all nonnegative, nonmissing values are used in the correspondence analysis. Observations with one or more negative values are removed from the analysis.

The WEIGHT variable must be numeric. Observations with negative weights are treated as supplementary observations. The absolute values of the weights are used to weight the observations.

Types of Tables Used as Input

The following example explains correspondence analysis and illustrates some capabilities of PROC CORRESP.

```
data Neighbor;
   input Name $ 1-10 Age $ 12-18 Sex $ 19-25
         Height $ 26-30 Hair $ 32-37;
   datalines;
Jones      Old    Male   Short White
Smith      Young  Female Tall  Brown
Kasavitz   Old    Male   Short Brown
Ernst      Old    Female Tall  White
Zannoria   Old    Female Short Brown
Spangel    Young  Male   Tall  Blond
Myers      Young  Male   Tall  Brown
Kasinski   Old    Male   Short Blond
Colman     Young  Female Short Blond
Delafave   Old    Male   Tall  Brown
Singer     Young  Male   Tall  Brown
Igor       Old           Short
;
```

There are several types of tables, $\mathbf{N}$, that can be used as input to correspondence analysis —all tables can be defined using a binary matrix, $\mathbf{Z}$.

With the BINARY option, $\mathbf{N} = \mathbf{Z}$ is directly analyzed. The binary matrix has one column for each category and one row for each individual or case. A binary table constructed from m categorical variables has m partitions. The following table has four partitions, one for each of the four categorical variables. Each partition has a 1 in each row, and each row contains exactly four 1s since there are four categorical variables. More generally, the binary design matrix has exactly m 1s in each row. The 1s indicate the categories to which the observation applies.

Table 24.2. $\mathbf{Z}$, The Binary Coding of Neighbor Data Set

$\mathbf{Z}_{\mathbf{Hair}}$			$\mathbf{Z}_{\mathbf{Height}}$		$\mathbf{Z}_{\mathbf{Sex}}$		$\mathbf{Z}_{\mathbf{Age}}$	
Blond	Brown	White	Short	Tall	Female	Male	Old	Young
0	0	1	1	0	0	1	1	0
0	1	0	0	1	1	0	0	1
0	1	0	1	0	0	1	1	0
0	0	1	0	1	1	0	1	0
0	1	0	1	0	1	0	1	0
1	0	0	0	1	0	1	0	1
0	1	0	0	1	0	1	0	1
1	0	0	1	0	0	1	1	0
1	0	0	1	0	1	0	0	1
0	1	0	0	1	0	1	1	0
0	1	0	0	1	0	1	0	1

With the MCA option, the Burt table ($\mathbf{Z}'\mathbf{Z}$) is analyzed. A Burt table is a partitioned symmetric matrix containing all pairs of crosstabulations among a set of categorical variables. Each diagonal partition is a diagonal matrix containing marginal frequencies (a crosstabulation of a variable with itself). Each off-diagonal partition is an ordinary contingency table. Each contingency table above the diagonal has a transposed counterpart below the diagonal.

Table 24.3. $\mathbf{Z}'\mathbf{Z}$, The Burt Table

	Blond	Brown	White	Short	Tall	Female	Male	Old	Young
Blond	3	0	0	2	1	1	2	1	2
Brown	0	6	0	2	4	2	4	3	3
White	0	0	2	1	1	1	1	2	0
Short	2	2	1	5	0	2	3	4	1
Tall	1	4	1	0	6	2	4	2	4
Female	1	2	1	2	2	4	0	2	2
Male	2	4	1	3	4	0	7	4	3
Old	1	3	2	4	2	2	4	6	0
Young	2	3	0	1	4	2	3	0	5

This Burt table is composed of all pairs of crosstabulations among the variables Hair, Height, Sex, and Age. It is composed of sixteen individual subtables —the number of variables squared. Both the rows and the columns have the same nine categories (in this case Blond, Brown, White, Short, Tall, Female, Male, Old, and Young). The off-diagonal partitions are crosstabulations of each variable with every other variable. Below the diagonal are the following crosstabulations (from left to right, top to bottom): Height * Hair, Sex * Hair, Sex * Height, Age * Hair, Age * Height, and Age * Sex. Each crosstabulation below the diagonal has a transposed counterpart above the diagonal. Each diagonal partition contains a crosstabulation of a variable with itself (Hair * Hair, Height * Height, Sex * Sex, and Age * Age). The diagonal elements of the diagonal partitions contain marginal frequencies of the off-diagonal partitions.

For example, the table Hair * Height has three rows for Hair and two columns for Height. The values of the Hair * Height table, summed across rows, sum to the diagonal values of the Height * Height table, as displayed in the following table.

Table 24.4. $\mathbf{Z}_{\mathbf{Hair},\mathbf{Height}}{}'\mathbf{Z}_{\mathbf{Height}}$
The (Hair Height) × Height Crosstabulation

	Short	Tall
Blond	2	1
Brown	2	4
White	1	1
Short	5	0
Tall	0	6

A simple crosstabulation of Hair × Height is $\mathbf{N} = \mathbf{Z}_{\mathbf{Hair}}{}'\mathbf{Z}_{\mathbf{Height}}$. Crosstabulations such as this, involving only two variables, are the input to simple correspondence analysis.

Table 24.5. $\mathbf{Z}_{\mathbf{Hair}}{}'\mathbf{Z}_{\mathbf{Height}}$
The Hair × Height Crosstabulation

	Short	Tall
Blond	2	1
Brown	2	4
White	1	1

Tables such as the following ($\mathbf{N} = \mathbf{Z}_{\mathbf{Hair}}{}'\mathbf{Z}_{\mathbf{Height},\mathbf{Sex}}$), made up of several crosstabulations, can also be analyzed in simple correspondence analysis.

Table 24.6. $\mathbf{Z}_{\mathbf{Hair}}{}'\mathbf{Z}_{\mathbf{Height},\mathbf{Sex}}$
The Hair × (Height Sex) Crosstabulation

	Short	Tall	Female	Male
Blond	2	1	1	2
Brown	2	4	2	4
White	1	1	1	1

Coding, Fuzzy Coding, and Doubling

You can use an indicator matrix as input to PROC CORRESP using the VAR statement. An indicator matrix is composed of several submatrices, each of which is a design matrix with one column for each category of a categorical variable. In order to create an indicator matrix, you must code an indicator variable for each level of each categorical variable. For example, the categorical variable Sex, with two levels (Female and Male), would be coded using two indicator variables.

A binary indicator variable is coded 1 to indicate the presence of an attribute and 0 to indicate its absence. For the variable Sex, a male would be coded Female=0 and Male=1, and a female would be coded Female=1 and Male=0. The indicator variables representing a categorical variable must sum to 1.0. You can specify the BINARY option to create a binary table.

Sometimes binary data such as Yes/No data are available. For example, 1 means “Yes, I have bought this brand in the last month” and 0 means “No, I have not bought this brand in the last month”.

```
title 'Doubling Yes/No Data';

proc format;
   value yn 0 = 'No '  1 = 'Yes';
   run;

data BrandChoice;
   input a b c;
   label a = 'Brand A' b = 'Brand B' c = 'Brand B';
   format a b c yn.;
   datalines;
0 0 1
1 1 0
0 1 1
0 1 0
1 0 0
;
```

Data such as these cannot be analyzed directly because the raw data do not consist of partitions, each with one column per level and exactly one 1 in each row. The data must be *doubled* so that both Yes and No are both represented by a column in the data matrix. The TRANSREG procedure provides one way of doubling. In the following statements, the DESIGN option specifies that PROC TRANSREG is being used only for coding, not analysis. The option SEPARATORS=': ' specifies that labels for the coded columns are constructed from input variable labels, followed by a colon and space, followed by the formatted value. The variables are designated in the MODEL statement as CLASS variables, and the ZERO=NONE option creates binary variables for all levels. The OUTPUT statement specifies the output data set and drops the _NAME_, _TYPE_, and Intercept variables. PROC TRANSREG stores a list of coded variable names in a macro variable &_TRGIND, which in this case has the value “aNo aYes bNo bYes cNo cYes”. This macro can be used directly in the VAR statement in PROC CORRESP.

```
proc transreg data=BrandChoice design separators=': ';
   model class(a b c / zero=none);
   output out=Doubled(drop=_: Intercept);
```

```
run;

proc print label;
run;

proc corresp data=Doubled norow short;
   var &_trgind;
run;
```

A fuzzy-coded indicator also sums to 1.0 across levels of the categorical variable, but it is coded with fractions rather than with 1 and 0. The fractions represent the distribution of the attribute across several levels of the categorical variable.

Ordinal variables, such as survey responses of 1 to 3 can be represented as two design variables.

Table 24.7. Coding an Ordinal Variable

Ordinal Values	Coding	
1	0.25	0.75
2	0.50	0.50
3	0.75	0.25

Values of the coding sum to one across the two coded variables.

This next example illustrates the use of binary and fuzzy-coded indicator variables. Fuzzy-coded indicators are used to represent missing data. Note that the missing values in the observation Igor are coded with equal proportions.

```
proc transreg data=Neighbor design cprefix=0;
   model class(Age Sex Height Hair / zero=none);
   output out=Neighbor2(drop=_: Intercept);
   id Name;
   run;

data Neighbor3;
   set Neighbor2;
   if Sex = ' ' then do;
      Female = 0.5;
      Male   = 0.5;
      end;
   if Hair = ' ' then do;
      White = 1/3;
      Brown = 1/3;
      Blond = 1/3;
      end;
run;

proc print label;
run;
```

```
         Age   Age     Sex    Sex  Height  Height    Hair     Hair     Hair
   Obs   Old  Young  Female  Male   Short   Tall     Blond    Brown    White    Age    Sex     Height  Hair   Name

     1    1     0      0.0    1.0     1       0     0.00000  0.00000  1.00000  Old    Male    Short   White  Jones
     2    0     1      1.0    0.0     0       1     0.00000  1.00000  0.00000  Young  Female  Tall    Brown  Smith
     3    1     0      0.0    1.0     1       0     0.00000  1.00000  0.00000  Old    Male    Short   Brown  Kasavitz
     4    1     0      1.0    0.0     0       1     0.00000  0.00000  1.00000  Old    Female  Tall    White  Ernst
     5    1     0      1.0    0.0     1       0     0.00000  1.00000  0.00000  Old    Female  Short   Brown  Zannoria
     6    0     1      0.0    1.0     0       1     1.00000  0.00000  0.00000  Young  Male    Tall    Blond  Spangel
     7    0     1      0.0    1.0     0       1     0.00000  1.00000  0.00000  Young  Male    Tall    Brown  Myers
     8    1     0      0.0    1.0     1       0     1.00000  0.00000  0.00000  Old    Male    Short   Blond  Kasinski
     9    0     1      1.0    0.0     1       0     1.00000  0.00000  0.00000  Young  Female  Short   Blond  Colman
    10    1     0      0.0    1.0     0       1     0.00000  1.00000  0.00000  Old    Male    Tall    Brown  Delafave
    11    0     1      0.0    1.0     0       1     0.00000  1.00000  0.00000  Young  Male    Tall    Brown  Singer
    12    1     0      0.5    0.5     1       0     0.33333  0.33333  0.33333  Old            Short          Igor
```

Figure 24.3. Fuzzy Coding of Missing Values

There is one set of coded variables for each input categorical variable. If observation 12 is excluded, each set is a binary design matrix. Each design matrix has one column for each category and exactly one 1 in each row.

Fuzzy-coding is shown in the final observation, Igor. The observation Igor has missing values for the variables Sex and Hair. The design matrix variables are coded with fractions that sum to one within each categorical variable.

An alternative way to represent missing data is to treat missing values as an additional level of the categorical variable. This alternative is available with the MISSING option in the PROC statement. This approach yields coordinates for missing responses, allowing the comparison of “missing” along with the other levels of the categorical variables.

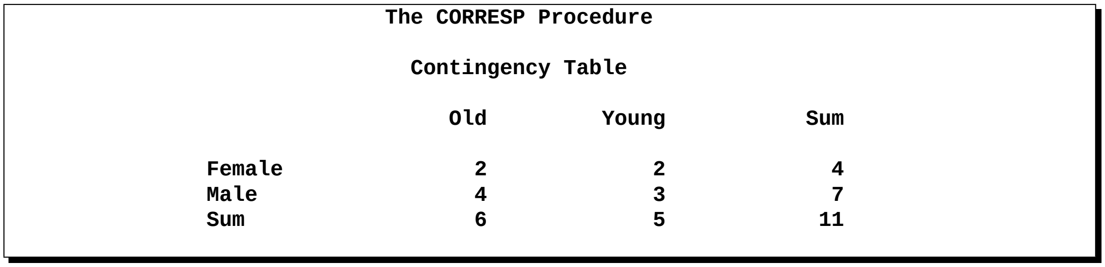
Greenacre and Hastie (1987) discuss additional coding schemes, including one for continuous variables. Continuous variables can be coded with PROC TRANSREG by specifying BSPLINE(*variables* / degree=1) in the MODEL statement.

Using the TABLES Statement

In the following TABLES statement, each variable list consists of a single variable:

```
proc corresp data=Neighbor dimens=1 observed short;
   ods select observed;
   tables Sex, Age;
run;
```

These statements create a contingency table with two rows (Female and Male) and two columns (Old and Young) and show the neighbors broken down by age and sex. The DIMENS=1 option overrides the default, which is DIMENS=2. The OBSERVED option displays the contingency table. The SHORT option limits the displayed output. Because it contains missing values, the observation where Name=’Igor’ is omitted from the analysis. Figure 24.4 displays the contingency table.

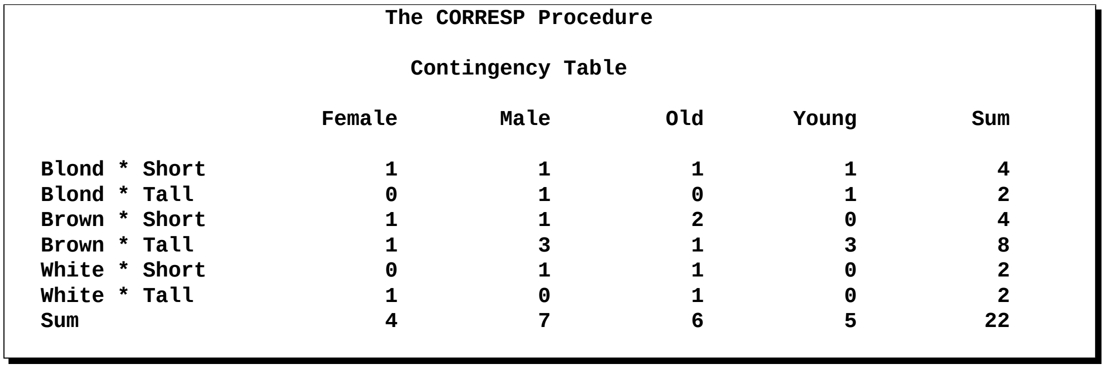


```
                          The CORRESP Procedure

                            Contingency Table

                               Old          Young            Sum

             Female              2              2              4
             Male                4              3              7
             Sum                 6              5             11
```

Figure 24.4. Contingency Table for Sex, Age

The following statements create a table with six rows (**Blond*Short**, **Blond*Tall**, **Brown*Short**, **Brown*Tall**, **White*Short**, and **White*Tall**), and four columns (**Female**, **Male**, **Old**, and **Young**). The levels of the row variables are crossed, forming mutually exclusive categories, whereas the categories of the column variables overlap.

```
   proc corresp data=Neighbor cross=row observed short;
      ods select observed;
      tables Hair Height, Sex Age;
   run;
```

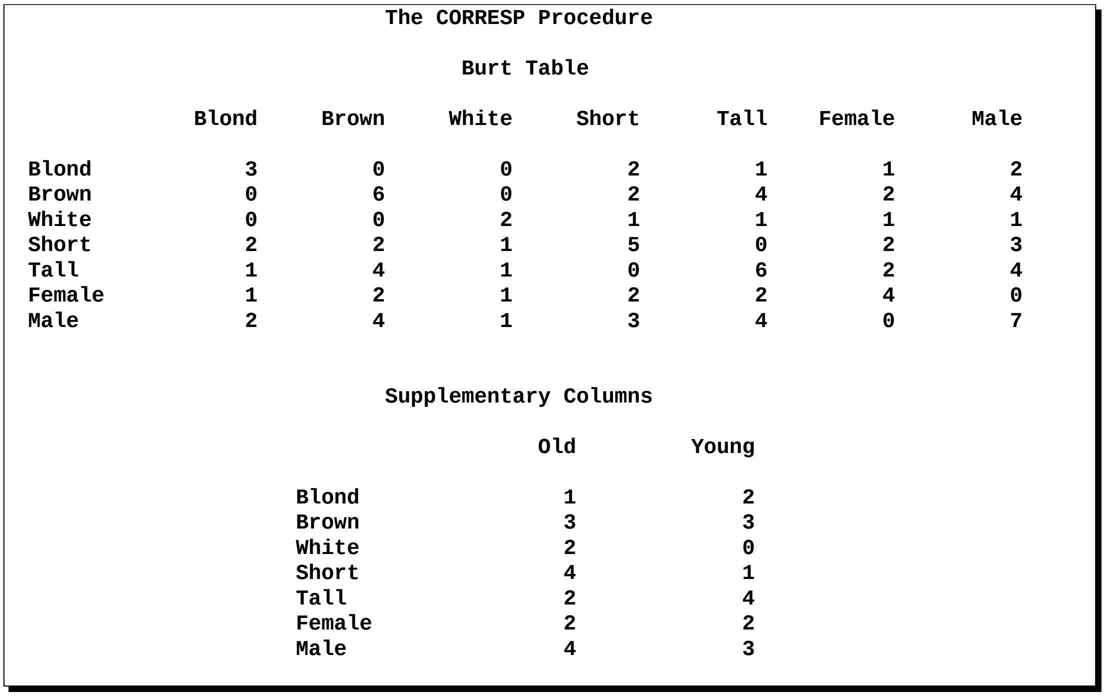


```
                          The CORRESP Procedure

                            Contingency Table

                      Female         Male          Old        Young          Sum

Blond * Short              1            1            1            1            4
Blond * Tall               0            1            0            1            2
Brown * Short              1            1            2            0            4
Brown * Tall               1            3            1            3            8
White * Short              0            1            1            0            2
White * Tall               1            0            1            0            2
Sum                        4            7            6            5           22
```

Figure 24.5. Contingency Table for Hair * Height, Sex Age

You can enter supplementary variables with TABLES input by including a SUPPLEMENTARY statement. Variables named in the SUPPLEMENTARY statement indicate TABLES variables with categories that are supplementary. In other words, the categories of the variable Age are represented in the row and column space, but they are not used in determining the scores of the categories of the variables Hair, Height, and Sex. The variable used in the SUPPLEMENTARY statement must be listed in the TABLES statement as well. For example, the following statements create a Burt table with seven active rows and columns (**Blond**, **Brown**, **White**, **Short**, **Tall**, **Female**, **Male**) and two supplementary rows and columns (**Old** and **Young**).

```
proc corresp data=Neighbor observed short mca;
   ods select burt supcols;
   tables Hair Height Sex Age;
   supplementary Age;
run;
```

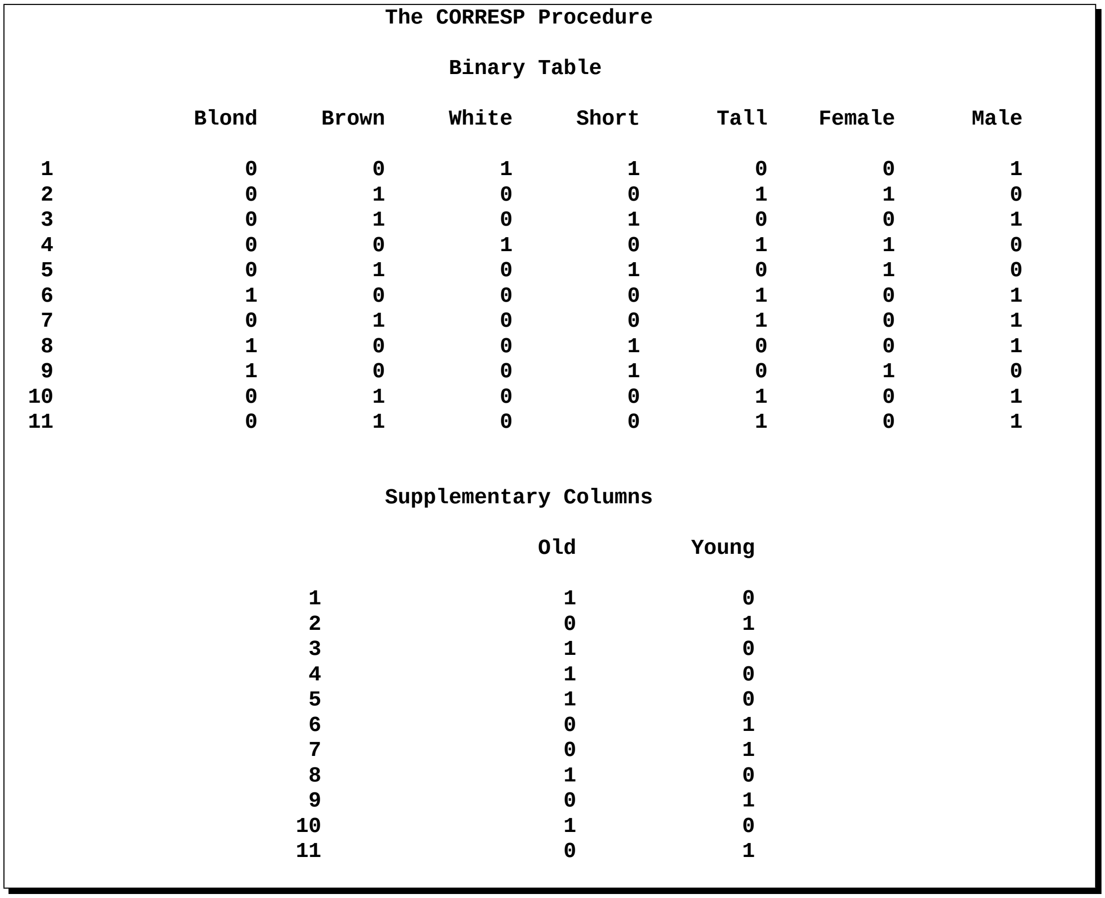


```
                              The CORRESP Procedure

                                    Burt Table

              Blond      Brown      White      Short       Tall     Female       Male

Blond             3          0          0          2          1          1          2
Brown             0          6          0          2          4          2          4
White             0          0          2          1          1          1          1
Short             2          2          1          5          0          2          3
Tall              1          4          1          0          6          2          4
Female            1          2          1          2          2          4          0
Male              2          4          1          3          4          0          7


                              Supplementary Columns

                                        Old         Young

                      Blond               1             2
                      Brown               3             3
                      White               2             0
                      Short               4             1
                      Tall                2             4
                      Female              2             2
                      Male                4             3
```

Figure 24.6. Burt Table from PROC CORRESP

The following statements create a binary table with 7 active columns (**Blond**, **Brown**, **White**, **Short**, **Tall**, **Female**, **Male**), 2 supplementary columns (**Old** and **Young**), and 11 rows for the 11 observations with nonmissing values.

```
proc corresp data=Neighbor observed short binary;
   ods select binary supcols;
   tables Hair Height Sex Age;
   supplementary Age;
run;
```



```
                              The CORRESP Procedure

                                  Binary Table

              Blond      Brown      White      Short       Tall     Female       Male

  1               0          0          1          1          0          0          1
  2               0          1          0          0          1          1          0
  3               0          1          0          1          0          0          1
  4               0          0          1          0          1          1          0
  5               0          1          0          1          0          1          0
  6               1          0          0          0          1          0          1
  7               0          1          0          0          1          0          1
  8               1          0          0          1          0          0          1
  9               1          0          0          1          0          1          0
 10               0          1          0          0          1          0          1
 11               0          1          0          0          1          0          1


                              Supplementary Columns

                                      Old         Young

                         1              1             0
                         2              0             1
                         3              1             0
                         4              1             0
                         5              1             0
                         6              0             1
                         7              0             1
                         8              1             0
                         9              0             1
                        10              1             0
                        11              0             1
```

Figure 24.7. Binary Table from PROC CORRESP

Using the VAR Statement

With VAR statement input, the rows of the contingency table correspond to the observations of the input data set, and the columns correspond to the VAR statement variables. The values of the variables typically contain the table frequencies. The example displayed in Figure 24.4 could be run with VAR statement input using the following code:

```
data Ages;
   input Sex $ Old Young;
   datalines;
Female  2 2
Male    4 3
;

proc corresp data=Ages dimens=1 observed short;
   var Old Young;
   id Sex;
run;
```

Only nonnegative values are accepted. Negative values are treated as missing, causing the observation to be excluded from the analysis. The values are not required to be integers. Row labels for the table are specified with an ID variable. Column labels are constructed from the variable name or variable label if one is specified. When you specify multiple correspondence analysis (MCA), the row and column labels are the same and are constructed from the variable names or labels, so you cannot include an ID statement. With MCA, the VAR statement must list the variables in the order in which the rows occur. For example, the table displayed in Figure 24.6, which was created with the following TABLES statement,

```
      tables Hair Height Sex Age;
```

is input as follows with the VAR statement:

```
   proc corresp data=table nvars=4 mca;
      var Blond Brown White Short Tall Female Male Old Young;
   run;
```

You must specify the NVARS= option to specify the number of original categorical variables with the MCA option. The option NVARS=n is needed to find boundaries between the subtables of the Burt table. If f is the sum of all elements in the Burt table $\mathbf{Z}'\mathbf{Z}$, then fn^{-2} is the number of rows in the binary matrix $\mathbf{Z}$. The sum of all elements in each diagonal subtable of the Burt table must be fn^{-2}.

To enter supplementary observations, include a WEIGHT statement with negative weights for those observations. Specify the SUPPLEMENTARY statement to include supplementary variables. You must list supplementary variables in both the VAR and SUPPLEMENTARY statements.

Missing and Invalid Data

With VAR statement input, observations with missing or negative frequencies are excluded from the analysis. Supplementary variables and supplementary observations with missing or negative frequencies are also excluded. Negative weights are valid with VAR statement input.

With TABLES statement input, observations with negative weights are excluded from the analysis. With this form of input, missing cell frequencies cannot occur. Observations with missing values on the categorical variables are excluded unless you specify the MISSING option. If you specify the MISSING option, ordinary missing values and special missing values are treated as additional levels of a categorical variable. In all cases, if any row or column of the constructed table contains only zeros, that row or column is excluded from the analysis.

Observations with missing weights are excluded from the analysis.

Creating a Data Set Containing the Crosstabulation

The CORRESP procedure can read or create a contingency or Burt table. PROC CORRESP is generally more efficient with VAR statement input than with TABLES statement input. TABLES statement input requires that the table be created from raw categorical variables, whereas the VAR statement is used to read an existing table. If PROC CORRESP runs out of memory, it may be possible to use some other method to create the table and then use VAR statement input with PROC CORRESP.

The following example uses the CORRESP, FREQ, and TRANSPOSE procedures to create rectangular tables from a SAS data set WORK.A that contains the categorical variables V1–V5. The Burt table examples assume that no categorical variable has a value found in any of the other categorical variables (that is, that each row and column label is unique).

You can use PROC CORRESP and ODS to create a rectangular two-way contingency table from two categorical variables.

```
proc corresp data=a observed short;
   ods listing close;
   ods output Observed=Obs(drop=Sum where=(Label ne 'Sum'));
   tables v1, v2;
run;

ods listing;
```

You can use PROC FREQ and PROC TRANSPOSE to create a rectangular two-way contingency table from two categorical variables.

```
proc freq data=a;
   tables v1 * v2 / sparse noprint out=freqs;
run;

proc transpose data=freqs out=rfreqs;
   id  v2;
   var count;
   by  v1;
run;
```

You can use PROC CORRESP and ODS to create a Burt table from five categorical variables.

```
proc corresp data=a observed short mca;
   ods listing close;
   ods output Burt=Obs;
   tables v1-v5;
run;

ods listing;
```

You can use a DATA step, PROC FREQ, and PROC TRANSPOSE to create a Burt table from five categorical variables.

```
data b;
   set a;
   array v[5] $ v1-v5;
   do i = 1 to 5;
      row = v[i];
      do j = 1 to 5;
         column = v[j];
         output;
         end;
      end;
   keep row column;
run;

proc freq data=b;
   tables row * column / sparse  noprint out=freqs;
run;

proc transpose data=freqs out=rfreqs;
   id  column;
   var count;
   by  row;
run;
```

Output Data Sets

The OUTC= Data Set

The OUTC= data set contains two or three character variables and $4n + 4$ numeric variables, where n is the number of axes from DIMENS=n (two by default). The OUTC= data set contains one observation for each row, column, supplementary row, and supplementary column point, and one observation for inertias.

The first variable is named _TYPE_ and identifies the type of observation. The values of _TYPE_ are as follows:

- The ‘INERTIA’ observation contains the total inertia in the INERTIA variable, and each dimension’s inertia in the Contr1–Contr$n$ variables.
- The ‘OBS’ observations contain the coordinates and statistics for the rows of the table.
- The ‘SUPOBS’ observations contain the coordinates and statistics for the supplementary rows of the table.
- The ‘VAR’ observations contain the coordinates and statistics for the columns of the table.
- The ‘SUPVAR’ observations contain the coordinates and statistics for the supplementary columns of the table.

If you specify the SOURCE option, then the data set also contains a variable _VAR_ containing the name or label of the input variable from which that row originates. The name of the next variable is either _NAME_ or (if you specify an ID statement) the name of the ID variable.

For observations with a value of ‘OBS’ or ‘SUPOBS’ for the _TYPE_ variable, the values of the second variable are constructed as follows:

- When you use a VAR statement without an ID statement, the values are ‘Row1’, ‘Row2’, and so on.
- When you specify a VAR statement with an ID statement, the values are set equal to the values of the ID variable.
- When you specify a TABLES statement, the _NAME_ variable has values formed from the appropriate row variable values.

For observations with a value of ‘VAR’ or ‘SUPVAR’ for the _TYPE_ variable, the values of the second variable are equal to the names or labels of the VAR (or SUPPLEMENTARY) variables. When you specify a TABLES statement, the values are formed from the appropriate column variable values.

The third and subsequent variables contain the numerical results of the correspondence analysis.

- Quality contains the quality of each point’s representation in the DIMENS=n dimensional display, which is the sum of squared cosines over the first n dimensions.
- Mass contains the masses or marginal sums of the relative frequency matrix.
- Inertia contains each point’s relative contribution to the total inertia.
- Dim1–Dim$n$ contain the point coordinates.
- Contr1–Contr$n$ contain the partial contributions to inertia.
- SqCos1–SqCos$n$ contain the squared cosines.
- Best1–Best$n$ and Best contain the summaries of the partial contributions to inertia.

The OUTF= Data Set

The OUTF= data set contains frequencies and percentages. It is similar to a PROC FREQ output data set. The OUTF= data set begins with a variable called _TYPE_, which contains the observation type. If the SOURCE option is specified, the data set contains two variables _ROWVAR_ and _COLVAR_ that contain the names or labels of the row and column input variables from which each cell originates. The next two variables are classification variables that contain the row and column levels. If you use TABLES statement input and each variable list consists of a single variable, the names of the first two variables match the names of the input variables; otherwise, these variables are named Row and Column. The next two variables are Count and Percent, which contain frequencies and percentages.

The _TYPE_ variable can have the following values:

- ‘OBSERVED’ observations contain the contingency table.
- ‘SUPOBS’ observations contain the supplementary rows.
- ‘SUPVAR’ observations contain the supplementary columns.
- ‘EXPECTED’ observations contain the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequency matrix under the hypothesis of row and column independence.
- ‘DEVIATION’ observations contain the matrix of deviations between the observed frequency matrix and the product of its row marginals and column marginals divided by its grand frequency. For ordinary two-way contingency tables, these are the observed minus expected frequencies under the hypothesis of row and column independence.
- ‘CELLCHI2’ observations contain contributions to the total chi-square test statistic.
- ‘RP’ observations contain the row profiles.
- ‘SUPRP’ observations contain supplementary row profiles.
- ‘CP’ observations contain the column profiles.
- ‘SUPCP’ observations contain supplementary column profiles.

Computational Resources

Let

$$\begin{aligned} n_r &= \text{number of rows in the table} \\ n_c &= \text{number of columns in the table} \\ n &= \text{number of observations} \\ v &= \text{number of VAR statement variables} \\ t &= \text{number of TABLES statement variables} \\ c &= \max(n_r, n_c) \\ d &= \min(n_r, n_c) \end{aligned}$$

For TABLES statement input, more than

$$32(t+1) + 8(\max(2tn, (n_r+3)(n_c+3)))$$

bytes of array space are required.

For VAR statement input, more than

$$16(v+2) + 8(n_r+3)(n_c+3)$$

bytes of array space are required.

Memory

The computational resources formulas are underestimates of the amounts of memory needed to handle most problems. If you use a utility data set, and if memory could be used with perfect efficiency, then roughly the stated amount of memory would be needed. In reality, most problems require at least two or three times the minimum.

PROC CORRESP tries to store the raw data (TABLES input) and the contingency table in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time.

Time

The time required to perform the generalized singular value decomposition is roughly proportional to $2cd^2 + 5d^3$. Overall computation time increases with table size at a rate roughly proportional to $(n_r n_c)^{\frac{3}{2}}$.

Algorithm and Notation

This section is primarily based on the theory of correspondence analysis found in Greenacre (1984). If you are interested in other references, see the "Background" section on page 1069.

Let **N** be the contingency table formed from those observations and variables that are not supplementary and from those observations that have no missing values and have a positive weight. This table is an $(n_r \times n_c)$ rank q matrix of nonnegative numbers with nonzero row and column sums. If $\mathbf{Z}_a$ is the binary coding for variable A, and $\mathbf{Z}_b$ is the binary coding for variable B, then $\mathbf{N} = \mathbf{Z}'_a\mathbf{Z}_b$ is a contingency table. Similarly, if $\mathbf{Z}_{b,c}$ contains the binary coding for both variables B and C, then $\mathbf{N} = \mathbf{Z}'_a\mathbf{Z}_{b,c}$ can also be input to a correspondence analysis. With the BINARY option, $\mathbf{N} = \mathbf{Z}$, and the analysis is based on a binary table. In multiple correspondence analysis, the analysis is based on a Burt table, $\mathbf{Z}'\mathbf{Z}$.

Let **1** be a vector of 1s of the appropriate order, let **I** be an identity matrix, and let diag$(\cdot)$ be a matrix-valued function that creates a diagonal matrix from a vector. Let

$$\begin{aligned}
f &= \mathbf{1}'\mathbf{N}\mathbf{1} \\
\mathbf{P} &= \frac{1}{f}\mathbf{N} \\
\mathbf{r} &= \mathbf{P}\mathbf{1} \\
\mathbf{c} &= \mathbf{P}'\mathbf{1} \\
\mathbf{D_r} &= \text{diag}(\mathbf{r}) \\
\mathbf{D_c} &= \text{diag}(\mathbf{c}) \\
\mathbf{R} &= \mathbf{D_r^{-1}P} \\
\mathbf{C}' &= \mathbf{D_c^{-1}P}'
\end{aligned}$$

The scalar f is the sum of all elements in $\mathbf{N}$. The matrix $\mathbf{P}$ is a matrix of relative frequencies. The vector $\mathbf{r}$ contains row marginal proportions or row "masses." The vector $\mathbf{c}$ contains column marginal proportions or column masses. The matrices $\mathbf{D_r}$ and $\mathbf{D_c}$ are diagonal matrices of marginals.

The rows of $\mathbf{R}$ contain the "row profiles." The elements of each row of $\mathbf{R}$ sum to one. Each (i, j) element of $\mathbf{R}$ contains the observed probability of being in column j given membership in row i. Similarly, the columns of $\mathbf{C}$ contain the column profiles. The coordinates in correspondence analysis are based on the generalized singular value decomposition of $\mathbf{P}$,

$$\mathbf{P} = \mathbf{A}\mathbf{D_u}\mathbf{B}'$$

where

$$\mathbf{A}'\mathbf{D_r^{-1}}\mathbf{A} = \mathbf{B}'\mathbf{D_c^{-1}}\mathbf{B} = \mathbf{I}$$

In multiple correspondence analysis,

$$\mathbf{P} = \mathbf{B}\mathbf{D_u^2}\mathbf{B}'$$

The matrix $\mathbf{A}$, which is the rectangular matrix of left generalized singular vectors, has n_r rows and q columns; the matrix $\mathbf{D_u}$, which is a diagonal matrix of singular values, has q rows and columns; and the matrix $\mathbf{B}$, which is the rectangular matrix of right generalized singular vectors, has n_c rows and q columns. The columns of $\mathbf{A}$ and $\mathbf{B}$ define the principal axes of the column and row point clouds, respectively.

The generalized singular value decomposition of $\mathbf{P} - \mathbf{rc}'$, discarding the last singular value (which is zero) and the last left and right singular vectors, is exactly the same as a generalized singular value decomposition of $\mathbf{P}$, discarding the first singular value (which is one), the first left singular vector, $\mathbf{r}$, and the first right singular vector, $\mathbf{c}$. The first (trivial) column of $\mathbf{A}$ and $\mathbf{B}$ and the first singular value in $\mathbf{D_u}$ are discarded before any results are displayed. You can obtain the generalized singular value decomposition of $\mathbf{P} - \mathbf{rc}'$ from the ordinary singular value decomposition of $\mathbf{D_r^{-1/2}}(\mathbf{P} - \mathbf{rc}')\mathbf{D_c^{-1/2}}$.

$$\mathbf{D_r^{-1/2}}(\mathbf{P} - \mathbf{rc}')\mathbf{D_c^{-1/2}} = \mathbf{U}\mathbf{D_u}\mathbf{V}' = (\mathbf{D_r^{-1/2}}\mathbf{A})\mathbf{D_u}(\mathbf{D_c^{-1/2}}\mathbf{B})'$$

$$\mathbf{P} - \mathbf{rc}' = \mathbf{D_r^{1/2}}\mathbf{U}\mathbf{D_u}\mathbf{V}'\mathbf{D_c^{1/2}} = (\mathbf{D_r^{1/2}}\mathbf{U})\mathbf{D_u}(\mathbf{D_c^{1/2}}\mathbf{V})' = \mathbf{A}\mathbf{D_u}\mathbf{B}'$$

Hence, $\mathbf{A} = \mathbf{D_r^{1/2}}\mathbf{U}$ and $\mathbf{B} = \mathbf{D_c^{1/2}}\mathbf{V}$.

The default row coordinates are $\mathbf{D_r^{-1}}\mathbf{A}\mathbf{D_u}$, and the default column coordinates are $\mathbf{D_c^{-1}}\mathbf{B}\mathbf{D_u}$. Typically the first two columns of $\mathbf{D_r^{-1}}\mathbf{A}\mathbf{D_u}$ and $\mathbf{D_c^{-1}}\mathbf{B}\mathbf{D_u}$ are plotted to display graphically associations between the row and column categories. The plot consists of two overlaid plots, one for rows and one for columns. The row points are row profiles, rescaled so that distances between profiles can be displayed as ordinary Euclidean distances, then orthogonally rotated to a principal axes orientation. The column points are column profiles, rescaled so that distances between profiles can be displayed as ordinary Euclidean distances, then orthogonally rotated to a principal axes orientation. Distances between row points and other row points have meaning. Distances between column points and other column points have meaning. However, distances between column points and row points are not interpretable.

The PROFILE=, ROW=, and COLUMN= Options

The PROFILE=, ROW=, and COLUMN= options standardize the coordinates before they are displayed and placed in the output data set. The options PROFILE=BOTH, PROFILE=ROW, and PROFILE=COLUMN provide the standardizations that are typically used in correspondence analysis. There are six choices each for row and column coordinates. However, most of the combinations of the ROW= and COLUMN= options are not useful. The ROW= and COLUMN= options are provided for completeness, but they are not intended for general use.

ROW=	**Matrix Formula**
A	$\mathbf{A}$
AD	$\mathbf{A}\mathbf{D_u}$
DA	$\mathbf{D_r^{-1}A}$
DAD	$\mathbf{D_r^{-1}AD_u}$
DAD1/2	$\mathbf{D_r^{-1}AD_u^{1/2}}$
DAID1/2	$\mathbf{D_r^{-1}A(I+D_u)^{1/2}}$

COLUMN=	**Matrix Formula**
B	$\mathbf{B}$
BD	$\mathbf{BD_u}$
DB	$\mathbf{D_c^{-1}B}$
DBD	$\mathbf{D_c^{-1}BD_u}$
DBD1/2	$\mathbf{D_c^{-1}BD_u^{1/2}}$
DBID1/2	$\mathbf{D_c^{-1}B(I+D_u)^{1/2}}$

When PROFILE=ROW (ROW=DAD and COLUMN=DB), the row coordinates $\mathbf{D_r^{-1}AD_u}$ and column coordinates $\mathbf{D_c^{-1}B}$ provide a correspondence analysis based on the row profile matrix. The row profile (conditional probability) matrix is defined as $\mathbf{R} = \mathbf{D_r^{-1}P} = \mathbf{D_r^{-1}AD_uB'}$. The elements of each row of $\mathbf{R}$ sum to one. Each (i, j) element of $\mathbf{R}$ contains the observed probability of being in column j given membership in row i. The "principal" row coordinates $\mathbf{D_r^{-1}AD_u}$ and "standard" column coordinates $\mathbf{D_c^{-1}B}$ provide a decomposition of $\mathbf{D_r^{-1}AD_uB'D_c^{-1}} = \mathbf{D_r^{-1}PD_c^{-1}} = \mathbf{RD_c^{-1}}$. Since $\mathbf{D_r^{-1}AD_u} = \mathbf{RD_c^{-1}B}$, the row coordinates are weighted centroids of the column coordinates. Each column point, with coordinates scaled to standard coordinates, defines a vertex in $(n_c - 1)$-dimensional space. All of the principal row coordinates are located in the space defined by the standard column coordinates. Distances among row points have meaning, but distances among column points and distances between row and column points are not interpretable.

The option PROFILE=COLUMN can be described as applying the PROFILE=ROW formulas to the transpose of the contingency table. When PROFILE=COLUMN (ROW=DA and COLUMN=DBD), the principal column coordinates $\mathbf{D_c^{-1}BD_u}$ are weighted centroids of the standard row coordinates $\mathbf{D_r^{-1}A}$. Each row point, with coordinates scaled to standard coordinates, defines a vertex in $(n_r - 1)$-dimensional space. All of the principal column coordinates are located in the space defined by the standard row coordinates. Distances among column points have meaning, but distances among row points and distances between row and column points are not interpretable.

The usual sets of coordinates are given by the default PROFILE=BOTH (ROW=DAD and COLUMN=DBD). All of the summary statistics, such as the squared cosines and contributions to inertia, apply to these two sets of points. One advantage to using these coordinates is that both sets ($\mathbf{D_r^{-1}AD_u}$ and $\mathbf{D_c^{-1}BD_u}$) are postmultiplied by the diagonal matrix $\mathbf{D_u}$, which has diagonal values that are all less than or equal to one. When $\mathbf{D_u}$ is a part of the definition of only one set of coordinates, that set forms a tight cluster near the centroid whereas the other set of points is more widely dispersed. Including $\mathbf{D_u}$ in both sets makes a better graphical display. However, care must be taken in interpreting such a plot. No correct interpretation of distances between row points and column points can be made.

Another property of this choice of coordinates concerns the geometry of distances between points within each set. The default row coordinates can be decomposed into $\mathbf{D_r^{-1}AD_u} = \mathbf{D_r^{-1}AD_uB'D_c^{-1}B} = (\mathbf{D_r^{-1}P})(\mathbf{D_c^{-1/2}})(\mathbf{D_c^{-1/2}B})$. The row coordinates are row profiles $(\mathbf{D_r^{-1}P})$, rescaled by $\mathbf{D_c^{-1/2}}$ (rescaled so that distances between profiles are transformed from a chi-square metric to a Euclidean metric), then orthogonally rotated (with $\mathbf{D_c^{-1/2}B}$) to a principal axes orientation. Similarly, the column coordinates are column profiles rescaled to a Euclidean metric and orthogonally rotated to a principal axes orientation.

The rationale for computing distances between row profiles using the non-Euclidean chi-square metric is as follows. Each row of the contingency table can be viewed as a realization of a multinomial distribution conditional on its row marginal frequency. The null hypothesis of row and column independence is equivalent to the hypothesis of homogeneity of the row profiles. A significant chi-square statistic is geometrically interpreted as a significant deviation of the row profiles from their centroid, $\mathbf{c}'$. The chi-square metric is the Mahalanobis metric between row profiles based on their estimated covariance matrix under the homogeneity assumption (Greenacre and Hastie 1987). A parallel argument can be made for the column profiles.

When ROW=DAD1/2 and COLUMN=DBD1/2 (Gifi 1990; van der Heijden and de Leeuw 1985), the row coordinates $\mathbf{D_r^{-1}AD_u^{1/2}}$ and column coordinates $\mathbf{D_c^{-1}BD_u^{1/2}}$ are a decomposition of $\mathbf{D_r^{-1}PD_c^{-1}}$.

In all of the preceding pairs, distances between row and column points are not meaningful. This prompted Carroll, Green, and Schaffer (1986) to propose that row coordinates $\mathbf{D_r^{-1}A(I + D_u)^{1/2}}$ and column coordinates $\mathbf{D_c^{-1}B(I + D_u)^{1/2}}$ be used.

These coordinates are (except for a constant scaling) the coordinates from a multiple correspondence analysis of a Burt table created from two categorical variables. This standardization is available with ROW=DAID1/2 and COLUMN=DBID1/2. However, this approach has been criticized on both theoretical and empirical grounds by Greenacre (1989). The Carroll, Green, and Schaffer standardization relies on the assumption that the chi-square metric is an appropriate metric for measuring the distance between the columns of a bivariate indicator matrix. See the section "Types of Tables Used as Input" on page 1083 for a description of indicator matrices. Greenacre (1989) showed that this assumption cannot be justified.

The MCA Option

The MCA option performs a multiple correspondence analysis (MCA). This option requires a Burt table. You can specify the MCA option with a table created from a design matrix with fuzzy coding schemes as long as every row of every partition of the design matrix has the same marginal sum. For example, each row of each partition could contain the probabilities that the observation is a member of each level. Then the Burt table constructed from this matrix no longer contains all integers, and the diagonal partitions are no longer diagonal matrices, but MCA is still valid.

A TABLES statement with a single variable list creates a Burt table. Thus, you can always specify the MCA option with this type of input. If you use the MCA option when reading an existing table with a VAR statement, you must ensure that the table is a Burt table.

If you perform MCA on a table that is not a Burt table, the results of the analysis are invalid. If the table is not symmetric, or if the sums of all elements in each diagonal partition are not equal, PROC CORRESP displays an error message and quits.

A subset of the columns of a Burt table is not necessarily a Burt table, so in MCA it is not appropriate to designate arbitrary columns as supplementary. You can, however, designate all columns from one or more categorical variables as supplementary.

The results of a multiple correspondence analysis of a Burt table $\mathbf{Z'Z}$ are the same as the column results from a simple correspondence analysis of the binary (or fuzzy) matrix $\mathbf{Z}$. Multiple correspondence analysis is not a simple correspondence analysis of the Burt table. It is not appropriate to perform a simple correspondence analysis of a Burt table. The MCA option is based on $\mathbf{P} = \mathbf{B}\mathbf{D}_\mathbf{u}^\mathbf{2}\mathbf{B}'$, whereas a simple correspondence analysis of the Burt table would be based on $\mathbf{P} = \mathbf{B}\mathbf{D}_\mathbf{u}\mathbf{B}'$.

Since the rows and columns of the Burt table are the same, no row information is displayed or written to the output data sets. The resulting inertias and the default (COLUMN=DBD) column coordinates are the appropriate inertias and coordinates for an MCA. The supplementary column coordinates, cosines, and quality of representation formulas for MCA differ from the simple correspondence analysis formulas because the design matrix column profiles and left singular vectors are not available.

The following statements create a Burt table and perform a multiple correspondence analysis:

```
proc corresp data=Neighbor observed short mca;
   tables Hair Height Sex Age;
run;
```

Both the rows and the columns have the same nine categories (Blond, Brown, White, Short, Tall, Female, Male, Old, and Young).

MCA Adjusted Inertias

The usual principal inertias of a Burt Table constructed from m categorical variables in MCA are the eigenvalues u_k from $\mathbf{D_u^2}$. The problem with these inertias is that they provide a pessimistic indication of fit. Benzécri (1979) proposed the following inertia adjustment, which is also described by Greenacre (1984, p. 145):

$$\left(\frac{m}{m-1}\right)^2 \times \left(u_k - \frac{1}{m}\right)^2 \qquad \text{for } u_k > \frac{1}{m}$$

The Benzécri adjustment is available with the BENZECRI option.

Greenacre (1994, p. 156) argues that the Benzécri adjustment overestimates the quality of fit. Greenacre proposes instead the following inertia adjustment:

$$\left(\frac{m}{m-1}\right)^2 \times \left(\sqrt{u_k} - \frac{1}{m}\right)^2 \qquad \text{for } \sqrt{u_k} > \frac{1}{m}$$

The Greenacre adjustment is available with the GREENACRE option.

Ordinary unadjusted inertias are printed by default with MCA when neither the BENZECRI nor the GREENACRE option is specified. However, the unadjusted inertias are not printed by default when either the BENZECRI or the GREENACRE option is specified. To display both adjusted and unadjusted inertias, specify the UNADJUSTED option in addition to the relevant adjusted inertia option (BENZECRI, GREENACRE, or both).

Supplementary Rows and Columns

Supplementary rows and columns are represented as points in the joint row and column space, but they are not used when determining the locations of the other active rows and columns of the table. The formulas that are used to compute coordinates for the supplementary rows and columns depend on the PROFILE= option or on the ROW= and COLUMN= options. Let $\mathbf{S_o}$ be the matrix with rows that contain the supplementary observations and $\mathbf{S_v}$ be a matrix with rows that contain the supplementary variables. Note that $\mathbf{S_v}$ is defined to be the transpose of the supplementary variable partition of the table. Let $\mathbf{R_s} = \text{diag}(\mathbf{S_o 1})^{\mathbf{-1}}\mathbf{S_o}$ be the supplementary observation profile matrix and $\mathbf{C_s} = \text{diag}(\mathbf{S_v 1})^{\mathbf{-1}}\mathbf{S_v}$ be the supplementary variable profile matrix. Note that the notation $\text{diag}(\cdot)^{\mathbf{-1}}$ means to convert the vector to a diagonal matrix, then invert the diagonal matrix. The coordinates for the supplementary observations and variables are as follows.

ROW=	**Matrix Formula**
A	$\frac{1}{f}\mathbf{S_o D_c^{-1} B D_u^{-1}}$
AD	$\frac{1}{f}\mathbf{S_o D_c^{-1} B}$
DA	$\mathbf{R_s D_c^{-1} B D_u^{-1}}$
DAD	$\mathbf{R_s D_c^{-1} B}$
DAD1/2	$\mathbf{R_s D_c^{-1} B D_u^{-1/2}}$
DAID1/2	$\mathbf{R_s D_c^{-1} B D_u^{-1} (I + D_u)^{1/2}}$

COLUMN=	**Matrix Formula**
B	$\frac{1}{f}\mathbf{S_v D_r^{-1} A D_u^{-1}}$
BD	$\frac{1}{f}\mathbf{S_v D_r^{-1} A}$
DB	$\mathbf{C_s D_r^{-1} A D_u^{-1}}$
DBD	$\mathbf{C_s D_r^{-1} A}$
DBD1/2	$\mathbf{C_s D_r^{-1} A D_u^{-1/2}}$
DBID1/2	$\mathbf{C_s D_r^{-1} A D_u^{-1} (I + D_u)^{1/2}}$

MCA COLUMN=	**Matrix Formula**
B	not allowed
BD	not allowed
DB	$\mathbf{C_s D_r^{-1} B D_u^{-2}}$
DBD	$\mathbf{C_s D_r^{-1} B D_u^{-1}}$
DBD1/2	$\mathbf{C_s D_r^{-1} B D_u^{-3/2}}$
DBID1/2	$\mathbf{C_s D_r^{-1} B D_u^{-2} (I + D_u)^{1/2}}$

Statistics that Aid Interpretation

The partial contributions to inertia, squared cosines, quality of representation, inertia, and mass provide additional information about the coordinates. These statistics are displayed by default. Include the SHORT or NOPRINT option in the PROC CORRESP statement to avoid having these statistics displayed.

These statistics pertain to the default PROFILE=BOTH coordinates, no matter what values you specify for the ROW=, COLUMN=, or PROFILE= option. Let sq$(\cdot)$ be a matrix-valued function denoting element-wise squaring of the argument matrix. Let t be the total inertia (the sum of the elements in $\mathbf{D_u^2}$).

In MCA, let $\mathbf{D_s}$ be the Burt table partition containing the intersection of the supplementary columns and the supplementary rows. The matrix $\mathbf{D_s}$ is a diagonal matrix of marginal frequencies of the supplemental columns of the binary matrix **Z**. Let p be the number of rows in this design matrix.

Statistic	**Matrix Formula**
Row partial contributions to inertia	$\mathbf{D_r^{-1}}\text{sq}(\mathbf{A})$
Column partial contributions to inertia	$\mathbf{D_c^{-1}}\text{sq}(\mathbf{B})$
Row squared cosines	$\text{diag}(\text{sq}(\mathbf{AD_u})\mathbf{1})^{-1}\text{sq}(\mathbf{AD_u})$
Column squared cosines	$\text{diag}(\text{sq}(\mathbf{BD_u})\mathbf{1})^{-1}\text{sq}(\mathbf{BD_u})$
Row mass	$\mathbf{r}$
Column mass	$\mathbf{c}$
Row inertia	$\frac{1}{t}\mathbf{D_r^{-1}}\ \text{sq}(\mathbf{AD_u})\mathbf{1}$
Column inertia	$\frac{1}{t}\mathbf{D_c^{-1}}\ \text{sq}(\mathbf{BD_u})\mathbf{1}$
Supplementary row squared cosines	$\text{diag}(\text{sq}(\mathbf{R_s} - \mathbf{1c'})\mathbf{D_c^{-1}1})^{-1}\text{sq}(\mathbf{R_sD_c^{-1}B})$
Supplementary column squared cosines	$\text{diag}(\text{sq}(\mathbf{C_s} - \mathbf{1r'})\mathbf{D_r^{-1}1})^{-1}\text{sq}(\mathbf{C_sD_r^{-1}A})$
MCA supplementary column squared cosines	$\mathbf{D_s}(p\mathbf{I} - \mathbf{D_s})^{-1}\ \text{sq}(\mathbf{C_sD_r^{-1}BD_u^{-1}})$

The quality of representation in the DIMENS=n dimensional display of any point is the sum of its squared cosines over only the n dimensions. Inertia and mass are not defined for supplementary points.

A table that summarizes the partial contributions to inertia table is also computed. The points that best explain the inertia of each dimension and the dimension to which each point contributes the most inertia are indicated. The output data set variable names for this table are Best1–Best$n$ (where DIMENS=n) and Best. The Best column contains the dimension number of the largest partial contribution to inertia for each point (the index of the maximum value in each row of $\mathbf{D_r^{-1}}\text{sq}(\mathbf{A})$ or $\mathbf{D_c^{-1}}\text{sq}(\mathbf{B})$).

For each row, the Best1–Best$n$ columns contain either the corresponding value of Best if the point is one of the biggest contributors to the dimension's inertia or 0 if it is not. Specifically, Best1 contains the value of Best for the point with the largest contribution to dimension one's inertia. A cumulative proportion sum is initialized to this point's partial contribution to the inertia of dimension one. If this sum is less than the value for the MININERTIA= option, then Best1 contains the value of Best for the point with the second largest contribution to dimension one's inertia. Otherwise, this point's Best1 is 0. This point's partial contribution to inertia is added to the sum. This process continues for the point with the third largest partial contribution, and so on, until adding a point's contribution to the sum increases the sum beyond the value of the MININERTIA= option. This same algorithm is then used for Best2, and so on.

For example, the following table contains contributions to inertia and the corresponding Best variables. The contribution to inertia variables are proportions that sum to 1 within each column. The first point makes its greatest contribution to the inertia of dimension two, so Best for point one is set to 2 and Best1–Best3 for point one must all be 0 or 2. The second point also makes its greatest contribution to the inertia of dimension two, so Best for point two is set to 2 and Best1–Best3 for point two must all be 0 or 2, and so on.

Assume MININERTIA=0.8, the default. In dimension one, the largest contribution is 0.41302 for the fourth point, so Best1 is set to 1, the value of Best for the fourth point. Because this value is less than 0.8, the second largest value (0.36456 for point five) is found and its Best1 is set to its Best's value of 1. Because $0.41302 + 0.36456 = 0.77758$ is less than 0.8, the third point (0.0882 at point eight) is found and Best1 is set to 3 since the contribution to dimension 3 for that point is greater than the contribution to dimension 1. This increases the sum of the partial contributions to greater than 0.8, so the remaining Best1 values are all 0.

Contr1	Contr2	Contr3	Best1	Best2	Best3	Best
0.01593	0.32178	0.07565	0	2	2	2
0.03014	0.24826	0.07715	0	2	2	2
0.00592	0.02892	0.02698	0	0	0	2
0.41302	0.05191	0.05773	1	0	0	1
0.36456	0.00344	0.15565	1	0	1	1
0.03902	0.30966	0.11717	0	2	2	2
0.00019	0.01840	0.00734	0	0	0	2
0.08820	0.00527	0.16555	3	0	3	3
0.01447	0.00024	0.03851	0	0	0	3
0.02855	0.01213	0.27827	0	0	3	3

Displayed Output

The display options control the amount of displayed output. By default, the following information is displayed:

- an inertia and chi-square decomposition table including the total inertia, the principal inertias of each dimension (eigenvalues), the singular values (square roots of the eigenvalues), each dimension's percentage of inertia, a horizontal bar chart of the percentages, and the total chi-square with its degrees of freedom and decomposition. The chi-square statistics and degrees of freedom are valid only when the constructed table is an ordinary two-way contingency table.
- the coordinates of the rows and columns on the dimensions
- the mass, relative contribution to the total inertia, and quality of representation in the DIMENS=n dimensional display of each row and column
- the squared cosines of the angles between each axis and a vector from the origin to the point
- the partial contributions of each point to each dimension's inertia

- the Best table, indicators of which points best explain the inertia of each dimension

Specific display options and combinations of options display output as follows.

If you specify the OBSERVED or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays

- the contingency table including the row and column marginal frequencies; or with BINARY, the binary table; or the Burt table in MCA
- the supplementary rows
- the supplementary columns

If you specify the OBSERVED or ALL option, with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays

- the contingency table or Burt table in MCA, scaled to percentages, including the row and column marginal percentages
- the supplementary rows, scaled to percentages
- the supplementary columns, scaled to percentages

If you specify the EXPECTED or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence.

If you specify the EXPECTED or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the product of the row marginals and the column marginals divided by the grand frequency of the observed percentages table. For ordinary two-way contingency tables, these are the expected percentages under the hypothesis of row and column independence.

If you specify the DEVIATION or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the observed minus expected frequencies. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence.

If you specify the DEVIATION or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the observed minus expected percentages. For ordinary two-way contingency tables, these are the expected percentages under the hypothesis of row and column independence.

If you specify the CELLCHI2 or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays contributions to the total chi-square test statistic, including the row and column marginals. The intersection of the marginals contains the total chi-square statistic.

If you specify the CELLCHI2 or ALL option with the PRINT=PERCENT or the PRINT=BOTH option, PROC CORRESP displays contributions to the total chi-square, scaled to percentages, including the row and column marginals.

If you specify the RP or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the row profiles and the supplementary row profiles.

If you specify the RP or ALL option with the PRINT=PERCENT or the PRINT=BOTH option, PROC CORRESP displays the row profiles (scaled to percentages) and the supplementary row profiles (scaled to percentages).

If you specify the CP or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the column profiles and the supplementary column profiles.

If you specify the CP or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the column profiles (scaled to percentages) and the supplementary column profiles (scaled to percentages).

If you do not specify the NOPRINT option, PROC CORRESP displays the inertia and chi-square decomposition table. This includes the nonzero singular values of the contingency table (or, in MCA, the binary matrix **Z** used to create the Burt table), the nonzero principal inertias (or eigenvalues) for each dimension, the total inertia, the total chi-square, the decomposition of chi-square, the chi-square degrees of freedom (appropriate only when the table is an ordinary two-way contingency table), the percent of the total chi-square and inertia for each dimension, and a bar chart of the percents.

If you specify the MCA option and you do not specify the NOPRINT option, PROC CORRESP displays the adjusted inertias. This includes the nonzero adjusted inertias, percents, cumulative percents, and a bar chart of the percents.

If you do not specify the NOROW, NOPRINT, or MCA option, PROC CORRESP displays the row coordinates and the supplementary row coordinates (displayed when there are supplementary row points).

If you do not specify the NOROW, NOPRINT, MCA, or SHORT option, PROC CORRESP displays

- the summary statistics for the row points including the quality of representation of the row points in the n-dimensional display, the mass, and the relative contributions to inertia
- the quality of representation of the supplementary row points in the n-dimensional display (displayed when there are supplementary row points)
- the partial contributions to inertia for the row points
- the row Best table, indicators of which row points best explain the inertia of each dimension
- the squared cosines for the row points
- the squared cosines for the supplementary row points (displayed when there are supplementary row points)

If you do not specify the NOCOLUMN or NOPRINT option, PROC CORRESP displays the column coordinates and the supplementary column coordinates (displayed when there are supplementary column points).

If you do not specify the NOCOLUMN, NOPRINT, or SHORT option, PROC CORRESP displays

- the summary statistics for the column points including the quality of representation of the column points in the n-dimensional display, the mass, and the relative contributions to inertia for the supplementary column points
- the quality of representation of the supplementary column points in the n-dimensional display (displayed when there are supplementary column points)
- the partial contributions to inertia for the column points
- the column Best table, indicators of which column points best explain the inertia of each dimension
- the squared cosines for the column points
- the squared cosines for the supplementary column points

ODS Table Names

PROC CORRESP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 24.8. ODS Tables Produced in PROC CORRESP

ODS Table Name	**Description**	**Option**
AdjInGreenacre	Greenacre Inertia Adjustment	GREENACRE
AdjInBenzecri	Benzécri Inertia Adjustment	BENZECRI
Binary	Binary table	OBSERVED, BINARY
BinaryPct	Binary table percents	OBSERVED, BINARY *
Burt	Burt table	OBSERVED, MCA
BurtPct	Burt table percents	OBSERVED, MCA *
CellChiSq	Contributions to Chi Square	CELLCHI2
CellChiSqPct	Contributions, pcts	CELLCHI2 *
ColBest	Col best indicators	default
ColContr	Col contributions to inertia	default
ColCoors	Col coordinates	default
ColProfiles	Col profiles	CP
ColProfilesPct	Col profiles, pcts	CP *
ColQualMassIn	Col quality, mass, inertia	default
ColSqCos	Col squared cosines	default
DF	DF, Chi Square (not displayed)	default
Deviations	Observed - expected freqs	DEVIATIONS
DeviationsPct	Observed - expected pcts	DEVIATIONS *

Table 24.8. (continued)

ODS Table Name	Description	Option
Expected	Expected frequencies	EXPECTED
ExpectedPct	Expected percents	EXPECTED *
Inertias	Inertia decomposition table	default
Observed	Observed frequencies	OBSERVED
ObservedPct	Observed percents	OBSERVED *
RowBest	Row best indicators	default
RowContr	Row contributions to inertia	default
RowCoors	Row coordinates	default
RowProfiles	Row profiles	RP
RowProfilesPct	Row profiles, pcts	RP *
RowQualMassIn	Row quality, mass, inertia	default
RowSqCos	Row squared cosines	default
SupColCoors	Supp col coordinates	default
SupColProfiles	Supp col profiles	CP
SupColProfilesPct	Supp col profiles, pcts	CP *
SupColQuality	Supp col quality	default
SupCols	Supplementary col freq	OBSERVED
SupColsPct	Supplementary col pcts	OBSERVED *
SupColSqCos	Supp col squared cosines	default
SupRows	Supplementary row freqs	OBSERVED
SupRowCoors	Supp row coordinates	default
SupRowProfiles	Supp row profiles	RP
SupRowProfilesPct	Supp row profiles, pcts	RP *
SupRowQuality	Supp row quality	default
SupRowsPct	Supplementary row pcts	OBSERVED *
SupRowSqCos	Supp row squared cosines	default

*Percents are displayed when you specify the PRINT=PERCENT or PRINT=BOTH option.

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the CORRESP procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request a graph you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

ODS Graph Names

PROC CORRESP assigns a name to the graph it creates using ODS. You can use this name to reference the graph when using ODS. The name is listed in Table 24.9.

To request a graph you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 24.9. ODS Graphics Produced by PROC CORRESP

ODS Graph Name	Plot Description
CorrespPlot	Correspondence analysis plot

Examples

Example 24.1. Simple Correspondence Analysis of Cars and Their Owners

In this example, PROC CORRESP creates a contingency table from categorical data and performs a simple correspondence analysis. The data are from a sample of individuals who were asked to provide information about themselves and their cars. The questions included origin of the car (American, Japanese, European) and family status (single, married, single and living with children, and married living with children). These data are used again in Example 24.2.

The first steps read the input data and assign formats. PROC CORRESP is used to perform the simple correspondence analysis. The ALL option displays all tables including the contingency table, chi-square information, profiles, and all results of the correspondence analysis. The OUTC= option creates an output coordinate data set. The TABLES statement specifies the row and column categorical variables. The %PLOTIT macro is used to plot the results.

Normally, you only need to tell the %PLOTIT macro the name of the input data set, DATA=Coor, and the type of analysis performed on the data, DATATYPE=CORRESP.

The following statements produce Output 24.1.1:

```
title ’Car Owners and Car Origin’;

proc format;
   value Origin  1 = ’American’ 2 = ’Japanese’ 3 = ’European’;
   value Size    1 = ’Small’    2 = ’Medium’   3 = ’Large’;
   value Type    1 = ’Family’   2 = ’Sporty’   3 = ’Work’;
   value Home    1 = ’Own’      2 = ’Rent’;
   value Sex     1 = ’Male’     2 = ’Female’;
   value Income  1 = ’1 Income’ 2 = ’2 Incomes’;
   value Marital 1 = ’Single with Kids’ 2 = ’Married with Kids’
                 3 = ’Single’           4 = ’Married’;
   run;

data Cars;
   missing a;
   input (Origin Size Type Home Income Marital Kids Sex) (1.) @@;
   * Check for End of Line;
   if n(of Origin -- Sex) eq 0 then do; input; return; end;
   marital = 2 * (kids le 0) + marital;
   format Origin Origin. Size Size. Type Type. Home Home.
          Sex Sex. Income Income. Marital Marital.;
   output;
   datalines;
131112212121110121112201131211011211221122112121131122123211222212212201
121122023121221232211101122122022121110122112102131112211121110112311101
211112113211223121122202221122111311123131211102321122223221220221221101
122122022121220211212201221122021122110132112202213112111331226122221101
1212110231AA2202321122121131121121212202122122021121110222221102121212221
211211012211222212211101313112113121220121112212121112212211222221112211
221111011112220122212201131211013121220113112222131112012131110221112211
121112212211121121112201321122311311221113112212213211013121220221221101
133211011212220233311102213111023211122121312222212212111111222121112211
133112011212112212112212212222022131222222121101111122022211220113112212
211112012232220121221102213211011131220121212201211122112331220233312202
222122012111220212112201221122112212220222212211311122012111110112212212
112222011131112221212202322211021222110121221101333211012232110132212101
223222013111220112211101211211022112110212211102221122021111220112111211
111122022121110113311122322111122221210222211101212122021211221232112202
133111011311221121322201213122121111221222112202133122021212111212121.2212
121122.221212102331122122221210113111221212111022111221121211101212121010
311212022231221112112211211211312221221213112212221122022222110131212202
213122211311221212112222113122221221220213111221121211221211221221221102
131122211211220221222101223112012111221212111102223122111311222121111102
2121110121112202133122222311122121312212112.21013121220121111221121112202
111212023121110111112221212111012211220221321101221211122121220112111112
212211022111110122221101121111211212211012212223222112221221122121212112202
213122112211110212121201113211012221110232111102212211012112220121212202
221112011211220121221101211211022211221112121101111112212121221111221201
211122122122111212112221111122312132110113121101121122222111220222121102
221211012122110221221102312111012122220121121101121122221111222212221102
212122021222120113112202121122212121110113111101123112212111220113111101
221112211321210131212211121211011222110122112222123122023121223112212202
311211012131110131221102112211021131220213122201222111022121221221312202
131.225232211101222122221131112412211220221121112131222022122220122122201
212111011311220221312202221122123221210121222202223122121211221221111112
211111121211221221212201113122122131220222112222211122011311110112312211
211222013221220121211211312122122221220122112201111222011211110122311112
312111021231220122121101211112112.221102221122121211221222111101211112101
121211013211222121112222321112112112110121321101113111012221220121312201
213211012212220221211101321122121111220221121101122211021122110213112212
21212201121112213122110112121102221222021212101
;

*---Perform Simple Correspondence Analysis---;
proc corresp all data=Cars outc=Coor;
   tables Marital, Origin;
   run;

*---Plot the Simple Correspondence Analysis Results---;
%plotit(data=Coor, datatype=corresp)
```

Correspondence analysis locates all the categories in a Euclidean space. The first two dimensions of this space are plotted to examine the associations among the categories. Since the smallest dimension of this table is three, there is no loss of information when only two dimensions are plotted. The plot should be thought of as two different overlaid plots, one for each categorical variable. Distances between points within a variable have meaning, but distances between points from different variables do not.

Output 24.1.1. Simple Correspondence Analysis of a Contingency Table

```
                              Car Owners and Car Origin

                                The CORRESP Procedure

                                  Contingency Table

                              American      European      Japanese           Sum

Married                             37            14            51           102
Married with Kids                   52            15            44           111
Single                              33            15            63           111
Single with Kids                     6             1             8            15
Sum                                128            45           166           339


                         Chi-Square Statistic Expected Values

                                     American      European      Japanese

       Married                        38.5133       13.5398       49.9469
       Married with Kids              41.9115       14.7345       54.3540
       Single                         41.9115       14.7345       54.3540
       Single with Kids                5.6637        1.9912        7.3451


                            Observed Minus Expected Values

                                     American      European      Japanese

       Married                        -1.5133        0.4602        1.0531
       Married with Kids              10.0885        0.2655      -10.3540
       Single                         -8.9115        0.2655        8.6460
       Single with Kids                0.3363       -0.9912        0.6549


                   Contributions to the Total Chi-Square Statistic

                              American      European      Japanese           Sum

Married                        0.05946       0.01564       0.02220       0.09730
Married with Kids              2.42840       0.00478       1.97235       4.40553
Single                         1.89482       0.00478       1.37531       3.27492
Single with Kids               0.01997       0.49337       0.05839       0.57173
Sum                            4.40265       0.51858       3.42825       8.34947
```

Output 24.1.1. (continued)

```
                           Car Owners and Car Origin

                            The CORRESP Procedure

                                 Row Profiles

                                American      European      Japanese

        Married                 0.362745      0.137255      0.500000
        Married with Kids       0.468468      0.135135      0.396396
        Single                  0.297297      0.135135      0.567568
        Single with Kids        0.400000      0.066667      0.533333


                               Column Profiles

                                American      European      Japanese

        Married                 0.289063      0.311111      0.307229
        Married with Kids       0.406250      0.333333      0.265060
        Single                  0.257813      0.333333      0.379518
        Single with Kids        0.046875      0.022222      0.048193
```

Output 24.1.1. (continued)

```
                            Car Owners and Car Origin

                              The CORRESP Procedure

                       Inertia and Chi-Square Decomposition

Singular  Principal     Chi-            Cumulative
   Value    Inertia   Square  Percent      Percent    19   38   57   76   95
                                                  ----+----+----+----+----+---
 0.15122    0.02287  7.75160    92.84        92.84  ************************
 0.04200    0.00176  0.59787     7.16       100.00  **

   Total    0.02463  8.34947   100.00

Degrees of Freedom = 6


                                 Row Coordinates

                                                 Dim1       Dim2

                 Married                      -0.0278     0.0134
                 Married with Kids             0.1991     0.0064
                 Single                       -0.1716     0.0076
                 Single with Kids             -0.0144    -0.1947


                      Summary Statistics for the Row Points

                                        Quality       Mass    Inertia

           Married                       1.0000     0.3009     0.0117
           Married with Kids             1.0000     0.3274     0.5276
           Single                        1.0000     0.3274     0.3922
           Single with Kids              1.0000     0.0442     0.0685
```

Output 24.1.1. (continued)

```
                          Car Owners and Car Origin

                            The CORRESP Procedure

              Partial Contributions to Inertia for the Row Points

                                               Dim1       Dim2

                Married                      0.0102     0.0306
                Married with Kids            0.5678     0.0076
                Single                       0.4217     0.0108
                Single with Kids             0.0004     0.9511


Indices of the Coordinates that Contribute Most to Inertia for the Row Points

                                         Dim1       Dim2       Best

          Married                           0          0          2
          Married with Kids                 1          0          1
          Single                            1          0          1
          Single with Kids                  0          2          2


                      Squared Cosines for the Row Points

                                               Dim1       Dim2

                Married                      0.8121     0.1879
                Married with Kids            0.9990     0.0010
                Single                       0.9980     0.0020
                Single with Kids             0.0054     0.9946
```

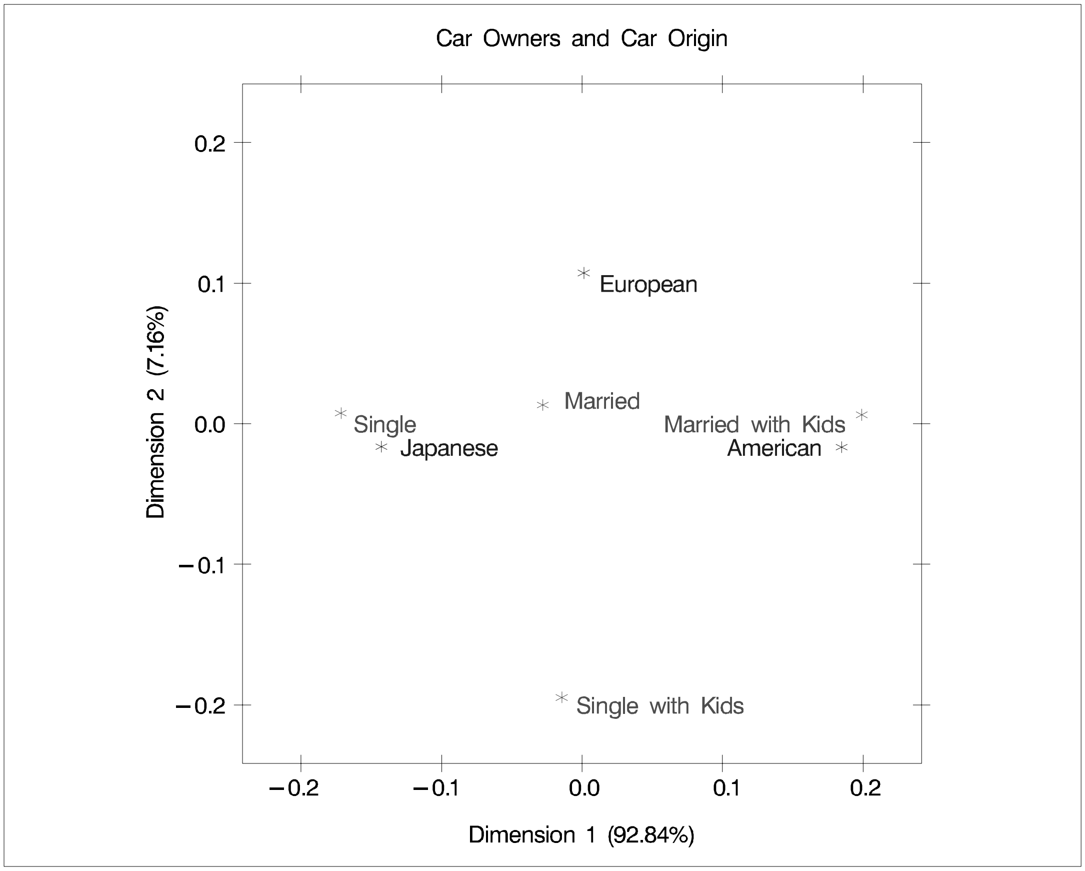
Output 24.1.1. (continued)

```
                          Car Owners and Car Origin

                            The CORRESP Procedure

                              Column Coordinates

                                       Dim1       Dim2

                        American     0.1847    -0.0166
                        European     0.0013     0.1073
                        Japanese    -0.1428    -0.0163


                  Summary Statistics for the Column Points

                              Quality       Mass    Inertia

                  American     1.0000     0.3776     0.5273
                  European     1.0000     0.1327     0.0621
                  Japanese     1.0000     0.4897     0.4106
```

Output 24.1.1. (continued)

```
                          Car Owners and Car Origin

                            The CORRESP Procedure

             Partial Contributions to Inertia for the Column Points

                                      Dim1       Dim2

                        American     0.5634     0.0590
                        European     0.0000     0.8672
                        Japanese     0.4366     0.0737


Indices of the Coordinates that Contribute Most to Inertia for the Column Points

                                 Dim1       Dim2       Best

                  American          1          0          1
                  European          0          2          2
                  Japanese          1          0          1


                     Squared Cosines for the Column Points

                                      Dim1       Dim2

                        American     0.9920     0.0080
                        European     0.0001     0.9999
                        Japanese     0.9871     0.0129
```

Output 24.1.1. (continued)



To interpret the plot, start by interpreting the row points separately from the column points. The European point is near and to the left of the centroid, so it makes a relatively small contribution to the chi-square statistic (because it is near the centroid), it contributes almost nothing to the inertia of dimension one (since its coordinate on dimension one has a small absolute value relative to the other column points), and it makes a relatively large contribution to the inertia of dimension two (since its coordinate on dimension two has a large absolute value relative to the other column points). Its squared cosines for dimension one and two, approximately 0 and 1, respectively, indicate that its position is almost completely determined by its location on dimension two. Its quality of display is 1.0, indicating perfect quality, since the table is two-dimensional after the centering. The American and Japanese points are far from the centroid, and they lie along dimension one. They make relatively large contributions to the chi-square statistic and the inertia of dimension one. The horizontal dimension seems to be largely determined by Japanese versus American car ownership.

In the row points, the Married point is near the centroid, and the Single with Kids point has a small coordinate on dimension one that is near zero. The horizontal dimension seems to be largely determined by the Single versus the Married with

Kids points. The two interpretations of dimension one show the association with being Married with Kids and owning an American car, and being single and owning a Japanese car. The fact that the Married with Kids point is close to the American point and the fact that the Japanese point is near the Single point should be ignored. Distances between row and column points are not defined. The plot shows that more people who are married with kids than you would expect if the rows and columns were independent drive an American car, and more people who are single than you would expect if the rows and columns were independent drive a Japanese car.

Example 24.2. Multiple Correspondence Analysis of Cars and Their Owners

In this example, PROC CORRESP creates a Burt table from categorical data and performs a multiple correspondence analysis. The data are from a sample of individuals who were asked to provide information about themselves and their cars. The questions included origin of the car (American, Japanese, European), size of car (Small, Medium, Large), type of car (Family, Sporty, Work Vehicle), home ownership (Owns, Rents), marital/family status (single, married, single and living with children, and married living with children), and sex (Male, Female).

The data are read and formats assigned in a previous step, displayed in Example 24.1. The variables used in this example are Origin, Size, Type, Income, Home, Marital, and Sex. MCA specifies multiple correspondence analysis, OBSERVED displays the Burt table, and the OUTC= option creates an output coordinate data set. The TABLES statement with only a single variable list and no comma creates the Burt table. The %PLOTIT macro is used to plot the results with vertical and horizontal reference lines.

The data used to produce Output 24.2.1 and Output 24.2.2 can be found in Example 24.1.

```
title ’MCA of Car Owners and Car Attributes’;

*---Perform Multiple Correspondence Analysis---;
proc corresp mca observed data=Cars outc=Coor;
   tables Origin Size Type Income Home Marital Sex;
run;

*---Plot the Multiple Correspondence Analysis Results---;
%plotit(data=Coor, datatype=corresp, href=0, vref=0)
```

Output 24.2.1. Multiple Correspondence Analysis of a Burt Table

```
                                           MCA of Car Owners and Car Attributes

                                                  The CORRESP Procedure

                                                        Burt Table

                             American European Japanese    Large   Medium    Small   Family   Sporty     Work 1 Income

    American                      125        0        0       36       60       29       81       24       20       58
    European                        0       44        0        4       20       20       17       23        4       18
    Japanese                        0        0      165        2       61      102       76       59       30       74
    Large                          36        4        2       42        0        0       30        1       11       20
    Medium                         60       20       61        0      141        0       89       39       13       57
    Small                          29       20      102        0        0      151       55       66       30       73
    Family                         81       17       76       30       89       55      174        0        0       69
    Sporty                         24       23       59        1       39       66        0      106        0       55
    Work                           20        4       30       11       13       30        0        0       54       26
    1 Income                       58       18       74       20       57       73       69       55       26      150
    2 Incomes                      67       26       91       22       84       78      105       51       28        0
    Own                            93       38      111       35      106      101      130       71       41       80
    Rent                           32        6       54        7       35       50       44       35       13       70
    Married                        37       13       51        9       42       50       50       35       16       10
    Married with Kids              50       15       44       21       51       37       79       12       18       27
    Single                         32       15       62       11       40       58       35       57       17       99
    Single with Kids                6        1        8        1        8        6       10        2        3       14
    Female                         58       21       70       17       70       62       83       44       22       47
    Male                           67       23       95       25       71       89       91       62       32      103

                                                       Burt Table

                                                                      Married             Single
                                    2                                    with               with
                              Incomes       Own      Rent   Married      Kids    Single      Kids    Female      Male

    American                       67        93        32        37        50        32         6        58        67
    European                       26        38         6        13        15        15         1        21        23
    Japanese                       91       111        54        51        44        62         8        70        95
    Large                          22        35         7         9        21        11         1        17        25
    Medium                         84       106        35        42        51        40         8        70        71
    Small                          78       101        50        50        37        58         6        62        89
    Family                        105       130        44        50        79        35        10        83        91
    Sporty                         51        71        35        35        12        57         2        44        62
    Work                           28        41        13        16        18        17         3        22        32
    1 Income                        0        80        70        10        27        99        14        47       103
    2 Incomes                     184       162        22        91        82        10         1       102        82
    Own                           162       242         0        76       106        52         8       114       128
    Rent                           22         0        92        25         3        57         7        35        57
    Married                        91        76        25       101         0         0         0        53        48
    Married with Kids              82       106         3         0       109         0         0        48        61
    Single                         10        52        57         0         0       109         0        35        74
    Single with Kids                1         8         7         0         0         0        15        13         2
    Female                        102       114        35        53        48        35        13       149         0
    Male                           82       128        57        48        61        74         2         0       185
```

Output 24.2.1. (continued)

```
                      MCA of Car Owners and Car Attributes

                             The CORRESP Procedure

                      Inertia and Chi-Square Decomposition

Singular  Principal     Chi-           Cumulative
   Value    Inertia    Square  Percent     Percent      4    8   12   16   20
                                                    ----+----+----+----+----+---
 0.56934    0.32415    970.77    18.91       18.91  ************************
 0.48352    0.23380    700.17    13.64       32.55  *****************
 0.42716    0.18247    546.45    10.64       43.19  *************
 0.41215    0.16987    508.73     9.91       53.10  ************
 0.38773    0.15033    450.22     8.77       61.87  ***********
 0.38520    0.14838    444.35     8.66       70.52  ***********
 0.34066    0.11605    347.55     6.77       77.29  ********
 0.32983    0.10879    325.79     6.35       83.64  ********
 0.31517    0.09933    297.47     5.79       89.43  *******
 0.28069    0.07879    235.95     4.60       94.03  ******
 0.26115    0.06820    204.24     3.98       98.01  *****
 0.18477    0.03414    102.24     1.99      100.00  **

   Total    1.71429   5133.92   100.00

Degrees of Freedom = 324
```

Output 24.2.1. (continued)

```
                     MCA of Car Owners and Car Attributes

                            The CORRESP Procedure

                             Column Coordinates

                                            Dim1        Dim2

          American                       -0.4035      0.8129
          European                       -0.0568     -0.5552
          Japanese                        0.3208     -0.4678
          Large                          -0.6949      1.5666
          Medium                         -0.2562      0.0965
          Small                           0.4326     -0.5258
          Family                         -0.4201      0.3602
          Sporty                          0.6604     -0.6696
          Work                            0.0575      0.1539
          1 Income                        0.8251      0.5472
          2 Incomes                      -0.6727     -0.4461
          Own                            -0.3887     -0.0943
          Rent                            1.0225      0.2480
          Married                        -0.4169     -0.7954
          Married with Kids              -0.8200      0.3237
          Single                          1.1461      0.2930
          Single with Kids                0.4373      0.8736
          Female                         -0.3365     -0.2057
          Male                            0.2710      0.1656


                  Summary Statistics for the Column Points

                                 Quality        Mass     Inertia

    American                      0.4925      0.0535      0.0521
    European                      0.0473      0.0188      0.0724
    Japanese                      0.3141      0.0706      0.0422
    Large                         0.4224      0.0180      0.0729
    Medium                        0.0548      0.0603      0.0482
    Small                         0.3825      0.0646      0.0457
    Family                        0.3330      0.0744      0.0399
    Sporty                        0.4112      0.0453      0.0569
    Work                          0.0052      0.0231      0.0699
    1 Income                      0.7991      0.0642      0.0459
    2 Incomes                     0.7991      0.0787      0.0374
    Own                           0.4208      0.1035      0.0230
    Rent                          0.4208      0.0393      0.0604
    Married                       0.3496      0.0432      0.0581
    Married with Kids             0.3765      0.0466      0.0561
    Single                        0.6780      0.0466      0.0561
    Single with Kids              0.0449      0.0064      0.0796
    Female                        0.1253      0.0637      0.0462
    Male                          0.1253      0.0791      0.0372
```

Output 24.2.1. (continued)

```
                  MCA of Car Owners and Car Attributes

                         The CORRESP Procedure

         Partial Contributions to Inertia for the Column Points

                                          Dim1       Dim2

           American                     0.0268     0.1511
           European                     0.0002     0.0248
           Japanese                     0.0224     0.0660
           Large                        0.0268     0.1886
           Medium                       0.0122     0.0024
           Small                        0.0373     0.0764
           Family                       0.0405     0.0413
           Sporty                       0.0610     0.0870
           Work                         0.0002     0.0023
           1 Income                     0.1348     0.0822
           2 Incomes                    0.1099     0.0670
           Own                          0.0482     0.0039
           Rent                         0.1269     0.0103
           Married                      0.0232     0.1169
           Married with Kids            0.0967     0.0209
           Single                       0.1889     0.0171
           Single with Kids             0.0038     0.0209
           Female                       0.0223     0.0115
           Male                         0.0179     0.0093
```

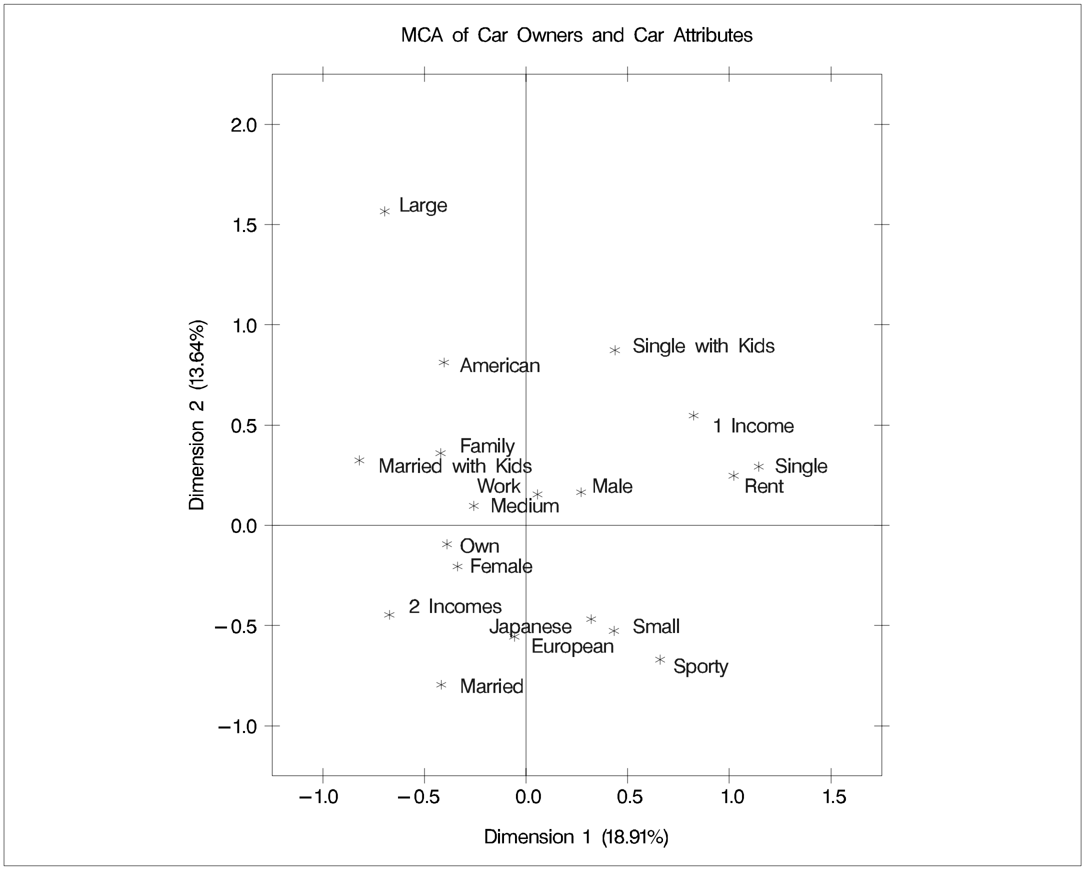
Output 24.2.1. (continued)

```
                       MCA of Car Owners and Car Attributes

                              The CORRESP Procedure

Indices of the Coordinates that Contribute Most to Inertia for the Column Points

                                          Dim1       Dim2       Best

           American                          0          2          2
           European                          0          0          2
           Japanese                          0          2          2
           Large                             0          2          2
           Medium                            0          0          1
           Small                             0          2          2
           Family                            2          0          2
           Sporty                            2          2          2
           Work                              0          0          2
           1 Income                          1          1          1
           2 Incomes                         1          1          1
           Own                               1          0          1
           Rent                              1          0          1
           Married                           0          2          2
           Married with Kids                 1          0          1
           Single                            1          0          1
           Single with Kids                  0          0          2
           Female                            0          0          1
           Male                              0          0          1


                      Squared Cosines for the Column Points

                                                Dim1       Dim2

                 American                     0.0974     0.3952
                 European                     0.0005     0.0468
                 Japanese                     0.1005     0.2136
                 Large                        0.0695     0.3530
                 Medium                       0.0480     0.0068
                 Small                        0.1544     0.2281
                 Family                       0.1919     0.1411
                 Sporty                       0.2027     0.2085
                 Work                         0.0006     0.0046
                 1 Income                     0.5550     0.2441
                 2 Incomes                    0.5550     0.2441
                 Own                          0.3975     0.0234
                 Rent                         0.3975     0.0234
                 Married                      0.0753     0.2742
                 Married with Kids            0.3258     0.0508
                 Single                       0.6364     0.0416
                 Single with Kids             0.0090     0.0359
                 Female                       0.0912     0.0341
                 Male                         0.0912     0.0341
```

Multiple correspondence analysis locates all the categories in a Euclidean space. The first two dimensions of this space are plotted to examine the associations among the categories. The top-right quadrant of the plot shows that the categories single, single with kids, 1 income, and renting a home are associated. Proceeding clockwise, the categories sporty, small, and Japanese are associated. The bottom-left quadrant shows the association between being married, owning your own home, and having two incomes. Having children is associated with owning a large American family car. Such information could be used in market research to identify target audiences for advertisements.

This interpretation is based on points found in approximately the same direction from the origin and in approximately the same region of the space. Distances between points do not have a straightforward interpretation in multiple correspondence analysis. The geometry of multiple correspondence analysis is not a simple generalization of the geometry of simple correspondence analysis (Greenacre and Hastie 1987; Greenacre 1988).

Output 24.2.2. Plot of Multiple Correspondence Analysis of a Burt Table



If you want to perform a multiple correspondence analysis and get scores for the individuals, you can specify the BINARY option to analyze the binary table. In the interest of space, only the first ten rows of coordinates are printed.

```
title 'Car Owners and Car Attributes';
title2 'Binary Table';

*---Perform Multiple Correspondence Analysis---;
proc corresp data=Cars binary;
```

```
   ods select RowCoors;
   tables Origin Size Type Income Home Marital Sex;
run;
```

Output 24.2.3. Correspondence Analysis of a Binary Table

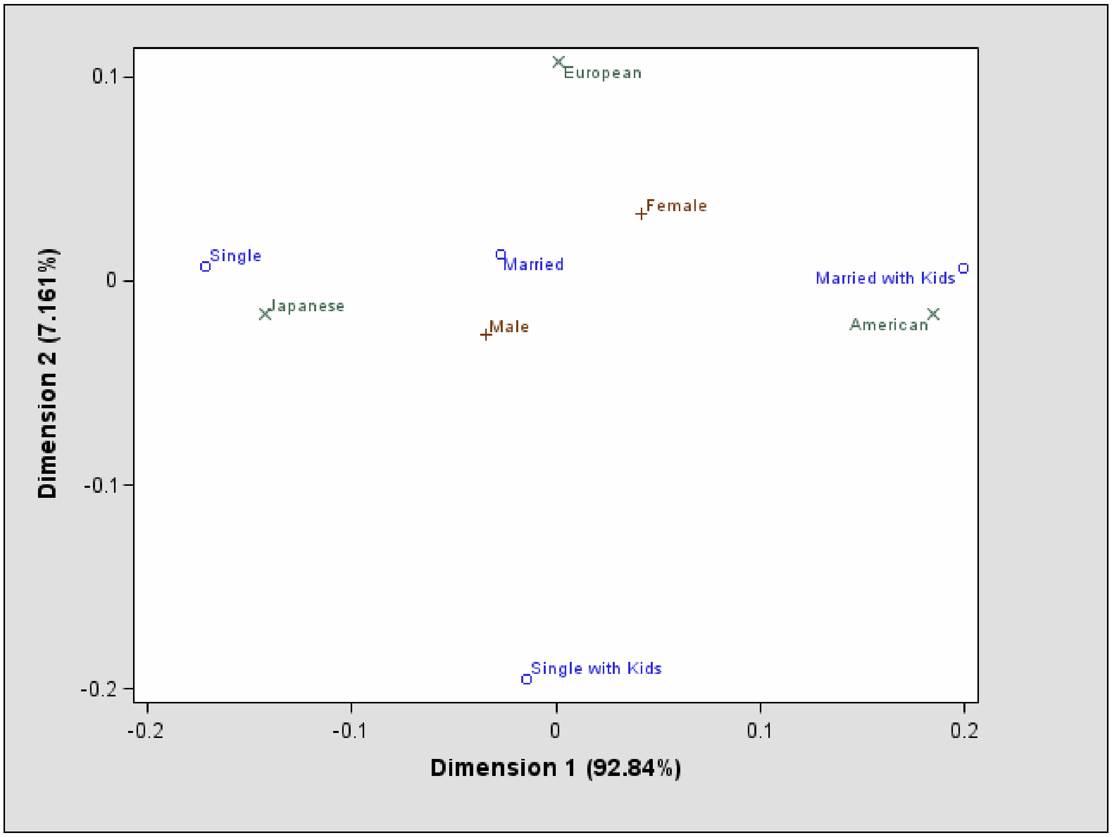
```
                         Car Owners and Car Attributes
                                  Binary Table

                             The Corresp Procedure

                                Row Coordinates

                                     Dim1        Dim2

                         1        -0.4093      1.0878
                         2         0.8198     -0.2221
                         3        -0.2193     -0.5328
                         4         0.4382      1.1799
                         5        -0.6750      0.3600
                         6        -0.1778      0.1441
                         7        -0.9375      0.6846
                         8        -0.7405     -0.1539
                         9        -0.3027     -0.2749
                        10        -0.7263     -0.0803
```

Example 24.3. Cars and Their Owners, ODS Graphics (Experimental)

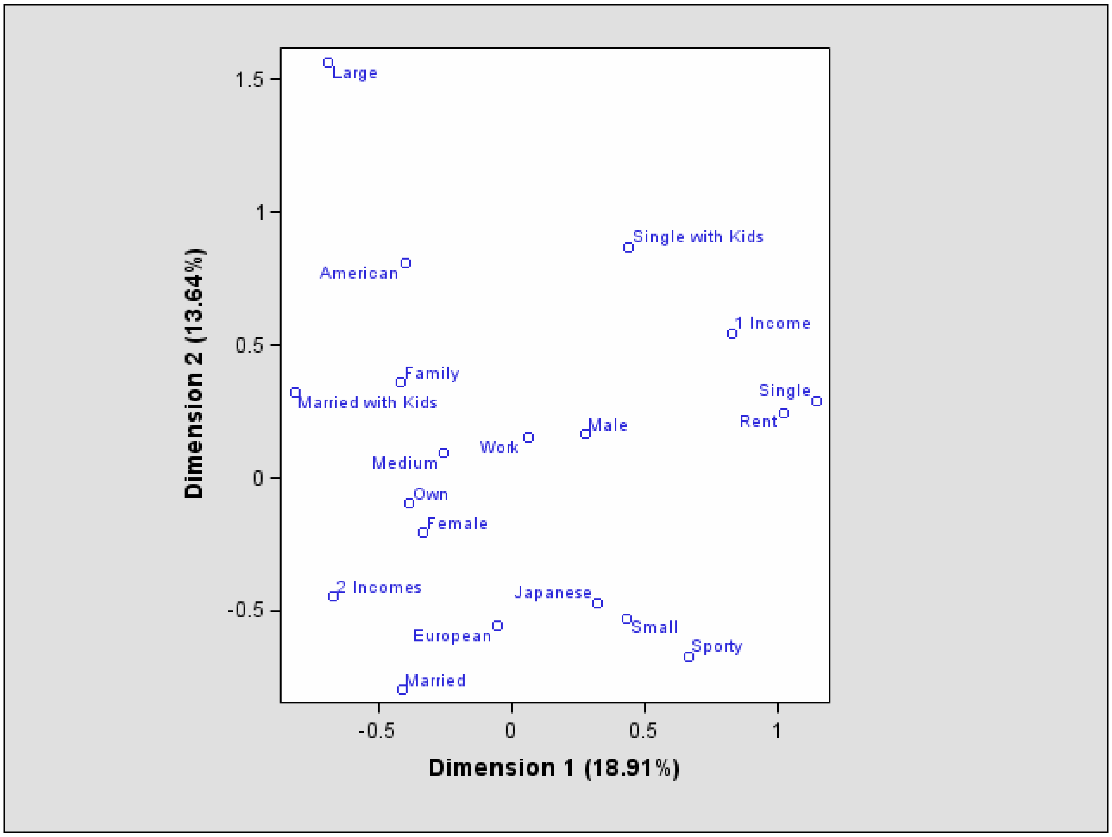
These graphical displays are requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the CORRESP procedure, see the “ODS Graphics” section on page 1109.

```
ods html;
ods graphics on;

*---Perform Simple Correspondence Analysis---;
proc corresp short data=Cars;
   tables Sex Marital, Origin;
   supvar Sex;
run;

*---Perform Multiple Correspondence Analysis---;
proc corresp mca short data=Cars;
   tables Origin Size Type Income Home Marital Sex;
run;

ods graphics off;
ods html close;
```

Output 24.3.1. Simple Correspondence Analysis (Experimental)



Output 24.3.2. Multiple Correspondence Analysis (Experimental)



Example 24.4. Simple Correspondence Analysis of U.S. Population

In this example, PROC CORRESP reads an existing contingency table with supplementary observations and performs a simple correspondence analysis. The data are populations of the fifty states, grouped into regions, for each of the census years from 1920 to 1970 (U.S. Bureau of the Census 1979). Alaska and Hawaii are treated as supplementary regions. They were not states during this entire period and they are not physically connected to the other 48 states. Consequently, it is reasonable to expect that population changes in these two states operate differently from population changes in the other states. The correspondence analysis is performed giving the supplementary points negative weight, then the coordinates for the supplementary points are computed in the solution defined by the other points.

The initial DATA step reads the table, provides labels for the years, flags the supplementary rows with negative weights, and specifies absolute weights of 1000 for all observations since the data were originally reported in units of 1000 people.

In the PROC CORRESP statement, PRINT=PERCENT and the display options display the table of cell percentages (OBSERVED), cell contributions to the total chi-square scaled to sum to 100 (CELLCHI2), row profile rows that sum to 100 (RP), and column profile columns that sum to 100 (CP). The SHORT option specifies that the correspondence analysis summary statistics, contributions to inertia, and squared cosines should not be displayed. The option OUTC=COOR creates the output coordinate data set. Since the data are already in table form, a VAR statement is used to read the table. Row labels are specified with the ID statement, and column labels come from the variable labels. The WEIGHT statement flags the supplementary observations and restores the table values to populations.

The %PLOTIT macro is used to plot the results. Normally, you only need to tell the %PLOTIT macro the name of the input data set, DATA=Coor, and the type of analysis performed on the data, DATATYPE=CORRESP. In this case, PLOTVARS=Dim1 Dim2 is also specified to indicate that Dim1 is the vertical axis variable, as opposed to the default PLOTVARS=Dim2 Dim1.

For an essentially one-dimensional plot such as this, specifying PLOTVARS=Dim1 Dim2 improves the graphical display.

The following statements produce Output 24.4.1 and Output 24.4.2:

```
title 'United States Population';

data USPop;

   * Regions:
   * New England     - ME, NH, VT, MA, RI, CT.
   * Great Lake      - OH, IN, IL, MI, WI.
   * South Atlantic  - DE, MD, DC, VA, WV, NC, SC, GA, FL.
   * Mountain        - MT, ID, WY, CO, NM, AZ, UT, NV.
   * Pacific         - WA, OR, CA.
   *
   * Note: Multiply data values by 1000 to get populations.;

   input Region $14. y1920 y1930 y1940 y1950 y1960 y1970;

   label y1920 = '1920'    y1930 = '1930'    y1940 = '1940'
         y1950 = '1950'    y1960 = '1960'    y1970 = '1970';

   if region = 'Hawaii' or region = 'Alaska'
      then w = -1000;       /* Flag Supplementary Observations */
      else w =  1000;

   datalines;
New England       7401  8166  8437  9314 10509 11842
NY, NJ, PA       22261 26261 27539 30146 34168 37199
Great Lake       21476 25297 26626 30399 36225 40252
Midwest          12544 13297 13517 14061 15394 16319
South Atlantic   13990 15794 17823 21182 25972 30671
KY, TN, AL, MS    8893  9887 10778 11447 12050 12803
AR, LA, OK, TX   10242 12177 13065 14538 16951 19321
Mountain          3336  3702  4150  5075  6855  8282
Pacific           5567  8195  9733 14486 20339 25454
Alaska              55    59    73   129   226   300
Hawaii             256   368   423   500   633   769
```

```
;

*---Perform Simple Correspondence Analysis---;
proc corresp print=percent observed cellchi2 rp cp
     short outc=Coor;
   var y1920 -- y1970;
   id Region;
   weight w;
   run;

*---Plot the Simple Correspondence Analysis Results---;
%plotit(data=Coor, datatype=corresp, plotvars=Dim1 Dim2)
```

The contingency table shows that the population of all regions increased over this time period. The row profiles show that population is increasing at a different rate for the different regions. There is a small increase in population in the Midwest, for example, but the population has more than quadrupled in the Pacific region over the same period. The column profiles show that in 1920, the US population was concentrated in the NY, NJ, PA, Great Lakes, Midwest, and South Atlantic regions. With time, the population is shifting more to the South Atlantic, Mountain, and Pacific regions. This is also clear from the correspondence analysis. The inertia and chi-square decomposition table shows that there are five nontrivial dimensions in the table, but the association between the rows and columns is almost entirely one-dimensional.

Output 24.4.1. Supplementary Observations Example

```
                            United States Population

                             The CORRESP Procedure

                               Contingency Table

Percents                1920     1930     1940     1950     1960     1970      Sum

New England            0.830    0.916    0.946    1.045    1.179    1.328    6.245
NY, NJ, PA             2.497    2.946    3.089    3.382    3.833    4.173   19.921
Great Lake             2.409    2.838    2.987    3.410    4.064    4.516   20.224
Midwest                1.407    1.492    1.516    1.577    1.727    1.831    9.550
South Atlantic         1.569    1.772    1.999    2.376    2.914    3.441   14.071
KY, TN, AL, MS         0.998    1.109    1.209    1.284    1.352    1.436    7.388
AR, LA, OK, TX         1.149    1.366    1.466    1.631    1.902    2.167    9.681
Mountain               0.374    0.415    0.466    0.569    0.769    0.929    3.523
Pacific                0.625    0.919    1.092    1.625    2.282    2.855    9.398
Sum                   11.859   13.773   14.771   16.900   20.020   22.677  100.000


                               Supplementary Rows

Percents         1920         1930         1940         1950         1960         1970

Alaska       0.006170     0.006619     0.008189     0.014471     0.025353     0.033655
Hawaii       0.028719     0.041283     0.047453     0.056091     0.071011     0.086268


                Contributions to the Total Chi-Square Statistic

Percents                1920     1930     1940     1950     1960     1970      Sum

New England            0.937    0.314    0.054    0.009    0.352    0.469    2.135
NY, NJ, PA             0.665    1.287    0.633    0.006    0.521    2.265    5.378
Great Lake             0.004    0.085    0.000    0.001    0.005    0.094    0.189
Midwest                5.749    2.039    0.684    0.072    1.546    4.472   14.563
South Atlantic         0.509    1.231    0.259    0.000    0.285    1.688    3.973
KY, TN, AL, MS         1.454    0.711    1.098    0.087    0.946    2.945    7.242
AR, LA, OK, TX         0.000    0.069    0.077    0.001    0.059    0.030    0.238
Mountain               0.391    0.868    0.497    0.098    0.498    1.834    4.187
Pacific               18.591    9.380    5.458    0.074    7.346   21.248   62.096
Sum                   28.302   15.986    8.761    0.349   11.558   35.046  100.000
```

Output 24.4.1. (continued)

```
                             United States Population

                              The CORRESP Procedure

                                  Row Profiles

  Percents                1920      1930      1940      1950      1960      1970

  New England          13.2947   14.6688   15.1557   16.7310   18.8777   21.2722
  NY, NJ, PA           12.5362   14.7888   15.5085   16.9766   19.2416   20.9484
  Great Lake           11.9129   14.0325   14.7697   16.8626   20.0943   22.3281
  Midwest              14.7348   15.6193   15.8777   16.5167   18.0825   19.1691
  South Atlantic       11.1535   12.5917   14.2093   16.8872   20.7060   24.4523
  KY, TN, AL, MS       13.5033   15.0126   16.3655   17.3813   18.2969   19.4403
  AR, LA, OK, TX       11.8687   14.1111   15.1401   16.8471   19.6433   22.3897
  Mountain             10.6242   11.7898   13.2166   16.1624   21.8312   26.3758
  Pacific               6.6453    9.7823   11.6182   17.2918   24.2784   30.3841


                            Supplementary Row Profiles

Percents        1920         1930         1940         1950         1960         1970

Alaska        6.5321       7.0071       8.6698      15.3207      26.8409      35.6295
Hawaii        8.6809      12.4788      14.3438      16.9549      21.4649      26.0766


                                 Column Profiles

  Percents                1920      1930      1940      1950      1960      1970

  New England           7.0012    6.6511    6.4078    6.1826    5.8886    5.8582
  NY, NJ, PA           21.0586   21.3894   20.9155   20.0109   19.1457   18.4023
  Great Lake           20.3160   20.6042   20.2221   20.1788   20.2983   19.9126
  Midwest              11.8664   10.8303   10.2660    9.3337    8.6259    8.0730
  South Atlantic       13.2343   12.8641   13.5363   14.0606   14.5532   15.1729
  KY, TN, AL, MS        8.4126    8.0529    8.1857    7.5985    6.7521    6.3336
  AR, LA, OK, TX        9.6888    9.9181    9.9227    9.6503    9.4983    9.5581
  Mountain              3.1558    3.0152    3.1519    3.3688    3.8411    4.0971
  Pacific               5.2663    6.6748    7.3921    9.6158   11.3968   12.5921
```

Output 24.4.1. (continued)

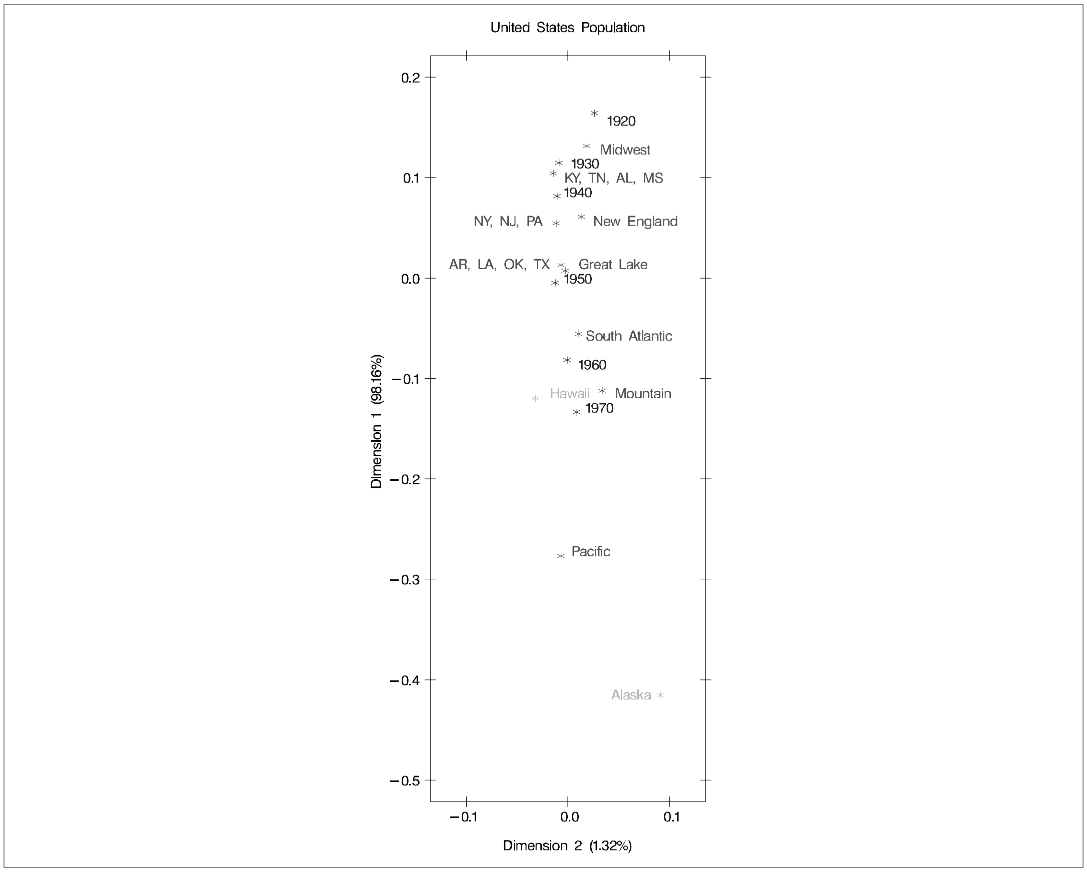
```
                            United States Population

                             The CORRESP Procedure

                      Inertia and Chi-Square Decomposition

Singular  Principal     Chi-            Cumulative
   Value    Inertia   Square  Percent     Percent     20   40   60   80  100
                                                   ----+----+----+----+---
 0.10664    0.01137  1.014E7    98.16       98.16  *************************
 0.01238    0.00015   136586     1.32       99.48
 0.00658    0.00004    38540     0.37       99.85
 0.00333    0.00001   9896.6     0.10       99.95
 0.00244    0.00001   5309.9     0.05      100.00

   Total    0.01159  1.033E7   100.00

Degrees of Freedom = 40


                                Row Coordinates

                                            Dim1       Dim2

                     New England          0.0611     0.0132
                     NY, NJ, PA           0.0546    -0.0117
                     Great Lake           0.0074    -0.0028
                     Midwest              0.1315     0.0186
                     South Atlantic      -0.0553     0.0105
                     KY, TN, AL, MS       0.1044    -0.0144
                     AR, LA, OK, TX       0.0131    -0.0067
                     Mountain            -0.1121     0.0338
                     Pacific             -0.2766    -0.0070


                         Supplementary Row Coordinates

                                        Dim1       Dim2

                         Alaska      -0.4152     0.0912
                         Hawaii      -0.1198    -0.0321


                               Column Coordinates

                                        Dim1       Dim2

                         1920         0.1642     0.0263
                         1930         0.1149    -0.0089
                         1940         0.0816    -0.0108
                         1950        -0.0046    -0.0125
                         1960        -0.0815    -0.0007
                         1970        -0.1335     0.0086
```

The plot shows that the first dimension correctly orders the years. There is nothing in the correspondence analysis that forces this to happen; PROC CORRESP knows nothing about the inherent ordering of the column categories. The ordering of the regions and the ordering of the years reflect the shift over time of the U.S. population from the Northeast quadrant of the country to the South and to the West. The results show that the West and Southeast are growing faster than the rest of the contiguous 48 states.

The plot also shows that the growth pattern for Hawaii is similar to the growth pattern for the mountain states and that Alaska's growth is even more extreme than the Pacific states' growth. The row profiles confirm this interpretation.

The Pacific region is farther from the origin than all other active points. The Midwest is the extreme region in the other direction. The table of contributions to the total chi-square shows that 62% of the total chi-square statistic is contributed by the Pacific region, which is followed by the Midwest at over 14%. Similarly the two extreme years, 1920 and 1970, together contribute over 63% to the total chi-square, whereas the years nearer the origin of the plot contribute less.

Output 24.4.2. Supplementary Observations Example



References

Benzécri, J.P. (1973), *L'Analyse des Données: T. 2, l'Analyse des Correspondances*, Paris: Dunod.

Benzécri, J.P. (1979), *Sur le Calcul des taux d'inertie dans l'analyse d'un questionaire*, Addendum et erratum á [BIN.MULT.]. Cahiers de l'Analyse des Données 4, 377–378.

Burt, C. (1950), "The Factorial Analysis of Qualitative Data," *British Journal of Psychology*, 3, 166–185.

Carroll, J.D, Green, P.E., and Schaffer, C.M. (1986), "Interpoint Distance Comparisons in Correspondence Analysis," *Journal of Marketing Research*, 23, 271–280.

Fisher, R.A. (1940), "The Precision of Discriminant Functions," *Annals of Eugenics*, 10, 422–429.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Greenacre, M.J. (1984), *Theory and Applications of Correspondence Analysis*, London: Academic Press.

Greenacre, M.J. (1988), "Correspondence Analysis of Multivariate Categorical Data by Weighted Least-Squares," *Biometrika*, 75, 457–467.

Greenacre, M.J. (1989), "The Carroll-Green-Schaffer Scaling in Correspondence Analysis: A Theoretical and Empirical Appraisal," *Journal of Market Research*,26, 358–365.

Greenacre, M.J. (1994), "Multiple and Joint Correspondence Analysis," in Greenacre, M.J. and Blasius, J. (ed) *Correspondence Analysis in the Social Sciences*, London: Academic Press.

Greenacre, M.J. and Hastie, T. (1987), "The Geometric Interpretation of Correspondence Analysis," *Journal of the American Statistical Association*, 82, 437–447.

Guttman, L. (1941), "The Quantification of a Class of Attributes: A Theory and Method of Scale Construction," in P. Horst, et al. (ed)., *The Prediction of Personal Adjustment*, New York: Social Science Research Council.

Hayashi, C. (1950), "On the Quantification of Qualitative Data from the Mathematico-Statistical Point of View," *Annals of the Institute of Statistical Mathematics*, 2 (1), 35–47.

van der Heijden, P.G.M, and de Leeuw, J. (1985), "Correspondence Analysis Used Complementary to Loglinear Analysis," *Psychometrika*, 50, 429–447.

Hirshfield, H.O. (1935), "A Connection Between Correlation and Contingency," *Cambridge Philosophical Society Proceedings*, 31, 520–524.

Hoffman, D.L. and Franke, G.R. (1986), “Correspondence Analysis: Graphical Representation of Categorical Data in Marketing Research,” *Journal of Marketing Research*, 23, 213–227.

Horst, P. (1935), “Measuring Complex Attitudes,” *Journal of Social Psychology*, 6, 369–374.

Kobayashi, R. (1981), *An Introduction to Quantification Theory*, Tokyo: Japan Union of Scientists and Engineers.

Komazawa, T. (1982), *Quantification Theory and Data Processing*, Tokyo: Asakura-shoten.

Lebart, L., Morineau, A., and Tabard, N. (1977), *Techniques de la Description Statistique*, Paris: Dunod.

Lebart, L., Morineau, A., and Warwick, K.M. (1984), *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*, New York: John Wiley & Sons, Inc.

Nishisato, S. (1980), *Analysis of Categorical Data: Dual Scaling and Its Applications*, Toronto: University of Toronto Press.

Nishisato, S. (1982), *Quantification of Qualitative Data - Dual Scaling and Its Applications*, Tokyo: Asakura-shoten.

Richardson, M., and Kuder, G.F. (1933), “Making a Rating Scale that Measures,” *Personnel Journal*, 12, 36–40.

Tenenhaus, M. and Young, F.W. (1985), “An Analysis and Synthesis of Multiple Correspondence Analysis, Optimal Scaling, Dual Scaling, Homogeneity Analysis, and Other Methods of Quantifying Categorical Multivariate Data,” *Psychometrika*, 50, 91–119.

U.S. Bureau of the Census (1979), *Statistical Abstract of the United States*, (100th Edition), Washington DC.

Chapter 25
The DISCRIM Procedure

Chapter Contents

Chapter 25
The DISCRIM Procedure

Overview

For a set of observations containing one or more quantitative variables and a classification variable defining groups of observations, the DISCRIM procedure develops a discriminant criterion to classify each observation into one of the groups. The derived discriminant criterion from this data set can be applied to a second data set during the same execution of PROC DISCRIM. The data set that PROC DISCRIM uses to derive the discriminant criterion is called the *training* or *calibration* data set.

When the distribution within each group is assumed to be multivariate normal, a parametric method can be used to develop a discriminant function. The discriminant function, also known as a classification criterion, is determined by a measure of generalized squared distance (Rao 1973). The classification criterion can be based on either the individual within-group covariance matrices (yielding a quadratic function) or the pooled covariance matrix (yielding a linear function); it also takes into account the prior probabilities of the groups. The calibration information can be stored in a special SAS data set and applied to other data sets.

When no assumptions can be made about the distribution within each group, or when the distribution is assumed not to be multivariate normal, nonparametric methods can be used to estimate the group-specific densities. These methods include the kernel and k-nearest-neighbor methods (Rosenblatt 1956; Parzen 1962). The DISCRIM procedure uses uniform, normal, Epanechnikov, biweight, or triweight kernels for density estimation.

Either Mahalanobis or Euclidean distance can be used to determine proximity. Mahalanobis distance can be based on either the full covariance matrix or the diagonal matrix of variances. With a k-nearest-neighbor method, the pooled covariance matrix is used to calculate the Mahalanobis distances. With a kernel method, either the individual within-group covariance matrices or the pooled covariance matrix can be used to calculate the Mahalanobis distances. With the estimated group-specific densities and their associated prior probabilities, the posterior probability estimates of group membership for each class can be evaluated.

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. Given a classification variable and several quantitative variables, PROC DISCRIM derives canonical variables (linear combinations of the quantitative variables) that summarize between-class variation in much the same way that principal components summarize total variation. (See Chapter 21, “The CANDISC Procedure,” for more information on canonical discriminant analysis.) A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of a discriminant criterion, you should use the CANDISC procedure.

The DISCRIM procedure can produce an output data set containing various statistics such as means, standard deviations, and correlations. If a parametric method is used, the discriminant function is also stored in the data set to classify future observations. When canonical discriminant analysis is performed, the output data set includes canonical coefficients that can be rotated by the FACTOR procedure. PROC DISCRIM can also create a second type of output data set containing the classification results for each observation. When canonical discriminant analysis is performed, this output data set also includes canonical variable scores. A third type of output data set containing the group-specific density estimates at each observation can also be produced.

PROC DISCRIM evaluates the performance of a discriminant criterion by estimating error rates (probabilities of misclassification) in the classification of future observations. These error-rate estimates include error-count estimates and posterior probability error-rate estimates. When the input data set is an ordinary SAS data set, the error rate can also be estimated by cross validation.

Do not confuse discriminant analysis with cluster analysis. All varieties of discriminant analysis require prior knowledge of the classes, usually in the form of a sample from each class. In cluster analysis, the data do not include information on class membership; the purpose is to construct a classification.

See Chapter 6, “Introduction to Discriminant Procedures,” for a discussion of discriminant analysis and the SAS/STAT procedures available.

Getting Started

The data in this example are measurements taken on 159 fish caught in Finland’s lake Laengelmavesi. The species, weight, three different length measurements, height, and width of each fish are tallied. The full data set is displayed in Chapter 67, “The STEPDISC Procedure.” The STEPDISC procedure identifies all the variables as significant indicators of the differences among the seven fish species. The goal now is to find a discriminant function based on these six variables that best classifies the fish into species.

First, assume that the data are normally distributed within each group with equal covariances across groups. The following program uses PROC DISCRIM to analyze the Fish data and create Figure 25.1 through Figure 25.5.

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
data fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
   symbol = put(Species, specfmt.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4
1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1
1  363.0 26.3 29.0 33.5 38.0 13.3
 ...[155 more records]
;
proc discrim data=fish;
   class Species;
run;
```

The DISCRIM procedure begins by displaying summary information about the variables in the analysis. This information includes the number of observations, the number of quantitative variables in the analysis (specified with the VAR statement), and the number of classes in the classification variable (specified with the CLASS statement). The frequency of each class, its weight, proportion of the total sample, and prior probability are also displayed. Equal priors are assigned by default.

```
                                 Fish Measurement Data

                                The DISCRIM Procedure

                Observations     158          DF Total               157
                Variables          6          DF Within Classes      151
                Classes            7          DF Between Classes       6


                               Class Level Information

                   Variable                                                 Prior
          Species      Name          Frequency       Weight    Proportion    Probability

          Bream        Bream                34      34.0000      0.215190       0.142857
          Parkki       Parkki               11      11.0000      0.069620       0.142857
          Perch        Perch                56      56.0000      0.354430       0.142857
          Pike         Pike                 17      17.0000      0.107595       0.142857
          Roach        Roach                20      20.0000      0.126582       0.142857
          Smelt        Smelt                14      14.0000      0.088608       0.142857
          Whitefish    Whitefish             6       6.0000      0.037975       0.142857
```

Figure 25.1. Summary Information

The natural log of the determinant of the pooled covariance matrix is displayed next (Figure 25.2). The squared distances between the classes are shown in Figure 25.3.

```
                        Fish Measurement Data

                        The DISCRIM Procedure

                 Pooled Covariance Matrix Information

                                    Natural Log of the
                      Covariance    Determinant of the
                      Matrix Rank    Covariance Matrix

                                6              4.17613
```

Figure 25.2. Pooled Covariance Matrix Information

```
                            Fish Measurement Data

                            The DISCRIM Procedure

                         2         _   _       -1  _   _
                        D (i|j) = (X - X )' COV   (X - X )
                                    i   j           i   j


                    Generalized Squared Distance to Species

From
Species          Bream      Parkki       Perch        Pike       Roach       Smelt   Whitefish

Bream                0    83.32523   243.66688   310.52333   133.06721   252.75503   132.05820
Parkki        83.32523           0    57.09760   174.20918    27.00096    60.52076    26.54855
Perch        243.66688    57.09760           0   101.06791    29.21632    29.26806    20.43791
Pike         310.52333   174.20918   101.06791           0    92.40876   127.82177    99.90673
Roach        133.06721    27.00096    29.21632    92.40876           0    33.84280     6.31997
Smelt        252.75503    60.52076    29.26806   127.82177    33.84280           0    46.37326
Whitefish    132.05820    26.54855    20.43791    99.90673     6.31997    46.37326           0
```

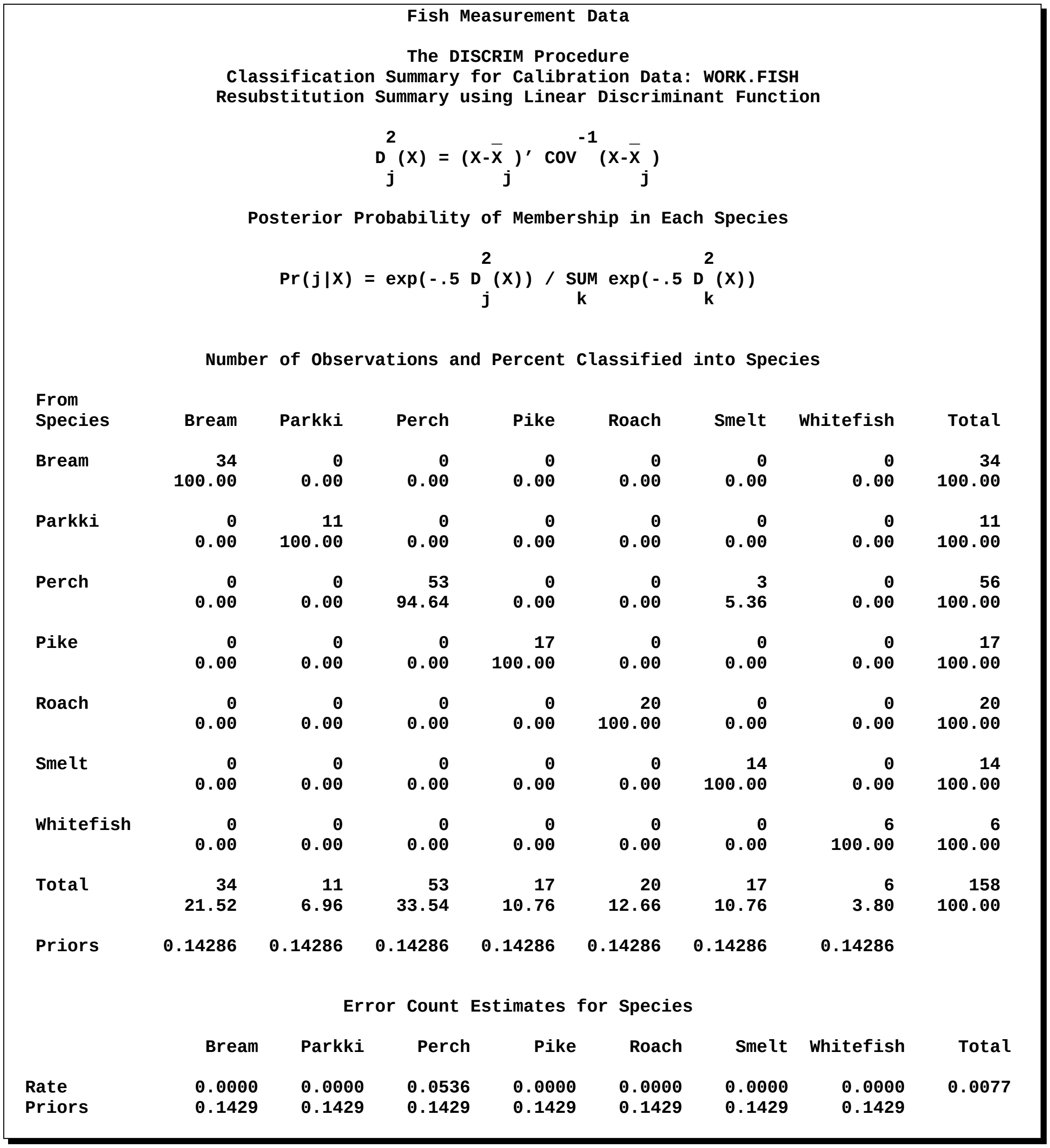
Figure 25.3. Squared Distances

The coefficients of the linear discriminant function are displayed (in Figure 25.4) with the default options METHOD=NORMAL and POOL=YES.

```
                            Fish Measurement Data

                            The DISCRIM Procedure

                          Linear Discriminant Function

                     _     -1 _                                -1 _
     Constant = -.5 X' COV   X       Coefficient Vector = COV    X
                     j        j                                   j


                   Linear Discriminant Function for Species

Variable        Bream      Parkki       Perch        Pike       Roach       Smelt   Whitefish

Constant   -185.91682   -64.92517   -48.68009  -148.06402   -62.65963   -19.70401   -67.44603
Weight       -0.10912    -0.09031    -0.09418    -0.13805    -0.09901    -0.05778    -0.09948
Length1     -23.02273   -13.64180   -19.45368   -20.92442   -14.63635    -4.09257   -22.57117
Length2     -26.70692    -5.38195    17.33061     6.19887    -7.47195    -3.63996     3.83450
Length3      50.55780    20.89531     5.25993    22.94989    25.00702    10.60171    21.12638
Height       13.91638     8.44567    -1.42833    -8.99687    -0.26083    -1.84569     0.64957
Width       -23.71895   -13.38592     1.32749    -9.13410    -3.74542    -3.43630    -2.52442
```

Figure 25.4. Linear Discriminant Function

A summary of how the discriminant function classifies the data used to develop the function is displayed last. In Figure 25.5, you see that only three of the observations are misclassified. The error-count estimates give the proportion of misclassified observations in each group. Since you are classifying the same data that are used to derive the discriminant function, these error-count estimates are biased. One way to reduce the bias of the error-count estimates is to split the Fish data into two sets, use one set to derive the discriminant function, and use the other to run validation tests; Example 25.4 on page 1231 shows how to analyze a test data set. Another method of reducing bias is to classify each observation using a discriminant function computed from all of the other observations; this method is invoked with the CROSSVALIDATE option.



```
                                Fish Measurement Data

                                The DISCRIM Procedure
                    Classification Summary for Calibration Data: WORK.FISH
                   Resubstitution Summary using Linear Discriminant Function

                                 2         _       -1   _
                                D (X) = (X-X )' COV  (X-X )
                                 j          j           j

                        Posterior Probability of Membership in Each Species

                                            2                    2
                        Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                            j        k           k


                     Number of Observations and Percent Classified into Species

   From
   Species         Bream    Parkki     Perch      Pike     Roach     Smelt  Whitefish     Total

   Bream              34         0         0         0         0         0          0        34
                  100.00      0.00      0.00      0.00      0.00      0.00       0.00    100.00

   Parkki              0        11         0         0         0         0          0        11
                    0.00    100.00      0.00      0.00      0.00      0.00       0.00    100.00

   Perch               0         0        53         0         0         3          0        56
                    0.00      0.00     94.64      0.00      0.00      5.36       0.00    100.00

   Pike                0         0         0        17         0         0          0        17
                    0.00      0.00      0.00    100.00      0.00      0.00       0.00    100.00

   Roach               0         0         0         0        20         0          0        20
                    0.00      0.00      0.00      0.00    100.00      0.00       0.00    100.00

   Smelt               0         0         0         0         0        14          0        14
                    0.00      0.00      0.00      0.00      0.00    100.00       0.00    100.00

   Whitefish           0         0         0         0         0         0          6         6
                    0.00      0.00      0.00      0.00      0.00      0.00     100.00    100.00

   Total              34        11        53        17        20        17          6       158
                   21.52      6.96     33.54     10.76     12.66     10.76       3.80    100.00

   Priors        0.14286   0.14286   0.14286   0.14286   0.14286   0.14286    0.14286


                              Error Count Estimates for Species

                     Bream    Parkki     Perch      Pike     Roach     Smelt  Whitefish     Total

  Rate              0.0000    0.0000    0.0536    0.0000    0.0000    0.0000     0.0000    0.0077
  Priors            0.1429    0.1429    0.1429    0.1429    0.1429    0.1429     0.1429
```

Figure 25.5. Resubstitution Misclassification Summary

Syntax

The following statements are available in PROC DISCRIM.

PROC DISCRIM < *options* > **;**

CLASS *variable* **;**

BY *variables* **;**

FREQ *variable* **;**

ID *variable* **;**

PRIORS *probabilities* **;**

TESTCLASS *variable* **;**

TESTFREQ *variable* **;**

TESTID *variable* **;**

VAR *variables* **;**

WEIGHT *variable* **;**

Only the PROC DISCRIM and CLASS statements are required. The following sections describe the PROC DISCRIM statement and then describe the other statements in alphabetical order.

PROC DISCRIM Statement

PROC DISCRIM < *options* > **;**

This statement invokes the DISCRIM procedure. You can specify the following options in the PROC DISCRIM statement.

Tasks	**Options**
Specify Input Data Set	DATA= TESTDATA=
Specify Output Data Set	OUTSTAT= OUT= OUTCROSS= OUTD= TESTOUT= TESTOUTD=
Discriminant Analysis	METHOD= POOL= SLPOOL=
Nonparametric Methods	K= R= KERNEL= METRIC=

Tasks	Options
Classification Rule	THRESHOLD=
Determine Singularity	SINGULAR=
Canonical Discriminant Analysis	CANONICAL CANPREFIX= NCAN=
Resubstitution Classification	LIST LISTERR NOCLASSIFY
Cross Validation Classification	CROSSLIST CROSSLISTERR CROSSVALIDATE
Test Data Classification	TESTLIST TESTLISTERR
Estimate Error Rate	POSTERR
Control Displayed Output	
Correlations	BCORR PCORR TCORR WCORR
Covariances	BCOV PCOV TCOV WCOV
SSCP Matrix	BSSCP PSSCP TSSCP WSSCP
Miscellaneous	ALL ANOVA DISTANCE MANOVA SIMPLE STDMEAN
Suppress output	NOPRINT SHORT

ALL

activates all options that control displayed output. When the derived classification criterion is used to classify observations, the ALL option also activates the POSTERR option.

ANOVA

displays univariate statistics for testing the hypothesis that the class means are equal in the population for each variable.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. You should interpret the between-class covariances in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

CANONICAL
CAN

performs canonical discriminant analysis.

CANPREFIX=*name*

specifies a prefix for naming the canonical variables. By default, the names are Can1, Can2, . . . , Can$n$. If you specify CANPREFIX=ABC, the components are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix, plus the number of digits required to designate the canonical variables, should not exceed 32. The prefix is truncated if the combined length exceeds 32.

The CANONICAL option is activated when you specify either the NCAN= or the CANPREFIX= option. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of discriminant criteria, you should use PROC CANDISC.

CROSSLIST

displays the cross validation classification results for each observation.

CROSSLISTERR

displays the cross validation classification results for misclassified observations only.

CROSSVALIDATE

specifies the cross validation classification of the input DATA= data set. When a parametric method is used, PROC DISCRIM classifies each observation in the DATA= data set using a discriminant function computed from the other observations in the DATA= data set, excluding the observation being classified. When a nonparametric method is used, the covariance matrices used to compute the distances are based on all observations in the data set and do not exclude the observation being classified. However, the observation being classified is excluded from the nonparametric density estimation (if you specify the R= option) or the k nearest neighbors (if you specify the K= option) of that observation. The CROSSVALIDATE option is set when you specify the CROSSLIST, CROSSLISTERR, or OUTCROSS= option.

DATA=*SAS-data-set*

specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS/STAT procedures. These specially structured data sets include TYPE=CORR, TYPE=COV, TYPE=CSSCP, TYPE=SSCP, TYPE=LINEAR, TYPE=QUAD, and TYPE=MIXED. The input data set must be an ordinary SAS data set if you specify METHOD=NPAR. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DISTANCE
MAHALANOBIS

displays the squared Mahalanobis distances between the group means, F statistics, and the corresponding probabilities of greater Mahalanobis squared distances between the group means. The squared distances are based on the specification of the POOL= and METRIC= options.

K=*k*

specifies a k value for the k-nearest-neighbor rule. An observation $\mathbf{x}$ is classified into a group based on the information from the k nearest neighbors of $\mathbf{x}$. Do not specify both the K= and R= options.

KERNEL=BIWEIGHT | BIW
KERNEL=EPANECHNIKOV | EPA
KERNEL=NORMAL | NOR
KERNEL=TRIWEIGHT | TRI
KERNEL=UNIFORM | UNI

specifies a kernel density to estimate the group-specific densities. You can specify the KERNEL= option only when the R= option is specified. The default is KERNEL=UNIFORM.

LIST

displays the resubstitution classification results for each observation. You can specify this option only when the input data set is an ordinary SAS data set.

LISTERR

displays the resubstitution classification results for misclassified observations only. You can specify this option only when the input data set is an ordinary SAS data set.

MANOVA

displays multivariate statistics for testing the hypothesis that the class means are equal in the population.

METHOD=NORMAL | NPAR

determines the method to use in deriving the classification criterion. When you specify METHOD=NORMAL, a parametric method based on a multivariate normal distribution within each class is used to derive a linear or quadratic discriminant function. The default is METHOD=NORMAL. When you specify METHOD=NPAR, a nonparametric method is used and you must also specify either the K= or R= option.

METRIC=DIAGONAL | FULL | IDENTITY

specifies the metric in which the computations of squared distances are performed. If you specify METRIC=FULL, PROC DISCRIM uses either the pooled covariance matrix (POOL=YES) or individual within-group covariance matrices (POOL=NO) to compute the squared distances. If you specify METRIC=DIAGONAL, PROC DISCRIM uses either the diagonal matrix of the pooled covariance matrix (POOL=YES) or diagonal matrices of individual within-group covariance matrices (POOL=NO) to compute the squared distances. If you specify METRIC=IDENTITY, PROC DISCRIM uses Euclidean distance. The default is METRIC=FULL. When you specify METHOD=NORMAL, the option METRIC=FULL is used.

NCAN=*number*

specifies the number of canonical variables to compute. The value of *number* must be less than or equal to the number of variables. If you specify the option NCAN=0, the procedure displays the canonical correlations but not the canonical coefficients, structures, or means. Let v be the number of variables in the VAR statement and c be the number of classes. If you omit the NCAN= option, only $\min(v, c-1)$ canonical variables are generated. If you request an output data set (OUT=, OUTCROSS=, TESTOUT=), v canonical variables are generated. In this case, the last $v - (c-1)$ canonical variables have missing values.

The CANONICAL option is activated when you specify either the NCAN= or the CANPREFIX= option. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of discriminant criterion, you should use PROC CANDISC.

NOCLASSIFY

suppresses the resubstitution classification of the input DATA= data set. You can specify this option only when the input data set is an ordinary SAS data set.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, “Using the Output Delivery System,” for more information.

OUT=*SAS-data-set*

creates an output SAS data set containing all the data from the DATA= data set, plus the posterior probabilities and the class into which each observation is classified by resubstitution. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the “OUT= Data Set” section on page 1170.

OUTCROSS=*SAS-data-set*

creates an output SAS data set containing all the data from the DATA= data set, plus the posterior probabilities and the class into which each observation is classified by cross validation. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the “OUT= Data Set” section on page 1170.

OUTD=*SAS-data-set*

creates an output SAS data set containing all the data from the DATA= data set, plus the group-specific density estimates for each observation. See the "OUT= Data Set" section on page 1170.

OUTSTAT=*SAS-data-set*

creates an output SAS data set containing various statistics such as means, standard deviations, and correlations. When the input data set is an ordinary SAS data set or when TYPE=CORR, TYPE=COV, TYPE=CSSCP, or TYPE=SSCP, this option can be used to generate discriminant statistics. When you specify the CANONICAL option, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class are included in the data set. If you specify METHOD=NORMAL, the output data set also includes coefficients of the discriminant functions, and the output data set is TYPE=LINEAR (POOL=YES), TYPE=QUAD (POOL=NO), or TYPE=MIXED (POOL=TEST). If you specify METHOD=NPAR, this output data set is TYPE=CORR. This data set also holds calibration information that can be used to classify new observations. See the "Saving and Using Calibration Information" section on page 1167 and the "OUT= Data Set" section on page 1170.

PCORR

displays pooled within-class correlations.

PCOV

displays pooled within-class covariances.

POOL=NO | TEST | YES

determines whether the pooled or within-group covariance matrix is the basis of the measure of the squared distance. If you specify POOL=YES, PROC DISCRIM uses the pooled covariance matrix in calculating the (generalized) squared distances. Linear discriminant functions are computed. If you specify POOL=NO, the procedure uses the individual within-group covariance matrices in calculating the distances. Quadratic discriminant functions are computed. The default is POOL=YES.

When you specify METHOD=NORMAL, the option POOL=TEST requests Bartlett's modification of the likelihood ratio test (Morrison 1976; Anderson 1984) of the homogeneity of the within-group covariance matrices. The test is unbiased (Perlman 1980). However, it is not robust to nonnormality. If the test statistic is significant at the level specified by the SLPOOL= option, the within-group covariance matrices are used. Otherwise, the pooled covariance matrix is used. The discriminant function coefficients are displayed only when the pooled covariance matrix is used.

POSTERR

displays the posterior probability error-rate estimates of the classification criterion based on the classification results.

PSSCP

displays the pooled within-class corrected SSCP matrix.

R=*r*

specifies a radius r value for kernel density estimation. With uniform, Epanechnikov, biweight, or triweight kernels, an observation $\mathbf{x}$ is classified into a group based on the information from observations $\mathbf{y}$ in the training set within the radius r of $\mathbf{x}$, that is, the group t observations $\mathbf{y}$ with squared distance $d_t^2(\mathbf{x},\mathbf{y}) \leq r^2$. When a normal kernel is used, the classification of an observation $\mathbf{x}$ is based on the information of the estimated group-specific densities from all observations in the training set. The matrix $r^2\mathbf{V}_t$ is used as the group t covariance matrix in the normal-kernel density, where $\mathbf{V}_t$ is the matrix used in calculating the squared distances. Do not specify both the K= and R= options. For more information on selecting r, see the "Nonparametric Methods" section on page 1158.

SHORT

suppresses the display of certain items in the default output. If you specify METHOD= NORMAL, PROC DISCRIM suppresses the display of determinants, generalized squared distances between-class means, and discriminant function coefficients. When you specify the CANONICAL option, PROC DISCRIM suppresses the display of canonical structures, canonical coefficients, and class means on canonical variables; only tables of canonical correlations are displayed.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the criterion for determining the singularity of a matrix, where $0 < p < 1$. The default is SINGULAR=1E$-$8.

Let $\mathbf{S}$ be the total-sample correlation matrix. If the R^2 for predicting a quantitative variable in the VAR statement from the variables preceding it exceeds $1 - p$, then $\mathbf{S}$ is considered singular. If $\mathbf{S}$ is singular, the probability levels for the multivariate test statistics and canonical correlations are adjusted for the number of variables with R^2 exceeding $1 - p$.

Let $\mathbf{S}_t$ be the group t covariance matrix and $\mathbf{S}_p$ be the pooled covariance matrix. In group t, if the R^2 for predicting a quantitative variable in the VAR statement from the variables preceding it exceeds $1 - p$, then $\mathbf{S}_t$ is considered singular. Similarly, if the partial R^2 for predicting a quantitative variable in the VAR statement from the variables preceding it, after controlling for the effect of the CLASS variable, exceeds $1 - p$, then $\mathbf{S}_p$ is considered singular.

If PROC DISCRIM needs to compute either the inverse or the determinant of a matrix that is considered singular, then it uses a quasi-inverse or a quasi-determinant. For details, see the "Quasi-Inverse" section on page 1164.

SLPOOL=*p*

specifies the significance level for the test of homogeneity. You can specify the SLPOOL= option only when POOL=TEST is also specified. If you specify POOL=

TEST but omit the SLPOOL= option, PROC DISCRIM uses 0.10 as the significance level for the test.

STDMEAN

displays total-sample and pooled within-class standardized class means.

TCORR

displays total-sample correlations.

TCOV

displays total-sample covariances.

TESTDATA=*SAS-data-set*

names an ordinary SAS data set with observations that are to be classified. The quantitative variable names in this data set must match those in the DATA= data set. When you specify the TESTDATA= option, you can also specify the TESTCLASS, TESTFREQ, and TESTID statements. When you specify the TESTDATA= option, you can use the TESTOUT= and TESTOUTD= options to generate classification results and group-specific density estimates for observations in the test data set. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

TESTLIST

lists classification results for all observations in the TESTDATA= data set.

TESTLISTERR

lists only misclassified observations in the TESTDATA= data set but only if a TESTCLASS statement is also used.

TESTOUT=*SAS-data-set*

creates an output SAS data set containing all the data from the TESTDATA= data set, plus the posterior probabilities and the class into which each observation is classified. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the "OUT= Data Set" section on page 1170.

TESTOUTD=*SAS-data-set*

creates an output SAS data set containing all the data from the TESTDATA= data set, plus the group-specific density estimates for each observation. See the "OUT= Data Set" section on page 1170.

THRESHOLD=*p*

specifies the minimum acceptable posterior probability for classification, where $0 \leq p \leq 1$. If the largest posterior probability of group membership is less than the THRESHOLD value, the observation is classified into group OTHER. The default is THRESHOLD=0.

TSSCP

displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC DISCRIM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the DISCRIM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, see the discussion in the *SAS Procedures Guide*.

If you specify the TESTDATA= option and the TESTDATA= data set does not contain any of the BY variables, then the entire TESTDATA= data set is classified according to the discriminant functions computed in each BY group in the DATA= data set.

If the TESTDATA= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the TESTDATA= data set as in the DATA= data set, then PROC DISCRIM displays an error message and stops.

If all BY variables appear in the TESTDATA= data set with the same type and length as in the DATA= data set, then each BY group in the TESTDATA= data set is classified by the discriminant function from the corresponding BY group in the DATA= data set. The BY groups in the TESTDATA= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be exactly the same BY groups in the same order in both data sets. If you omit the NOTSORTED option, some BY groups may appear in one data set but not in the other. If some BY groups appear in the TESTDATA= data set but not in the DATA= data set, and you request an output test data set using the TESTOUT= or TESTOUTD= option, these BY groups are not included in the output data set.

CLASS Statement

CLASS *variable* **;**

The values of the classification variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The specified variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, it is truncated to an integer.

ID Statement

ID *variable* **;**

The ID statement is effective only when you specify the LIST or LISTERR option in the PROC DISCRIM statement. When the DISCRIM procedure displays the classification results, the ID variable (rather than the observation number) is displayed for each observation.

PRIORS Statement

PRIORS EQUAL;

PRIORS PROPORTIONAL | PROP;

PRIORS *probabilities* **;**

The PRIORS statement specifies the prior probabilities of group membership. To set the prior probabilities equal, use

```
priors equal;
```

To set the prior probabilities proportional to the sample sizes, use

```
priors proportional;
```

For other than equal or proportional priors, specify the prior probability for each level of the classification variable. Each class level can be written as either a SAS name or a quoted string, and it must be followed by an equal sign and a numeric constant between zero and one. A SAS name begins with a letter or an underscore and can contain digits as well. Lowercase character values and data values with leading blanks must be enclosed in quotes. For example, to define prior probabilities for each level of Grade, where Grade's values are A, B, C, and D, the PRIORS statement can be

```
priors A=0.1 B=0.3 C=0.5 D=0.1;
```

If Grade's values are 'a', 'b', 'c', and 'd', each class level must be written as a quoted string:

```
priors 'a'=0.1  'b'=0.3  'c'=0.5  'd'=0.1;
```

If Grade is numeric, with formatted values of '1', '2', and '3', the PRIORS statement can be

```
priors '1'=0.3  '2'=0.6  '3'=0.1;
```

The specified class levels must exactly match the formatted values of the CLASS variable. For example, if a CLASS variable C has the format 4.2 and a value 5, the PRIORS statement must specify '5.00', not '5.0' or '5'. If the prior probabilities do not sum to one, these probabilities are scaled proportionally to have the sum equal to one. The default is PRIORS EQUAL.

TESTCLASS Statement

TESTCLASS *variable* **;**

The TESTCLASS statement names the variable in the TESTDATA= data set that is used to determine whether an observation in the TESTDATA= data set is misclassified. The TESTCLASS variable should have the same type (character or numeric) and length as the variable given in the CLASS statement. PROC DISCRIM considers an observation misclassified when the formatted value of the TESTCLASS variable does not match the group into which the TESTDATA= observation is classified. When the TESTCLASS statement is missing and the TESTDATA= data set contains the variable given in the CLASS statement, the CLASS variable is used as the TESTCLASS variable. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

TESTFREQ Statement

TESTFREQ *variable* **;**

If a variable in the TESTDATA= data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a TESTFREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the TESTFREQ variable for the observation.

If the value of the TESTFREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, it is truncated to an integer.

TESTID Statement

TESTID *variable* **;**

The TESTID statement is effective only when you specify the TESTLIST or TESTLISTERR option in the PROC DISCRIM statement. When the DISCRIM procedure displays the classification results for the TESTDATA= data set, the TESTID variable (rather than the observation number) is displayed for each observation. The variable given in the TESTID statement must be in the TESTDATA= data set.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the quantitative variables to be included in the analysis. The default is all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is used.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details

Missing Values

Observations with missing values for variables in the analysis are excluded from the development of the classification criterion. When the values of the classification variable are missing, the observation is excluded from the development of the classification criterion, but if no other variables in the analysis have missing values for that observation, the observation is classified and displayed with the classification results.

Background

The following notation is used to describe the classification methods:

$\mathbf{x}$	a p-dimensional vector containing the quantitative variables of an observation
$\mathbf{S}_p$	the pooled covariance matrix
t	a subscript to distinguish the groups
n_t	the number of training set observations in group t

$\mathbf{m}_t$	the p-dimensional vector containing variable means in group t
$\mathbf{S}_t$	the covariance matrix within group t
$\lvert\mathbf{S}_t\rvert$	the determinant of $\mathbf{S}_t$
q_t	the prior probability of membership in group t
$p(t\|\mathbf{x})$	the posterior probability of an observation $\mathbf{x}$ belonging to group t
f_t	the probability density function for group t
$f_t(\mathbf{x})$	the group-specific density estimate at $\mathbf{x}$ from group t
$f(\mathbf{x})$	$\sum_t q_t f_t(\mathbf{x})$, the estimated unconditional density at $\mathbf{x}$
e_t	the classification error rate for group t

Bayes' Theorem

Assuming that the prior probabilities of group membership are known and that the group-specific densities at $\mathbf{x}$ can be estimated, PROC DISCRIM computes $p(t|\mathbf{x})$, the probability of $\mathbf{x}$ belonging to group t, by applying Bayes' theorem:

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

PROC DISCRIM partitions a p-dimensional vector space into regions R_t, where the region R_t is the subspace containing all p-dimensional vectors $\mathbf{y}$ such that $p(t|\mathbf{y})$ is the largest among all groups. An observation is classified as coming from group t if it lies in region R_t.

Parametric Methods

Assuming that each group has a multivariate normal distribution, PROC DISCRIM develops a discriminant function or classification criterion using a measure of generalized squared distance. The classification criterion is based on either the individual within-group covariance matrices or the pooled covariance matrix; it also takes into account the prior probabilities of the classes. Each observation is placed in the class from which it has the smallest generalized squared distance. PROC DISCRIM also computes the posterior probability of an observation belonging to each class.

The squared Mahalanobis distance from $\mathbf{x}$ to group t is

$$d_t^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_t)'\mathbf{V}_t^{-1}(\mathbf{x} - \mathbf{m}_t)$$

where $\mathbf{V}_t = \mathbf{S}_t$ if the within-group covariance matrices are used, or $\mathbf{V}_t = \mathbf{S}_p$ if the pooled covariance matrix is used.

The group-specific density estimate at $\mathbf{x}$ from group t is then given by

$$f_t(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} |\mathbf{V}_t|^{-\frac{1}{2}} \exp\left(-0.5 d_t^2(\mathbf{x})\right)$$

Using Bayes' theorem, the posterior probability of $\mathbf{x}$ belonging to group t is

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{\sum_u q_u f_u(\mathbf{x})}$$

where the summation is over all groups.

The generalized squared distance from $\mathbf{x}$ to group t is defined as

$$D_t^2(\mathbf{x}) = d_t^2(\mathbf{x}) + g_1(t) + g_2(t)$$

where

$$g_1(t) \quad = \quad \begin{cases} \ln|\mathbf{S}_t| & \text{if the within-group covariance matrices are used} \\ 0 & \text{if the pooled covariance matrix is used} \end{cases}$$

and

$$g_2(t) \quad = \quad \begin{cases} -2\ln(q_t) & \text{if the prior probabilities are not all equal} \\ 0 & \text{if the prior probabilities are all equal} \end{cases}$$

The posterior probability of $\mathbf{x}$ belonging to group t is then equal to

$$p(t|\mathbf{x}) = \frac{\exp\left(-0.5D_t^2(\mathbf{x})\right)}{\sum_u \exp\left(-0.5D_u^2(\mathbf{x})\right)}$$

The discriminant scores are $-0.5D_u^2(\mathbf{x})$. An observation is classified into group u if setting $t = u$ produces the largest value of $p(t|\mathbf{x})$ or the smallest value of $D_t^2(\mathbf{x})$. If this largest posterior probability is less than the threshold specified, $\mathbf{x}$ is classified into group OTHER.

Nonparametric Methods

Nonparametric discriminant methods are based on nonparametric estimates of group-specific probability densities. Either a kernel method or the k-nearest-neighbor method can be used to generate a nonparametric density estimate in each group and to produce a classification criterion. The kernel method uses uniform, normal, Epanechnikov, biweight, or triweight kernels in the density estimation.

Either Mahalanobis distance or Euclidean distance can be used to determine proximity. When the k-nearest-neighbor method is used, the Mahalanobis distances are based on the pooled covariance matrix. When a kernel method is used, the Mahalanobis distances are based on either the individual within-group covariance matrices or the pooled covariance matrix. Either the full covariance matrix or the diagonal matrix of variances can be used to calculate the Mahalanobis distances.

The squared distance between two observation vectors, $\mathbf{x}$ and $\mathbf{y}$, in group t is given by

$$d_t^2(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})' V_t^{-1} (\mathbf{x}-\mathbf{y})$$

where $\mathbf{V}_t$ has one of the following forms:

$$V_t = \begin{cases} \mathbf{S}_p & \text{the pooled covariance matrix} \\ \text{diag}(\mathbf{S}_p) & \text{the diagonal matrix of the pooled covariance matrix} \\ \mathbf{S}_t & \text{the covariance matrix within group } t \\ \text{diag}(\mathbf{S}_t) & \text{the diagonal matrix of the covariance matrix within group } t \\ \mathbf{I} & \text{the identity matrix} \end{cases}$$

The classification of an observation vector $\mathbf{x}$ is based on the estimated group-specific densities from the training set. From these estimated densities, the posterior probabilities of group membership at $\mathbf{x}$ are evaluated. An observation $\mathbf{x}$ is classified into group u if setting $t = u$ produces the largest value of $p(t|\mathbf{x})$. If there is a tie for the largest probability or if this largest probability is less than the threshold specified, $\mathbf{x}$ is classified into group OTHER.

The kernel method uses a fixed radius, r, and a specified kernel, K_t, to estimate the group t density at each observation vector $\mathbf{x}$. Let $\mathbf{z}$ be a p-dimensional vector. Then the volume of a p-dimensional unit sphere bounded by $\mathbf{z}'\mathbf{z} = 1$ is

$$v_0 = \frac{\pi^{\frac{p}{2}}}{\Gamma\left(\frac{p}{2}+1\right)}$$

where Γ represents the gamma function (refer to *SAS Language Reference: Dictionary*).

Thus, in group t, the volume of a p-dimensional ellipsoid bounded by $\{\mathbf{z} \mid \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} = r^2\}$ is

$$v_r(t) = r^p |V_t|^{\frac{1}{2}} v_0$$

The kernel method uses one of the following densities as the kernel density in group t.

Uniform Kernel

$$K_t(\mathbf{z}) = \begin{cases} \dfrac{1}{v_r(t)} & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

Normal Kernel (with mean zero, variance $r^2\mathbf{V}_t$)

$$K_t(\mathbf{z}) = \frac{1}{c_0(t)} \exp\left(-\frac{1}{2r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)$$

where $c_0(t) = (2\pi)^{\frac{p}{2}} r^p |\mathbf{V}_t|^{\frac{1}{2}}$.

Epanechnikov Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_1(t)\left(1 - \frac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right) & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_1(t) = \frac{1}{v_r(t)}\left(1 + \frac{p}{2}\right)$.

Biweight Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_2(t)\left(1 - \frac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)^2 & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_2(t) = \left(1 + \frac{p}{4}\right) c_1(t)$.

Triweight Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_3(t)\left(1 - \frac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)^3 & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_3(t) = \left(1 + \frac{p}{6}\right) c_2(t)$.

The group t density at $\mathbf{x}$ is estimated by

$$f_t(\mathbf{x}) = \frac{1}{n_t}\sum_{\mathbf{y}} K_t(\mathbf{x} - \mathbf{y})$$

where the summation is over all observations $\mathbf{y}$ in group t, and K_t is the specified kernel function. The posterior probability of membership in group t is then given by

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

where $f(\mathbf{x}) = \sum_u q_u f_u(\mathbf{x})$ is the estimated unconditional density. If $f(\mathbf{x})$ is zero, the observation $\mathbf{x}$ is classified into group OTHER.

The uniform-kernel method treats $K_t(\mathbf{z})$ as a multivariate uniform function with density uniformly distributed over $\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2$. Let k_t be the number of training set observations $\mathbf{y}$ from group t within the closed ellipsoid centered at $\mathbf{x}$ specified by $d_t^2(\mathbf{x},\mathbf{y}) \le r^2$. Then the group t density at $\mathbf{x}$ is estimated by

$$f_t(\mathbf{x}) = \frac{k_t}{n_t v_r(t)}$$

When the identity matrix or the pooled within-group covariance matrix is used in calculating the squared distance, $v_r(t)$ is a constant, independent of group membership. The posterior probability of $\mathbf{x}$ belonging to group t is then given by

$$p(t|\mathbf{x}) = \frac{\frac{q_t k_t}{n_t}}{\sum_u \frac{q_u k_u}{n_u}}$$

If the closed ellipsoid centered at $\mathbf{x}$ does not include any training set observations, $f(\mathbf{x})$ is zero and $\mathbf{x}$ is classified into group OTHER. When the prior probabilities are equal, $p(t|\mathbf{x})$ is proportional to k_t/n_t and $\mathbf{x}$ is classified into the group that has the highest proportion of observations in the closed ellipsoid. When the prior probabilities are proportional to the group sizes, $p(t|\mathbf{x}) = k_t/\sum_u k_u$, $\mathbf{x}$ is classified into the group that has the largest number of observations in the closed ellipsoid.

The nearest-neighbor method fixes the number, k, of training set points for each observation $\mathbf{x}$. The method finds the radius $r_k(\mathbf{x})$ that is the distance from $\mathbf{x}$ to the kth nearest training set point in the metric $\mathbf{V}_t^{-1}$. Consider a closed ellipsoid centered at $\mathbf{x}$ bounded by $\{\mathbf{z} \mid (\mathbf{z}-\mathbf{x})'\mathbf{V}_t^{-1}(\mathbf{z}-\mathbf{x}) = r_k^2(\mathbf{x})\}$; the nearest-neighbor method is equivalent to the uniform-kernel method with a location-dependent radius $r_k(\mathbf{x})$. Note that, with ties, more than k training set points may be in the ellipsoid.

Using the k-nearest-neighbor rule, the k_n (or more with ties) smallest distances are saved. Of these k distances, let k_t represent the number of distances that are associated with group t. Then, as in the uniform-kernel method, the estimated group t density at $\mathbf{x}$ is

$$f_t(\mathbf{x}) = \frac{k_t}{n_t v_k(\mathbf{x})}$$

where $v_k(\mathbf{x})$ is the volume of the ellipsoid bounded by $\{\mathbf{z} \mid (\mathbf{z}-\mathbf{x})'\mathbf{V}_t^{-1}(\mathbf{z}-\mathbf{x}) = r_k^2(\mathbf{x})\}$. Since the pooled within-group covariance matrix is used to calculate the distances used in the nearest-neighbor method, the volume $v_k(\mathbf{x})$ is a constant independent of group membership. When $k = 1$ is used in the nearest-neighbor rule, $\mathbf{x}$ is classified into the group associated with the $\mathbf{y}$ point that yields the smallest squared distance $d_t^2(\mathbf{x}, \mathbf{y})$. Prior probabilities affect nearest-neighbor results in the same way that they affect uniform-kernel results.

With a specified squared distance formula (METRIC=, POOL=), the values of r and k determine the degree of irregularity in the estimate of the density function, and they are called smoothing parameters. Small values of r or k produce jagged density estimates, and large values of r or k produce smoother density estimates. Various methods for choosing the smoothing parameters have been suggested, and there is as yet no simple solution to this problem.

For a fixed kernel shape, one way to choose the smoothing parameter r is to plot estimated densities with different values of r and to choose the estimate that is most in accordance with the prior information about the density. For many applications, this approach is satisfactory.

Another way of selecting the smoothing parameter r is to choose a value that optimizes a given criterion. Different groups may have different sets of optimal values. Assume that the unknown density has bounded and continuous second derivatives and that the kernel is a symmetric probability density function. One criterion is to minimize an approximate mean integrated square error of the estimated density (Rosenblatt 1956). The resulting optimal value of r depends on the density function and the kernel. A reasonable choice for the smoothing parameter r is to optimize the criterion with the assumption that group t has a normal distribution with covariance matrix $\mathbf{V}_t$. Then, in group t, the resulting optimal value for r is given by

$$\left(\frac{A(K_t)}{n_t} \right)^{1/(p+4)}$$

where the optimal constant $A(K_t)$ depends on the kernel K_t (Epanechnikov 1969). For some useful kernels, the constants $A(K_t)$ are given by

$$\begin{aligned} A(K_t) &= \frac{1}{p} 2^{p+1} (p+2) \Gamma \left(\frac{p}{2} \right) && \text{with a uniform kernel} \\ A(K_t) &= \frac{4}{2p+1} && \text{with a normal kernel} \\ A(K_t) &= \frac{2^{p+2} p^2 (p+2)(p+4)}{2p+1} \Gamma \left(\frac{p}{2} \right) && \text{with an Epanechnikov kernel} \end{aligned}$$

These selections of $A(K_t)$ are derived under the assumption that the data in each group are from a multivariate normal distribution with covariance matrix $\mathbf{V}_t$. However, when the Euclidean distances are used in calculating the squared distance

$(\mathbf{V}_t = I)$, the smoothing constant should be multiplied by s, where s is an estimate of standard deviations for all variables. A reasonable choice for s is

$$s = \left(\frac{1}{p} \sum s_{jj} \right)^{\frac{1}{2}}$$

where s_{jj} are group t marginal variances.

The DISCRIM procedure uses only a single smoothing parameter for all groups. However, with the selection of the matrix to be used in the distance formula (using the METRIC= or POOL= option), individual groups and variables can have different scalings. When $\mathbf{V}_t$, the matrix used in calculating the squared distances, is an identity matrix, the kernel estimate on each data point is scaled equally for all variables in all groups. When $\mathbf{V}_t$ is the diagonal matrix of a covariance matrix, each variable in group t is scaled separately by its variance in the kernel estimation, where the variance can be the pooled variance $(\mathbf{V}_t = \mathbf{S}_p)$ or an individual within-group variance $(\mathbf{V}_t = \mathbf{S}_t)$. When $\mathbf{V}_t$ is a full covariance matrix, the variables in group t are scaled simultaneously by $\mathbf{V}_t$ in the kernel estimation.

In nearest-neighbor methods, the choice of k is usually relatively uncritical (Hand 1982). A practical approach is to try several different values of the smoothing parameters within the context of the particular application and to choose the one that gives the best cross validated estimate of the error rate.

Classification Error-Rate Estimates

A classification criterion can be evaluated by its performance in the classification of future observations. PROC DISCRIM uses two types of error-rate estimates to evaluate the derived classification criterion based on parameters estimated by the training sample:

- error-count estimates
- posterior probability error-rate estimates.

The error-count estimate is calculated by applying the classification criterion derived from the training sample to a test set and then counting the number of misclassified observations. The group-specific error-count estimate is the proportion of misclassified observations in the group. When the test set is independent of the training sample, the estimate is unbiased. However, it can have a large variance, especially if the test set is small.

When the input data set is an ordinary SAS data set and no independent test sets are available, the same data set can be used both to define and to evaluate the classification criterion. The resulting error-count estimate has an optimistic bias and is called an *apparent error rate*. To reduce the bias, you can split the data into two sets, one set for deriving the discriminant function and the other set for estimating the error rate. Such a split-sample method has the unfortunate effect of reducing the effective sample size.

Another way to reduce bias is cross validation (Lachenbruch and Mickey 1968). Cross validation treats $n - 1$ out of n training observations as a training set. It determines the discriminant functions based on these $n - 1$ observations and then applies them to classify the one observation left out. This is done for each of the n training observations. The misclassification rate for each group is the proportion of sample observations in that group that are misclassified. This method achieves a nearly unbiased estimate but with a relatively large variance.

To reduce the variance in an error-count estimate, smoothed error-rate estimates are suggested (Glick 1978). Instead of summing terms that are either zero or one as in the error-count estimator, the smoothed estimator uses a continuum of values between zero and one in the terms that are summed. The resulting estimator has a smaller variance than the error-count estimate. The posterior probability error-rate estimates provided by the POSTERR option in the PROC DISCRIM statement (see the following section, “Posterior Probability Error-Rate Estimates”) are smoothed error-rate estimates. The posterior probability estimates for each group are based on the posterior probabilities of the observations classified into that same group. The posterior probability estimates provide good estimates of the error rate when the posterior probabilities are accurate. When a parametric classification criterion (linear or quadratic discriminant function) is derived from a nonnormal population, the resulting posterior probability error-rate estimators may not be appropriate.

The overall error rate is estimated through a weighted average of the individual group-specific error-rate estimates, where the prior probabilities are used as the weights.

To reduce both the bias and the variance of the estimator, Hora and Wilcox (1982) compute the posterior probability estimates based on cross validation. The resulting estimates are intended to have both low variance from using the posterior probability estimate and low bias from cross validation. They use Monte Carlo studies on two-group multivariate normal distributions to compare the cross validation posterior probability estimates with three other estimators: the apparent error rate, cross validation estimator, and posterior probability estimator. They conclude that the cross validation posterior probability estimator has a lower mean squared error in their simulations.

Quasi-Inverse

Consider the plot shown in Figure 25.6 with two variables, X1 and X2, and two classes, A and B. The within-class covariance matrix is diagonal, with a positive value for X1 but zero for X2. Using a Moore-Penrose pseudo-inverse would effectively ignore X2 in doing the classification, and the two classes would have a zero generalized distance and could not be discriminated at all. The quasi-inverse used by PROC DISCRIM replaces the zero variance for X2 by a small positive number to remove the singularity. This allows X2 to be used in the discrimination and results correctly in a large generalized distance between the two classes and a zero error rate. It also allows new observations, such as the one indicated by N, to be classified in a reasonable way. PROC CANDISC also uses a quasi-inverse when the total-sample covariance matrix is considered to be singular and Mahalanobis distances are requested. This problem with singular within-class covariance matrices is discussed in Ripley (1996, p. 38). The use of the quasi-inverse is an innovation introduced by SAS Institute Inc.

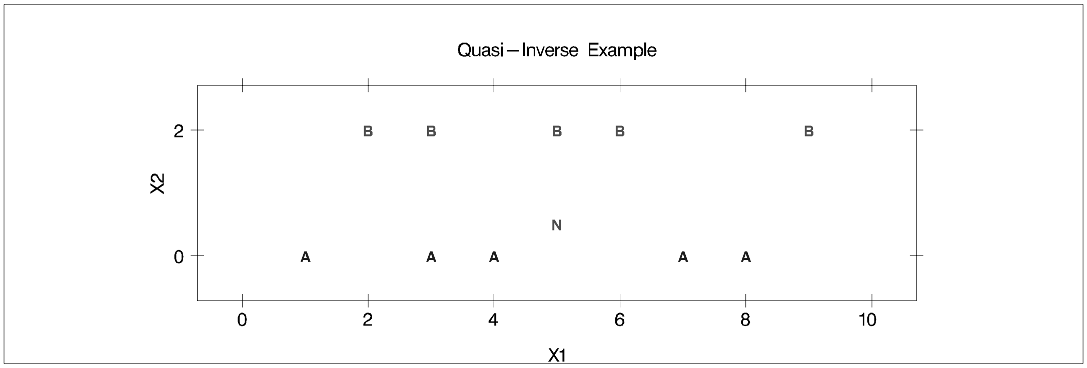


Figure 25.6. Plot of Data with Singular Within-Class Covariance Matrix

Let $\mathbf{S}$ be a singular covariance matrix. The matrix $\mathbf{S}$ can be either a within-group covariance matrix, a pooled covariance matrix, or a total-sample covariance matrix. Let v be the number of variables in the VAR statement and the nullity n be the number of variables among them with (partial) R^2 exceeding $1 - p$. If the determinant of $\mathbf{S}$ (Testing of Homogeneity of Within Covariance Matrices) or the inverse of $\mathbf{S}$ (Squared Distances and Generalized Squared Distances) is required, a quasi-determinant or quasi-inverse is used instead. PROC DISCRIM scales each variable to unit total-sample variance before calculating this quasi-inverse. The calculation is based on the spectral decomposition $\mathbf{S} = \mathbf{\Gamma\Lambda\Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_j, $j = 1, \ldots, v$, where $\lambda_i \geq \lambda_j$ when $i < j$, and $\mathbf{\Gamma}$ is a matrix with the corresponding orthonormal eigenvectors of $\mathbf{S}$ as columns. When the nullity n is less than v, set $\lambda_j^0 = \lambda_j$ for $j = 1, \ldots, v - n$, and $\lambda_j^0 = p\bar{\lambda}$ for $j = v - n + 1, \ldots, v$, where

$$\bar{\lambda} = \frac{1}{v-n} \sum_{k=1}^{v-n} \lambda_k$$

When the nullity n is equal to v, set $\lambda_j^0 = p$, for $j = 1, \ldots, v$. A quasi-determinant is then defined as the product of λ_j^0, $j = 1, \ldots, v$. Similarly, a quasi-inverse is then defined as $\mathbf{S}^* = \mathbf{\Gamma\Lambda}^*\mathbf{\Gamma}'$, where $\mathbf{\Lambda}^*$ is a diagonal matrix of values $1/\lambda_j^0$, $j = 1, \ldots, v$.

Posterior Probability Error-Rate Estimates

The posterior probability error-rate estimates (Fukunaga and Kessell 1973; Glick 1978; Hora and Wilcox 1982) for each group are based on the posterior probabilities of the observations classified into that same group.

A sample of observations with classification results can be used to estimate the posterior error rates. The following notation is used to describe the sample.

$\mathcal{S}$	the set of observations in the (training) sample
n	the number of observations in $\mathcal{S}$
n_t	the number of observations in $\mathcal{S}$ in group t
$\mathcal{R}_t$	the set of observations such that the posterior probability belonging to group t is the largest
$\mathcal{R}_{ut}$	the set of observations from group u such that the posterior probability belonging to group t is the largest.

The classification error rate for group t is defined as

$$e_t = 1 - \int_{\mathcal{R}_t} f_t(\mathbf{x}) d\mathbf{x}$$

The posterior probability of $\mathbf{x}$ for group t can be written as

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

where $f(\mathbf{x}) = \sum_u q_u f_u(\mathbf{x})$ is the unconditional density of $\mathbf{x}$.

Thus, if you replace $f_t(\mathbf{x})$ with $p(t|\mathbf{x})f(\mathbf{x})/q_t$, the error rate is

$$e_t = 1 - \frac{1}{q_t}\int_{\mathcal{R}_t} p(t|\mathbf{x})f(\mathbf{x})d\mathbf{x}$$

An estimator of e_t, unstratified over the groups from which the observations come, is then given by

$$\hat{e}_t \text{ (unstratified)} = 1 - \frac{1}{nq_t}\sum_{\mathcal{R}_t} p(t|\mathbf{x})$$

where $p(t|\mathbf{x})$ is estimated from the classification criterion, and the summation is over all sample observations of $\mathcal{S}$ classified into group t. The true group membership of each observation is not required in the estimation. The term nq_t is the number of observations that are expected to be classified into group t, given the priors. If more observations than expected are classified into group t, then $\hat{e}_t$ can be negative.

Further, if you replace $f(\mathbf{x})$ with $\sum_u q_u f_u(\mathbf{x})$, the error rate can be written as

$$e_t = 1 - \frac{1}{q_t}\sum_u q_u \int_{\mathcal{R}_{ut}} p(t|\mathbf{x})f_u(\mathbf{x})d\mathbf{x}$$

and an estimator stratified over the group from which the observations come is given by

$$\hat{e}_t \text{ (stratified)} = 1 - \frac{1}{q_t}\sum_u q_u \frac{1}{n_u}\left(\sum_{\mathcal{R}_{ut}} p(t|\mathbf{x})\right)$$

The inner summation is over all sample observations of $\mathcal{S}$ coming from group u and classified into group t, and n_u is the number of observations originally from group u. The stratified estimate uses only the observations with known group membership. When the prior probabilities of the group membership are proportional to the group sizes, the stratified estimate is the same as the unstratified estimator.

The estimated group-specific error rates can be less than zero, usually due to a large discrepancy between prior probabilities of group membership and group sizes. To have a reliable estimate for group-specific error rate estimates, you should use group sizes that are at least approximately proportional to the prior probabilities of group membership.

A total error rate is defined as a weighted average of the individual group error rates

$$e = \sum_t q_t e_t$$

and can be estimated from

$$\hat{e} \text{ (unstratified)} = \sum_t q_t \hat{e}_t \text{ (unstratified)}$$

or

$$\hat{e}\ (\text{stratified}) = \sum_t q_t \hat{e}_t\ (\text{stratified})$$

The total unstratified error-rate estimate can also be written as

$$\hat{e}\ (\text{unstratified}) = 1 - \frac{1}{n} \sum_t \sum_{\mathcal{R}_t} p(t|\mathbf{x})$$

which is one minus the average value of the maximum posterior probabilities for each observation in the sample. The prior probabilities of group membership do not appear explicitly in this overall estimate.

Saving and Using Calibration Information

When you specify METHOD=NORMAL to derive a linear or quadratic discriminant function, you can save the calibration information developed by the DISCRIM procedure in a SAS data set by using the OUTSTAT= option in the procedure. PROC DISCRIM then creates a specially structured SAS data set of TYPE=LINEAR, TYPE=QUAD, or TYPE=MIXED that contains the calibration information. For more information on these data sets, see Appendix A, "Special SAS Data Sets." Calibration information cannot be saved when METHOD=NPAR, but you can classify a TESTDATA= data set in the same step. For an example of this, see Example 25.1 on page 1180.

To use this calibration information to classify observations in another data set, specify both of the following:

- the name of the calibration data set after the DATA= option in the PROC DISCRIM statement
- the name of the data set to be classified after the TESTDATA= option in the PROC DISCRIM statement.

Here is an example:

```
data original;
   input position x1 x2;
   datalines;
 ...[data lines]
;

proc discrim outstat=info;
   class position;
run;

data check;
   input position x1 x2;
   datalines;
 ...[second set of data lines]
;

proc discrim data=info testdata=check testlist;
   class position;
run;
```

The first DATA step creates the SAS data set Original, which the DISCRIM procedure uses to develop a classification criterion. Specifying OUTSTAT=INFO in the PROC DISCRIM statement causes the DISCRIM procedure to store the calibration information in a new data set called Info. The next DATA step creates the data set Check. The second PROC DISCRIM statement specifies DATA=INFO and TESTDATA=CHECK so that the classification criterion developed earlier is applied to the Check data set. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

Input Data Sets

DATA= Data Set

When you specify METHOD=NPAR, an ordinary SAS data set is required as the input DATA= data set. When you specify METHOD=NORMAL, the DATA= data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS/STAT procedures. These specially structured data sets include

- TYPE=CORR data sets created by PROC CORR using a BY statement
- TYPE=COV data sets created by PROC PRINCOMP using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG using both the OUTSSCP= option and a BY statement
- TYPE=LINEAR, TYPE=QUAD, and TYPE=MIXED data sets produced by previous runs of PROC DISCRIM that used both METHOD=NORMAL and OUTSTAT= options

When the input data set is TYPE=CORR, TYPE=COV, TYPE=CSSCP, or TYPE=SSCP, the BY variable in these data sets becomes the CLASS variable in the DISCRIM procedure.

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, PROC DISCRIM reads the number of observations for each class from the observations with _TYPE_='N' and reads the variable means in each class from the observations with _TYPE_='MEAN'. PROC DISCRIM then reads the within-class correlations from the observations with _TYPE_='CORR' and reads the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and cross products from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When you specify POOL=YES and the data set does not include any observations with _TYPE_='CSSCP' (data set TYPE=CSSCP), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CORR' (data set TYPE=CORR) for each class, PROC DISCRIM reads the pooled within-class information from the data set. In this case, PROC DISCRIM reads the pooled within-class covariances from the observations with _TYPE_='PCOV' (data set TYPE=COV) or reads the pooled within-class correlations from the observations with _TYPE_='PCORR' and the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR) or the pooled within-class corrected SSCP matrix from the observations with _TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, the DISCRIM procedure reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations for each class from the variable INTERCEP in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and cross products from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='variablenames'.

When the input data set is TYPE=LINEAR, TYPE=QUAD, or TYPE=MIXED, PROC DISCRIM reads the prior probabilities for each class from the observations with variable _TYPE_='PRIOR'.

When the input data set is TYPE=LINEAR, PROC DISCRIM reads the coefficients of the linear discriminant functions from the observations with variable _TYPE_='LINEAR' (see page 1173).

When the input data set is TYPE=QUAD, PROC DISCRIM reads the coefficients of the quadratic discriminant functions from the observations with variable _TYPE_='QUAD' (see page 1173).

When the input data set is TYPE=MIXED, PROC DISCRIM reads the coefficients of the linear discriminant functions from the observations with variable _TYPE_='LINEAR'. If there are no observations with _TYPE_='LINEAR', PROC DISCRIM then reads the coefficients of the quadratic discriminant functions from the observations with variable _TYPE_='QUAD' (see page 1173).

TESTDATA= Data Set

The TESTDATA= data set is an ordinary SAS data set with observations that are to be classified. The quantitative variable names in this data set must match those in the DATA= data set. The TESTCLASS statement can be used to specify the variable containing group membership information of the TESTDATA= data set observations. When the TESTCLASS statement is missing and the TESTDATA= data set contains the variable given in the CLASS statement, this variable is used as the TESTCLASS variable. The TESTCLASS variable should have the same type (character or numeric) and length as the variable given in the CLASS statement. PROC DISCRIM considers an observation misclassified when the value of the TESTCLASS variable does not match the group into which the TESTDATA= observation is classified.

Output Data Sets

When an output data set includes variables containing the posterior probabilities of group membership (OUT=, OUTCROSS=, or TESTOUT= data sets) or group-specific density estimates (OUTD= or TESTOUTD= data sets), the names of these variables are constructed from the formatted values of the class levels converted to valid SAS variable names.

OUT= Data Set

The OUT= data set contains all the variables in the DATA= data set, plus new variables containing the posterior probabilities and the resubstitution classification results. The names of the new variables containing the posterior probabilities are constructed from the formatted values of the class levels converted to SAS names. A new variable, _INTO_, with the same attributes as the CLASS variable, specifies the class to which each observation is assigned. If an observation is classified into group OTHER, the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of canonical variables. The names of the canonical variables are constructed as described in the CANPREFIX= option. The canonical variables have means equal to zero and pooled within-class variances equal to one.

An OUT= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTD= Data Set

The OUTD= data set contains all the variables in the DATA= data set, plus new variables containing the group-specific density estimates. The names of the new variables containing the density estimates are constructed from the formatted values of the class levels.

An OUTD= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTCROSS= Data Set

The OUTCROSS= data set contains all the variables in the DATA= data set, plus new variables containing the posterior probabilities and the classification results of cross validation. The names of the new variables containing the posterior probabilities are constructed from the formatted values of the class levels. A new variable, _INTO_, with the same attributes as the CLASS variable, specifies the class to which each observation is assigned. When an observation is classified into group OTHER, the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of new variables. The names of the new variables are constructed as described in the CANPREFIX= option. The new variables have mean zero and pooled within-class variance equal to one.

An OUTCROSS= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

TESTOUT= Data Set

The TESTOUT= data set contains all the variables in the TESTDATA= data set, plus new variables containing the posterior probabilities and the classification results. The names of the new variables containing the posterior probabilities are formed from the formatted values of the class levels. A new variable, _INTO_, with the same attributes as the CLASS variable, gives the class to which each observation is assigned. If an observation is classified into group OTHER, the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of new variables. The names of the new variables are formed as described in the CANPREFIX= option.

TESTOUTD= Data Set

The TESTOUTD= data set contains all the variables in the TESTDATA= data set, plus new variables containing the group-specific density estimates. The names of the new variables containing the density estimates are formed from the formatted values of the class levels.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure. The data set contains various statistics such as means, standard deviations, and correlations. For an example of an OUTSTAT= data set, see Example 25.3 on page 1222. When you specify the CANONICAL option, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class are included in the data set.

If you specify METHOD=NORMAL, the output data set also includes coefficients of the discriminant functions, and the data set is TYPE=LINEAR (POOL=YES), TYPE=QUAD (POOL=NO), or TYPE=MIXED (POOL=TEST). If you specify METHOD=NPAR, this output data set is TYPE=CORR.

The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- the CLASS variable
- _TYPE_, a character variable of length 8 that identifies the type of statistic
- _NAME_, a character variable of length 32 that identifies the row of the matrix, the name of the canonical variable, or the type of the discriminant function coefficients
- the quantitative variables, that is, those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement

The observations, as identified by the variable _TYPE_, have the following _TYPE_ values:

TYPE	**Contents**
N	number of observations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
SUMWGT	sum of weights both for the total sample (CLASS variable missing) and within each class (CLASS variable present), if a WEIGHT statement is specified
MEAN	means both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PRIOR	prior probability for each class
STDMEAN	total-standardized class means
PSTDMEAN	pooled within-class standardized class means
STD	standard deviations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSTD	pooled within-class standard deviations
BSTD	between-class standard deviations
RSQUARED	univariate R^2s
LNDETERM	the natural log of the determinant or the natural log of the quasi-determinant of the within-class covariance matrix either pooled (CLASS variable missing) or not pooled (CLASS variable present)

The following kinds of observations are identified by the combination of the variables _TYPE_ and _NAME_. When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the row of the matrix.

TYPE	**Contents**
CSSCP	corrected SSCP matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSSCP	pooled within-class corrected SSCP matrix
BSSCP	between-class SSCP matrix
COV	covariance matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCOV	pooled within-class covariance matrix
BCOV	between-class covariance matrix
CORR	correlation matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCORR	pooled within-class correlation matrix
BCORR	between-class correlation matrix

When you request canonical discriminant analysis, the _TYPE_ variable can have one of the following values. The _NAME_ variable identifies a canonical variable.

TYPE	**Contents**
CANCORR	canonical correlations
STRUCTUR	canonical structure
BSTRUCT	between canonical structure
PSTRUCT	pooled within-class canonical structure
SCORE	standardized canonical coefficients
RAWSCORE	raw canonical coefficients
CANMEAN	means of the canonical variables for each class

When you specify METHOD=NORMAL, the _TYPE_ variable can have one of the following values. The _NAME_ variable identifies different types of coefficients in the discriminant function.

TYPE	**Contents**
LINEAR	coefficients of the linear discriminant functions
QUAD	coefficients of the quadratic discriminant functions

The values of the _NAME_ variable are as follows:

NAME	**Contents**
variable names	quadratic coefficients of the quadratic discriminant functions (a symmetric matrix for each class)
LINEAR	linear coefficients of the discriminant functions
CONST	constant coefficients of the discriminant functions

Computational Resources

In the following discussion, let

$n =$	number of observations in the training data set
$v =$	number of variables
$c =$	number of class levels
$k =$	number of canonical variables
$l =$	length of the CLASS variable

Memory Requirements

The amount of temporary storage required depends on the discriminant method used and the options specified. The least amount of temporary storage in bytes needed to process the data is approximately

$$c(32v + 3l + 128) + 8v^2 + 104v + 4l$$

A parametric method (METHOD=NORMAL) requires an additional temporary memory of $12v^2 + 100v$ bytes. When you specify the CROSSVALIDATE option, this temporary storage must be increased by $4v^2 + 44v$ bytes. When a nonparametric method (METHOD=NPAR) is used, an additional temporary storage of $10v^2 + 94v$ bytes is needed if you specify METRIC=FULL to evaluate the distances.

With the MANOVA option, the temporary storage must be increased by $8v^2 + 96v$ bytes. The CANONICAL option requires a temporary storage of $2v^2+94v+8k(v+c)$ bytes. The POSTERR option requires a temporary storage of $8c^2 + 64c + 96$ bytes. Additional temporary storage is also required for classification summary and for each output data set.

For example, in the following statements,

```
proc discrim manova;
   class gp;
   var x1 x2 x3;
run;
```

if the CLASS variable gp has a length of eight and the input data set contains two class levels, the procedure requires a temporary storage of 1992 bytes. This includes 1104 bytes for data processing, 480 bytes for using a parametric method, and 408 bytes for specifying the MANOVA option.

Time Requirements

The following factors determine the time requirements of discriminant analysis.

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. PROC DISCRIM must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n\ln(c)$.
- The time for inverting a covariance matrix is proportional to v^3.
- With a parametric method, the time required to classify each observation is proportional to cv for a linear discriminant function and is proportional to cv^2 for a quadratic discriminant function. When you specify the CROSSVALIDATE option, the discriminant function is updated for each observation in the classification. A substantial amount of time is required.
- With a nonparametric method, the data are stored in a tree structure (Friedman, Bentley, and Finkel 1977). The time required to organize the observations into the tree structure is proportional to $nv\ln(n)$. The time for performing each tree

search is proportional to $\ln(n)$. When you specify the normal KERNEL= option, all observations in the training sample contribute to the density estimation and more computer time is needed.

- The time required for the canonical discriminant analysis is proportional to v^3.

Each of the preceding factors has a different machine-dependent constant of proportionality.

Displayed Output

The displayed output from PROC DISCRIM includes the following:

- Class Level Information, including the values of the classification variable, Variable Name constructed from each class value, the Frequency and Weight of each value, its Proportion in the total sample, and the Prior Probability for each class level.

Optional output includes the following:

- Within-Class SSCP Matrices for each group
- Pooled Within-Class SSCP Matrix
- Between-Class SSCP Matrix
- Total-Sample SSCP Matrix
- Within-Class Covariance Matrices, $\mathbf{S}_t$, for each group
- Pooled Within-Class Covariance Matrix, $\mathbf{S}_p$
- Between-Class Covariance Matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-Sample Covariance Matrix
- Within-Class Correlation Coefficients and $\Pr > |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled Within-Class Correlation Coefficients and $\Pr > |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-Class Correlation Coefficients and $\Pr > |r|$ to test the hypothesis that the between-class population correlation coefficients are zero
- Total-Sample Correlation Coefficients and $\Pr > |r|$ to test the hypothesis that the total population correlation coefficients are zero
- Simple descriptive Statistics including N (the number of observations), Sum, Mean, Variance, and Standard Deviation both for the total sample and within each class
- Total-Sample Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the total sample standard deviation

- Pooled Within-Class Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation
- Pairwise Squared Distances Between Groups
- Univariate Test Statistics, including Total-Sample Standard Deviations, Pooled Within-Class Standard Deviations, Between-Class Standard Deviations, R^2, $R^2/(1 - R^2)$, F, and $\mathrm{Pr} > F$ (univariate F values and probability levels for one-way analyses of variance)
- Multivariate Statistics and F Approximations, including Wilks' Lambda, Pillai's Trace, Hotelling-Lawley Trace, and Roy's Greatest Root with F approximations, degrees of freedom (Num DF and Den DF), and probability values $(\mathrm{Pr} > F)$. Each of these four multivariate statistics tests the hypothesis that the class means are equal in the population. See Chapter 2, "Introduction to Regression Procedures," for more information.

If you specify METHOD=NORMAL, the following three statistics are displayed:

- Covariance Matrix Information, including Covariance Matrix Rank and Natural Log of Determinant of the Covariance Matrix for each group (POOL=TEST, POOL=NO) and for the pooled within-group (POOL=TEST, POOL=YES)
- Optionally, Test of Homogeneity of Within Covariance Matrices (the results of a chi-square test of homogeneity of the within-group covariance matrices) (Morrison 1976; Kendall, Stuart, and Ord 1983; Anderson 1984)
- Pairwise Generalized Squared Distances Between Groups

If the CANONICAL option is specified, the displayed output contains these statistics:

- Canonical Correlations
- Adjusted Canonical Correlations (Lawley 1959). These are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations may not be computable and are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.
- Approximate Standard Error of the canonical correlations
- Squared Canonical Correlations
- Eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Each eigenvalue is equal to $\rho^2/(1 - \rho^2)$, where ρ^2 is the corresponding squared canonical correlation and can be interpreted as the ratio of between-class variation to within-class variation for the corresponding canonical variable. The table includes Eigenvalues, Differences between successive eigenvalues, the Proportion of the sum of the eigenvalues, and the Cumulative proportion.

- Likelihood Ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for all canonical correlations equals Wilks' lambda.
- Approximate F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Num DF (numerator degrees of freedom), Den DF (denominator degrees of freedom), and $\mathrm{Pr} > F$, the probability level associated with the F statistic

The following statistic concerns the classification criterion:

- the Linear Discriminant Function, but only if you specify METHOD=NORMAL and the pooled covariance matrix is used to calculate the (generalized) squared distances

When the input DATA= data set is an ordinary SAS data set, the displayed output includes the following:

- Optionally, the Resubstitution Results including Obs, the observation number (if an ID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation, the group into which the developed criterion would classify it, and the Posterior Probability of its Membership in each group
- Resubstitution Summary, a summary of the performance of the classification criterion based on resubstitution classification results
- Error Count Estimate of the resubstitution classification results
- Optionally, Posterior Probability Error Rate Estimates of the resubstitution classification results

If you specify the CROSSVALIDATE option, the displayed output contains these statistics:

- Optionally, the Cross-validation Results including Obs, the observation number (if an ID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation, the group into which the developed criterion would classify it, and the Posterior Probability of its Membership in each group
- Cross-validation Summary, a summary of the performance of the classification criterion based on cross validation classification results
- Error Count Estimate of the cross validation classification results
- Optionally, Posterior Probability Error Rate Estimates of the cross validation classification results

If you specify the TESTDATA= option, the displayed output contains these statistics:

- Optionally, the Classification Results including Obs, the observation number (if a TESTID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation (if a TESTCLASS statement is included), the group into which the developed criterion would classify it, and the Posterior Probability of its Membership in each group
- Classification Summary, a summary of the performance of the classification criterion
- Error Count Estimate of the test data classification results
- Optionally, Posterior Probability Error Rate Estimates of the test data classification results

ODS Table Names

PROC DISCRIM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 25.1. ODS Tables Produced by PROC DISCRIM

ODS Table Name	Description	PROC DISCRIM Option
ANOVA	Univariate statistics	ANOVA
AvePostCrossVal	Average posterior probabilities, cross validation	POSTERR & CROSSVALIDATE
AvePostResub	Average posterior probabilities, resubstitution	POSTERR
AvePostTestClass	Average posterior probabilities, test classification	POSTERR & TEST=
AveRSquare	Average R-Square	ANOVA
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
BStruc	Between canonical structure	CANONICAL
CanCorr	Canonical correlations	CANONICAL
CanonicalMeans	Class means on canonical variables	CANONICAL
ChiSq	Chi-square information	POOL=TEST
ClassifiedCrossVal	Number of observations and percent classified, cross validation	CROSSVALIDATE
ClassifiedResub	Number of observations and percent classified, resubstitution	default
ClassifiedTestClass	Number of observations and percent classified, test classification	TEST=
Counts	Number of observations, variables, classes, df	default
CovDF	DF for covariance matrices, not displayed	any *COV option

Table 25.1. (continued)

ODS Table Name	**Description**	**PROC DISCRIM Option**
Dist	Squared distances	MAHALANOBIS
DistFValues	F values based on squared distances	MAHALANOBIS
DistGeneralized	Generalized squared distances	default
DistProb	Probabilities for F values from squared distances	MAHALANOBIS
ErrorCrossVal	Error count estimates, cross validation	CROSSVALIDATE
ErrorResub	Error count estimates, resubstitution	default
ErrorTestClass	Error count estimates, test classification	TEST=
Levels	Class level information	default
LinearDiscFunc	Linear discriminant function	POOL=YES
LogDet	Log determinant of the covariance matrix	default
MultStat	MANOVA	MANOVA
PCoef	Pooled standard canonical coefficients	CANONICAL
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
PStruc	Pooled within canonical structure	CANONICAL
PostCrossVal	Posterior probabilities, cross validation	CROSSLIST or CROSSLISTERR
PostErrCrossVal	Posterior error estimates, cross validation	POSTERR & CROSSVALIDATE
PostErrResub	Posterior error estimates, resubstitution	POSTERR
PostErrTestClass	Posterior error estimates, test classification	POSTERR & TEST=
PostResub	Posterior probabilities, resubstitution	LIST or LISTERR
PostTestClass	Posterior probabilities, test classification	TESTLIST or TESTLISTERR
RCoef	Raw canonical coefficients	CANONICAL
SimpleStatistics	Simple statistics	SIMPLE
TCoef	Total-sample standard canonical coefficients	CANONICAL
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN

Table 25.1. (continued)

ODS Table Name	Description	PROC DISCRIM Option
TStruc	Total canonical structure	CANONICAL
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Examples

The iris data published by Fisher (1936) are widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species, *Iris setosa, I. versicolor, and I. virginica*. The iris data are used in Example 25.1 through Example 25.3.

Example 25.4 and Example 25.5 use remote-sensing data on crops. In this data set, the observations are grouped into five crops: clover, corn, cotton, soybeans, and sugar beets. Four measures called X1 through X4 make up the descriptive variables.

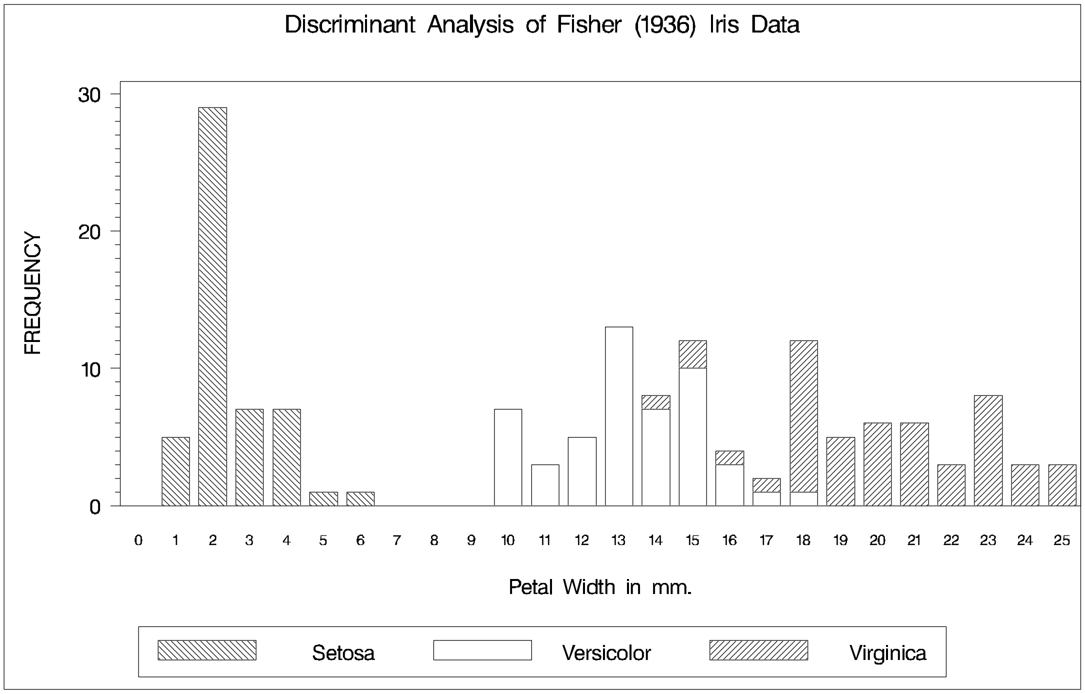
Example 25.1. Univariate Density Estimates and Posterior Probabilities

In this example, several discriminant analyses are run with a single quantitative variable, petal width, so that density estimates and posterior probabilities can be plotted easily. The example produces Output 25.1.1 through Output 25.1.5. The GCHART procedure is used to display the sample distribution of petal width in the three species. Note the overlap between species *I. versicolor* and *I. virginica* that the bar chart shows. These statements produce Output 25.1.1:

```
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
run;

data iris;
   title ’Discriminant Analysis of Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   symbol = put(Species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
```

```
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;

pattern1 c=red    /*v=l1   */;
pattern2 c=yellow /*v=empty*/;
pattern3 c=blue   /*v=r1   */;
axis1 label=(angle=90);
axis2 value=(height=.6);
legend1 frame label=none;

proc gchart data=iris;
   vbar PetalWidth / subgroup=Species midpoints=0 to 25
        raxis=axis1 maxis=axis2 legend=legend1 cframe=ligr;
run;
```

Output 25.1.1. Sample Distribution of Petal Width in Three Species



In order to plot the density estimates and posterior probabilities, a data set called plotdata is created containing equally spaced values from -5 to 30, covering the range of petal width with a little to spare on each end. The plotdata data set is used with the TESTDATA= option in PROC DISCRIM.

```
data plotdata;
   do PetalWidth=-5 to 30 by .5;
      output;
   end;
run;
```

The same plots are produced after each discriminant analysis, so a macro can be used to reduce the amount of typing required. The macro PLOT uses two data sets. The data set plotd, containing density estimates, is created by the TESTOUTD= option in PROC DISCRIM. The data set plotp, containing posterior probabilities, is created by the TESTOUT= option. For each data set, the macro PLOT removes uninteresting values (near zero) and does an overlay plot showing all three species on a single plot. The following statements create the macro PLOT

```
%macro plot;
   data plotd;
      set plotd;
      if setosa<.002 then setosa=.;
      if versicolor<.002 then versicolor=.;
      if virginica <.002 then virginica=.;
      label PetalWidth='Petal Width in mm.';
   run;

   symbol1 i=join v=none c=red    l=1 /*l=21*/;
   symbol2 i=join v=none c=yellow l=1 /*l= 1*/;
   symbol3 i=join v=none c=blue   l=1 /*l= 2*/;
   legend1 label=none frame;
   axis1 label=(angle=90 'Density') order=(0 to .6 by .1);

   proc gplot data=plotd;
      plot setosa*PetalWidth
           versicolor*PetalWidth
           virginica*PetalWidth
           / overlay vaxis=axis1 legend=legend1 frame
             cframe=ligr;
      title3 'Plot of Estimated Densities';
   run;

   data plotp;
      set plotp;
      if setosa<.01 then setosa=.;
      if versicolor<.01 then versicolor=.;
      if virginica<.01 then virginica=.;
      label PetalWidth='Petal Width in mm.';
   run;

   axis1 label=(angle=90 'Posterior Probability')
         order=(0 to 1 by .2);

   proc gplot data=plotp;
      plot setosa*PetalWidth
           versicolor*PetalWidth
           virginica*PetalWidth
           / overlay vaxis=axis1 legend=legend1 frame
             cframe=ligr;
      title3 'Plot of Posterior Probabilities';
   run;
%mend;
```

The first analysis uses normal-theory methods (METHOD=NORMAL) assuming equal variances (POOL=YES) in the three classes. The NOCLASSIFY option suppresses the resubstitution classification results of the input data set observations. The CROSSLISTERR option lists the observations that are misclassified under cross validation and displays cross validation error-rate estimates. The following statements produce Output 25.1.2:

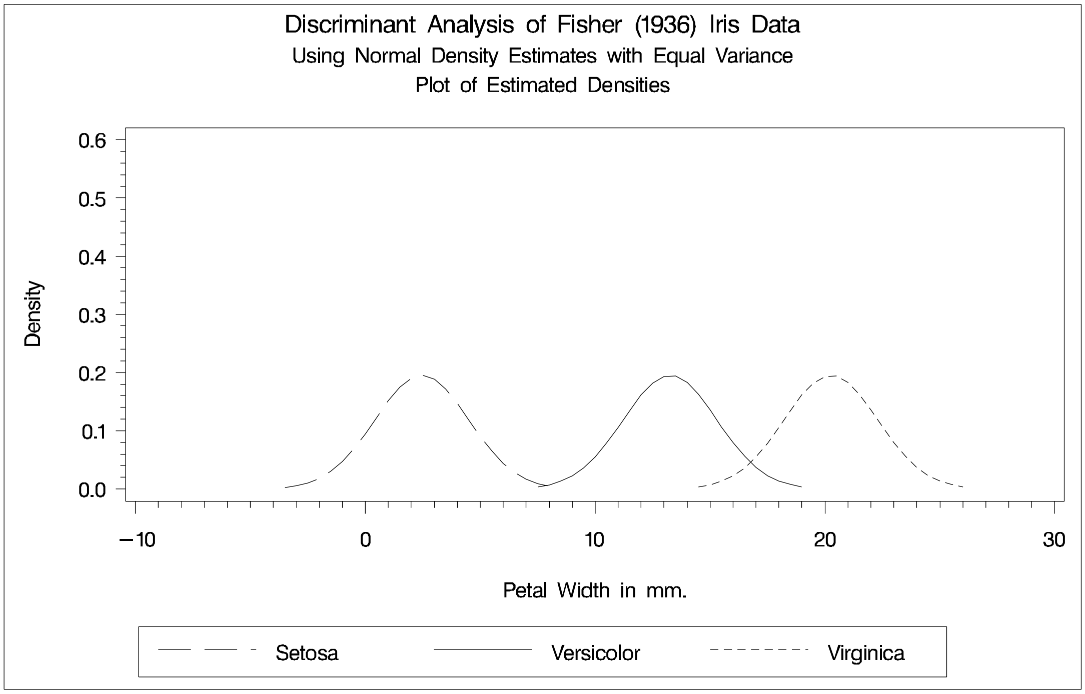
```
   proc discrim data=iris method=normal pool=yes
                testdata=plotdata testout=plotp testoutd=plotd
                short noclassify crosslisterr;
      class Species;
      var PetalWidth;
      title2 'Using Normal Density Estimates with Equal Variance';
   run;
   %plot
```

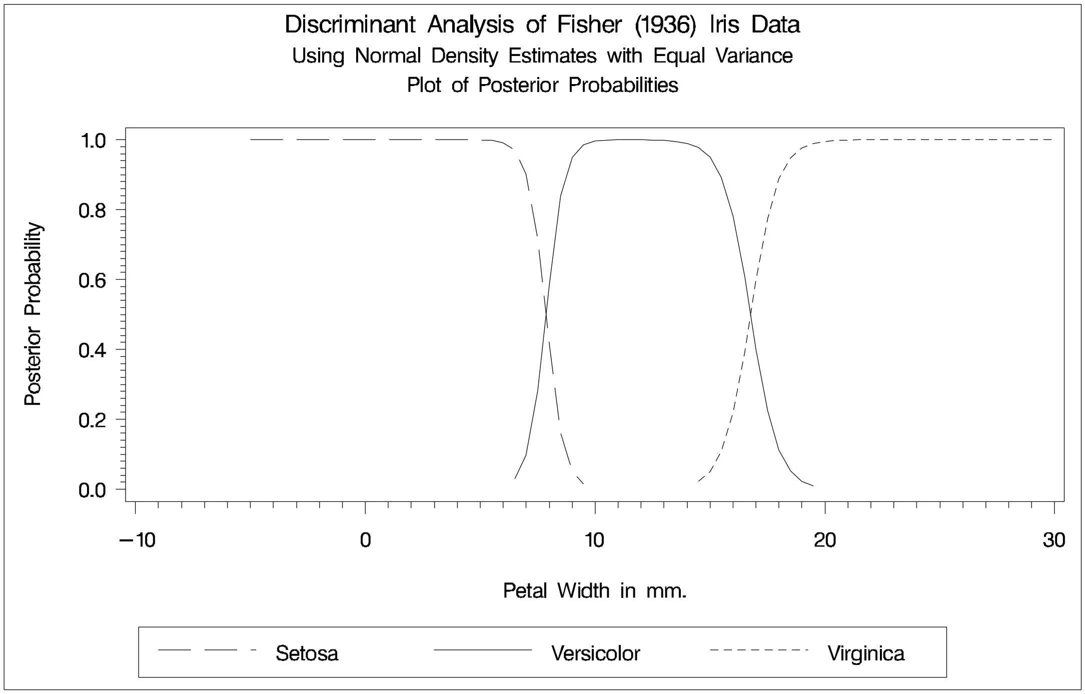
Output 25.1.2. Normal Density Estimates with Equal Variance

```
                 Discriminant Analysis of Fisher (1936) Iris Data
                Using Normal Density Estimates with Equal Variance

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          1          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                   Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
                Using Normal Density Estimates with Equal Variance

                             The DISCRIM Procedure
               Classification Results for Calibration Data: WORK.IRIS
           Cross-validation Results using Linear Discriminant Function

                       Generalized Squared Distance Function

                         2          _          -1     _
                        D (X) = (X-X    )' COV    (X-X    )
                         j          (X)j      (X)     (X)j

                Posterior Probability of Membership in Each Species

                                    2                    2
                   Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                    j        k           k


                  Posterior Probability of Membership in Species

          From          Classified
   Obs    Species       into Species        Setosa    Versicolor     Virginica

     5    Virginica     Versicolor *        0.0000        0.9610        0.0390
     9    Versicolor    Virginica  *        0.0000        0.0952        0.9048
    57    Virginica     Versicolor *        0.0000        0.9940        0.0060
    78    Virginica     Versicolor *        0.0000        0.8009        0.1991
    91    Virginica     Versicolor *        0.0000        0.9610        0.0390
   148    Versicolor    Virginica  *        0.0000        0.3828        0.6172

                         * Misclassified observation
                Discriminant Analysis of Fisher (1936) Iris Data
                Using Normal Density Estimates with Equal Variance

                             The DISCRIM Procedure
                 Classification Summary for Calibration Data: WORK.IRIS
              Cross-validation Summary using Linear Discriminant Function

                       Generalized Squared Distance Function

                          2         _          -1    _
                         D (X) = (X-X    )' COV    (X-X    )
                          j          (X)j      (X)     (X)j

                 Posterior Probability of Membership in Each Species

                                2                    2
               Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                j         k          k


            Number of Observations and Percent Classified into Species

       From
       Species           Setosa      Versicolor      Virginica         Total

       Setosa                50               0              0            50
                         100.00            0.00           0.00        100.00

       Versicolor             0              48              2            50
                           0.00           96.00           4.00        100.00

       Virginica              0               4             46            50
                           0.00            8.00          92.00        100.00

       Total                 50              52             48           150
                          33.33           34.67          32.00        100.00

       Priors           0.33333         0.33333        0.33333


                         Error Count Estimates for Species

                            Setosa    Versicolor    Virginica       Total

         Rate               0.0000        0.0400       0.0800      0.0400
         Priors             0.3333        0.3333       0.3333
```

```
          Discriminant Analysis of Fisher (1936) Iris Data
         Using Normal Density Estimates with Equal Variance

                       The DISCRIM Procedure
        Classification Summary for Test Data: WORK.PLOTDATA
     Classification Summary using Linear Discriminant Function

               Generalized Squared Distance Function

                     2         _       -1   _
                    D (X) = (X-X )' COV  (X-X )
                     j          j            j

        Posterior Probability of Membership in Each Species

                              2                    2
           Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                              j         k          k


    Number of Observations and Percent Classified into Species

                 Setosa      Versicolor      Virginica        Total

Total                26              18             27           71
                  36.62           25.35          38.03       100.00

Priors          0.33333         0.33333        0.33333
```





The next analysis uses normal-theory methods assuming unequal variances (POOL=NO) in the three classes. The following statements produce Output 25.1.3:

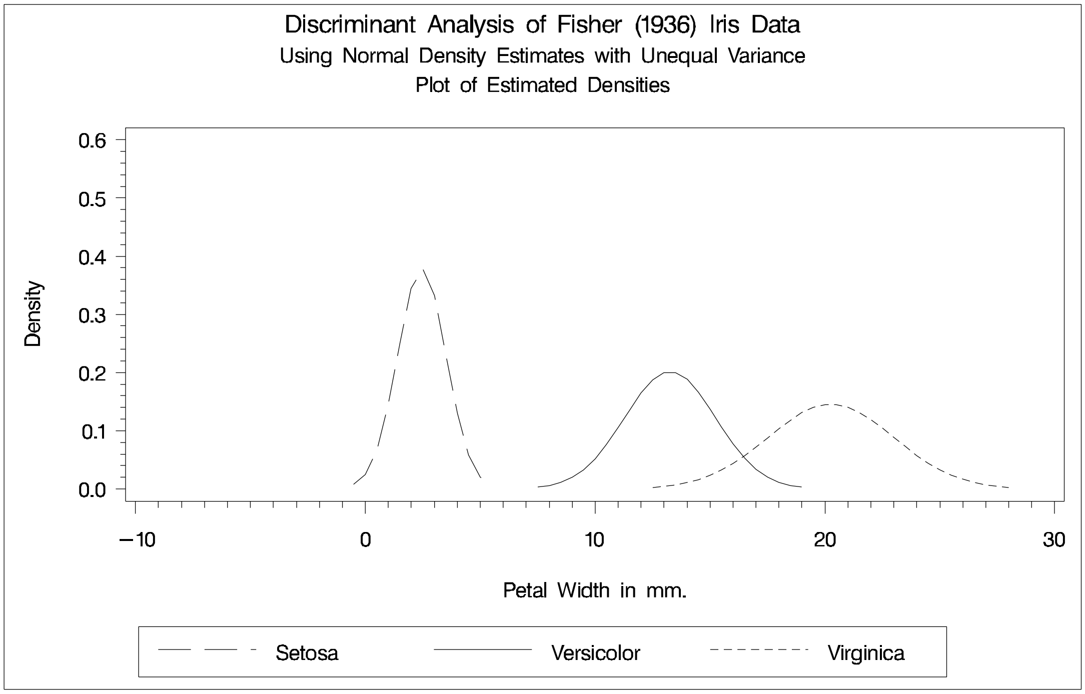
```
proc discrim data=iris method=normal pool=no
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
   title2 'Using Normal Density Estimates with Unequal Variance';
run;
%plot
```

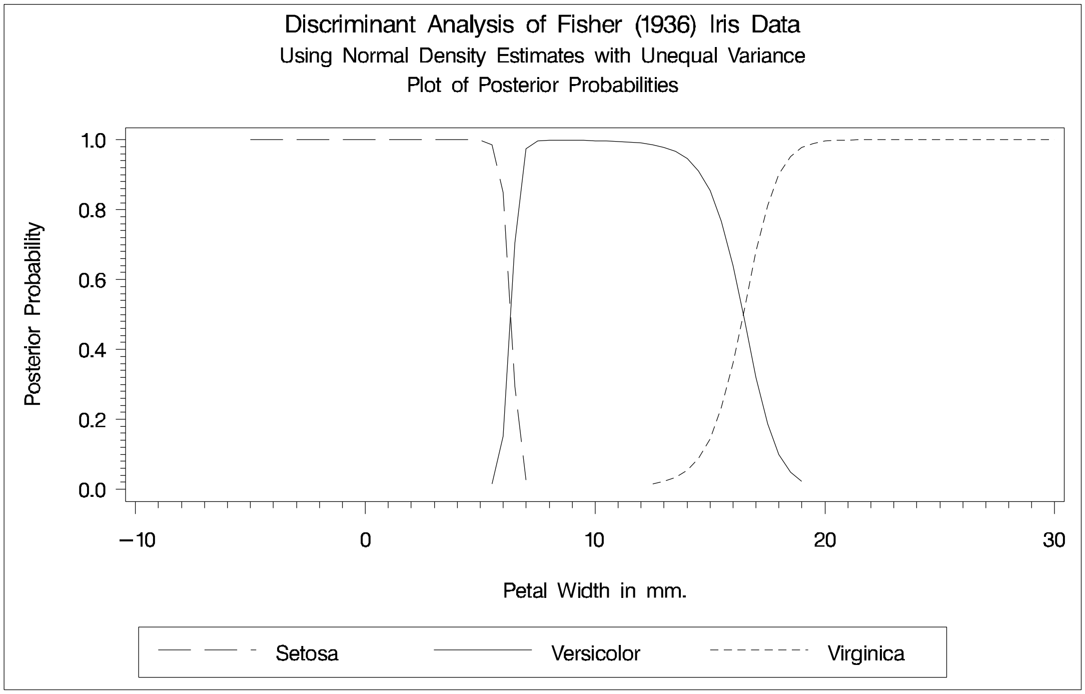
Output 25.1.3. Normal Density Estimates with Unequal Variance

```
                Discriminant Analysis of Fisher (1936) Iris Data
              Using Normal Density Estimates with Unequal Variance

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          1          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                 Prior
Species       Name            Frequency       Weight    Proportion    Probability

Setosa        Setosa                 50      50.0000      0.333333       0.333333
Versicolor    Versicolor             50      50.0000      0.333333       0.333333
Virginica     Virginica              50      50.0000      0.333333       0.333333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
                Using Normal Density Estimates with Unequal Variance

                              The DISCRIM Procedure
               Classification Results for Calibration Data: WORK.IRIS
             Cross-validation Results using Quadratic Discriminant Function

                        Generalized Squared Distance Function

                  2          _         -1     _
                 D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
                  j          (X)j      (X)j    (X)j           (X)j

                 Posterior Probability of Membership in Each Species

                                    2                    2
                   Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                    j        k           k


                  Posterior Probability of Membership in Species

       From          Classified
Obs    Species       into Species        Setosa    Versicolor     Virginica

  5    Virginica     Versicolor *        0.0000        0.8740        0.1260
  9    Versicolor    Virginica  *        0.0000        0.0686        0.9314
 42    Setosa        Versicolor *        0.4923        0.5073        0.0004
 57    Virginica     Versicolor *        0.0000        0.9602        0.0398
 78    Virginica     Versicolor *        0.0000        0.6558        0.3442
 91    Virginica     Versicolor *        0.0000        0.8740        0.1260
148    Versicolor    Virginica  *        0.0000        0.2871        0.7129

                          * Misclassified observation
                 Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Unequal Variance

                              The DISCRIM Procedure
               Classification Summary for Calibration Data: WORK.IRIS
            Cross-validation Summary using Quadratic Discriminant Function

                        Generalized Squared Distance Function

                  2         _           -1     _
                 D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
                  j          (X)j      (X)j    (X)j           (X)j

               Posterior Probability of Membership in Each Species

                                    2                    2
                   Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                    j        k            k


           Number of Observations and Percent Classified into Species

     From
     Species          Setosa      Versicolor      Virginica        Total

     Setosa               49               1              0           50
                          98.00            2.00           0.00       100.00

     Versicolor            0              48              2           50
                        0.00           96.00           4.00       100.00

     Virginica             0               4             46           50
                        0.00            8.00          92.00       100.00

     Total                49              53             48          150
                          32.67           35.33          32.00       100.00

     Priors          0.33333         0.33333        0.33333


                          Error Count Estimates for Species

                             Setosa    Versicolor    Virginica       Total

          Rate               0.0200        0.0400       0.0800      0.0467
          Priors             0.3333        0.3333       0.3333
```

```
               Discriminant Analysis of Fisher (1936) Iris Data
             Using Normal Density Estimates with Unequal Variance

                            The DISCRIM Procedure
             Classification Summary for Test Data: WORK.PLOTDATA
         Classification Summary using Quadratic Discriminant Function

                    Generalized Squared Distance Function

                     2         _         -1   _
                    D (X) = (X-X )' COV  (X-X ) + ln |COV |
                     j          j      j     j           j

              Posterior Probability of Membership in Each Species

                                  2                    2
                 Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                   j        k            k


         Number of Observations and Percent Classified into Species

                     Setosa      Versicolor      Virginica        Total

     Total               23              20             28           71
                      32.39           28.17          39.44       100.00

     Priors         0.33333         0.33333        0.33333
```





Two more analyses are run with nonparametric methods (METHOD=NPAR), specifically kernel density estimates with normal kernels (KERNEL=NORMAL). The first of these uses equal bandwidths (smoothing parameters) (POOL=YES) in each class. The use of equal bandwidths does not constrain the density estimates to be of equal variance. The value of the radius parameter that, assuming normality, minimizes an approximate mean integrated square error is 0.48 (see the "Nonparametric Methods" section on page 1158). Choosing $r = 0.4$ gives a more detailed look at the irregularities in the data. The following statements produce Output 25.1.4:

```
proc discrim data=iris method=npar kernel=normal
                r=.4 pool=yes
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
   title2 'Using Kernel Density Estimates with Equal
           Bandwidth';
run;
%plot
```

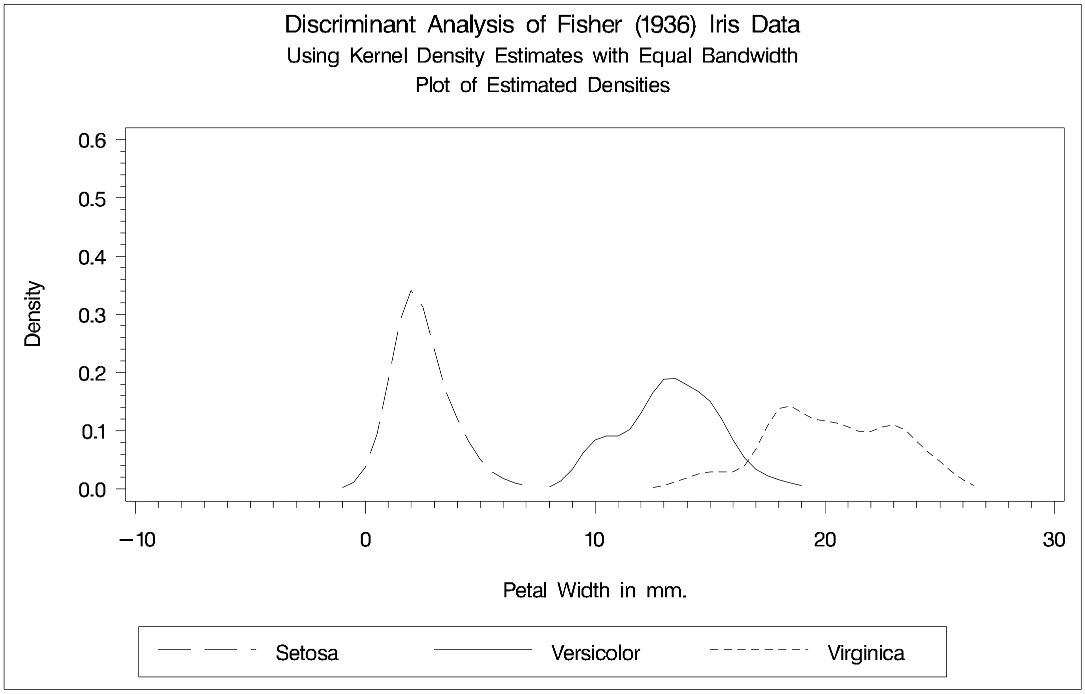
Output 25.1.4. Kernel Density Estimates with Equal Bandwidth

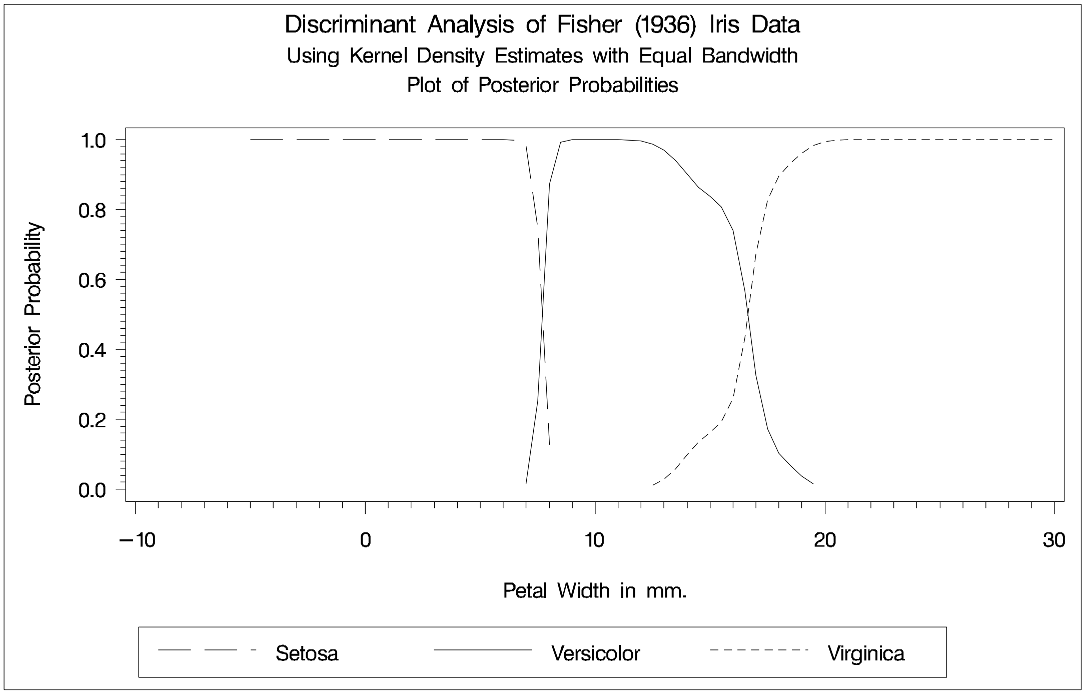
```
                Discriminant Analysis of Fisher (1936) Iris Data
              Using Kernel Density Estimates with Equal Bandwidth

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          1          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```

```
                Discriminant Analysis of Fisher (1936) Iris Data
              Using Kernel Density Estimates with Equal Bandwidth

                             The DISCRIM Procedure
             Classification Results for Calibration Data: WORK.IRIS
              Cross-validation Results using Normal Kernel Density

                           Squared Distance Function

                             2                  -1
                            D (X,Y) = (X-Y)' COV  (X-Y)

              Posterior Probability of Membership in Each Species

                               -1               2           2
                    F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                               j   i                 ji

                  Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                                 j            k       k


                Posterior Probability of Membership in Species

         From          Classified
  Obs    Species       into Species        Setosa    Versicolor     Virginica

    5    Virginica     Versicolor *        0.0000        0.8827        0.1173
    9    Versicolor    Virginica  *        0.0000        0.0438        0.9562
   57    Virginica     Versicolor *        0.0000        0.9472        0.0528
   78    Virginica     Versicolor *        0.0000        0.8061        0.1939
   91    Virginica     Versicolor *        0.0000        0.8827        0.1173
  148    Versicolor    Virginica  *        0.0000        0.2586        0.7414

                         * Misclassified observation
```

```
          Discriminant Analysis of Fisher (1936) Iris Data
        Using Kernel Density Estimates with Equal Bandwidth

                       The DISCRIM Procedure
       Classification Summary for Calibration Data: WORK.IRIS
        Cross-validation Summary using Normal Kernel Density

                     Squared Distance Function

                     2                  -1
                    D (X,Y) = (X-Y)’ COV  (X-Y)

        Posterior Probability of Membership in Each Species

                       -1               2           2
             F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                       j   i                 ji

            Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                           j           k       k


    Number of Observations and Percent Classified into Species

From
Species          Setosa      Versicolor      Virginica        Total

Setosa               50               0              0           50
                 100.00            0.00           0.00       100.00

Versicolor            0              48              2           50
                   0.00           96.00           4.00       100.00

Virginica             0               4             46           50
                   0.00            8.00          92.00       100.00

Total                50              52             48          150
                  33.33           34.67          32.00       100.00

Priors          0.33333         0.33333        0.33333


                 Error Count Estimates for Species

                    Setosa    Versicolor    Virginica       Total

  Rate              0.0000        0.0400       0.0800      0.0400
  Priors            0.3333        0.3333       0.3333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
               Using Kernel Density Estimates with Equal Bandwidth

                              The DISCRIM Procedure
               Classification Summary for Test Data: WORK.PLOTDATA
                Classification Summary using Normal Kernel Density

                            Squared Distance Function

                         2                  -1
                        D (X,Y) = (X-Y)' COV  (X-Y)

               Posterior Probability of Membership in Each Species

                              -1              2          2
                    F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                              j   i                ji

                   Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                                  j            k       k


           Number of Observations and Percent Classified into Species

                        Setosa      Versicolor      Virginica        Total

       Total                26              18             27           71
                         36.62           25.35          38.03       100.00

       Priors          0.33333         0.33333        0.33333
```





Another nonparametric analysis is run with unequal bandwidths (POOL=NO). These statements produce Output 25.1.5:

```
proc discrim data=iris method=npar kernel=normal
                r=.4 pool=no
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
   title2 'Using Kernel Density Estimates with Unequal
           Bandwidth';
run;
%plot
```

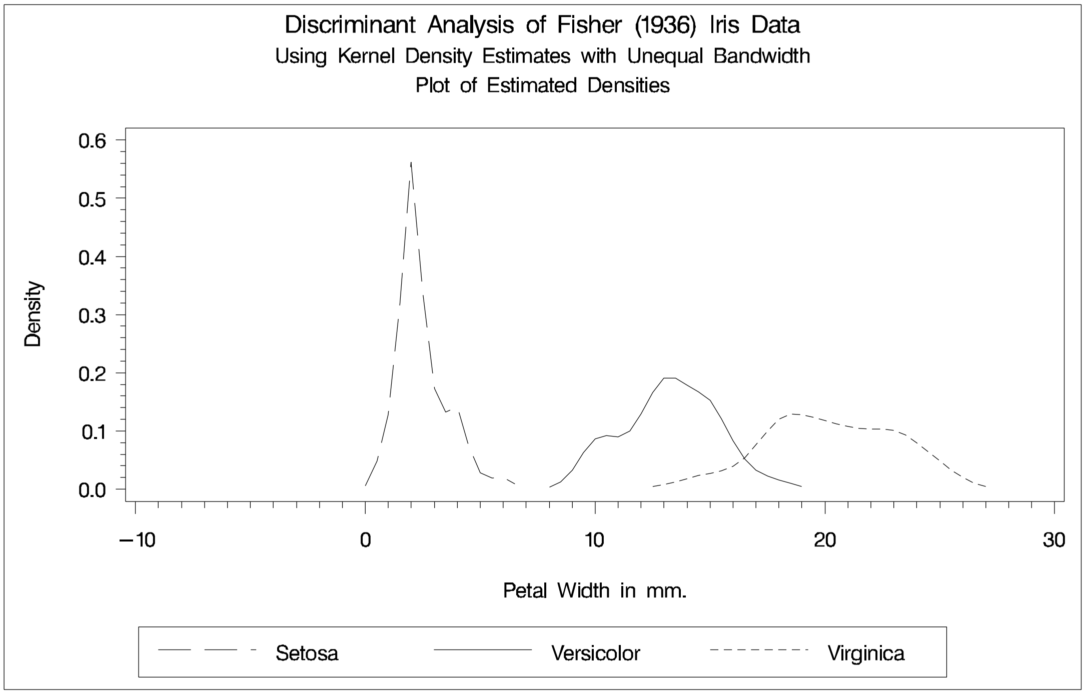
Output 25.1.5. Kernel Density Estimates with Unequal Bandwidth

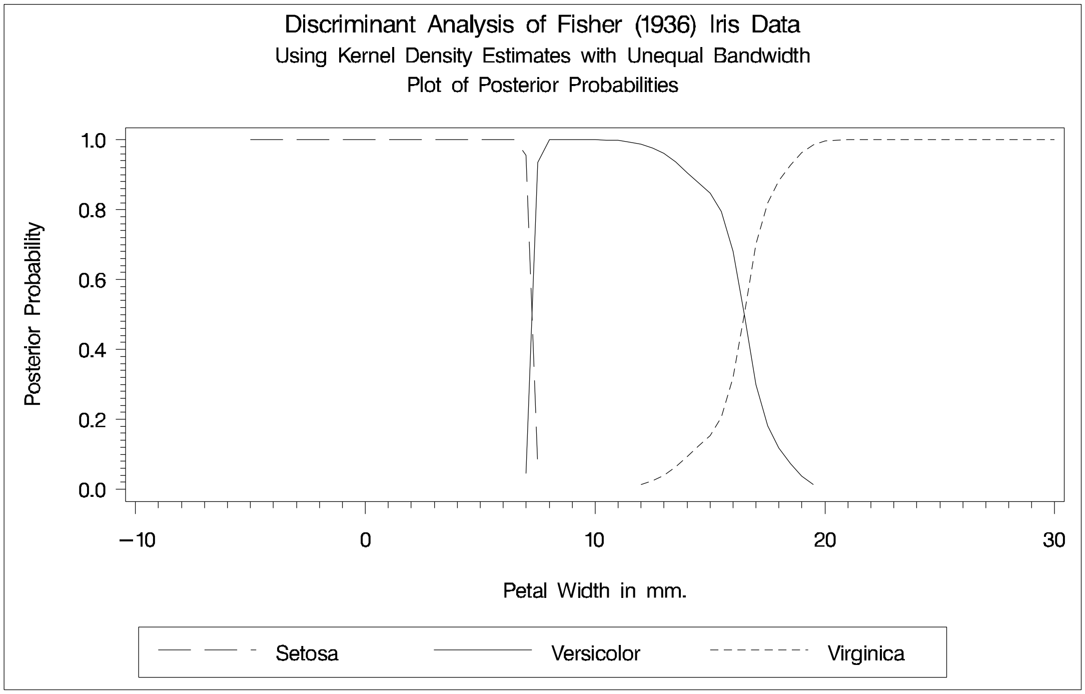
```
                Discriminant Analysis of Fisher (1936) Iris Data
             Using Kernel Density Estimates with Unequal Bandwidth

                            The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          1          DF Within Classes      147
            Classes            3          DF Between Classes       2


                           Class Level Information

              Variable                                                 Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```

```
                Discriminant Analysis of Fisher (1936) Iris Data
             Using Kernel Density Estimates with Unequal Bandwidth

                            The DISCRIM Procedure
             Classification Results for Calibration Data: WORK.IRIS
              Cross-validation Results using Normal Kernel Density

                          Squared Distance Function

                           2                  -1
                          D (X,Y) = (X-Y)' COV  (X-Y)
                                              j

              Posterior Probability of Membership in Each Species

                              -1                2           2
                    F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                              j   i                 ji

                   Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                                  j           k       k


                Posterior Probability of Membership in Species

         From          Classified
  Obs    Species       into Species        Setosa    Versicolor     Virginica

    5    Virginica     Versicolor *        0.0000        0.8805        0.1195
    9    Versicolor    Virginica  *        0.0000        0.0466        0.9534
   57    Virginica     Versicolor *        0.0000        0.9394        0.0606
   78    Virginica     Versicolor *        0.0000        0.7193        0.2807
   91    Virginica     Versicolor *        0.0000        0.8805        0.1195
  148    Versicolor    Virginica  *        0.0000        0.2275        0.7725

                         * Misclassified observation
```

```
          Discriminant Analysis of Fisher (1936) Iris Data
       Using Kernel Density Estimates with Unequal Bandwidth

                       The DISCRIM Procedure
       Classification Summary for Calibration Data: WORK.IRIS
        Cross-validation Summary using Normal Kernel Density

                     Squared Distance Function

                     2                  -1
                    D (X,Y) = (X-Y)’ COV  (X-Y)
                                        j

        Posterior Probability of Membership in Each Species

                       -1               2           2
             F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                       j   i                 ji

            Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                           j           k       k


    Number of Observations and Percent Classified into Species

From
Species          Setosa      Versicolor      Virginica        Total

Setosa               50               0              0           50
                 100.00            0.00           0.00       100.00

Versicolor            0              48              2           50
                   0.00           96.00           4.00       100.00

Virginica             0               4             46           50
                   0.00            8.00          92.00       100.00

Total                50              52             48          150
                  33.33           34.67          32.00       100.00

Priors          0.33333         0.33333        0.33333


                 Error Count Estimates for Species

                    Setosa    Versicolor    Virginica       Total

  Rate              0.0000        0.0400       0.0800      0.0400
  Priors            0.3333        0.3333       0.3333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
              Using Kernel Density Estimates with Unequal Bandwidth

                             The DISCRIM Procedure
               Classification Summary for Test Data: WORK.PLOTDATA
                Classification Summary using Normal Kernel Density

                           Squared Distance Function

                            2                   -1
                           D (X,Y) = (X-Y)' COV  (X-Y)
                                                j

               Posterior Probability of Membership in Each Species

                              -1                2           2
                    F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                              j   i                  ji

                   Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                                  j            k       k


           Number of Observations and Percent Classified into Species

                       Setosa      Versicolor      Virginica        Total

       Total               25              18             28           71
                        35.21           25.35          39.44       100.00

       Priors         0.33333         0.33333        0.33333
```



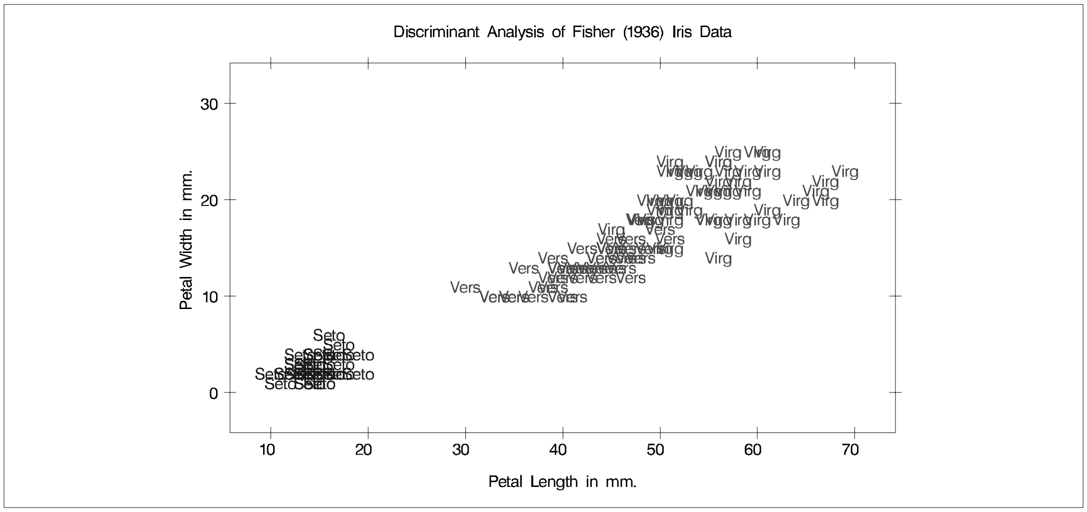

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth
Plot of Posterior Probabilities
Posterior Probability
1.0
0.8
0.6
0.4
0.2
0.0
−10
0
10
20
30
Petal Width in mm.
Setosa
Versicolor
Virginica

Example 25.2. Bivariate Density Estimates and Posterior Probabilities

In this example, four more discriminant analyses of iris data are run with two quantitative variables: petal width and petal length. The example produces Output 25.2.1 through Output 25.2.5. A scatter plot shows the joint sample distribution. See Appendix B, "Using the %PLOTIT Macro," for more information on the %PLOTIT macro.

```
%plotit(data=iris, plotvars=PetalWidth PetalLength,
        labelvar=_blank_, symvar=symbol, typevar=symbol,
        symsize=0.35, symlen=4, exttypes=symbol, ls=100);
```

Output 25.2.1. Joint Sample Distribution of Petal Width and Petal Length in Three Species



Another data set is created for plotting, containing a grid of points suitable for contour plots. The large number of points in the grid makes the following analyses very time-consuming. If you attempt to duplicate these examples, begin with a small number of points in the grid.

```
data plotdata;
   do PetalLength=-2 to 72 by 0.25;
      h + 1;    * Number of horizontal cells;
      do PetalWidth=-5 to 32 by 0.25;
         n + 1; * Total number of cells;
         output;
      end;
   end;
   * Make variables to contain H and V grid sizes;
   call symput('hnobs', compress(put(h    , best12.)));
   call symput('vnobs', compress(put(n / h, best12.)));
   drop n h;
run;
```

A macro CONTOUR is defined to make contour plots of density estimates and posterior probabilities. Classification results are also plotted on the same grid.

```
%macro contour;
   data contour(keep=PetalWidth PetalLength symbol density);
      set plotd(in=d) iris;
      if d then density = max(setosa,versicolor,virginica);
   run;

   title3 'Plot of Estimated Densities';
   %plotit(data=contour, plotvars=PetalWidth PetalLength,
           labelvar=_blank_, symvar=symbol, typevar=symbol,
           symlen=4, exttypes=symbol contour, ls=100,
           paint=density white black, rgbtypes=contour,
           hnobs=&hnobs, vnobs=&vnobs, excolors=white,
           rgbround=-16 1 1 1,  extend=close, options=noclip,
           types  =Setosa Versicolor Virginica  '',
           symtype=symbol symbol     symbol     contour,
           symsize=0.6    0.6        0.6        1,
           symfont=swiss  swiss      swiss      solid)

   data posterior(keep=PetalWidth PetalLength symbol
        prob _into_);
      set plotp(in=d) iris;
      if d then prob = max(setosa,versicolor,virginica);
   run;

   title3 'Plot of Posterior Probabilities '
          '(Black to White is Low to High Probability)';
   %plotit(data=posterior, plotvars=PetalWidth PetalLength,
           labelvar=_blank_, symvar=symbol, typevar=symbol,
           symlen=4, exttypes=symbol contour, ls=100,
           paint=prob black white 0.3 0.999, rgbtypes=contour,
           hnobs=&hnobs, vnobs=&vnobs,  excolors=white,
           rgbround=-16 1 1 1, extend=close, options=noclip,
           types  =Setosa Versicolor Virginica  '',
           symtype=symbol symbol     symbol     contour,
           symsize=0.6    0.6        0.6        1,
           symfont=swiss  swiss      swiss      solid)

   title3 'Plot of Classification Results';
   %plotit(data=posterior, plotvars=PetalWidth PetalLength,
           labelvar=_blank_, symvar=symbol, typevar=symbol,
           symlen=4, exttypes=symbol contour, ls=100,
           paint=_into_ CXCCCCCC CXDDDDDD white,
              rgbtypes=contour, hnobs=&hnobs, vnobs=&vnobs,
              excolors=white,
           extend=close, options=noclip,
           types  =Setosa Versicolor Virginica  '',
           symtype=symbol symbol     symbol     contour,
           symsize=0.6    0.6        0.6        1,
           symfont=swiss  swiss      swiss      solid)

%mend;
```

A normal-theory analysis (METHOD=NORMAL) assuming equal covariance matrices (POOL=YES) illustrates the linearity of the classification boundaries. These statements produce Output 25.2.2:

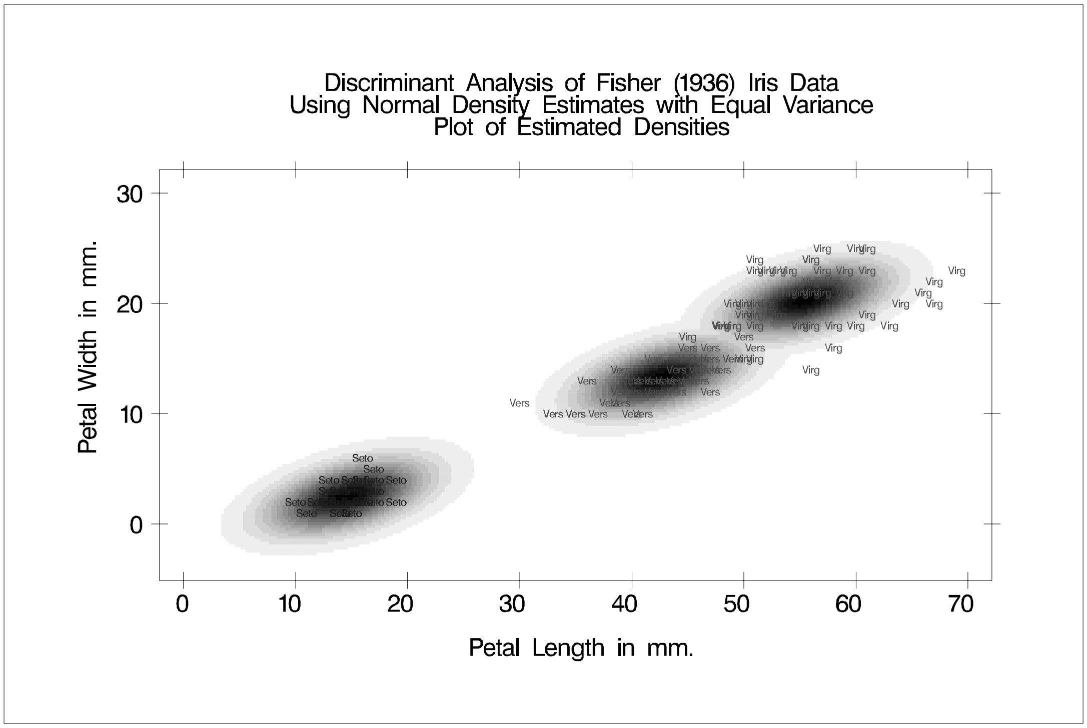
```
proc discrim data=iris method=normal pool=yes
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
   title2 ’Using Normal Density Estimates with Equal
           Variance’;
run;
%contour
```

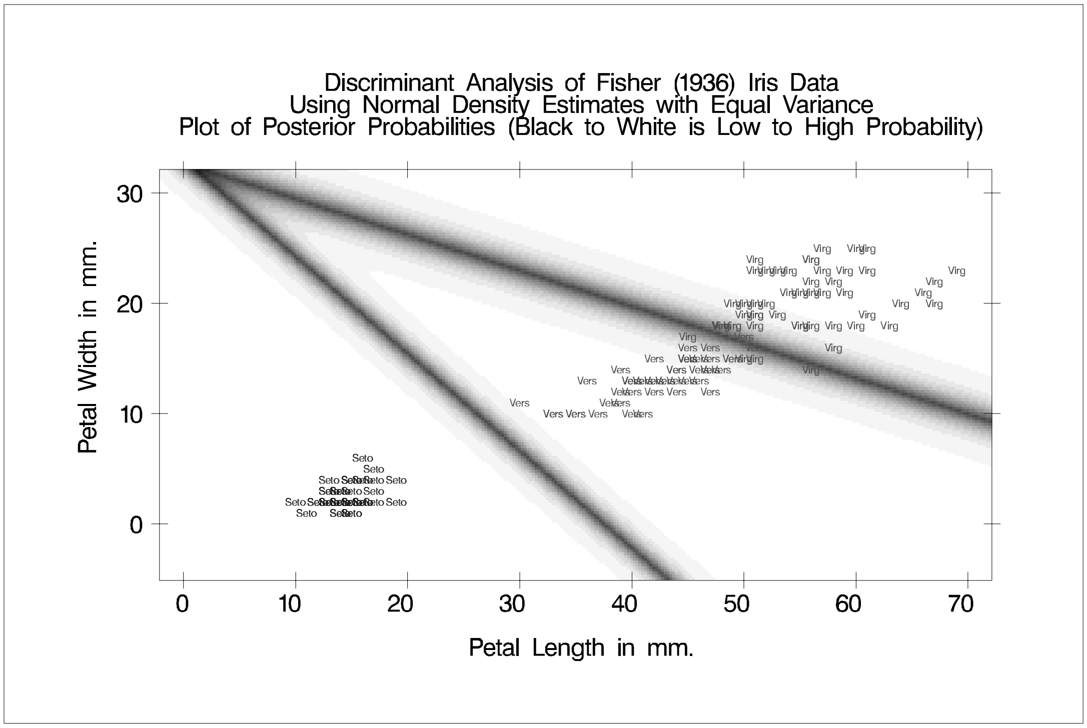
Output 25.2.2. Normal Density Estimates with Equal Variance

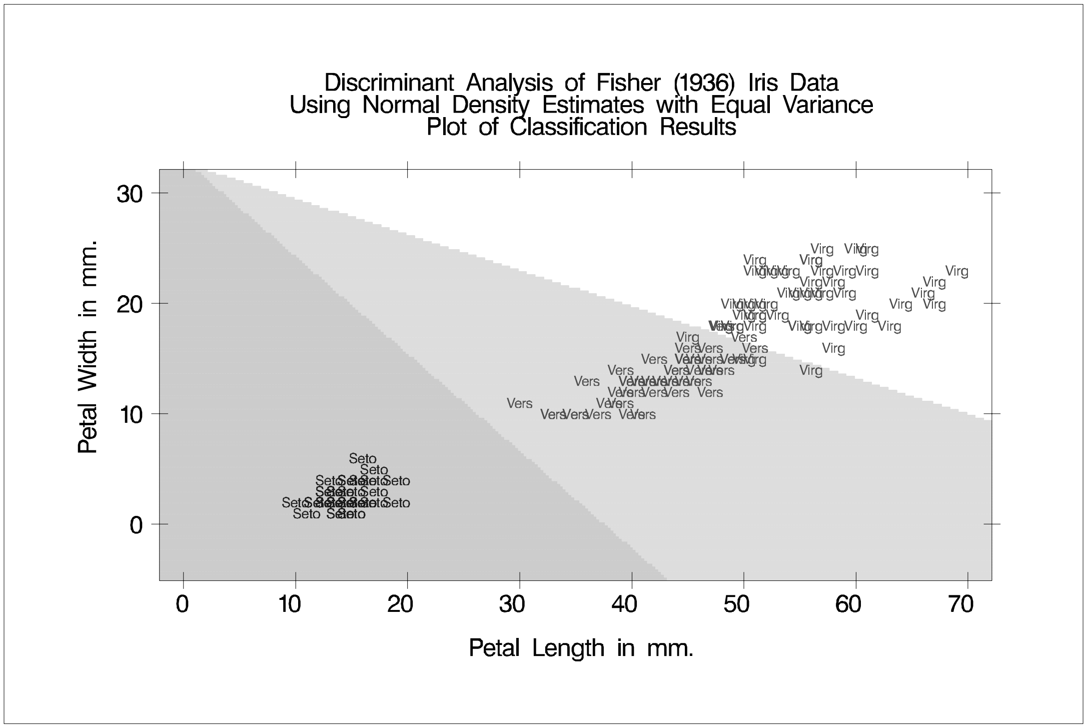
```
                Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Equal Variance

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          2          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                       Prior
Species       Name           Frequency        Weight    Proportion    Probability

Setosa        Setosa                50       50.0000      0.333333       0.333333
Versicolor    Versicolor            50       50.0000      0.333333       0.333333
Virginica     Virginica             50       50.0000      0.333333       0.333333
```

```
                Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Equal Variance

                            The DISCRIM Procedure
              Classification Results for Calibration Data: WORK.IRIS
           Cross-validation Results using Linear Discriminant Function

                      Generalized Squared Distance Function

                         2         _          -1    _
                        D (X) = (X-X    )' COV    (X-X    )
                         j          (X)j      (X)     (X)j

                 Posterior Probability of Membership in Each Species

                                      2                    2
                Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                      j        k           k


                   Posterior Probability of Membership in Species

         From          Classified
  Obs    Species       into Species        Setosa    Versicolor     Virginica

    5    Virginica     Versicolor *        0.0000        0.8453        0.1547
    9    Versicolor    Virginica  *        0.0000        0.2130        0.7870
   25    Virginica     Versicolor *        0.0000        0.8322        0.1678
   57    Virginica     Versicolor *        0.0000        0.8057        0.1943
   91    Virginica     Versicolor *        0.0000        0.8903        0.1097
  148    Versicolor    Virginica  *        0.0000        0.3118        0.6882

                           * Misclassified observation
                Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Equal Variance

                             The DISCRIM Procedure
             Classification Summary for Calibration Data: WORK.IRIS
          Cross-validation Summary using Linear Discriminant Function

                     Generalized Squared Distance Function

                       2         _          -1     _
                      D (X) = (X-X    )' COV    (X-X    )
                       j          (X)j      (X)     (X)j

              Posterior Probability of Membership in Each Species

                                    2                    2
                 Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                    j        k           k


          Number of Observations and Percent Classified into Species

      From
      Species          Setosa      Versicolor      Virginica        Total

      Setosa               50               0              0           50
                       100.00            0.00           0.00       100.00

      Versicolor            0              48              2           50
                         0.00           96.00           4.00       100.00

      Virginica             0               4             46           50
                         0.00            8.00          92.00       100.00

      Total                50              52             48          150
                        33.33           34.67          32.00       100.00

      Priors          0.33333         0.33333        0.33333


                       Error Count Estimates for Species

                          Setosa    Versicolor    Virginica       Total

        Rate              0.0000        0.0400       0.0800      0.0400
        Priors            0.3333        0.3333       0.3333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
                Using Normal Density Estimates with Equal Variance

                              The DISCRIM Procedure
               Classification Summary for Test Data: WORK.PLOTDATA
            Classification Summary using Linear Discriminant Function

                      Generalized Squared Distance Function

                            2          _       -1   _
                           D (X) = (X-X )' COV  (X-X )
                            j          j            j

               Posterior Probability of Membership in Each Species

                                     2                    2
                  Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                     j        k           k


           Number of Observations and Percent Classified into Species

                        Setosa      Versicolor      Virginica        Total

       Total             14507           16888          12858        44253
                         32.78           38.16          29.06       100.00

       Priors          0.33333         0.33333        0.33333
```




Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance
Plot of Posterior Probabilities (Black to White is Low to High Probability)
Petal Width in mm.
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg

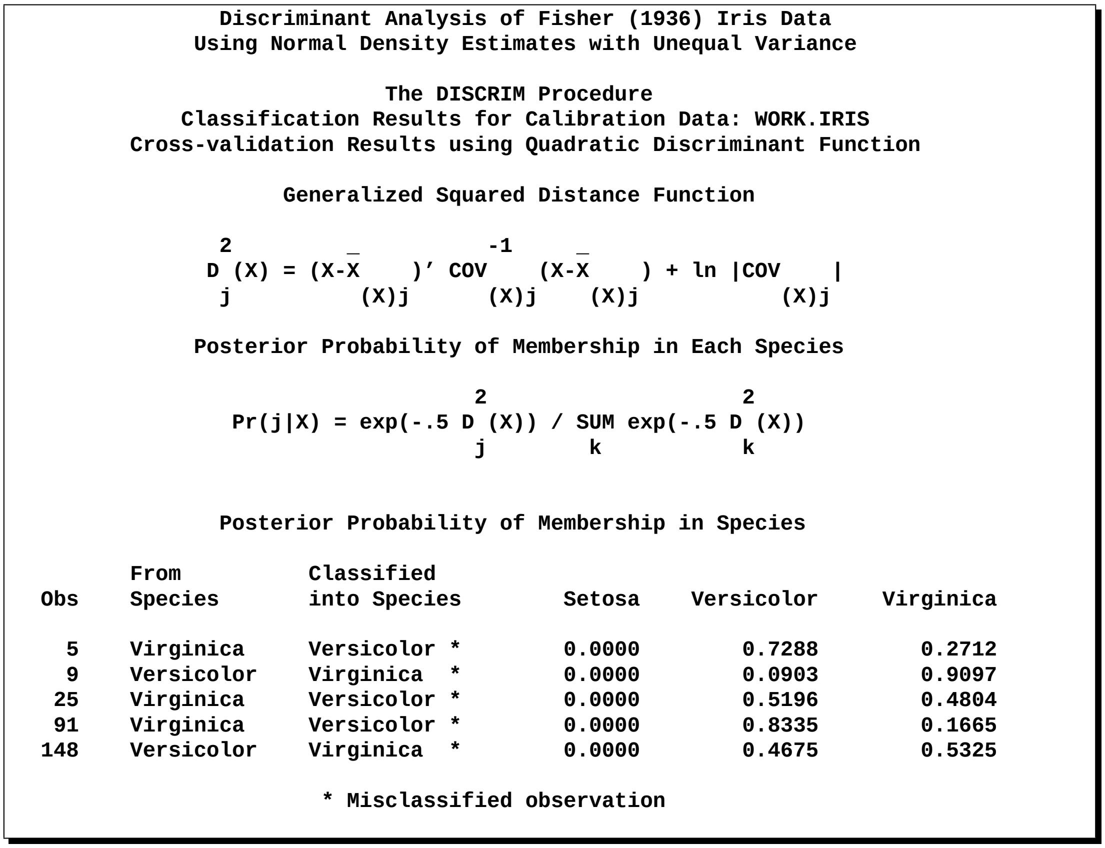

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance
Plot of Classification Results
Petal Width in mm.
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg

A normal-theory analysis assuming unequal covariance matrices (POOL=NO) illustrates quadratic classification boundaries. These statements produce Output 25.2.3:

```
proc discrim data=iris method=normal pool=no
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
   title2 'Using Normal Density Estimates with Unequal
           Variance';
run;
%contour
```

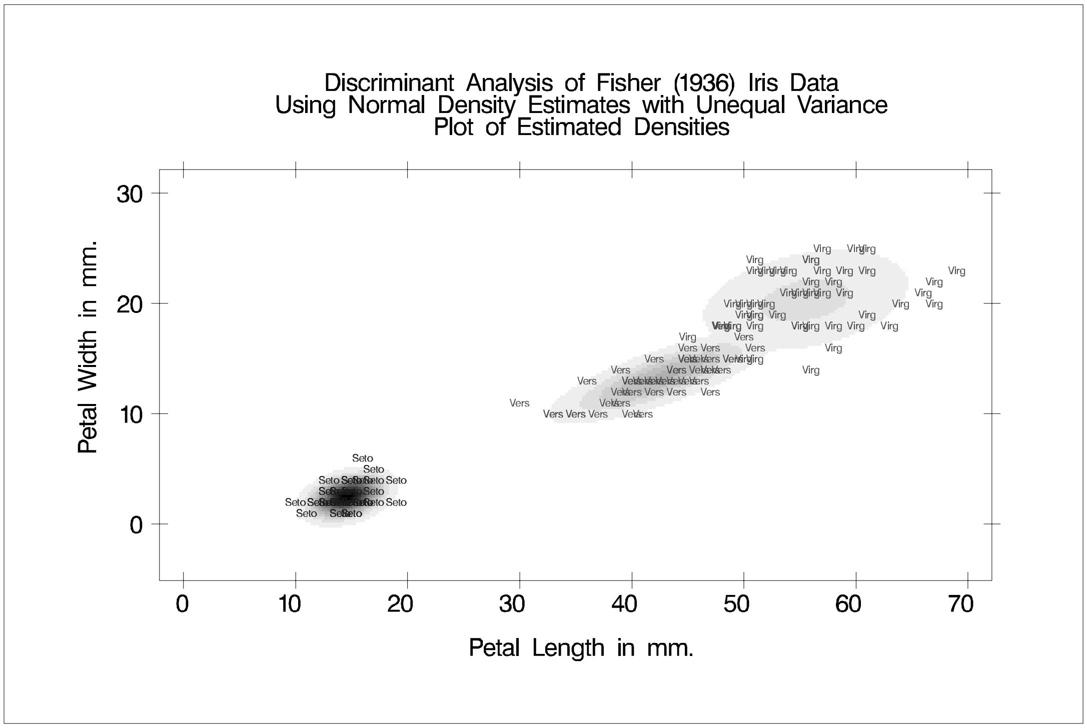
Output 25.2.3. Normal Density Estimates with Unequal Variance

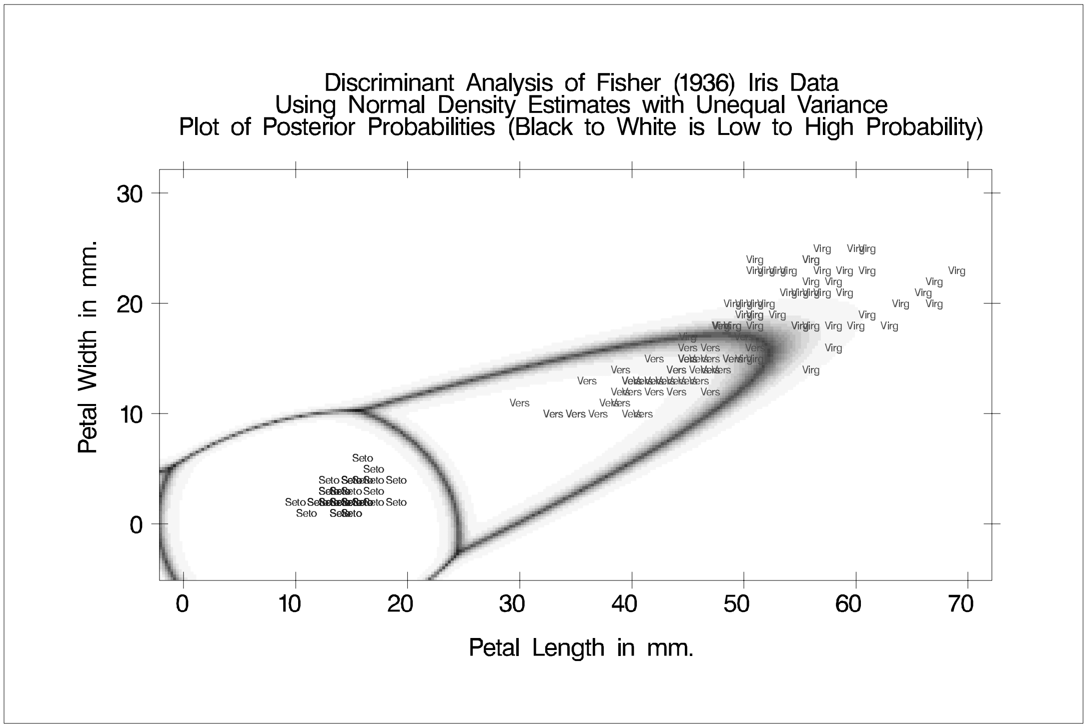
```
                Discriminant Analysis of Fisher (1936) Iris Data
              Using Normal Density Estimates with Unequal Variance

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          2          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                    Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```



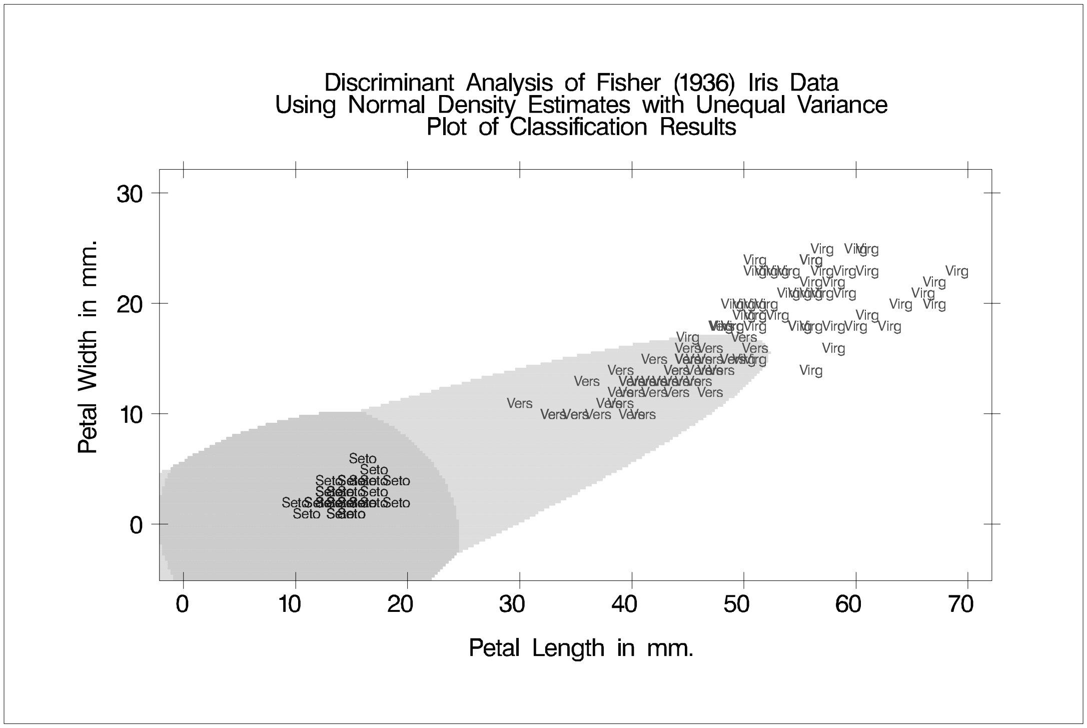
```
                Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Unequal Variance

                              The DISCRIM Procedure
              Classification Results for Calibration Data: WORK.IRIS
            Cross-validation Results using Quadratic Discriminant Function

                       Generalized Squared Distance Function

                 2         _         -1    _
                D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
                 j          (X)j      (X)j    (X)j           (X)j

               Posterior Probability of Membership in Each Species

                                   2                     2
                  Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                   j        k            k


                 Posterior Probability of Membership in Species

      From          Classified
Obs   Species       into Species       Setosa    Versicolor     Virginica

  5   Virginica     Versicolor *       0.0000        0.7288        0.2712
  9   Versicolor    Virginica  *       0.0000        0.0903        0.9097
 25   Virginica     Versicolor *       0.0000        0.5196        0.4804
 91   Virginica     Versicolor *       0.0000        0.8335        0.1665
148   Versicolor    Virginica  *       0.0000        0.4675        0.5325

                         * Misclassified observation
                 Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Unequal Variance

                              The DISCRIM Procedure
              Classification Summary for Calibration Data: WORK.IRIS
            Cross-validation Summary using Quadratic Discriminant Function

                      Generalized Squared Distance Function

                 2         _           -1    _
                D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
                 j          (X)j      (X)j    (X)j           (X)j

               Posterior Probability of Membership in Each Species

                                   2                    2
                  Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                   j        k           k


           Number of Observations and Percent Classified into Species

       From
       Species          Setosa      Versicolor      Virginica        Total

       Setosa               50               0              0           50
                        100.00            0.00           0.00       100.00

       Versicolor            0              48              2           50
                          0.00           96.00           4.00       100.00

       Virginica             0               3             47           50
                          0.00            6.00          94.00       100.00

       Total                50              51             49          150
                         33.33           34.00          32.67       100.00

       Priors          0.33333         0.33333        0.33333


                          Error Count Estimates for Species

                             Setosa    Versicolor    Virginica       Total

         Rate                0.0000        0.0400       0.0600      0.0333
         Priors              0.3333        0.3333       0.3333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
               Using Normal Density Estimates with Unequal Variance

                              The DISCRIM Procedure
               Classification Summary for Test Data: WORK.PLOTDATA
           Classification Summary using Quadratic Discriminant Function

                        Generalized Squared Distance Function

                         2          _         -1   _
                        D (X) = (X-X )' COV  (X-X ) + ln |COV |
                         j          j      j     j           j

               Posterior Probability of Membership in Each Species

                                     2                    2
                  Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                     j        k           k


           Number of Observations and Percent Classified into Species

                        Setosa      Versicolor      Virginica        Total

       Total              5461            5354          33438        44253
                         12.34           12.10          75.56       100.00

       Priors          0.33333         0.33333        0.33333
```





Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance
Plot of Posterior Probabilities (Black to White is Low to High Probability)
Petal Width in mm.
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg



Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance
Plot of Classification Results
Petal Width in mm.
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg

A nonparametric analysis (METHOD=NPAR) follows, using normal kernels (KERNEL=NORMAL) and equal bandwidths (POOL=YES) in each class. The value of the radius parameter r that, assuming normality, minimizes an approximate mean integrated square error is 0.50 (see the "Nonparametric Methods" section on page 1158). These statements produce Output 25.2.4:

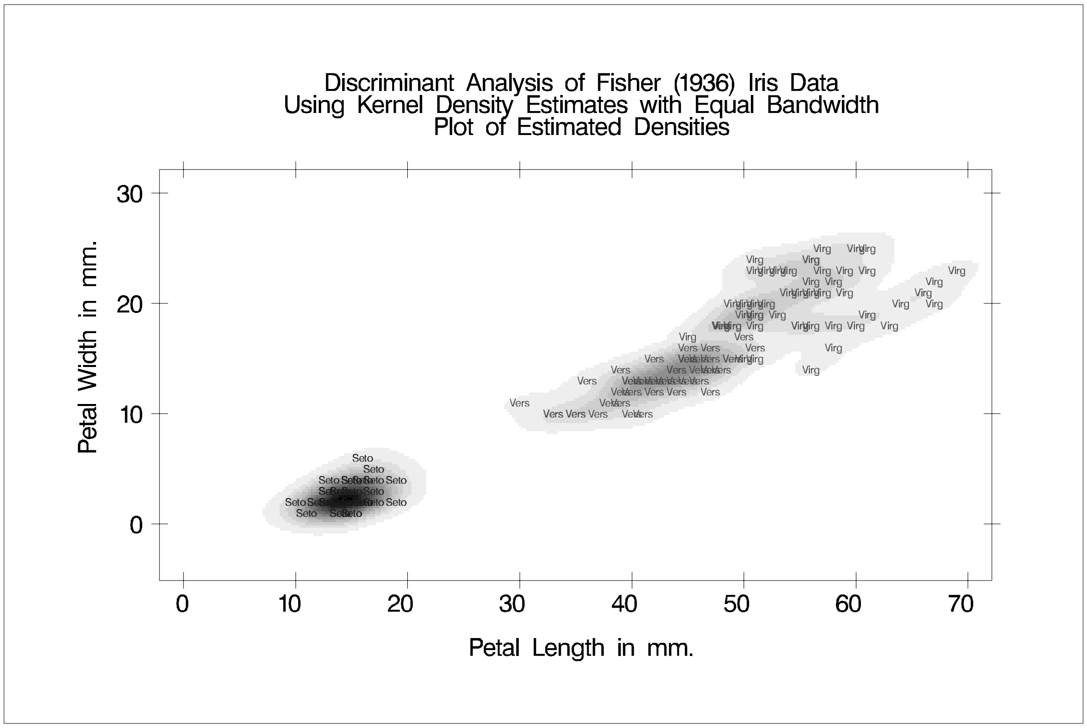
```
proc discrim data=iris method=npar kernel=normal
                r=.5 pool=yes
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
   title2 'Using Kernel Density Estimates with Equal
           Bandwidth';
run;
%contour
```

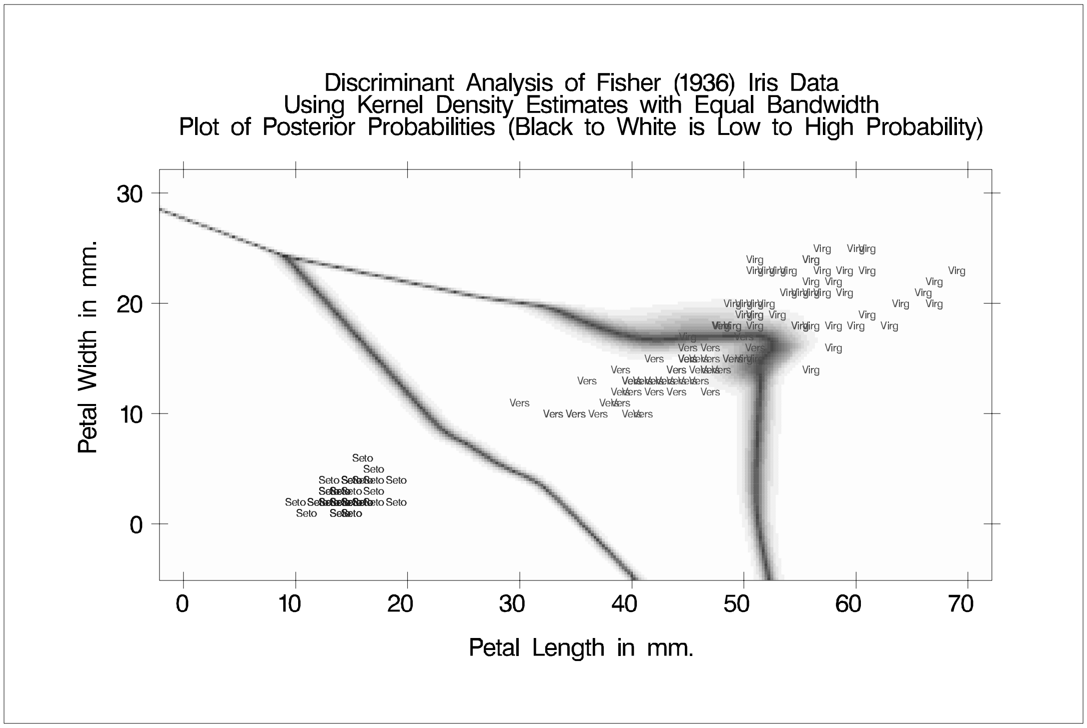
Output 25.2.4. Kernel Density Estimates with Equal Bandwidth

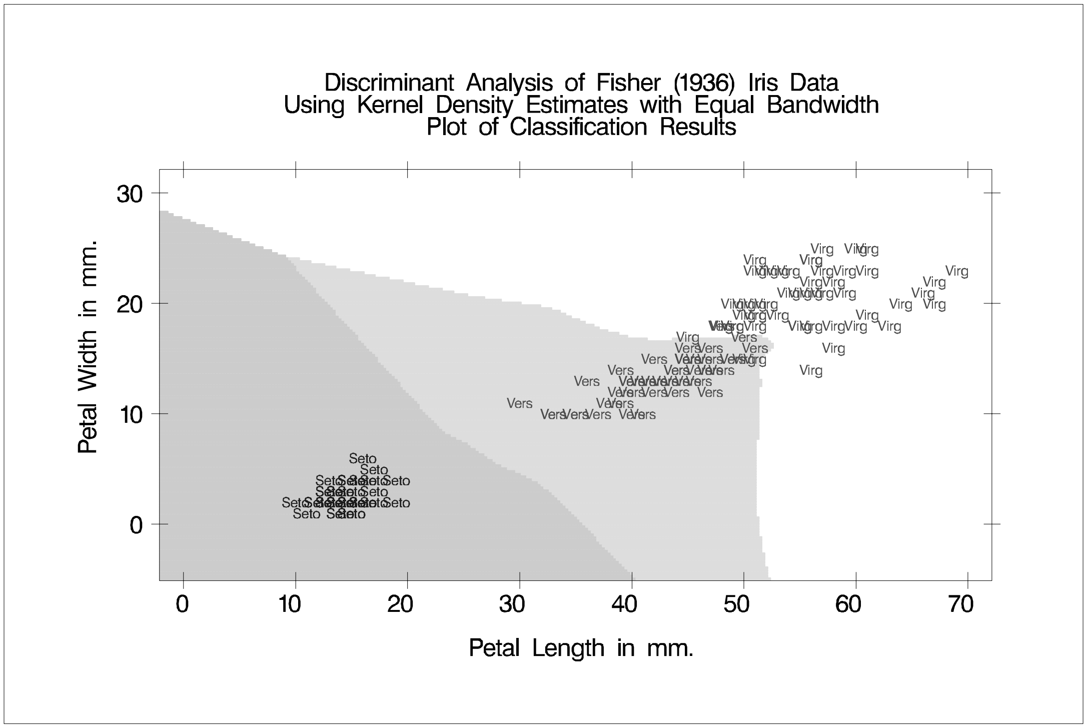
```
                Discriminant Analysis of Fisher (1936) Iris Data
              Using Kernel Density Estimates with Equal Bandwidth

                             The DISCRIM Procedure

            Observations     150          DF Total               149
            Variables          2          DF Within Classes      147
            Classes            3          DF Between Classes       2


                            Class Level Information

              Variable                                                 Prior
Species       Name           Frequency       Weight    Proportion    Probability

Setosa        Setosa                50      50.0000      0.333333       0.333333
Versicolor    Versicolor            50      50.0000      0.333333       0.333333
Virginica     Virginica             50      50.0000      0.333333       0.333333
```

```
              Discriminant Analysis of Fisher (1936) Iris Data
            Using Kernel Density Estimates with Equal Bandwidth

                           The DISCRIM Procedure
           Classification Results for Calibration Data: WORK.IRIS
            Cross-validation Results using Normal Kernel Density

                         Squared Distance Function

                         2                  -1
                        D (X,Y) = (X-Y)' COV  (X-Y)

            Posterior Probability of Membership in Each Species

                           -1                2           2
                 F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                           j   i                 ji

                Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                               j           k       k


              Posterior Probability of Membership in Species

       From          Classified
Obs    Species       into Species        Setosa    Versicolor     Virginica

  5    Virginica     Versicolor *        0.0000        0.7474        0.2526
  9    Versicolor    Virginica  *        0.0000        0.0800        0.9200
 25    Virginica     Versicolor *        0.0000        0.5863        0.4137
 91    Virginica     Versicolor *        0.0000        0.8358        0.1642
148    Versicolor    Virginica  *        0.0000        0.4123        0.5877

                      * Misclassified observation
              Discriminant Analysis of Fisher (1936) Iris Data
            Using Kernel Density Estimates with Equal Bandwidth

                           The DISCRIM Procedure
           Classification Summary for Calibration Data: WORK.IRIS
            Cross-validation Summary using Normal Kernel Density

                         Squared Distance Function

                         2                  -1
                        D (X,Y) = (X-Y)' COV  (X-Y)

            Posterior Probability of Membership in Each Species

                           -1               2           2
                 F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                           j   i                 ji

                Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                               j           k       k


        Number of Observations and Percent Classified into Species

    From
    Species          Setosa      Versicolor      Virginica        Total

    Setosa               50               0              0           50
                     100.00            0.00           0.00       100.00

    Versicolor            0              48              2           50
                       0.00           96.00           4.00       100.00

    Virginica             0               3             47           50
                       0.00            6.00          94.00       100.00

    Total                50              51             49          150
                      33.33           34.00          32.67       100.00

    Priors          0.33333         0.33333        0.33333


                     Error Count Estimates for Species

                        Setosa    Versicolor    Virginica       Total

      Rate              0.0000        0.0400       0.0600      0.0333
      Priors            0.3333        0.3333       0.3333
```

```
              Discriminant Analysis of Fisher (1936) Iris Data
             Using Kernel Density Estimates with Equal Bandwidth

                          The DISCRIM Procedure
             Classification Summary for Test Data: WORK.PLOTDATA
              Classification Summary using Normal Kernel Density

                         Squared Distance Function

                         2                   -1
                        D (X,Y) = (X-Y)' COV  (X-Y)

             Posterior Probability of Membership in Each Species

                           -1                2           2
                  F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                           j   i                  ji

                 Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                               j              k       k


         Number of Observations and Percent Classified into Species

                      Setosa      Versicolor      Virginica        Total

     Total             12631            9941          21681        44253
                       28.54           22.46          48.99       100.00

     Priors          0.33333         0.33333        0.33333
```




Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth
Plot of Posterior Probabilities (Black to White is Low to High Probability)
Petal Width in mm.
30
20
10
0
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg


Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth
Plot of Classification Results
Petal Width in mm.
30
20
10
0
Petal Length in mm.
0
10
20
30
40
50
60
70
Seto
Vers
Virg

Another nonparametric analysis is run with unequal bandwidths (POOL=NO). These statements produce Output 25.2.5:

```
proc discrim data=iris method=npar kernel=normal
                r=.5 pool=no
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
   title2 'Using Kernel Density Estimates with Unequal
           Bandwidth';
run;
%contour
```

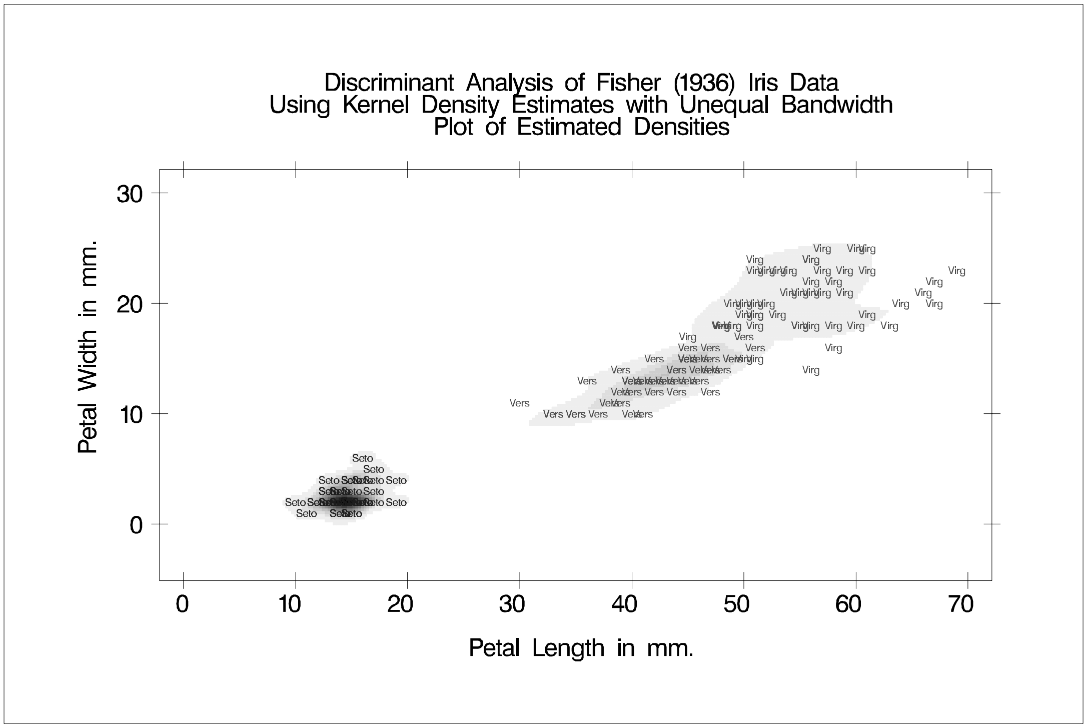
Output 25.2.5. Kernel Density Estimates with Unequal Bandwidth

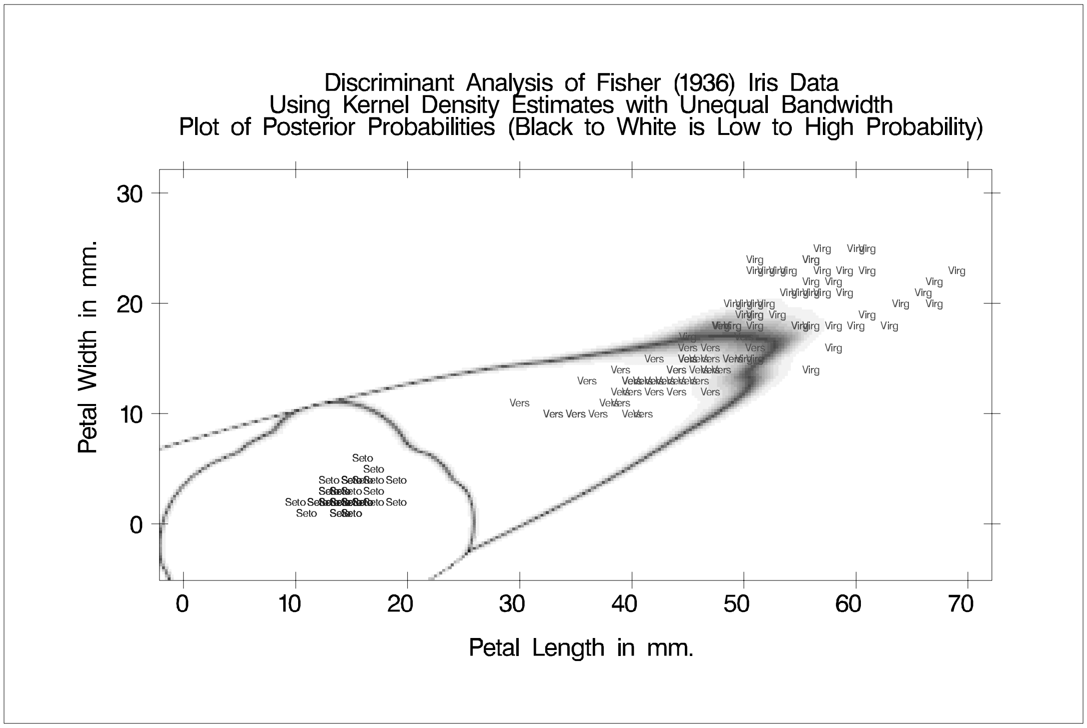
```
                Discriminant Analysis of Fisher (1936) Iris Data
             Using Kernel Density Estimates with Unequal Bandwidth

                             The DISCRIM Procedure

            Observations     150          DF Total              149
            Variables          2          DF Within Classes     147
            Classes            3          DF Between Classes      2


                            Class Level Information

              Variable                                                 Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```

```
              Discriminant Analysis of Fisher (1936) Iris Data
           Using Kernel Density Estimates with Unequal Bandwidth

                           The DISCRIM Procedure
           Classification Results for Calibration Data: WORK.IRIS
            Cross-validation Results using Normal Kernel Density

                         Squared Distance Function

                         2                  -1
                        D (X,Y) = (X-Y)' COV  (X-Y)
                                            j

            Posterior Probability of Membership in Each Species

                           -1               2           2
                 F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                           j   i                 ji

                Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                               j           k       k


              Posterior Probability of Membership in Species

       From          Classified
Obs    Species       into Species        Setosa    Versicolor     Virginica

  5    Virginica     Versicolor *        0.0000        0.7826        0.2174
  9    Versicolor    Virginica  *        0.0000        0.0506        0.9494
 91    Virginica     Versicolor *        0.0000        0.8802        0.1198
148    Versicolor    Virginica  *        0.0000        0.3726        0.6274

                      * Misclassified observation
```

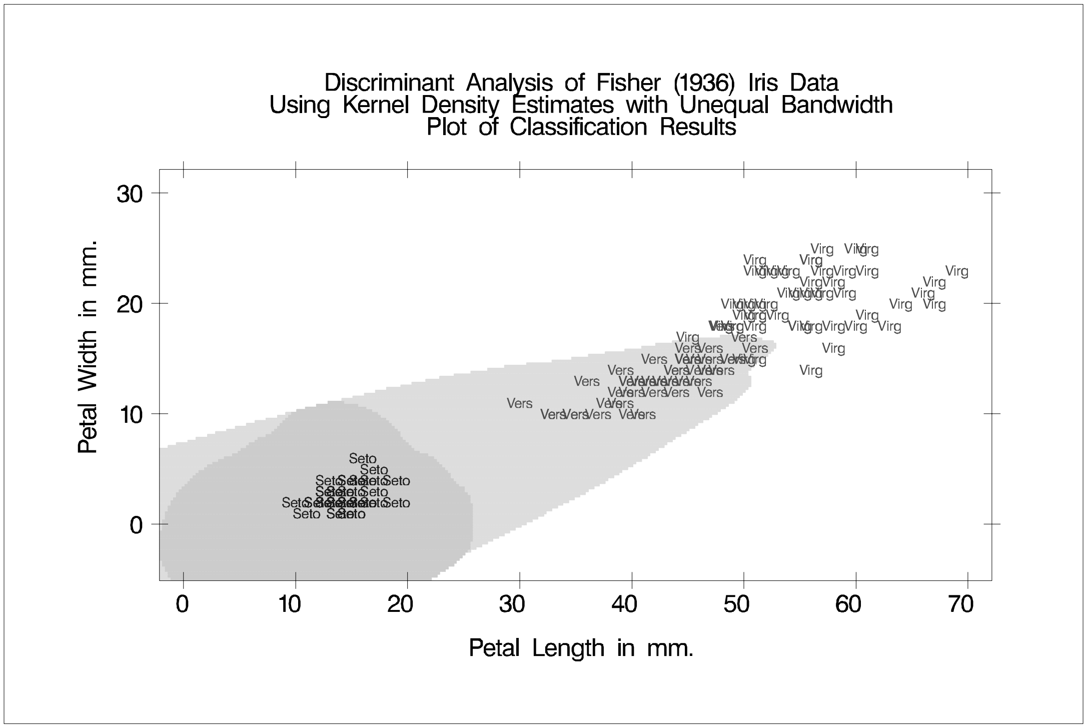
```
          Discriminant Analysis of Fisher (1936) Iris Data
       Using Kernel Density Estimates with Unequal Bandwidth

                       The DISCRIM Procedure
       Classification Summary for Calibration Data: WORK.IRIS
        Cross-validation Summary using Normal Kernel Density

                     Squared Distance Function

                     2                  -1
                    D (X,Y) = (X-Y)’ COV  (X-Y)
                                        j

        Posterior Probability of Membership in Each Species

                       -1               2           2
             F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                       j   i                 ji

            Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                           j           k       k


    Number of Observations and Percent Classified into Species

From
Species          Setosa      Versicolor      Virginica        Total

Setosa               50               0              0           50
                 100.00            0.00           0.00       100.00

Versicolor            0              48              2           50
                   0.00           96.00           4.00       100.00

Virginica             0               2             48           50
                   0.00            4.00          96.00       100.00

Total                50              50             50          150
                  33.33           33.33          33.33       100.00

Priors          0.33333         0.33333        0.33333


                 Error Count Estimates for Species

                    Setosa    Versicolor    Virginica       Total

  Rate              0.0000        0.0400       0.0400      0.0267
  Priors            0.3333        0.3333       0.3333
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
              Using Kernel Density Estimates with Unequal Bandwidth

                             The DISCRIM Procedure
               Classification Summary for Test Data: WORK.PLOTDATA
                Classification Summary using Normal Kernel Density

                            Squared Distance Function

                           2                   -1
                          D (X,Y) = (X-Y)' COV  (X-Y)
                                                j

               Posterior Probability of Membership in Each Species

                              -1               2           2
                     F(X|j) = n   SUM exp( -.5 D (X,Y  ) / R  )
                              j   i                  ji

                    Pr(j|X) = PRIOR  F(X|j) / SUM PRIOR  F(X|k)
                                   j            k       k


           Number of Observations and Percent Classified into Species

                        Setosa      Versicolor      Virginica        Total

       Total              5447            5984          32822        44253
                         12.31           13.52          74.17       100.00

       Priors          0.33333         0.33333        0.33333
```





Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth
Plot of Posterior Probabilities (Black to White is Low to High Probability)



Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth
Plot of Classification Results

Example 25.3. Normal-Theory Discriminant Analysis of Iris Data

In this example, PROC DISCRIM uses normal-theory methods to classify the iris data used in Example 25.1. The POOL=TEST option tests the homogeneity of the within-group covariance matrices (Output 25.3.3). Since the resulting test statistic is significant at the 0.10 level, the within-group covariance matrices are used to derive the quadratic discriminant criterion. The WCOV and PCOV options display the within-group covariance matrices and the pooled covariance matrix (Output 25.3.2). The DISTANCE option displays squared distances between classes (Output 25.3.4). The ANOVA and MANOVA options test the hypothesis that the class means are equal, using univariate statistics and multivariate statistics; all statistics are significant at the 0.0001 level (Output 25.3.5). The LISTERR option lists the misclassified observations under resubstitution (Output 25.3.6). The CROSSLISTERR option lists the observations that are misclassified under cross validation and displays cross validation error-rate estimates (Output 25.3.7). The resubstitution error count estimate, 0.02, is not larger than the cross validation error count estimate, 0.0267, as would be expected because the resubstitution estimate is optimistically biased. The OUTSTAT= option generates a TYPE=MIXED (because POOL=TEST) output data set containing various statistics such as means, covariances, and coefficients of the discriminant function (Output 25.3.8).

The following statements produce Output 25.3.1 through Output 25.3.8:

```
proc discrim data=iris outstat=irisstat
             wcov pcov method=normal pool=test
             distance anova manova listerr crosslisterr;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
   title2 'Using Quadratic Discriminant Function';
run;

proc print data=irisstat;
   title2 'Output Discriminant Statistics';
run;
```

Output 25.3.1. Quadratic Discriminant Analysis of Iris Data

```
                  Discriminant Analysis of Fisher (1936) Iris Data
                        Using Quadratic Discriminant Function

                                The DISCRIM Procedure

             Observations     150          DF Total               149
             Variables          4          DF Within Classes      147
             Classes            3          DF Between Classes       2


                               Class Level Information

                Variable                                                Prior
  Species       Name             Frequency        Weight    Proportion    Probability

  Setosa        Setosa                  50       50.0000      0.333333       0.333333
  Versicolor    Versicolor              50       50.0000      0.333333       0.333333
  Virginica     Virginica               50       50.0000      0.333333       0.333333
```

Output 25.3.2. Covariance Matrices

```
                      Discriminant Analysis of Fisher (1936) Iris Data
                            Using Quadratic Discriminant Function

                                    The DISCRIM Procedure
                              Within-Class Covariance Matrices

                                Species = Setosa,     DF = 49

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.    12.42489796     9.92163265     1.63551020     1.03306122
 SepalWidth    Sepal Width in mm.      9.92163265    14.36897959     1.16979592     0.92979592
 PetalLength   Petal Length in mm.     1.63551020     1.16979592     3.01591837     0.60693878
 PetalWidth    Petal Width in mm.      1.03306122     0.92979592     0.60693878     1.11061224

----------------------------------------------------------------------------------------------


                              Species = Versicolor,     DF = 49

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.    26.64326531     8.51836735    18.28979592     5.57795918
 SepalWidth    Sepal Width in mm.      8.51836735     9.84693878     8.26530612     4.12040816
 PetalLength   Petal Length in mm.    18.28979592     8.26530612    22.08163265     7.31020408
 PetalWidth    Petal Width in mm.      5.57795918     4.12040816     7.31020408     3.91061224

----------------------------------------------------------------------------------------------


                              Species = Virginica,     DF = 49

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.    40.43428571     9.37632653    30.32897959     4.90938776
 SepalWidth    Sepal Width in mm.      9.37632653    10.40040816     7.13795918     4.76285714
 PetalLength   Petal Length in mm.    30.32897959     7.13795918    30.45877551     4.88244898
 PetalWidth    Petal Width in mm.      4.90938776     4.76285714     4.88244898     7.54326531

----------------------------------------------------------------------------------------------
```

```
                      Discriminant Analysis of Fisher (1936) Iris Data
                            Using Quadratic Discriminant Function

                                    The DISCRIM Procedure

                      Pooled Within-Class Covariance Matrix,     DF = 147

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.    26.50081633     9.27210884    16.75142857     3.84013605
 SepalWidth    Sepal Width in mm.      9.27210884    11.53877551     5.52435374     3.27102041
 PetalLength   Petal Length in mm.    16.75142857     5.52435374    18.51877551     4.26653061
 PetalWidth    Petal Width in mm.      3.84013605     3.27102041     4.26653061     4.18816327


                            Within Covariance Matrix Information

                                              Natural Log of the
                                Covariance   Determinant of the
                      Species   Matrix Rank   Covariance Matrix

                      Setosa              4             5.35332
                      Versicolor          4             7.54636
                      Virginica           4             9.49362
                      Pooled              4             8.46214
```

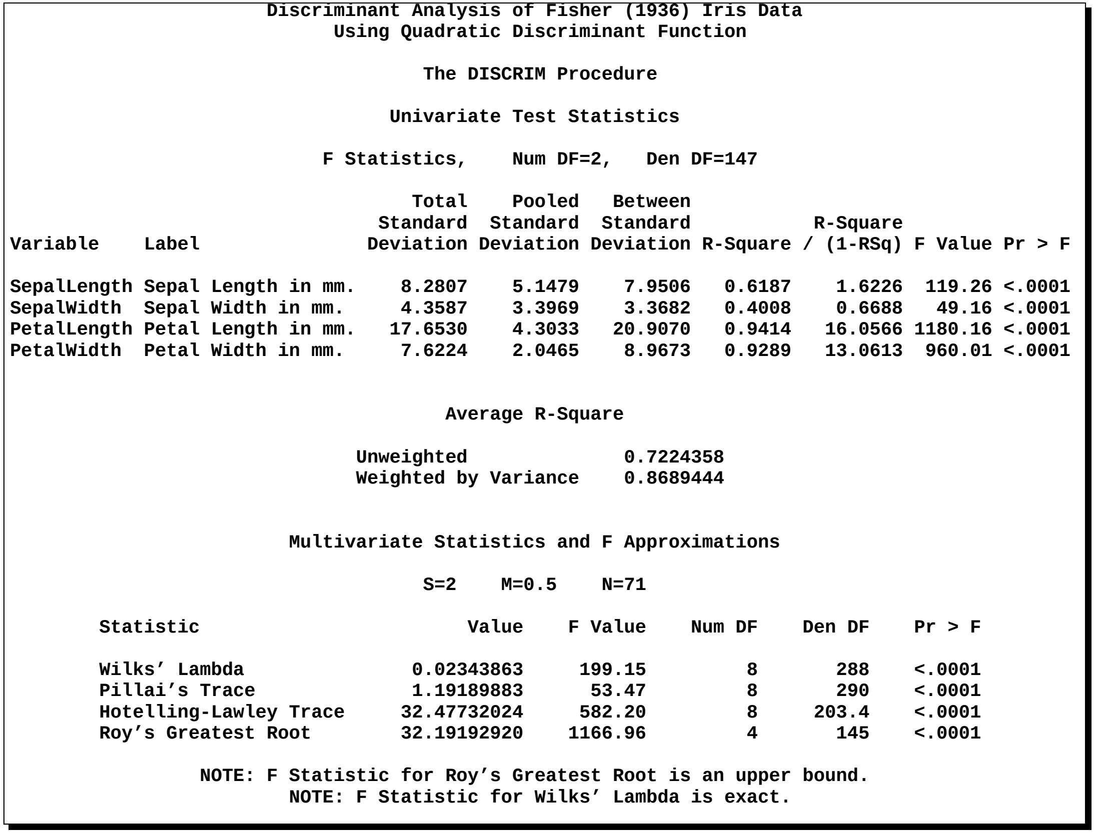
Output 25.3.3. Homogeneity Test

```
        Discriminant Analysis of Fisher (1936) Iris Data
              Using Quadratic Discriminant Function

                      The DISCRIM Procedure
        Test of Homogeneity of Within Covariance Matrices

Notation: K    = Number of Groups

          P    = Number of Variables

          N    = Total Number of Observations - Number of Groups

          N(i) = Number of Observations in the i'th Group - 1

                   __                        N(i)/2
                   ||  |Within SS Matrix(i)|
          V    = -----------------------------------
                                           N/2
                       |Pooled SS Matrix|

                        _                  _     2
                       |       1        1   |  2P + 3P - 1
          RHO  = 1.0 - | SUM -----  -  ---  | -------------
                       |_     N(i)      N  _|  6(P+1)(K-1)

          DF   = .5(K-1)P(P+1)
                                           _                   _
                                          |    PN/2             |
                                          |   N        V        |
Under the null hypothesis:      -2 RHO ln | ------------------ |
                                          |   __       PN(i)/2  |
                                          |_  ||  N(i)         _|

is distributed approximately as Chi-Square(DF).


                Chi-Square        DF    Pr > ChiSq

                140.943050        20        <.0001

Since the Chi-Square value is significant at the 0.1 level, the within
covariance matrices will be used in the discriminant function.
Reference: Morrison, D.F. (1976) Multivariate Statistical Methods
p252.
```

Output 25.3.4. Squared Distances

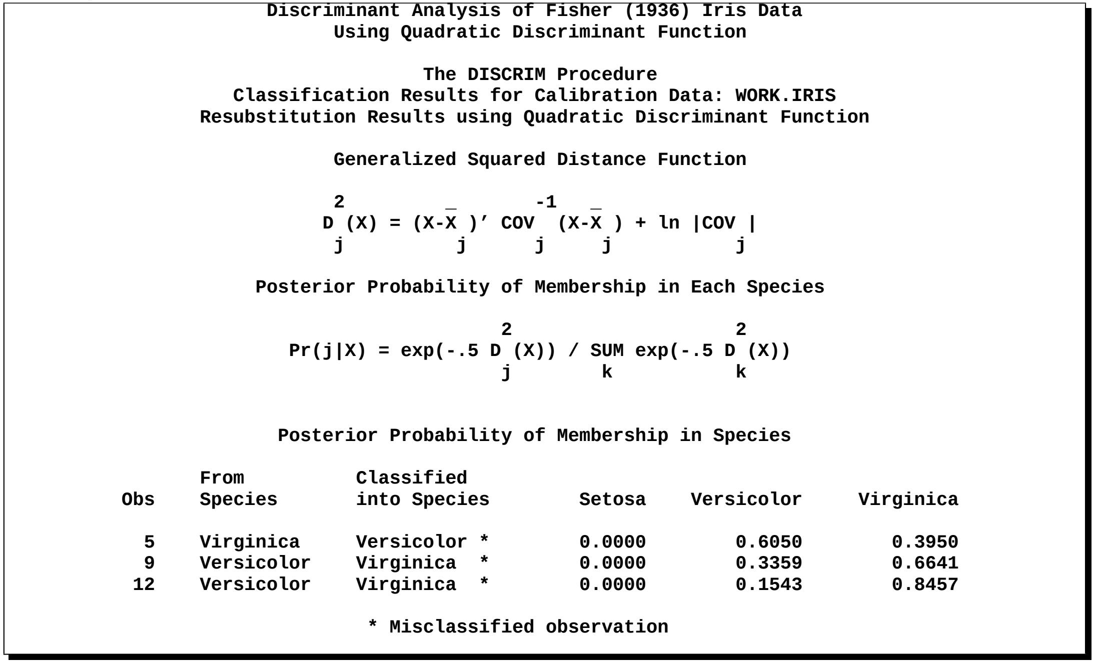
```
  Discriminant Analysis of Fisher (1936) Iris Data
        Using Quadratic Discriminant Function

                The DISCRIM Procedure

      Pairwise Squared Distances Between Groups

          2         _   _       -1  _   _
         D (i|j) = (X - X )' COV   (X - X )
                     i   j      j    i   j


            Squared Distance to Species

From
Species           Setosa    Versicolor     Virginica

Setosa                 0     103.19382     168.76759
Versicolor     323.06203             0      13.83875
Virginica      706.08494      17.86670             0

Pairwise Generalized Squared Distances Between Groups

    2         _   _       -1  _   _
   D (i|j) = (X - X )' COV   (X - X ) + ln |COV |
               i   j      j    i   j           j


      Generalized Squared Distance to Species

From
Species           Setosa    Versicolor     Virginica

Setosa           5.35332     110.74017     178.26121
Versicolor     328.41535       7.54636      23.33238
Virginica      711.43826      25.41306       9.49362
```

Output 25.3.5. Tests of Equal Class Means



```
                       Discriminant Analysis of Fisher (1936) Iris Data
                             Using Quadratic Discriminant Function

                                     The DISCRIM Procedure

                                  Univariate Test Statistics

                            F Statistics,    Num DF=2,   Den DF=147

                                    Total    Pooled   Between
                                 Standard  Standard  Standard           R-Square
Variable    Label               Deviation Deviation Deviation R-Square / (1-RSq) F Value Pr > F

SepalLength Sepal Length in mm.    8.2807    5.1479    7.9506   0.6187    1.6226  119.26 <.0001
SepalWidth  Sepal Width in mm.     4.3587    3.3969    3.3682   0.4008    0.6688   49.16 <.0001
PetalLength Petal Length in mm.   17.6530    4.3033   20.9070   0.9414   16.0566 1180.16 <.0001
PetalWidth  Petal Width in mm.     7.6224    2.0465    8.9673   0.9289   13.0613  960.01 <.0001


                                      Average R-Square

                               Unweighted              0.7224358
                               Weighted by Variance    0.8689444


                         Multivariate Statistics and F Approximations

                                   S=2    M=0.5    N=71

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.02343863     199.15         8       288    <.0001
        Pillai's Trace              1.19189883      53.47         8       290    <.0001
        Hotelling-Lawley Trace     32.47732024     582.20         8     203.4    <.0001
        Roy's Greatest Root        32.19192920    1166.96         4       145    <.0001

                 NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                         NOTE: F Statistic for Wilks' Lambda is exact.
```

Output 25.3.6. Misclassified Observations: Resubstitution



```
                       Discriminant Analysis of Fisher (1936) Iris Data
                             Using Quadratic Discriminant Function

                                     The DISCRIM Procedure
                          Classification Results for Calibration Data: WORK.IRIS
                       Resubstitution Results using Quadratic Discriminant Function

                             Generalized Squared Distance Function

                              2         _        -1   _
                             D (X) = (X-X )' COV  (X-X ) + ln |COV |
                              j          j      j     j           j

                        Posterior Probability of Membership in Each Species

                                            2                    2
                           Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                              j        k           k


                          Posterior Probability of Membership in Species

                 From          Classified
          Obs    Species       into Species        Setosa    Versicolor     Virginica

            5    Virginica     Versicolor *        0.0000        0.6050        0.3950
            9    Versicolor    Virginica  *        0.0000        0.3359        0.6641
           12    Versicolor    Virginica  *        0.0000        0.1543        0.8457

                               * Misclassified observation
```

```
         Discriminant Analysis of Fisher (1936) Iris Data
               Using Quadratic Discriminant Function

                       The DISCRIM Procedure
      Classification Summary for Calibration Data: WORK.IRIS
   Resubstitution Summary using Quadratic Discriminant Function

               Generalized Squared Distance Function

               2         _       -1   _
              D (X) = (X-X )’ COV  (X-X ) + ln |COV |
               j          j      j     j           j

        Posterior Probability of Membership in Each Species

                              2                    2
           Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                              j        k           k


    Number of Observations and Percent Classified into Species

From
Species          Setosa      Versicolor      Virginica        Total

Setosa               50               0              0           50
                 100.00            0.00           0.00       100.00

Versicolor            0              48              2           50
                   0.00           96.00           4.00       100.00

Virginica             0               1             49           50
                   0.00            2.00          98.00       100.00

Total                50              49             51          150
                  33.33           32.67          34.00       100.00

Priors          0.33333         0.33333        0.33333


                 Error Count Estimates for Species

                    Setosa    Versicolor    Virginica       Total

  Rate              0.0000        0.0400       0.0200      0.0200
  Priors            0.3333        0.3333       0.3333
```

Output 25.3.7. Misclassified Observations: Cross validation

```
             Discriminant Analysis of Fisher (1936) Iris Data
                   Using Quadratic Discriminant Function

                           The DISCRIM Procedure
          Classification Results for Calibration Data: WORK.IRIS
      Cross-validation Results using Quadratic Discriminant Function

                   Generalized Squared Distance Function

             2         _          -1      _
            D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
             j          (X)j      (X)j    (X)j           (X)j

            Posterior Probability of Membership in Each Species

                                  2                    2
               Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                  j        k           k


              Posterior Probability of Membership in Species

       From          Classified
Obs    Species       into Species        Setosa    Versicolor     Virginica

  5    Virginica     Versicolor *        0.0000        0.6632        0.3368
  8    Versicolor    Virginica  *        0.0000        0.3134        0.6866
  9    Versicolor    Virginica  *        0.0000        0.1616        0.8384
 12    Versicolor    Virginica  *        0.0000        0.0713        0.9287

                      * Misclassified observation
```

```
         Discriminant Analysis of Fisher (1936) Iris Data
               Using Quadratic Discriminant Function

                       The DISCRIM Procedure
      Classification Summary for Calibration Data: WORK.IRIS
  Cross-validation Summary using Quadratic Discriminant Function

               Generalized Squared Distance Function

         2         _          -1      _
        D (X) = (X-X    )' COV    (X-X    ) + ln |COV    |
         j          (X)j      (X)j    (X)j           (X)j

        Posterior Probability of Membership in Each Species

                              2                    2
           Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                              j        k           k


    Number of Observations and Percent Classified into Species

From
Species          Setosa      Versicolor      Virginica         Total

Setosa               50               0              0            50
                 100.00            0.00           0.00        100.00

Versicolor            0              47              3            50
                   0.00           94.00           6.00        100.00

Virginica             0               1             49            50
                   0.00            2.00          98.00        100.00

Total                50              48             52           150
                  33.33           32.00          34.67        100.00

Priors          0.33333         0.33333        0.33333


                 Error Count Estimates for Species

                    Setosa    Versicolor    Virginica       Total

  Rate              0.0000        0.0600       0.0200      0.0267
  Priors            0.3333        0.3333       0.3333
```

Output 25.3.8. Output Statistics from Iris Data

```
                    Discriminant Analysis of Fisher (1936) Iris Data
                             Output Discriminant Statistics

                                                 Sepal       Sepal        Petal        Petal
Obs     Species       _TYPE_        _NAME_       Length       Width       Length        Width

  1            .      N                          150.00      150.00       150.00       150.00
  2     Setosa        N                           50.00       50.00        50.00        50.00
  3     Versicolor    N                           50.00       50.00        50.00        50.00
  4     Virginica     N                           50.00       50.00        50.00        50.00
  5            .      MEAN                        58.43       30.57        37.58        11.99
  6     Setosa        MEAN                        50.06       34.28        14.62         2.46
  7     Versicolor    MEAN                        59.36       27.70        42.60        13.26
  8     Virginica     MEAN                        65.88       29.74        55.52        20.26
  9     Setosa        PRIOR                        0.33        0.33         0.33         0.33
 10     Versicolor    PRIOR                        0.33        0.33         0.33         0.33
 11     Virginica     PRIOR                        0.33        0.33         0.33         0.33
 12     Setosa        CSSCP       SepalLength    608.82      486.16        80.14        50.62
 13     Setosa        CSSCP       SepalWidth     486.16      704.08        57.32        45.56
 14     Setosa        CSSCP       PetalLength     80.14       57.32       147.78        29.74
 15     Setosa        CSSCP       PetalWidth      50.62       45.56        29.74        54.42
 16     Versicolor    CSSCP       SepalLength   1305.52      417.40       896.20       273.32
 17     Versicolor    CSSCP       SepalWidth     417.40      482.50       405.00       201.90
 18     Versicolor    CSSCP       PetalLength    896.20      405.00      1082.00       358.20
 19     Versicolor    CSSCP       PetalWidth     273.32      201.90       358.20       191.62
 20     Virginica     CSSCP       SepalLength   1981.28      459.44      1486.12       240.56
 21     Virginica     CSSCP       SepalWidth     459.44      509.62       349.76       233.38
 22     Virginica     CSSCP       PetalLength   1486.12      349.76      1492.48       239.24
 23     Virginica     CSSCP       PetalWidth     240.56      233.38       239.24       369.62
 24            .      PSSCP       SepalLength   3895.62     1363.00      2462.46       564.50
 25            .      PSSCP       SepalWidth    1363.00     1696.20       812.08       480.84
 26            .      PSSCP       PetalLength   2462.46      812.08      2722.26       627.18
 27            .      PSSCP       PetalWidth     564.50      480.84       627.18       615.66
 28            .      BSSCP       SepalLength   6321.21    -1995.27     16524.84      7127.93
 29            .      BSSCP       SepalWidth   -1995.27     1134.49     -5723.96     -2293.27
 30            .      BSSCP       PetalLength  16524.84    -5723.96     43710.28     18677.40
 31            .      BSSCP       PetalWidth    7127.93    -2293.27     18677.40      8041.33
 32            .      CSSCP       SepalLength  10216.83     -632.27     18987.30      7692.43
 33            .      CSSCP       SepalWidth    -632.27     2830.69     -4911.88     -1812.43
 34            .      CSSCP       PetalLength  18987.30    -4911.88     46432.54     19304.58
 35            .      CSSCP       PetalWidth    7692.43    -1812.43     19304.58      8656.99
 36            .      RSQUARED                     0.62        0.40         0.94         0.93
 37     Setosa        COV         SepalLength     12.42        9.92         1.64         1.03
 38     Setosa        COV         SepalWidth       9.92       14.37         1.17         0.93
 39     Setosa        COV         PetalLength      1.64        1.17         3.02         0.61
 40     Setosa        COV         PetalWidth       1.03        0.93         0.61         1.11
 41     Versicolor    COV         SepalLength     26.64        8.52        18.29         5.58
 42     Versicolor    COV         SepalWidth       8.52        9.85         8.27         4.12
 43     Versicolor    COV         PetalLength     18.29        8.27        22.08         7.31
 44     Versicolor    COV         PetalWidth       5.58        4.12         7.31         3.91
 45     Virginica     COV         SepalLength     40.43        9.38        30.33         4.91
 46     Virginica     COV         SepalWidth       9.38       10.40         7.14         4.76
 47     Virginica     COV         PetalLength     30.33        7.14        30.46         4.88
 48     Virginica     COV         PetalWidth       4.91        4.76         4.88         7.54
 49            .      PCOV        SepalLength     26.50        9.27        16.75         3.84
 50            .      PCOV        SepalWidth       9.27       11.54         5.52         3.27
 51            .      PCOV        PetalLength     16.75        5.52        18.52         4.27
 52            .      PCOV        PetalWidth       3.84        3.27         4.27         4.19
 53            .      BCOV        SepalLength     63.21      -19.95       165.25        71.28
 54            .      BCOV        SepalWidth     -19.95       11.34       -57.24       -22.93
 55            .      BCOV        PetalLength    165.25      -57.24       437.10       186.77
 56            .      BCOV        PetalWidth      71.28      -22.93       186.77        80.41
 57            .      COV         SepalLength     68.57       -4.24       127.43        51.63
 58            .      COV         SepalWidth      -4.24       19.00       -32.97       -12.16
 59            .      COV         PetalLength    127.43      -32.97       311.63       129.56
 60            .      COV         PetalWidth      51.63      -12.16       129.56        58.10
 61     Setosa        STD                          3.52        3.79         1.74         1.05
 62     Versicolor    STD                          5.16        3.14         4.70         1.98
 63     Virginica     STD                          6.36        3.22         5.52         2.75
 64            .      PSTD                         5.15        3.40         4.30         2.05
 65            .      BSTD                         7.95        3.37        20.91         8.97
 66            .      STD                          8.28        4.36        17.65         7.62
 67     Setosa        CORR        SepalLength      1.00        0.74         0.27         0.28
 68     Setosa        CORR        SepalWidth       0.74        1.00         0.18         0.23
 69     Setosa        CORR        PetalLength      0.27        0.18         1.00         0.33
 70     Setosa        CORR        PetalWidth       0.28        0.23         0.33         1.00
```

```
                 Discriminant Analysis of Fisher (1936) Iris Data
                         Output Discriminant Statistics

                                                Sepal       Sepal       Petal       Petal
Obs    Species      _TYPE_      _NAME_          Length      Width       Length      Width

 71   Versicolor    CORR        SepalLength      1.000       0.526       0.754       0.546
 72   Versicolor    CORR        SepalWidth       0.526       1.000       0.561       0.664
 73   Versicolor    CORR        PetalLength      0.754       0.561       1.000       0.787
 74   Versicolor    CORR        PetalWidth       0.546       0.664       0.787       1.000
 75   Virginica     CORR        SepalLength      1.000       0.457       0.864       0.281
 76   Virginica     CORR        SepalWidth       0.457       1.000       0.401       0.538
 77   Virginica     CORR        PetalLength      0.864       0.401       1.000       0.322
 78   Virginica     CORR        PetalWidth       0.281       0.538       0.322       1.000
 79            .    PCORR       SepalLength      1.000       0.530       0.756       0.365
 80            .    PCORR       SepalWidth       0.530       1.000       0.378       0.471
 81            .    PCORR       PetalLength      0.756       0.378       1.000       0.484
 82            .    PCORR       PetalWidth       0.365       0.471       0.484       1.000
 83            .    BCORR       SepalLength      1.000      -0.745       0.994       1.000
 84            .    BCORR       SepalWidth      -0.745       1.000      -0.813      -0.759
 85            .    BCORR       PetalLength      0.994      -0.813       1.000       0.996
 86            .    BCORR       PetalWidth       1.000      -0.759       0.996       1.000
 87            .    CORR        SepalLength      1.000      -0.118       0.872       0.818
 88            .    CORR        SepalWidth      -0.118       1.000      -0.428      -0.366
 89            .    CORR        PetalLength      0.872      -0.428       1.000       0.963
 90            .    CORR        PetalWidth       0.818      -0.366       0.963       1.000
 91   Setosa        STDMEAN                     -1.011       0.850      -1.301      -1.251
 92   Versicolor    STDMEAN                      0.112      -0.659       0.284       0.166
 93   Virginica     STDMEAN                      0.899      -0.191       1.016       1.085
 94   Setosa        PSTDMEAN                    -1.627       1.091      -5.335      -4.658
 95   Versicolor    PSTDMEAN                     0.180      -0.846       1.167       0.619
 96   Virginica     PSTDMEAN                     1.447      -0.245       4.169       4.039
 97            .    LNDETERM                     8.462       8.462       8.462       8.462
 98   Setosa        LNDETERM                     5.353       5.353       5.353       5.353
 99   Versicolor    LNDETERM                     7.546       7.546       7.546       7.546
100   Virginica     LNDETERM                     9.494       9.494       9.494       9.494
101   Setosa        QUAD        SepalLength     -0.095       0.062       0.023       0.024
102   Setosa        QUAD        SepalWidth       0.062      -0.078      -0.006       0.011
103   Setosa        QUAD        PetalLength      0.023      -0.006      -0.194       0.090
104   Setosa        QUAD        PetalWidth       0.024       0.011       0.090      -0.530
105   Setosa        QUAD        _LINEAR_         4.455      -0.762       3.356      -3.126
106   Setosa        QUAD        _CONST_       -121.826    -121.826    -121.826    -121.826
107   Versicolor    QUAD        SepalLength     -0.048       0.018       0.043      -0.032
108   Versicolor    QUAD        SepalWidth       0.018      -0.099      -0.011       0.097
109   Versicolor    QUAD        PetalLength      0.043      -0.011      -0.099       0.135
110   Versicolor    QUAD        PetalWidth      -0.032       0.097       0.135      -0.436
111   Versicolor    QUAD        _LINEAR_         1.801       1.596       0.327      -1.471
112   Versicolor    QUAD        _CONST_        -76.549     -76.549     -76.549     -76.549
113   Virginica     QUAD        SepalLength     -0.053       0.017       0.050      -0.009
114   Virginica     QUAD        SepalWidth       0.017      -0.079      -0.006       0.042
115   Virginica     QUAD        PetalLength      0.050      -0.006      -0.067       0.014
116   Virginica     QUAD        PetalWidth      -0.009       0.042       0.014      -0.097
117   Virginica     QUAD        _LINEAR_         0.737       1.325       0.623       0.966
118   Virginica     QUAD        _CONST_        -75.821     -75.821     -75.821     -75.821
```

Example 25.4. Linear Discriminant Analysis of Remote-Sensing Data on Crops

In this example, the remote-sensing data described at the beginning of the section are used. In the first PROC DISCRIM statement, the DISCRIM procedure uses normal-theory methods (METHOD=NORMAL) assuming equal variances (POOL=YES) in five crops. The PRIORS statement, PRIORS PROP, sets the prior probabilities proportional to the sample sizes. The LIST option lists the resubstitution classification results for each observation (Output 25.4.2). The CROSSVALIDATE option displays cross validation error-rate estimates (Output 25.4.3). The OUTSTAT= option stores the calibration information in a new data set to classify future observations. A second PROC DISCRIM statement uses this calibration information to classify a test data set. Note that the values of the identification variable, xvalues, are obtained by rereading the x1 through x4 fields in the data lines as a single character variable. The following statements produce Output 25.4.1 through Output 25.4.3.

```
data crops;
   title 'Discriminant Analysis of Remote Sensing Data
          on Five Crops';
   input Crop $ 4-13 x1-x4 xvalues $ 14-24;
   datalines;
Corn      16 27 31 33
Corn      15 23 30 30
Corn      16 27 27 26
Corn      18 20 25 23
Corn      15 15 31 32
Corn      15 32 32 15
Corn      12 15 16 73
Soybeans  20 23 23 25
Soybeans  24 24 25 32
Soybeans  21 25 23 24
Soybeans  27 45 24 12
Soybeans  12 13 15 42
Soybeans  22 32 31 43
Cotton    31 32 33 34
Cotton    29 24 26 28
Cotton    34 32 28 45
Cotton    26 25 23 24
Cotton    53 48 75 26
Cotton    34 35 25 78
Sugarbeets22 23 25 42
Sugarbeets25 25 24 26
Sugarbeets34 25 16 52
Sugarbeets54 23 21 54
Sugarbeets25 43 32 15
Sugarbeets26 54  2 54
Clover    12 45 32 54
Clover    24 58 25 34
Clover    87 54 61 21
Clover    51 31 31 16
Clover    96 48 54 62
Clover    31 31 11 11
Clover    56 13 13 71
Clover    32 13 27 32
Clover    36 26 54 32
Clover    53 08 06 54
Clover    32 32 62 16
;
proc discrim data=crops outstat=cropstat
             method=normal pool=yes
             list crossvalidate;
   class Crop;
   priors prop;
   id xvalues;
   var x1-x4;
   title2 'Using Linear Discriminant Function';
run;
```

Output 25.4.1. Linear Discriminant Function on Crop Data

```
                  Discriminant Analysis of Remote Sensing Data on Five Crops
                              Using Linear Discriminant Function

                                    The DISCRIM Procedure

                  Observations      36          DF Total                35
                  Variables          4          DF Within Classes       31
                  Classes            5          DF Between Classes       4


                                   Class Level Information

               Variable                                                        Prior
     Crop          Name            Frequency       Weight    Proportion    Probability

     Clover        Clover                 11      11.0000      0.305556       0.305556
     Corn          Corn                    7       7.0000      0.194444       0.194444
     Cotton        Cotton                  6       6.0000      0.166667       0.166667
     Soybeans      Soybeans                6       6.0000      0.166667       0.166667
     Sugarbeets    Sugarbeets              6       6.0000      0.166667       0.166667
```

```
                  Discriminant Analysis of Remote Sensing Data on Five Crops
                              Using Linear Discriminant Function

                                    The DISCRIM Procedure

                            Pooled Covariance Matrix Information

                                              Natural Log of the
                                Covariance    Determinant of the
                               Matrix Rank     Covariance Matrix

                                         4              21.30189
```

```
                  Discriminant Analysis of Remote Sensing Data on Five Crops
                              Using Linear Discriminant Function

                                    The DISCRIM Procedure

                         Pairwise Generalized Squared Distances Between Groups

                        2          _   _          -1  _   _
                       D (i|j) = (X - X )' COV   (X - X ) - 2 ln PRIOR
                                   i   j          i   j              j


                              Generalized Squared Distance to Crop

     From Crop           Clover          Corn        Cotton      Soybeans    Sugarbeets

     Clover             2.37125       7.52830       4.44969       6.16665       5.07262
     Corn               6.62433       3.27522       5.46798       4.31383       6.47395
     Cotton             3.23741       5.15968       3.58352       5.01819       4.87908
     Soybeans           4.95438       4.00552       5.01819       3.58352       4.65998
     Sugarbeets         3.86034       6.16564       4.87908       4.65998       3.58352
```

```
                Discriminant Analysis of Remote Sensing Data on Five Crops
                            Using Linear Discriminant Function

                                  The DISCRIM Procedure

                                Linear Discriminant Function

                            _      -1 _                                     -1 _
              Constant = -.5 X' COV   X  + ln PRIOR     Coefficient = COV   X
                              j        j           j    Vector                j


                         Linear Discriminant Function for Crop

       Variable        Clover          Corn        Cotton      Soybeans    Sugarbeets

       Constant     -10.98457      -7.72070     -11.46537      -7.28260      -9.80179
       x1             0.08907      -0.04180       0.02462     0.0000369       0.04245
       x2             0.17379       0.11970       0.17596       0.15896       0.20988
       x3             0.11899       0.16511       0.15880       0.10622       0.06540
       x4             0.15637       0.16768       0.18362       0.14133       0.16408
```

Output 25.4.2. Misclassified Observations: Resubstitution

```
                   Discriminant Analysis of Remote Sensing Data on Five Crops
                               Using Linear Discriminant Function

                                      The DISCRIM Procedure
                     Classification Results for Calibration Data: WORK.CROPS
                    Resubstitution Results using Linear Discriminant Function

                              Generalized Squared Distance Function

                             2         _       -1   _
                            D (X) = (X-X )' COV  (X-X ) - 2 ln PRIOR
                             j          j            j              j

                        Posterior Probability of Membership in Each Crop

                                             2                    2
                          Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                             j        k           k


                           Posterior Probability of Membership in Crop

                         Classified
xvalues      From Crop   into Crop         Clover        Corn      Cotton    Soybeans  Sugarbeets

16 27 31 33  Corn        Corn              0.0894      0.4054      0.1763      0.2392      0.0897
15 23 30 30  Corn        Corn              0.0769      0.4558      0.1421      0.2530      0.0722
16 27 27 26  Corn        Corn              0.0982      0.3422      0.1365      0.3073      0.1157
18 20 25 23  Corn        Corn              0.1052      0.3634      0.1078      0.3281      0.0955
15 15 31 32  Corn        Corn              0.0588      0.5754      0.1173      0.2087      0.0398
15 32 32 15  Corn        Soybeans   *      0.0972      0.3278      0.1318      0.3420      0.1011
12 15 16 73  Corn        Corn              0.0454      0.5238      0.1849      0.1376      0.1083
20 23 23 25  Soybeans    Soybeans          0.1330      0.2804      0.1176      0.3305      0.1385
24 24 25 32  Soybeans    Soybeans          0.1768      0.2483      0.1586      0.2660      0.1502
21 25 23 24  Soybeans    Soybeans          0.1481      0.2431      0.1200      0.3318      0.1570
27 45 24 12  Soybeans    Sugarbeets *      0.2357      0.0547      0.1016      0.2721      0.3359
12 13 15 42  Soybeans    Corn       *      0.0549      0.4749      0.0920      0.2768      0.1013
22 32 31 43  Soybeans    Cotton     *      0.1474      0.2606      0.2624      0.1848      0.1448
31 32 33 34  Cotton      Clover     *      0.2815      0.1518      0.2377      0.1767      0.1523
29 24 26 28  Cotton      Soybeans   *      0.2521      0.1842      0.1529      0.2549      0.1559
34 32 28 45  Cotton      Clover     *      0.3125      0.1023      0.2404      0.1357      0.2091
26 25 23 24  Cotton      Soybeans   *      0.2121      0.1809      0.1245      0.3045      0.1780
53 48 75 26  Cotton      Clover     *      0.4837      0.0391      0.4384      0.0223      0.0166
34 35 25 78  Cotton      Cotton            0.2256      0.0794      0.3810      0.0592      0.2548
22 23 25 42  Sugarbeets  Corn       *      0.1421      0.3066      0.1901      0.2231      0.1381
25 25 24 26  Sugarbeets  Soybeans   *      0.1969      0.2050      0.1354      0.2960      0.1667
34 25 16 52  Sugarbeets  Sugarbeets        0.2928      0.0871      0.1665      0.1479      0.3056
54 23 21 54  Sugarbeets  Clover     *      0.6215      0.0194      0.1250      0.0496      0.1845
25 43 32 15  Sugarbeets  Soybeans   *      0.2258      0.1135      0.1646      0.2770      0.2191
26 54  2 54  Sugarbeets  Sugarbeets        0.0850      0.0081      0.0521      0.0661      0.7887
12 45 32 54  Clover      Cotton     *      0.0693      0.2663      0.3394      0.1460      0.1789
24 58 25 34  Clover      Sugarbeets *      0.1647      0.0376      0.1680      0.1452      0.4845
87 54 61 21  Clover      Clover            0.9328      0.0003      0.0478      0.0025      0.0165
51 31 31 16  Clover      Clover            0.6642      0.0205      0.0872      0.0959      0.1322
96 48 54 62  Clover      Clover            0.9215      0.0002      0.0604      0.0007      0.0173
31 31 11 11  Clover      Sugarbeets *      0.2525      0.0402      0.0473      0.3012      0.3588
56 13 13 71  Clover      Clover            0.6132      0.0212      0.1226      0.0408      0.2023
32 13 27 32  Clover      Clover            0.2669      0.2616      0.1512      0.2260      0.0943
36 26 54 32  Clover      Cotton     *      0.2650      0.2645      0.3495      0.0918      0.0292
53 08 06 54  Clover      Clover            0.5914      0.0237      0.0676      0.0781      0.2392
32 32 62 16  Clover      Cotton     *      0.2163      0.3180      0.3327      0.1125      0.0206

                                 * Misclassified observation
                     Discriminant Analysis of Remote Sensing Data on Five Crops
                                 Using Linear Discriminant Function

                                       The DISCRIM Procedure
                      Classification Summary for Calibration Data: WORK.CROPS
                     Resubstitution Summary using Linear Discriminant Function

                                 Generalized Squared Distance Function

                                2         _        -1   _
                               D (X) = (X-X )' COV  (X-X ) - 2 ln PRIOR
                                j          j          j              j

                            Posterior Probability of Membership in Each Crop

                                           2                    2
                          Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                             j        k            k


                        Number of Observations and Percent Classified into Crop

    From Crop         Clover         Corn       Cotton      Soybeans      Sugarbeets        Total

    Clover                 6            0            3             0               2           11
                       54.55         0.00        27.27          0.00           18.18       100.00

    Corn                   0            6            0             1               0            7
                        0.00        85.71         0.00         14.29            0.00       100.00

    Cotton                 3            0            1             2               0            6
                       50.00         0.00        16.67         33.33            0.00       100.00

    Soybeans               0            1            1             3               1            6
                        0.00        16.67        16.67         50.00           16.67       100.00

    Sugarbeets             1            1            0             2               2            6
                       16.67        16.67         0.00         33.33           33.33       100.00

    Total                 10            8            5             8               5           36
                       27.78        22.22        13.89         22.22           13.89       100.00

    Priors           0.30556      0.19444      0.16667       0.16667         0.16667


                                     Error Count Estimates for Crop

                          Clover         Corn       Cotton     Soybeans     Sugarbeets       Total

       Rate               0.4545       0.1429       0.8333       0.5000         0.6667      0.5000
       Priors             0.3056       0.1944       0.1667       0.1667         0.1667
```

Output 25.4.3. Misclassified Observations: Cross Validation

```
                 Discriminant Analysis of Remote Sensing Data on Five Crops
                             Using Linear Discriminant Function

                                    The DISCRIM Procedure
                   Classification Summary for Calibration Data: WORK.CROPS
                 Cross-validation Summary using Linear Discriminant Function

                            Generalized Squared Distance Function

                       2          _          -1      _
                      D (X) = (X-X    )' COV    (X-X    ) - 2 ln PRIOR
                       j          (X)j       (X)     (X)j               j

                      Posterior Probability of Membership in Each Crop

                                           2                   2
                        Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                           j          k         k


                  Number of Observations and Percent Classified into Crop

From Crop        Clover         Corn       Cotton      Soybeans      Sugarbeets        Total

Clover                4            3            1             0               3           11
                  36.36        27.27         9.09          0.00           27.27       100.00

Corn                  0            4            1             2               0            7
                   0.00        57.14        14.29         28.57            0.00       100.00

Cotton                3            0            0             2               1            6
                  50.00         0.00         0.00         33.33           16.67       100.00

Soybeans              0            1            1             3               1            6
                   0.00        16.67        16.67         50.00           16.67       100.00

Sugarbeets            2            1            0             2               1            6
                  33.33        16.67         0.00         33.33           16.67       100.00

Total                 9            9            3             9               6           36
                  25.00        25.00         8.33         25.00           16.67       100.00

Priors          0.30556      0.19444      0.16667       0.16667         0.16667


                               Error Count Estimates for Crop

                     Clover        Corn      Cotton    Soybeans    Sugarbeets       Total

   Rate              0.6364      0.4286      1.0000      0.5000        0.8333      0.6667
   Priors            0.3056      0.1944      0.1667      0.1667        0.1667
```

Now use the calibration information stored in the Cropstat data set to classify a test data set. The TESTLIST option lists the classification results for each observation in the test data set. The following statements produce Output 25.4.4 and Output 25.4.5:

```
data test;
   input Crop $ 1-10 x1-x4 xvalues $ 11-21;
   datalines;
Corn      16 27 31 33
Soybeans  21 25 23 24
Cotton    29 24 26 28
Sugarbeets54 23 21 54
Clover    32 32 62 16
;

proc discrim data=cropstat testdata=test testout=tout
             testlist;
   class Crop;
   testid xvalues;
   var x1-x4;
   title2 ’Classification of Test Data’;
run;
proc print data=tout;
   title2 ’Output Classification Results of Test Data’;
run;
```

Output 25.4.4. Classification of Test Data

```
                    Discriminant Analysis of Remote Sensing Data on Five Crops
                                    Classification of Test Data

                                       The DISCRIM Procedure
                          Classification Results for Test Data: WORK.TEST
                       Classification Results using Linear Discriminant Function

                                Generalized Squared Distance Function

                                      2         _         -1   _
                                     D (X) = (X-X )' COV   (X-X )
                                      j          j            j

                             Posterior Probability of Membership in Each Crop

                                               2                    2
                              Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                                 j        k           k


                                Posterior Probability of Membership in Crop

                          Classified
xvalues      From Crop    into Crop          Clover        Corn      Cotton    Soybeans  Sugarbeets

16 27 31 33  Corn         Corn               0.0894      0.4054      0.1763      0.2392      0.0897
21 25 23 24  Soybeans     Soybeans           0.1481      0.2431      0.1200      0.3318      0.1570
29 24 26 28  Cotton       Soybeans   *       0.2521      0.1842      0.1529      0.2549      0.1559
54 23 21 54  Sugarbeets   Clover     *       0.6215      0.0194      0.1250      0.0496      0.1845
32 32 62 16  Clover       Cotton     *       0.2163      0.3180      0.3327      0.1125      0.0206

                                  * Misclassified observation
```

```
                 Discriminant Analysis of Remote Sensing Data on Five Crops
                                Classification of Test Data

                                   The DISCRIM Procedure
                       Classification Summary for Test Data: WORK.TEST
                  Classification Summary using Linear Discriminant Function

                              Generalized Squared Distance Function

                                   2         _        -1   _
                                  D (X) = (X-X )' COV  (X-X )
                                   j          j            j

                          Posterior Probability of Membership in Each Crop

                                            2                    2
                            Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                            j        k           k


                  Number of Observations and Percent Classified into Crop

 From Crop         Clover          Corn        Cotton      Soybeans      Sugarbeets         Total

 Clover                 0             0             1             0               0             1
                     0.00          0.00        100.00          0.00            0.00        100.00

 Corn                   0             1             0             0               0             1
                     0.00        100.00          0.00          0.00            0.00        100.00

 Cotton                 0             0             0             1               0             1
                     0.00          0.00          0.00        100.00            0.00        100.00

 Soybeans               0             0             0             1               0             1
                     0.00          0.00          0.00        100.00            0.00        100.00

 Sugarbeets             1             0             0             0               0             1
                   100.00          0.00          0.00          0.00            0.00        100.00

 Total                  1             1             1             2               0             5
                    20.00         20.00         20.00         40.00            0.00        100.00

 Priors           0.30556       0.19444       0.16667       0.16667         0.16667


                              Error Count Estimates for Crop

                       Clover          Corn        Cotton      Soybeans     Sugarbeets       Total

    Rate               1.0000        0.0000        1.0000        0.0000         1.0000      0.6389
    Priors             0.3056        0.1944        0.1667        0.1667         0.1667
```

Output 25.4.5. Output Data Set of the Classification Results for Test Data

```
                 Discriminant Analysis of Remote Sensing Data on Five Crops
                         Output Classification Results of Test Data

 Obs Crop       x1 x2 x3 x4   xvalues    Clover    Corn    Cotton Soybeans Sugarbeets _INTO_

  1  Corn       16 27 31 33 16 27 31 33 0.08935 0.40543 0.17632  0.23918   0.08972  Corn
  2  Soybeans   21 25 23 24 21 25 23 24 0.14811 0.24308 0.11999  0.33184   0.15698  Soybeans
  3  Cotton     29 24 26 28 29 24 26 28 0.25213 0.18420 0.15294  0.25486   0.15588  Soybeans
  4  Sugarbeets 54 23 21 54 54 23 21 54 0.62150 0.01937 0.12498  0.04962   0.18452  Clover
  5  Clover     32 32 62 16 32 32 62 16 0.21633 0.31799 0.33266  0.11246   0.02056  Cotton
```

Example 25.5. Quadratic Discriminant Analysis of Remote-Sensing Data on Crops

In this example, PROC DISCRIM uses normal-theory methods (METHOD=NORMAL) assuming unequal variances (POOL=NO) for the remote-sensing data of Example 25.4. The PRIORS statement, PRIORS PROP, sets the prior probabilities proportional to the sample sizes. The CROSSVALIDATE option displays cross validation error-rate estimates. Note that the total error count estimate by cross validation (0.5556) is much larger than the total error count estimate by resubstitution (0.1111). The following statements produce Output 25.5.1:

```
proc discrim data=crops
             method=normal pool=no
             crossvalidate;
   class Crop;
   priors prop;
   id xvalues;
   var x1-x4;
   title2 'Using Quadratic Discriminant Function';
run;
```

Output 25.5.1. Quadratic Discriminant Function on Crop Data

```
                  Discriminant Analysis of Remote Sensing Data on Five Crops
                           Using Quadratic Discriminant Function

                                  The DISCRIM Procedure

                   Observations      36          DF Total               35
                   Variables          4          DF Within Classes      31
                   Classes            5          DF Between Classes      4


                                 Class Level Information

              Variable                                                      Prior
 Crop         Name            Frequency       Weight    Proportion    Probability

 Clover       Clover                 11      11.0000      0.305556       0.305556
 Corn         Corn                    7       7.0000      0.194444       0.194444
 Cotton       Cotton                  6       6.0000      0.166667       0.166667
 Soybeans     Soybeans                6       6.0000      0.166667       0.166667
 Sugarbeets   Sugarbeets              6       6.0000      0.166667       0.166667
```

```
                  Discriminant Analysis of Remote Sensing Data on Five Crops
                           Using Quadratic Discriminant Function

                                  The DISCRIM Procedure

                          Within Covariance Matrix Information

                                                   Natural Log of the
                                     Covariance    Determinant of the
                  Crop              Matrix Rank     Covariance Matrix

                  Clover                      4              23.64618
                  Corn                        4              11.13472
                  Cotton                      4              13.23569
                  Soybeans                    4              12.45263
                  Sugarbeets                  4              17.76293
```

```
           Discriminant Analysis of Remote Sensing Data on Five Crops
                      Using Quadratic Discriminant Function

                              The DISCRIM Procedure

              Pairwise Generalized Squared Distances Between Groups

           2         _   _       -1  _   _
          D (i|j) = (X - X )' COV   (X - X ) + ln |COV | - 2 ln PRIOR
                      i   j      j    i   j           j              j


                      Generalized Squared Distance to Crop

From Crop         Clover          Corn        Cotton      Soybeans    Sugarbeets

Clover          26.01743          1320     104.18297     194.10546      31.40816
Corn            27.73809      14.40994     150.50763      38.36252      25.55421
Cotton          26.38544     588.86232      16.81921      52.03266      37.15560
Soybeans        27.07134      46.42131      41.01631      16.03615      23.15920
Sugarbeets      26.80188     332.11563      43.98280     107.95676      21.34645
```

```
                 Discriminant Analysis of Remote Sensing Data on Five Crops
                            Using Quadratic Discriminant Function

                                    The DISCRIM Procedure
                   Classification Summary for Calibration Data: WORK.CROPS
                Resubstitution Summary using Quadratic Discriminant Function

                            Generalized Squared Distance Function

                     2         _       -1   _
                    D (X) = (X-X )' COV  (X-X ) + ln |COV | - 2 ln PRIOR
                     j          j      j     j          j              j

                      Posterior Probability of Membership in Each Crop

                                           2                    2
                        Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                           j        k           k


                  Number of Observations and Percent Classified into Crop

From Crop        Clover         Corn       Cotton      Soybeans      Sugarbeets        Total

Clover                9            0            0             0               2           11
                  81.82         0.00         0.00          0.00           18.18       100.00

Corn                  0            7            0             0               0            7
                   0.00       100.00         0.00          0.00            0.00       100.00

Cotton                0            0            6             0               0            6
                   0.00         0.00       100.00          0.00            0.00       100.00

Soybeans              0            0            0             6               0            6
                   0.00         0.00         0.00        100.00            0.00       100.00

Sugarbeets            0            0            1             1               4            6
                   0.00         0.00        16.67         16.67           66.67       100.00

Total                 9            7            7             7               6           36
                  25.00        19.44        19.44         19.44           16.67       100.00

Priors          0.30556      0.19444      0.16667       0.16667         0.16667


                               Error Count Estimates for Crop

                     Clover        Corn      Cotton    Soybeans    Sugarbeets       Total

   Rate              0.1818      0.0000      0.0000      0.0000        0.3333      0.1111
   Priors            0.3056      0.1944      0.1667      0.1667        0.1667
```

```
                 Discriminant Analysis of Remote Sensing Data on Five Crops
                            Using Quadratic Discriminant Function

                                    The DISCRIM Procedure
                   Classification Summary for Calibration Data: WORK.CROPS
               Cross-validation Summary using Quadratic Discriminant Function

                            Generalized Squared Distance Function

               2         _          -1     _
              D (X) = (X-X    )' COV    (X-X    ) + ln |COV    | - 2 ln PRIOR
               j          (X)j      (X)j    (X)j           (X)j              j

                      Posterior Probability of Membership in Each Crop

                                           2                    2
                        Pr(j|X) = exp(-.5 D (X)) / SUM exp(-.5 D (X))
                                           j        k           k


                  Number of Observations and Percent Classified into Crop

From Crop        Clover         Corn       Cotton      Soybeans      Sugarbeets        Total

Clover                9            0            0             0               2           11
                  81.82         0.00         0.00          0.00           18.18       100.00

Corn                  3            2            0             0               2            7
                  42.86        28.57         0.00          0.00           28.57       100.00

Cotton                3            0            2             0               1            6
                  50.00         0.00        33.33          0.00           16.67       100.00

Soybeans              3            0            0             2               1            6
                  50.00         0.00         0.00         33.33           16.67       100.00

Sugarbeets            3            0            1             1               1            6
                  50.00         0.00        16.67         16.67           16.67       100.00

Total                21            2            3             3               7           36
                  58.33         5.56         8.33          8.33           19.44       100.00

Priors          0.30556      0.19444      0.16667       0.16667         0.16667


                               Error Count Estimates for Crop

                     Clover        Corn      Cotton    Soybeans    Sugarbeets       Total

   Rate              0.1818      0.7143      0.6667      0.6667        0.8333      0.5556
   Priors            0.3056      0.1944      0.1667      0.1667        0.1667
```

References

Anderson, T.W. (1984), *An Introduction to Multivariate Statistical Analysis, Second Edition*, New York: John Wiley & Sons, Inc.

Cover, T.M. and Hart, P.E. (1967), “Nearest Neighbor Pattern Classification,” *IEEE Transactions on Information Theory*, IT-13, 21–27.

Epanechnikov, V.A. (1969), “Nonparametric Estimation of a Multivariate Probability Density,” *Theory of Probability and Its Applications*, 14, 153–158.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Fix, E. and Hodges, J.L., Jr. (1959), “Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties,” *Report No. 4, Project No. 21-49-004*, School of Aviation Medicine, Randolph Air Force Base, TX.

Friedman, J.H., Bentley, J.L., and Finkel, R.A. (1977), “An Algorithm for Finding Best Matches in Logarithmic Expected Time,” *ACM Transactions on Mathematical Software*, 3, 209–226.

Fukunaga, K. and Kessel, D.L. (1973), “Nonparametric Bayes Error Estimation Using Unclassified Samples,” *IEEE Transactions on Information Theory*, 19, 434–440.

Glick, N. (1978), “Additive Estimators for Probabilities of Correct Classification,” *Pattern Recognition*, 10, 211–222.

Hand, D.J. (1981), *Discrimination and Classification*, New York: John Wiley & Sons, Inc.

Hand, D.J. (1982), *Kernel Discriminant Analysis*, New York: Research Studies Press.

Hand, D.J. (1986), “Recent Advances in Error Rate Estimation,” *Pattern Recognition Letters*, 4, 335–346.

Hora, S.C. and Wilcox, J.B. (1982), “Estimation of Error Rates in Several-Population Discriminant Analysis,” *Journal of Marketing Research*, XIX, 57–61.

Kendall, M.G., Stuart, A., and Ord, J.K. (1983), *The Advanced Theory of Statistics, Vol. 3, Fourth Edition*, New York: Macmillan Publishing Co., Inc.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Lachenbruch, P.A. and Mickey, M.A. (1968), “Estimation of Error Rates in Discriminant Analysis,” *Technometrics*, 10, 1–10.

Lawley, D.N. (1959), “Tests of Significance in Canonical Analysis,” *Biometrika*, 46, 59–66.

Morrison, D.F. (1976), *Multivariate Statistical Methods*, New York: McGraw-Hill.

Parzen, E. (1962), “On Estimation of a Probability Density Function and Mode,” *Annals of Mathematical Statistics*, 33, 1065–1076.

Perlman, M.D. (1980), "Unbiasedness of the Likelihood Ratio Tests for Equality of Several Covariance Matrices and Equality of Several Multivariate Normal Populations," *Annals of Statistics*, 8, 247–263.

Rao, C. R. (1973), *Linear Statistical Inference and Its Applications, Second Edition*, New York: John Wiley & Sons, Inc.

Ripley, B.D. (1996), *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press.

Rosenblatt, M. (1956), "Remarks on Some Nonparametric Estimates of a Density Function," *Annals of Mathematical Statistics*, 27, 832–837.

Silverman, B. W. (1986), *Density Estimation for Statistics and Data Analysis*, New York: Chapman and Hall.

Snapinn, S.M. and Knoke, J.D. (1985), "An Evaluation of Smoothed Classification Error-Rate Estimators," *Technometrics*, 27, 199–206.

Chapter 26
The DISTANCE Procedure

Chapter Contents

Chapter 26
The DISTANCE Procedure

Overview

The DISTANCE procedure computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. These proximity measures are stored as a lower triangular matrix or a square matrix in an output data set (depending on the SHAPE= option) that can then be used as input to the CLUSTER, MDS, and MODECLUS procedures. The input data set may contain numeric or character variables, or both, depending on which proximity measure is used.

The number of rows and columns in the output matrix equals the number of observations in the input data set. If there are BY groups, an output matrix is computed for each BY group with the size determined by the maximum number of observations in any BY group.

PROC DISTANCE also provides various non-parametric and parametric methods for standardizing variables. Different variables can be standardized with different methods.

Distance matrices are used frequently in data mining, genomics, marketing, financial analysis, management science, education, chemistry, psychology, biology, and various other fields.

Levels of Measurement

Measurement of some attribute of a set of objects is the process of assigning numbers or other symbols to the objects in such a way that properties of the numbers or symbols reflect properties of the attribute being measured. There are different *levels* of measurement that involve different properties (relations and operations) of the numbers or symbols. Associated with each level of measurement is a set of transformations of the measurements that preserve the relevant properties; these transformations are called *permissible* transformations. A particular way of assigning numbers or symbols to measure something is called a *scale* of measurement.

The most commonly discussed levels of measurement are:

Nominal — Two objects are assigned the same symbol if they have the same value of the attribute. Permissible transformations are any one-to-one or many-to-one transformation, although a many-to-one transformation loses information.

Ordinal — Objects are assigned numbers such that the order of the numbers reflects an order relation defined on the attribute. Two objects *x* and *y* with attribute values *a(x)* and *a(y)* are assigned numbers *m(x)* and *m(y)* such that if $m(x) > m(y)$, then $a(x) > a(y)$. Permissible transformations are any monotone increasing transformation, although a transformation that is not strictly increasing loses information.

Interval Objects are assigned numbers such that differences between the numbers reflect differences of the attribute. If $m(x) - m(y) > m(u) - m(v)$, then $a(x) - a(y) > a(u) - a(v)$. Permissible transformations are any affine transformation $t(m) = c * m + d$, where *c* and *d* are constants; another way of saying this is that the origin and unit of measurement are arbitrary.

Log-interval Objects are assigned numbers such that ratios between the numbers reflect ratios of the attribute. If $m(x)/m(y) > m(u)/m(v)$, then $a(x)/a(y) > a(u)/a(v)$. Permissible transformations are any power transformation $t(m) = c * m^d$, where *c* and *d* are constants.

Ratio Objects are assigned numbers such that differences and ratios between the numbers reflect differences and ratios of the attribute. Permissible transformations are any linear (similarity) transformation $t(m) = c * m$, where *c* is a constant; another way of saying this is that the unit of measurement is arbitrary.

Absolute Objects are assigned numbers such that all properties of the numbers reflect analogous properties of the attribute. The only permissible transformation is the identity transformation.

Proximity measures provided in the DISTANCE procedure accept four levels of measurement: nominal, ordinal, interval, and ratio. Ordinal variables are transformed to interval variables before processing. This is done by replacing the data with their rank scores, and by assuming that the classes of an ordinal variable are spaced equally along the interval scale. See the RANKSCORE= option in the PROC DISTANCE statement for choices on assigning scores to ordinal variables. There are also different approaches on how to transform an ordinal variable to an interval variable. Refer to Anderberg (1973) for alternatives.

Symmetric versus Asymmetric Nominal Variables

A binary variable contains two possible outcomes: 1 (positive/present) or 0 (negative/absent). If there is no preference for which outcome should be coded as 0 and which as 1, the binary variable is called *symmetric*. For example, the binary variable “is evergreen?” for a plant has the possible states “loses leaves in winter” and “does not lose leaves in winter.” Both are equally valuable and carry the same weight when a proximity measure is computed. Commonly used measures that accept symmetric binary variables include the Simple Matching, Hamann, Roger and Tanimoto, Sokal and Sneath 1, and Sokal and Sneath 3 coefficients.

If the outcomes of a binary variable are not equally important, the binary variable is called *asymmetric*. An example of such a variable is the presence or absence of a relatively rare attribute, such as “is color blind” for a human-being. While you say that two people who are color blind have something in common, you cannot say that people who are not color blind have something in common. The most important outcome is usually coded as 1 (present) and the other is coded as 0 (absent). The agreement of two 1's (a present-present match or a positive match) is more significant than the agreement of two 0's (an absent-absent match or a negative match.) Usually, the negative match is treated as irrelevant. Commonly used measures that accept asymmetric binary variables include Jaccard, Dice, Russell and Rao, Binary Lance and Williams nonmetric, and Kulcynski coefficients.

When nominal variables are employed, the comparison of one data unit with another can only be in terms of whether the data units score the same or different on the variables. If a variable is defined as an asymmetric nominal variable and two data units score the same but fall into the absent category, the absent-absent match is excluded from the computation of the proximity measure.

Standardization

Since variables with large variances tend to have more effect on the proximity measure than those with small variances, it is recommended to standardize the variables before the computation of the proximity measure. The DISTANCE procedure provides a convenient way to standardize each variable with its own method before the proximity measures are computed. The standardization can also be performed by the STDIZE procedure with the limitation that all variables must be standardized with the same method.

Mandatory Standardization

Variable standardization is not required if there is only one level of measurement, or if only asymmetric nominal and nominal levels are specified; otherwise, standardization is mandatory.

When standardization is mandatory and no standardization method is specified, a default method of standardization will be used. This default method is determined by the measurement level. In general, the default method is STD for interval variables and is MAXABS for ratio variables except when METHOD= GOWER or METHOD= DGOWER is specified. See the STD= option in the VAR statement for the default methods for GOWER and DGOWER as well as methods available for standardizing variables.

When standardization is mandatory, PROC DISTANCE suppresses the REPONLY option, if it is specified.

Getting Started

Creating a Distance Matrix as Input for a Subsequent Cluster Analysis

The following example demonstrates how you can use the DISTANCE procedure to obtain a distance matrix that will be used as input to a subsequent clustering procedure.

The following data, originated by A. Weber and cited in Hand et al. (1994, pp. 297), measure the amount of protein consumed for nine food groups in 25 European countries. The nine food groups are red meat (RedMeat), white meat (WhiteMeat), eggs (Eggs), milk (Milk), fish (Fish), cereal (Cereal), starch (Starch), nuts (Nuts), and fruits and vegetables (FruitVeg). Suppose you want to determine whether national figures in protein consumption can be used to determine certain types or categories of countries; specifically, you want to perform a cluster analysis to determine if these 25 countries can be formed into groups suggested by the data.

The following DATA step creates the SAS data set Protein:

```
title 'Protein Consumption in Europe';
data Protein;
input Country $14. RedMeat WhiteMeat Eggs Milk
                   Fish Cereal Starch Nuts FruitVeg;
datalines;
Albania        10.1  1.4  0.5   8.9  0.2  42.3  0.6  5.5  1.7
Austria         8.9 14.0  4.3  19.9  2.1  28.0  3.6  1.3  4.3
Belgium        13.5  9.3  4.1  17.5  4.5  26.6  5.7  2.1  4.0
Bulgaria        7.8  6.0  1.6   8.3  1.2  56.7  1.1  3.7  4.2
Czechoslovakia  9.7 11.4  2.8  12.5  2.0  34.3  5.0  1.1  4.0
Denmark        10.6 10.8  3.7  25.0  9.9  21.9  4.8  0.7  2.4
E Germany       8.4 11.6  3.7  11.1  5.4  24.6  6.5  0.8  3.6
Finland         9.5  4.9  2.7  33.7  5.8  26.3  5.1  1.0  1.4
France         18.0  9.9  3.3  19.5  5.7  28.1  4.8  2.4  6.5
Greece         10.2  3.0  2.8  17.6  5.9  41.7  2.2  7.8  6.5
Hungary         5.3 12.4  2.9   9.7  0.3  40.1  4.0  5.4  4.2
Ireland        13.9 10.0  4.7  25.8  2.2  24.0  6.2  1.6  2.9
Italy           9.0  5.1  2.9  13.7  3.4  36.8  2.1  4.3  6.7
Netherlands     9.5 13.6  3.6  23.4  2.5  22.4  4.2  1.8  3.7
Norway          9.4  4.7  2.7  23.3  9.7  23.0  4.6  1.6  2.7
Poland          6.9 10.2  2.7  19.3  3.0  36.1  5.9  2.0  6.6
Portugal        6.2  3.7  1.1   4.9 14.2  27.0  5.9  4.7  7.9
Romania         6.2  6.3  1.5  11.1  1.0  49.6  3.1  5.3  2.8
Spain           7.1  3.4  3.1   8.6  7.0  29.2  5.7  5.9  7.2
Sweden          9.9  7.8  3.5   4.7  7.5  19.5  3.7  1.4  2.0
Switzerland    13.1 10.1  3.1  23.8  2.3  25.6  2.8  2.4  4.9
UK             17.4  5.7  4.7  20.6  4.3  24.3  4.7  3.4  3.3
USSR            9.3  4.6  2.1  16.6  3.0  43.6  6.4  3.4  2.9
W Germany      11.4 12.5  4.1  18.8  3.4  18.6  5.2  1.5  3.8
Yugoslavia      4.4  5.0  1.2   9.5  0.6  55.9  3.0  5.7  3.2
;
```

The data set Protein contains the character variable Country and the nine numeric variables representing the food groups. The `$14.` in the INPUT statement specifies that the variable Country has a length of 14.

The following statements create the distance matrix and display part of it.

```
proc distance data=Protein out=Dist method=Euclid;
   var interval(RedMeat--FruitVeg / std=Std);
   id Country;
```

```
run;

options ls=120;
proc print data=Dist(Obs=10);
title2 'First 10 observations in the output data set from PROC DISTANCE';
run;
```

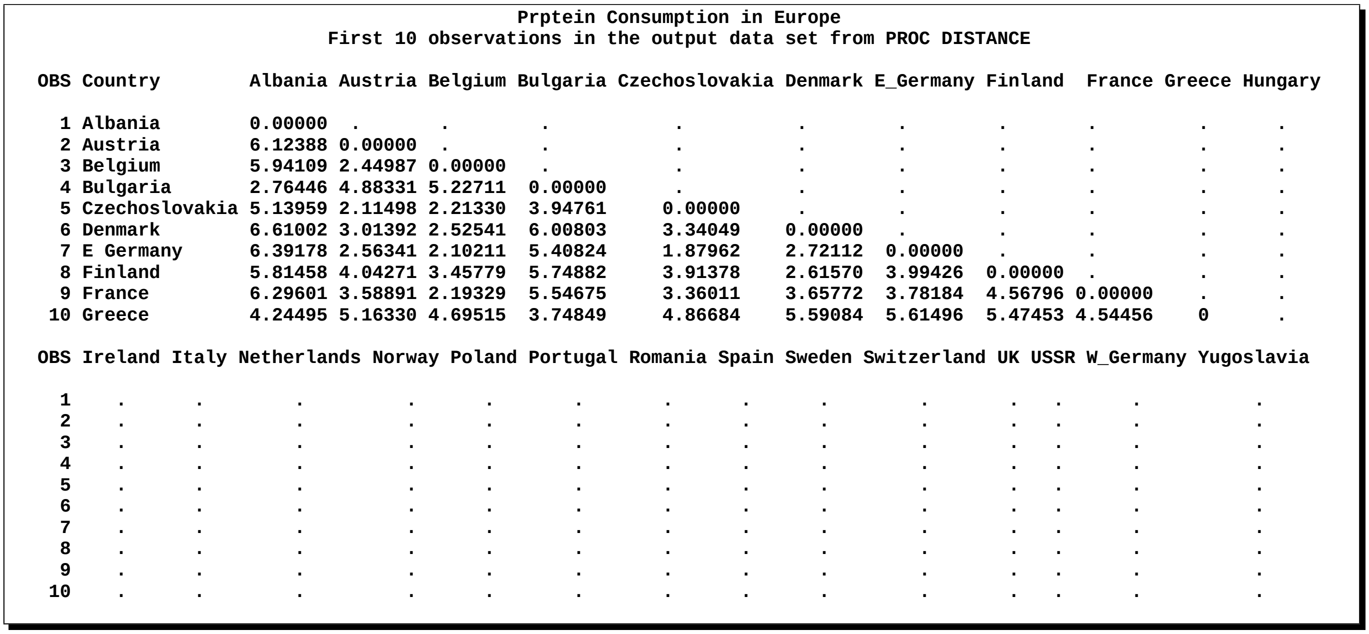
An output SAS data set called Dist that contains the distance matrix is created through the OUT= option. METHOD= Euclid requests that Euclidean (which also is the default) distances should be computed.

The VAR statement lists the variables (RedMeat—FruitVeg) along with their measurement level to be used in the analysis. An interval level of measurement is assigned to those variables. Since variables with large variances tend to have more effect on the proximity measure than those with small variances, each variable is standardized by the STD method to have a mean of 0 and a standard deviation of 1. This is done by adding "/" with the STD= Std option at the end of the variables list specification.

The ID statement specifies that the variable Country should be copied to the OUT= data set and used to generate names for the distance variables. The distance variables in the output data set are named by the values in the ID variable, and the maximum length for the names of these variables is 14.

There are 25 observations in the input data set; therefore, the output data set Dist contains a 25 by 25 lower triangle matrix.

The PROC PRINT statement displays the first 10 observations in the output data set Dist as shown in Figure 26.1.



```
                                           Prptein Consumption in Europe
                          First 10 observations in the output data set from PROC DISTANCE

OBS Country        Albania Austria Belgium Bulgaria Czechoslovakia Denmark E_Germany Finland  France Greece Hungary

  1 Albania        0.00000   .       .        .           .          .        .        .       .         .      .
  2 Austria        6.12388 0.00000   .        .           .          .        .        .       .         .      .
  3 Belgium        5.94109 2.44987 0.00000    .           .          .        .        .       .         .      .
  4 Bulgaria       2.76446 4.88331 5.22711  0.00000       .          .        .        .       .         .      .
  5 Czechoslovakia 5.13959 2.11498 2.21330  3.94761     0.00000      .        .        .       .         .      .
  6 Denmark        6.61002 3.01392 2.52541  6.00803     3.34049    0.00000    .        .       .         .      .
  7 E Germany      6.39178 2.56341 2.10211  5.40824     1.87962    2.72112  0.00000    .       .         .      .
  8 Finland        5.81458 4.04271 3.45779  5.74882     3.91378    2.61570  3.99426  0.00000   .         .      .
  9 France         6.29601 3.58891 2.19329  5.54675     3.36011    3.65772  3.78184  4.56796 0.00000     .      .
 10 Greece         4.24495 5.16330 4.69515  3.74849     4.86684    5.59084  5.61496  5.47453 4.54456     0      .

OBS Ireland Italy Netherlands Norway Poland Portugal Romania Spain Sweden Switzerland UK USSR W_Germany Yugoslavia

  1     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  2     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  3     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  4     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  5     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  6     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  7     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  8     .      .        .         .      .       .       .      .      .       .       .   .      .          .
  9     .      .        .         .      .       .       .      .      .       .       .   .      .          .
 10     .      .        .         .      .       .       .      .      .       .       .   .      .          .
```

Figure 26.1. First 10 Observations in the Output Data Set from PROC DISTANCE

The following statements produce the tree diagram in Figure 26.2:

```
proc cluster data=Dist method=Ward outtree=Tree noprint;
   id Country;
run;

axis1 order=(0 to 1 by 0.1);
proc tree data=Tree haxis=axis1 horizontal;
   height _rsq_;
   id Country;
run;
```

The CLUSTER procedure performs a Ward's Minimum-Variance cluster analysis based on the distance matrix created by the PROC DISTANCE. The printed output has been omitted, but the output data set Tree is created (through outtree= Tree) and used as input to the TREE procedure that produces the tree diagram as shown in Figure 26.2. The height statement specifies the variable _RSQ_ (R^2) as the height variable.

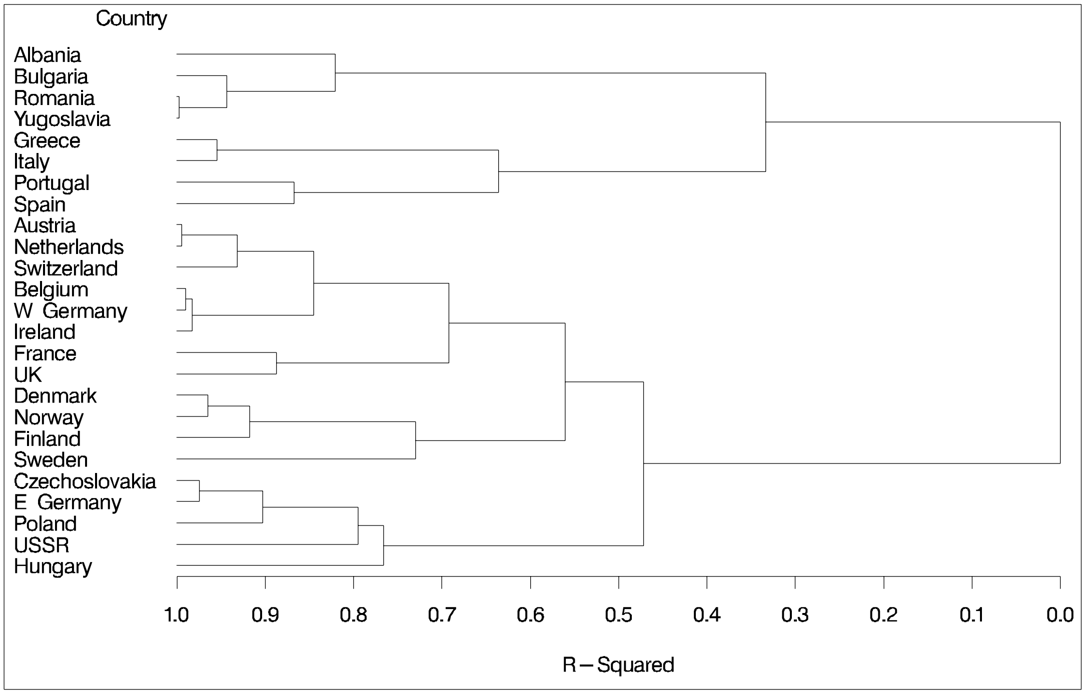


Figure 26.2. Tree Diagram of Clusters versus R-Square Values

After inspecting the tree diagram in Figure 26.2, you will see that when the countries are grouped into six clusters, the proportion of variance accounted for by these clusters is slightly less than 70% (69.3%).

Syntax

The following statements are available in the DISTANCE procedure.

PROC DISTANCE < *options* > **;**
  BY *variables* **;**
  COPY *variables* **;**
  FREQ *variable* **;**
  ID *variable* **;**
  VAR *level(variables* < */ opt-list* >*)* **;**
  WEIGHT *variable* **;**

Both the PROC DISTANCE statement and the VAR statement are required.

PROC DISTANCE Statement

PROC DISTANCE < *options* >

You can specify the following options in the PROC DISTANCE statement.

Table 26.1. Summary of PROC DISTANCE Statement Options

Task/Statement	Options	Description
standardizing variables	ADD=	specifies the constant to add to each value after standardizing and multiplying by the value specified in the MULT= option
	FUZZ=	specifies the relative fuzz factor for writing the output
	INITIAL=	specifies the method for computing initial estimates for the A-estimates
	MULT=	specifies the constant to multiply each value by after standardizing
	NORM	normalizes the scale estimator to be consistent for the standard deviation of a normal distribution
	SNORM	normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution
	VARDEF=	specifies the variances divisor
generating distance matrix	ABSENT=	specifies the value to be used as an absence value for all the asymmetric nominal variables
	METHOD=	specifies the method for computing proximity measures
	PREFIX=	specifies a prefix for naming the distance variables in the OUT= data set
	RANKSCORE=	specifies the method of assigning scores to ordinal variables

Table 26.1. (continued)

Task	Options	Description
	SHAPE=	specifies the shape of the proximity matrix to be stored in the OUT= data set
	UNDEF=	specifies the numeric constant used to replace undefined distances
	VARDEF=	specifies the variances divisor
missing values	NOMISS	replaces missing data by the location measure (does not standardize the data); generates missing distance for observations with missing values
	REPLACE	replaces missing data by zero in the standardized data
	REPONLY	replaces missing data by the location measure (does not standardize the data)
specifying data set details	DATA=	specifies the input data set
	OUT=	specifies the output data set
	OUTSDZ=	specifies the output data set for standardized scores

These options and their abbreviations are described, in alphabetical order, in the remainder of this section.

ABSENT= *num* **or** *qs*

specifies the value to be used as an absence value in an irrelevant absent-absent match for *all* of the asymmetric nominal variables. If you want to specify a different absence value for a particular variable, use the ABSENT= option in the VAR statement. See the ABSENT= option in the VAR statement later in this chapter for details.

An absence value for a variable can be either a numeric value or a quoted string consisting of combinations of characters. For instance, ., -999, “NA” are legal values for the ABSENT= option.

The default absence value for a character variable is “NONE” (notice that a blank value is considered a missing value), and the default absence value for a numeric variable is 0.

ADD= *c*

specifies a constant, *c*, to add to each value after standardizing and multiplying by the value you specify in the MULT= option. The default value is 0.

DATA= *SAS-data-set*

specifies the input data set containing observations from which the proximity is computed. If you omit the DATA= option, the most recently created SAS data set is used.

FUZZ= *c*

specifies the relative fuzz factor for computing the standardized scores. The default value is 1E-14. For the OUTSDZ= data set, the score is computed as follows:

$$\text{if } |\text{standardized scores}| < \text{scale measure} \times c, \text{ then standardized scores} = 0$$

INITIAL= *method*

specifies the method for computing initial estimates for the A-estimates (ABW, AWAVE, and AHUBER). The following methods are not allowed for the INITIAL= option: ABW, AHUBER, AWAVE, IN.

The default value is INITIAL=MAD.

METHOD= *method*

specifies the method for computing proximity measures.

For use in PROC CLUSTER, distance or dissimilarity measures such as METHOD= EUCLID or METHOD= DGOWER should be chosen.

The following six tables outline the proximity measures available for the METHOD= option. These tables are classified by levels of measurement accepted by each method. There are three to four columns in each table: the proximity measures (Method) column, the upper and lower bounds (Range) column(s), and the types of proximity (Type) column.

The Type column has two possible values: “sim” if a method generate similarity or “dis” if a method generates distance or dissimilarity measures.

For formulas and descriptions of these methods, see the “Details” section on page 1270.

Table 26.2 lists the GOWER and the DGOWER methods. These two methods accept all measurement levels including ratio, interval, ordinal, nominal, and asymmetric nominal. METHOD= GOWER or METHOD= DGOWER always implies standardization. Assuming all the numeric (ordinal, interval, and ratio) variables are standardized by their corresponding default methods, the possible range values for both methods in the second column of this table are on or between 0 and 1. To find out the default methods of standardization for METHOD= GOWER or METHOD= DGOWER, see the STD= option for the VAR statement later in this section. Entries in this table are:

GOWER	Gower’s similarity
DGOWER	1 minus GOWER

Table 26.2. Methods Accepting all Measurement Levels

Method	Range	Type
GOWER	0 to 1	sim
DGOWER	0 to 1	dis

Table 26.3 lists methods accepting ratio, interval, and ordinal variables. Entries in this table are:

EUCLID	Euclidean distance
SQEUCLID	Squared Euclidean distance
SIZE	Size distance
SHAPE	Shape distance
COV	Covariance
CORR	Correlation
DCORR	Correlation transformed to Euclidean distance
SQCORR	Squared correlation
DSQCORR	One minus squared correlation
L(*p*)	Minkowski (L_p) distance, where *p* is a positive numeric value
CITYBLOCK	L_1, City-block, or Manhattan distance
CHEBYCHEV	L_∞
POWER(p, r)	Generalized Euclidean distance where *p* is a positive numeric value, and *r* is a non-negative numeric value. The distance between two observations is the *r*th root of sum of the absolute differences to the *p*th power between the values for the observations.

Table 26.3. Methods Accepting Ratio, Interval, and Ordinal Variables

Method	Range	Type
EUCLID	≥ 0	dis
SQEUCLID	≥ 0	dis
SIZE	≥ 0	dis
SHAPE	≥ 0	dis
COV	≥ 0	sim
CORR	-1 to 1	sim
DCORR	0 to 2	dis
SQCORR	0 to 1	sim
DSQCORR	0 to 1	dis
L(*p*)	≥ 0	dis
CITYBLOCK	≥ 0	dis
CHEBYCHEV	≥ 0	dis
POWER(p, r)	≥ 0	dis

Table 26.4 lists methods accepting ratio variables. Notice that in the second column of this table, all of the possible range values are non-negative, because ratio variables are assumed to be positive. Entries in this table are:

SIMRATIO	Similarity ratio (if variables are binary, this is the Jaccard coefficient)
DISRATIO	One minus similarity ratio
NONMETRIC	Lance and Williams nonmetric coefficient
CANBERRA	Canberra metric distance coefficient
COSINE	Cosine
DOT	Dot (inner) product
OVERLAP	Overlap similarity
DOVERLAP	Overlap dissimilarity
CHISQ	Chi-squared
CHI	Squared root of Chi-squared
PHISQ	phi-squared
PHI	Squared root of phi-squared

Table 26.4. Methods Accepting Ratio Variables

Method	**Range**	**Type**
SIMRATIO	0 to 1	sim
DISRATIO	0 to 1	dis
NONMETRIC	0 to 1	dis
CANBERRA	0 to 1	dis
COSINE	0 to 1	sim
DOT	≥ 0	sim
OVERLAP	≥ 0	sim
DOVERLAP	≥ 0	dis
CHISQ	≥ 0	dis
CHI	≥ 0	dis
PHISQ	≥ 0	dis
PHI	≥ 0	dis

Table 26.5 lists methods accepting nominal variables. Entries in the previous table are:

HAMMING	Hamming distance
MATCH	Simple matching coefficient
DMATCH	Simple matching coefficient transformed to Euclidean distance
DSQMATCH	Simple matching coefficient transformed to squared Euclidean distance
HAMANN	Hamann coefficient
RT	Roger and Tanimoto

SS1	Sokal and Sneath 1
SS3	Sokal and Sneath 3

Table 26.5. Methods Accepting Nominal Variables

Method	**Range**	**Type**
HAMMING	0 to v^*	dis
MATCH	0 to 1	sim
DMATCH	0 to 1	dis
DSQMATCH	0 to 1	dis
HAMANN	-1 to 1	sim
RT	0 to 1	sim
SS1	0 to 1	sim
SS3	0 to 1	sim

[1] the number of variables or dimensionality.

Table 26.6 lists methods that accept asymmetric nominal variables. Use the ABSENT= option to create a value to be considered absent. Entries in the previous table are:

DICE	Dice coefficient or Czekanowski/Sorensen similarity coefficient
RR	Russell and Rao
BLWNM	Binary Lance and Williams nonmetric, or Bray-Curtis coefficient
K1	Kulcynski 1

Table 26.6. Methods Accepting Asymmetric Nominal Variables

Method	**Range**	**Type**
DICE	0 to 1	sim
RR	0 to 1	sim
BLWNM	0 to 1	dis
K1	≥ 0	sim

Table 26.7 lists methods accepting asymmetric nominal and ratio variables. Use the ABSENT= option to create a value to be considered absent. There are four instead of three columns in this table. The second column contains possible range values if only one level of measurement (either ratio or asymmetric nominal but not both) is specified; the third column contains possible range values if both levels are specified.

The JACCARD method is equivalent to the SIMRATIO method if there is no asymmetric nominal variable; if both ratio and asymmetric nominal variables are present, the coefficient is computed as the sum of the coefficient from the ratio variables and the coefficient from the asymmetric nominal variables. See “Proximity Measures” of the “Details” section on page 1270 for formulas and descriptions of the JACCARD method. Entries in this table are:

JACCARD	Jaccard similarity coefficient
DJACCARD	Jaccard dissimilarity coefficient

Table 26.7. Methods Accepting Asymmetric Nominal and Ratio Variables

Method	Range (one level)	Range (two levels)	Type
JACCARD	0 to 1	0 to 2	sim
DJACCARD	0 to 1	0 to 2	dis

MULT= *c*

specifies a constant, *c*, by which to multiply each value after standardizing. The default value is 1.

NOMISS

While standardizing variables, omit observations with missing values from computation of the location and scale measures. While computing distances, generate undefined (missing) distances for observations with missing values. Use the UNDEF= option to specify the undefined values.

If a distance matrix is created to be used as an input to PROC CLUSTER, the NOMISS option should not be used because the CLUSTER procedure will not accept distance matrices with missing values.

NORM

normalizes the scale estimator to be consistent for the standard deviation of a normal distribution when you specify the option STD= AGK, STD= IQR, STD= MAD, or STD= SPACING in the VAR statement.

PREFIX= *name*

specifies a prefix for naming the distance variables in the OUT= data set. By default, the names are Dist1, Dist2, . . . , Dist$n$. If you specify PREFIX=ABC, the variables are named ABC1, ABC2, ..., ABCn. If the ID statement is also specified, the variables are named by appending the value of the ID variable to the prefix.

OUT= *SAS-data-set*

specifies the name of the SAS data set created by PROC DISTANCE. The output data set contains the BY variables, the ID variable, computed distance variables, the COPY variables, the FREQ variable, and the WEIGHT variables.

If you omit the OUT= option, PROC DISTANCE creates an output data set named according to the DATAn convention.

OUTSDZ= *SAS-data-set*

specifies the name of the SAS data set containing the standardized scores. The output data set contains a copy of the DATA= data set, except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement.

RANKSCORE= MIDRANK | INDEX

specifies the method of assigning scores to ordinal variables. The available methods are listed as follows:

MIDRANK	assigns consecutive integers to each category with consideration of the frequency value. This is the default method.
INDEX	assigns consecutive integers to each category regardless of frequencies.

The following example explains how each method assigns the rank scores. Suppose the data contain an ordinal variable ABC with values A, B, C. There are two ways to assign numbers. One is to use midranks, which depend on the frequencies of each category. Another is to assign consecutive integers to each category, regardless of frequencies.

Table 26.8. Example of Assigning Rank Scores

ABC	MIDRANK	INDEX
A	1.5	1
A	1.5	1
B	4	2
B	4	2
B	4	2
C	6	3

REPLACE

replaces missing data by zero in the standardized data (which corresponds to the location measure before standardizing.) To replace missing data by something else, use the MISSING= option in the VAR statement. The REPLACE option implies standardization.

You can not specify both the REPLACE and the REPONLY options.

REPONLY

replaces missing data by the location measure specified by the MISSING= option or the STD= option (if the MISSING= option is not specified), but does *not* standardize the data. If the MISSING= option is not specified and METHOD= GOWER is specified, missing values are replaced by the location measure from the RANGE method (the minimum value), no matter what the value of the STD= option is.

When standardization is mandatory, PROC DISTANCE suppresses the REPONLY option.

You can not specify both the REPLACE and the REPONLY options.

SHAPE= TRIANGLE | TRI
SHAPE= SQUARE | SQU | SQR

specifies the shape of the proximity matrix to be stored in the OUT= data set. SHAPE= TRIANGLE requests the matrix to be stored as a lower triangular matrix; SHAPE= SQUARE requests the matrix to be stored as a squared matrix. Use SHAPE= SQUARE if the output data set is to be used as input to the MODECLUS procedures. The default is TRIANGLE.

SNORM

normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution when the STD= SPACING option is specified.

UNDEF= *n*

specifies the numeric constant used to replace undefined distances, for example, when an observation has all missing values, or if a divisor is zero.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of distance, dissimilarity, or similarity measures, and for standardizing variables whenever a variance or covariance is computed. By default, VARDEF=DF. The values and associated divisors are as follows:

Value	**Divisor**	**Formula**
DF	degrees of freedom	$n - 1$
N	number of observations	n
WDF	sum of weights minus 1	$(\sum_i w_i) - 1$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

VAR Statement

VAR | **VARIABLES** *level (variables < / opt-list >)*
< *level (variables < / opt-list >)*
level (variables < / opt-list >)
.
.
.
level (variables < / opt-list >) >

where the syntax for the *opt-list* is:

< ABSENT **=** value >
< MISSING **=** miss-method or value>
< ORDER **=** order-option >
< STD **=** std-method >
< WEIGHTS **=** weight-list >

The VAR statement lists variables from which distances are to be computed. The VAR statement is required. The variables can be numeric or character depending on their measurement levels. A variable may not appear more than once in either the same list or in a different list.

level is required. It declares the levels of measurement for those variables specified within the parentheses. Available values for *level* are:

ANOMINAL variables are asymmetric nominal and can be either numeric or character.

NOMINAL variables are symmetric nominal and can be either numeric or character.

ORDINAL variables are ordinal and can be either numeric or character. Values of ordinal variables will be replaced by their corresponding rank scores. If standardization is required, the standardized rank scores are output to the data set specified in the OUTSDZ= option.

See the RANKSCORE= option in the PROC DISTANCE statement for methods available for assigning rank scores to ordinal variables. After being replaced by scores, ordinal variables are considered interval.

INTERVAL variables are interval, and only numeric variables are allowed.

RATIO variables are ratio, and only numeric variables are allowed. Ratio variables should always contain positive measurements.

Each variable list can be followed by an option list. Use "/ " after the list of variables to start the option list. An option list contains options that are applied to the variables. The following options are available in the option list.

ABSENT= to specify the value to be used as an absence value in an irrelevant absent-absent match for asymmetric nominal variables.

MISSING= to specify the method (or value) with which to replace missing data

ORDER= to select the order for assigning scores to ordinal variables.

STD= to select the standardization method

WEIGHTS= to assign weights to the variables in the list

If an option is missing from the current attribute list, PROC DISTANCE provides default values for all the variables in the current list.

For example, in the following VAR statement:

```
var ratio(x1-x4/std= mad weights= .5 .5 .1 .5 missing= -99)
    interval(x5/std= range)
    ordinal(x6/order= desc);
```

the first option list defines x1 –x4 as ratio variables to be standardized by the MAD method. Also, any missing values in x1 –x4 should be replaced by -99. x1 is given a weight of 0.5, x2 is given a weight of 0.5, x3 is given a weight of 0.1, and x4 is given a weight of 0.5.

The second option list defines x5 as an interval variable to be standardized by the RANGE method. If the REPLACE option is specified in the PROC DISTANCE statement, missing values in x5 are replaced by the the location estimate from the RANGE method. By default, x5 is given a weight of 1.

The last option list defines x6 as an ordinal variable. The scores are assigned from highest-to-lowest by its unformatted values. Although the STD= option is not specified, x6 will be standardized by the default method (STD) because there is more than one level of measurements (ratio, interval, and ordinal) in the VAR statement. Again, if the REPLACE option is specified, missing values in x6 are replaced by the location estimate from the STD method. Finally, by default, x6 is given a weight of 1.

More details for the options are explained as follows.

STD= *std-method*

specifies the standardization method. Valid values for *std-method* are: MEAN, MEDIAN, SUM, EUCLEN, USTD, STD, RANGE, MIDRANGE, MAXABS, IQR, MAD, ABW, AHUBER, AWAVE, AGK, SPACING, and L. Table 26.9 lists available methods of standardization as well as their corresponding location and scale measures.

Table 26.9. Available Standardization Methods

Method	**Scale**	**Location**
MEAN	1	mean
MEDIAN	1	median
SUM	sum	0
EUCLEN	Euclidean length	0
USTD	standard deviation about origin	0
STD	standard deviation	mean
RANGE	range	minimum
MIDRANGE	range/2	midrange
MAXABS	maximum absolute value	0
IQR	interval quartile range	median
MAD	median abs. dev. from median	median
ABW(c)	biweight A-estimate	biweight 1-step M-estimate
AHUBER(c)	Huber A-estimate	Huber 1-step M-estimate
AWAVE(c)	Wave 1-step M-estimate	Wave A-estimate
AGK(p)	AGK estimate (ACECLUS)	mean
SPACING(p)	minimum spacing	mid minimum-spacing
L(p)	L_p	L_p

These standardization methods are further documented in the section on the METHOD= option in the PROC STDIZE statement of the STDIZE procedure (also see the "Standardization Methods" section on page 4136 in Chapter 66, "The STDIZE Procedure," .)

Standardization is not required if there is only one level of measurement, or if only asymmetric nominal and nominal levels are specified; otherwise, standardization is mandatory. When standardization is mandatory, a default method will be provided when the STD= option is not given. The default method is STD for standardizing interval variables and MAXABS for standardizing ratio variables unless METHOD=

GOWER or METHOD= DGOWER is specified. If METHOD= GOWER is specified, interval variables are standardized by the RANGE method, and whatever is specified in the STD= option is ignored; if METHOD= DGOWER is specified, the RANGE method is the default standardization method for interval variables. The MAXABS method is the default standardization method for ratio variables for both the GOWER and the DGOWER.

Notice that a ratio variable should always be positive.

Table 26.10 lists standardization methods and the levels of measurement that can be accepted by each method. For example, the SUM method can be used to standardize ratio variables but not interval, or ordinal variables. Also, the AGK and SPACING methods should not be used to standardize ordinal variables. If you apply AGK and SPACING to ranks, the results are degenerate because all the spacings of a given order are equal.

Table 26.10. Legitimate Levels of Measurements for Each Method

Standardization Method	**Legitimate Levels of Measurement**
MEAN	ratio, interval, ordinal
MEDIAN	ratio, interval, ordinal
SUM	ratio
EUCLEN	ratio
USTD	ratio
STD	ratio, interval, ordinal
RANGE	ratio, interval, ordinal
MIDRANGE	ratio, interval, ordinal
MAXABS	ratio
IQR	ratio, interval, ordinal
MAD	ratio, interval, ordinal
ABW(c)	ratio, interval, ordinal
AHUBER(c)	ratio, interval, ordinal
AWAVE(c)	ratio, interval, ordinal
AGK(p)	ratio, interval
SPACING(p)	ratio, interval
L(p)	ratio, interval, ordinal

ABSENT= *num* **or** *qs*

specifies the value to be used as an absence value in an irrelevant absent-absent match for asymmetric nominal variables. The absence value specified here overwrites the absence value specified through the ABSENT= option in the PROC DISTANCE statement for those variables in the current variable list.

An absence value for a variable can be either a numeric value or a quoted string consisting of combinations of characters. For instance, ., -999, “NA” are legal values for the ABSENT= option.

The default for an absence value for a character variable is “NONE” (notice that a blank value is considered a missing value), and the default for an absence value for a numeric variable is 0.

MISSING= *miss-method* **or** *value*

specifies the method or a numeric value for replacing missing values. If you omit the MISSING= option, the REPLACE option replaces missing values with the location measure given by the STD= option. Specify the MISSING= option when you want to replace missing values with a different value. You can specify any method that is valid in the STD= option. The corresponding location measure is used to replace missing values.

If a numeric value is given, the value replaces missing values after standardizing the data. However, when standardization is not mandatory, you can specify the REPONLY option with the MISSING= option to suppress standardization for cases in which you want only to replace missing values.

ORDER= ASCENDING | ASC
ORDER= DESCENDING | DESC
ORDER= ASCFORMATTED | ASCFMT
ORDER= DESFORMATTED | DESFMT
ORDER= DSORDER | DATA

specifies the order for assigning score to ordinal variables. The value for the ORDER= option can be one of the following:

ASCENDING scores are assigned in lowest-to-highest order of unformatted values.

DESCENDING scores are assigned in highest-to-lowest order of unformatted values.

ASCFORMATTED scores are assigned in ascending order by their formatted values. This option can be applied to character variables only, since unformatted values are always used for numeric variables.

DESFORMATTED scores are assigned in descending order by their formatted values. This option can be applied to character variables only, since unformatted values are always used for numeric variables.

DSORDER scores are assigned according to the order of their appearance in the input data set.

The default value is ASCENDING.

WEIGHTS= *weight-list*

specifies a list of values for weighting individual variables while computing the proximity. Values in this list can be separated by blanks or commas. You can include one or more items of the form *start* TO *stop* BY *increment*. This list should contain at least one weight. The maximum number of weights you can list is equal to the number of variables. If the number of weights is less than the number of variables, the last value in the *weight-list* is used for the rest of the variables; conversely, if the number of weights is greater than the number of variables, the trailing weights will be discarded.

The default value is 1.

ID Statement

ID *variable*

The ID statement specifies a single variable to be copied to the OUT= data set and used to generate names for the distance variables. The ID variable must be character.

Typically, each ID value occurs only once in the input data set or, if you use a BY statement, only once within a BY group.

If you specify both the ID and the BY statements, the ID variable must have the same values in the same order in each BY group.

COPY Statement

COPY *variables*

The COPY statement specifies a list of additional variables to be copied to the OUT= data set.

BY Statement

BY *variables*

You can specify a BY statement to obtain separate distance matrices for observations in groups defined by the BY variables.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*.

FREQ Statement

FREQ | **FREQUENCY** *variable*

The frequency variable is used for either standardizing variables or assigning rank scores to the ordinal variables. It has no direct effect on computing the distances.

For standardizing variables and assigning rank scores, PROC DISTANCE treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Non-integral values of the FREQ variable are truncated to the largest integer less than the FREQ value. If the FREQ variable has a value that is less than 1 or is missing, the observation is not used in the analysis.

WEIGHT Statement

WGT | **WEIGHT** *variable*

The WEIGHT statement specifies a numeric variable in the input data set with values that are used to weight each observation. This weight variable is used for standardizing variables rather than computing the distances. Only one variable can be specified.

The WEIGHT variable values can be non-integers. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero. The WEIGHT variable applies to variables that are standardized by the following options: STD=MEAN, STD=SUM, STD=EUCLEN, STD=USTD, STD=STD, STD=AGK, or STD=L. PROC DISTANCE uses the value of the WEIGHT variable w_i, as follows.

The sample mean and (uncorrected) sample variances are computed as

$$\overline{x}_w = \sum_i w_i x_i / \sum_i w_i$$

$${u_w}^2 = \sum_i w_i {x_i}^2 / d$$

$${s_w}^2 = \sum_i w_i (x_i - \overline{x}_w)^2 / d$$

where w_i is the weight value of the ith observation, x_i is the value of the ith observation, and d is the divisor controlled by the VARDEF= option (see the VARDEF= option in the PROC DISTANCE statement for details.)

PROC DISTANCE uses the value of the WEIGHT variable to calculate the following statistics:

MEAN the weighted mean, $\overline{x}_w$

SUM the weighted sum, $\sum_i w_i x_i$

USTD the weighted uncorrected standard deviation, $\sqrt{u_w^2}$

STD the weighted standard deviation, $\sqrt{s_w^2}$

EUCLEN the weighted Euclidean length, computed as the square root of the weighted uncorrected sum of squares:

$$\sqrt{\sum_i w_i {x_i}^2}$$

AGK the AGK estimate. This estimate is documented further in the ACECLUS procedure as the METHOD=COUNT option. See the discussion of the WEIGHT statement in Chapter 16, "The ACECLUS Procedure," for information on how the WEIGHT variable is applied to the AGK estimate.

L — the L_p estimate. This estimate is documented further in the FASTCLUS procedure as the LEAST= option. See the discussion of the WEIGHT statement in Chapter 28, “The FASTCLUS Procedure,” for information on how the WEIGHT variable is used to compute weighted cluster means. Note that the number of clusters is always 1.

Details

Proximity Measures

The following notation is used in this section:

v — the number of variables or the dimensionality

x_j — data for observation *x* and the *j*th variable, where *j*= 1 to *v*

y_j — data for observation *y* and the *j*th variable, where *j*= 1 to *v*

w_j — weight for the *j*th variable from the WEIGHTS= option in the VAR statement. $w_j = 0$ when either x_j or y_j is missing.

W — the sum of total weights. No matter if the observation is missing or not, its weight is added to this metric.

$\bar{x}$ — mean for observation *x*

$$\bar{x} = \sum_{i=1}^{v} w_j x_j / \sum_{i=1}^{v} w_j$$

$\bar{y}$ — mean for observation *y*

$$\bar{y} = \sum_{i=1}^{v} w_j y_j / \sum_{i=1}^{v} w_j$$

$d(x,y)$ — the distance or dissimilarity between observations *x* and *y*

$s(x,y)$ — the similarity between observations *x* and *y*

The factor $W/\sum_{i=1}^{v} w_j$ is used to adjust some of the proximity measures for missing values.

Methods Accepting All Measurement Levels

GOWER — Gower’s similarity

$$s_1(x,y) = \sum_{j=1}^{v} w_j \delta_{x,y}^{j} d_{x,y}^{j} / \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$$

To compute $\delta_{x,y}^{j}$: for nominal, ordinal, interval, or ratio variable,

$$\delta_{x,y}^{j} = 1;$$

for asymmetric nominal variable,

$$\delta_{x,y}^{j} = 1, \text{ if either } x_j \text{ or } y_j \text{ is present}$$

$$\delta_{x,y}^{j} = 0, \text{ if both } x_j \text{ and } y_j \text{ are absent}$$

To compute $d_{x,y}^{j}$: for nominal or asymmetric nominal variable,

$$\begin{aligned} d_{x,y}^{j} &= 1, \text{ if } x_j = y_j \\ d_{x,y}^{j} &= 0, \text{ if } x_j \neq y_j; \end{aligned}$$

for ordinal (where data are replaced by corresponding rank scores), interval, or ratio variable,

$$d_{x,y}^{j} = 1 - |x_j - y_j|$$

DGOWER
1 minus Gower
$d_2(x,y) = 1 - s_1(x,y)$

Methods Accepting Ratio, Interval, and Ordinal Variables:

EUCLID
Euclidean distance
$d_3(x,y) = \sqrt{(\sum_{j=1}^{v} w_j (x_j - y_j)^2) W / (\sum_{j=1}^{v} w_j)}$

SQEUCLID
Squared Euclidean distance
$d_4(x,y) = (\sum_{j=1}^{v} w_j (x_j - y_j)^2) W / (\sum_{j=1}^{v} w_j)$

SIZE
Size distance
$d_5(x,y) = |\sum_{j=1}^{v} w_j (x_j - y_j)| \sqrt{W} / (\sum_{j=1}^{v} w_j)$

SHAPE
Shape distance
$d_6(x,y) = \sqrt{(\sum_{j=1}^{v} w_j [(x_j - \bar{x}) - (y_j - \bar{y})]^2) W / (\sum_{j=1}^{v} w_j)}$

Note: squared shape distance plus squared size distance equals squared Euclidean distance.

COV
Covariance similarity coefficient
$s_7(x,y) = \sum_{j=1}^{v} w_j (x_j - \bar{x})(y_j - \bar{y}) / vardiv$, where

$$\begin{aligned} vardiv &= v \text{ if VARDEF=N} \\ &= v - 1 \text{ if VARDEF=DF} \\ &= \textstyle\sum_{j=1}^{v} w_j \text{ if VARDEF=WEIGHT} \\ &= \textstyle\sum_{j=1}^{v} w_j - 1 \text{ if VARDEF=WDF} \end{aligned}$$

CORR
Correlation similarity coefficient
$s_8(x,y) = \frac{\sum_{j=1}^{v} w_j (x_j - \bar{x})(y_j - \bar{y})}{\sqrt{\sum_{j=1}^{v} w_j (x_j - \bar{x})^2 \sum_{j=1}^{v} w_j (y_j - \bar{y})^2}}$

DCORR
Correlation transformed to Euclidean distance as sqrt(1-CORR)
$d_9(x,y) = \sqrt{1 - s_8(x,y)}$

SQCORR Squared correlation

$$s_{10}(x,y) = \frac{[\sum_{j=1}^{v} w_j(x_j-\bar{x})(y_j-\bar{y})]^2}{\sum_{j=1}^{v} w_j(x_j-\bar{x})^2 \sum_{j=1}^{v} w_j(y_j-\bar{y})^2}$$

DSQCORR Squared correlation transformed to squared Euclidean distance as (1-SQCORR)

$$d_{11}(x,y) = 1 - s_{10}(x,y)$$

L(p) Minkowski (L_p) distance, where p is a positive numeric value

$$d_{12}(x,y) = [(\sum_{j=1}^{v} w_j|x_j - y_j|^p)W/(\sum_{j=1}^{v} w_j)]^{1/p}$$

CITYBLOCK L$_1$

$$d_{13}(x,y) = (\sum_{j=1}^{v} w_j|x_j - y_j|)W/(\sum_{j=1}^{v} w_j)$$

CHEBYCHEV L$_\infty$

$$d_{14}(x,y) = \max_{j=1}^{v} w_j|x_j - y_j|$$

POWER(p,r) Generalized Euclidean distance, where *p* is a non-negative numeric value, and *r* is a positive numeric value. The distance between two observations is the *r*th root of sum of the absolute differences to the *p*th power between the values for the observations

$$d_{15}(x,y) = [(\sum_{j=1}^{v} w_j|x_j - y_j|^p)W/(\sum_{j=1}^{v} w_j)]^{1/r}$$

Methods Accepting Ratio Variables

SIMRATIO Similarity ratio

$$s_{16}(x,y) = \frac{\sum_{j}^{v} w_j(x_i y_j)}{\sum_{j=1}^{v} w_j(x_i y_j) + \sum_{j}^{v} w_j(x_j - y_j)^2}$$

DISRATIO one minus similarity ratio

$$d_{17}(x,y) = 1 - s_{16}(x,y)$$

NONMETRIC Lance-Williams nonmetric coefficient

$$d_{18}(x,y) = \frac{\sum_{j=1}^{v} w_j|x_j - y_j|}{\sum_{j=1}^{v} w_j(x_j + y_j)}$$

CANBERRA Canberra metric coefficient

$$d_{19}(x,y) = \sum_{j=1}^{v} (\frac{w_j|x_j - y_j|}{w_j(x_j + y_j)})$$

COSINE Cosine

$$s_{20}(x,y) = \frac{\sum_{j=1}^{v} w_j(x_i y_j)}{\sqrt{\sum_{j=1}^{v} w_j(x_j)^2 \sum_{j=1}^{v} w_j(y_j)^2}}$$

DOT

Dot (inner) product
$s_{21}(x,y) = \sum_{j=1}^{v} w_j(x_i y_j)/\sum_{j=1}^{v} w_j$

OVERLAP

Sum of the minimum values
$s_{22}(x,y) = \sum_{j=1}^{v} w_j[\min(x_j, y_j)]$

DOVERLAP

The maximum of the sum of the x and the sum of y minus overlap
$d_{23}(x,y) = \max(\sum_{j=1}^{v} w_i x_j, \sum_{j=1}^{v} w_i y_j) - s_{22}(x,y)$

CHISQ

chi-squared
If the data represent the frequency counts, chi-squared dissimilarity between two sets of frequencies can be computed. A 2 by *v* contingency table is illustrated to explain how the chi-squared dissimilarity is computed:

	Variable				**Row**
Observation	**Var 1**	**Var 2**	**...**	**Var v**	**sum**
X	x_1	x_2	...	x_v	r_x
Y	y_1	y_2	...	y_v	r_y
Column sum	c_1	c_2	...	c_v	T

where

$$\begin{aligned} r_x &= \sum_{j=1}^{v} w_j x_j \\ r_y &= \sum_{j=1}^{v} w_j y_j \\ c_j &= w_j(x_j + y_j) \\ T &= r_x + r_y = \sum_{j=1}^{v} c_j \end{aligned}$$

The chi-squared measure is computed as follows:
$d_{24}(x,y) = (\sum_{j=1}^{v} \frac{(w_j x_j - E(x_j))^2}{E(x_j)} + \sum_{j=1}^{v} \frac{(w_j y_j - E(y_j))^2}{E(y_j)})W/(\sum_{j=1}^{v} w_j)$
where for *j*= 1, 2, ..., *v*

$$\begin{aligned} E(x_j) &= r_x c_j/T \\ E(y_j) &= r_y c_j/T \end{aligned}$$

CHI

Squared root of chi-squared
$d_{25}(x,y) = \sqrt{d_{23}(x,y)}$

PHISQ

phi-squared
This is the CHISQ dissimilarity normalized by the sum of weights
$d_{26}(x,y) = d_{24}(x,y)/(\sum_{j=1}^{v} w_j)$

PHI

Squared root of phi-squared
$d_{27}(x,y) = \sqrt{d_{25}(x,y)}$

Methods Accepting Symmetric Nominal Variables

The following notation is used for computing $d_{28}(x,y)$ to $s_{35}(x,y)$. Notice that only the non-missing pairs are discussed below; all the pairs with at least one missing value will be excluded from any of the computations in the following section because $w_j = 0$, if either x_j or y_j is missing.

M non-missing matches

$M = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{array}{lcl} \delta_{x,y}^{j} & = & 1, \text{ if } \ x_j = y_j \\ \delta_{x,y}^{j} & = & 0, \text{ otherwise} \end{array}$$

X non-missing mismatches

$X = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{array}{lcl} \delta_{x,y}^{j} & = & 1, \text{ if } \ x_j \neq y_j \\ \delta_{x,y}^{j} & = & 0, \ \text{ otherwise} \end{array}$$

N total non-missing pairs

$N = \sum_{j=1}^{v} w_j$

HAMMING Hamming distance

$d_{28}(x,y) = X$

MATCH Simple matching coefficient

$s_{29}(x,y) = M/N$

DMATCH Simple matching coefficient transformed to Euclidean distance

$d_{30}(x,y) = \sqrt{1 - M/N} = \sqrt{(X/N)}$

DSQMATCH Simple matching coefficient transformed to squared Euclidean distance $d_{31}(x,y) = 1 - M/N = X/N$

HAMANN Hamann coefficient

$s_{32}(x,y) = (M - X)/N$

RT Roger and Tanimoto

$s_{33}(x,y) = M/(M + 2X)$

SS1 Sokal and Sneath 1

$s_{34}(x,y) = 2M/(2M + X)$

SS3 Sokal and Sneath 3. The coefficient between an observations and itself is always indeterminate (missing) since there is no mismatch.

$$s_{35}(x,y) = M/X$$

The following notation is used for computing $s_{36}(x,y)$ to $d_{41}(x,y)$. Notice that only the non-missing pairs are discussed below; all the pairs with at least one missing value will be excluded from any of the computations in the following section because $w_j = 0$, if either x_j or y_j is missing.

Also, the observed non-missing data of an asymmetric binary variable can possibly have only two outcomes: presence or absence. Therefore, the notation, *PX* (present mismatches), always has a value of zero for an asymmetric binary variable.

The following methods distinguish between the presence and absence of attributes.

X mismatches with at least one present
$X = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j \neq y_j \text{ and not both } x_j \text{ and } y_j \text{ are absent} \\ \delta_{x,y}^{j} &= 0, \text{ otherwise} \end{aligned}$$

PM present matches
$PM = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j = y_j \text{ and both } x_j \text{ and } y_j \text{ are present} \\ \delta_{x,y}^{j} &= 0, \text{ otherwise} \end{aligned}$$

PX present mismatches
$PX = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j \neq y_j \text{ and both } x_j \text{ and } y_j \text{ are present} \\ \delta_{x,y}^{j} &= 0, \text{ otherwise} \end{aligned}$$

PP both present = $PM + PX$

P at least one present = $PM + X$

PAX present-absent mismatches
$PAX = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j \neq y_j \text{ and either } x_j \text{ is present and } y_j \text{ is absent or} \\ & \quad x_j \text{ is absent and } y_j \text{ is present} \\ \delta_{x,y}^{j} &= 0 \text{ otherwise} \end{aligned}$$

N total non-missing pairs
$N = \sum_{j=1}^{v} w_j$

Methods Accepting Asymmetric Nominal and Ratio Variables

JACCARD Jaccard similarity coefficient

The JACCARD method is equivalent to the SIMRATIO method if there are only ratio variables; if there are both ratio and asymmetric nominal variables, the coefficient is computed as sum of the coefficient from the ratio variables (SIMRATIO) and the coefficient from the asymmetric nominal variables.

$s_{36}(x, y) = s_{16}(x, y) + PM/P$

DJACCARD Jaccard dissimilarity coefficient

The DJACCARD method is equivalent to the DISRATIO method if there are only ratio variables; if there are both ratio and asymmetric nominal variables, the coefficient is computed as sum of the coefficient from the ratio variables(DISRATIO) and the coefficient from the asymmetric nominal variables.

$d_{37}(x, y) = d_{17}x, y + X/P$

Methods Accepting Asymmetric Nominal Variables

DICE Dice coefficient or Czekanowski/Sorensen similarity coefficient
$s_{38}(x, y) = 2PM/(P + PM)$

RR Russell and Rao. This is the binary equivalent of the dot product coefficient.
$s_{39}(x, y) = PM/N$

BLWNM

BRAYCURTIS Binary Lance and Williams, also known as Bray and Curtis coefficient

$d_{40}(x, y) = X/(PAX + 2PP)$

K1 Kulcynski 1. The coefficient between an observations and itself is always indeterminate (missing) since there is no mismatch.
$d_{41}(x, y) = PM/X$

Missing Values

Standardizing Variables

Missing values can be replaced by the location measure or by any specified constant (see the REPLACE option in the PROC DISTANCE statement and the MISSING=

option in the VAR statement.) If standardization is not mandatory, you can also suppress standardization if you want only to replace missing values (see the REPONLY option in the PROC DISTANCE statement.)

If you specify the NOMISS option, PROC DISTANCE omits observations with any missing values in the analyzed variables from computation of the location and scale measures.

Distance Measures

If you specify the NOMISS option, PROC DISTANCE generates missing distance for observations with missing values. If the NOMISS option is not specified, the sum of total weights, no matter if an observation is missing or not, will be incorporated to the the computation of some of the proximity measures. See the "Details" section on page 1270 for formulas and descriptions.

Formatted versus Unformatted Values

PROC DISTANCE uses the formatted values from a character variable, if the variable has a format; for example, one assigned by a format statement. PROC DISTANCE uses the unformatted values from a numeric variable, even if it has a format.

Output Data Sets

OUT= Data Set

The DISTANCE procedure always produces an output data set, regardless of whether you specify the OUT= option in the PROC DISTANCE statement. PROC DISTANCE displays no output. Use PROC PRINT, PROC REPORT or some other SAS reporting tool to print the output data set.

The output data set contains the following variables:

- the ID variable, if any
- the BY variables, if any
- the COPY variables, if any
- the FREQ variable, if any
- the WEIGHT variable, if any
- the new distance variables, named from PREFIX= options along with the ID values, or from the default values.

OUTSDZ= Data Set

The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement.

Examples

Example 26.1. Divorce Grounds – the Jaccard Coefficient

A wide variety of distance and similarity measures are used in cluster analysis (Anderberg 1973, Sneath and Sokal 1973). If your data are in coordinate form and you want to use a non-Euclidean distance for clustering, you can compute a distance matrix using the DISTANCE procedure.

Similarity measures must be converted to dissimilarities before being used in PROC CLUSTER. Such conversion can be done in a variety of ways, such as taking reciprocals or subtracting from a large value. The choice of conversion method depends on the application and the similarity measure. If applicable, PROC DISTANCE provides a corresponding dissimilarity measure for each similarity measure.

In the following example, the observations are states. Binary-valued variables correspond to various grounds for divorce and indicate whether the grounds for divorce apply in each of the states of the USA. A value of “1” indicates that the ground for divorce applies and a value of “0” indicates the opposite. The 0-0 matches are treated totally irrelevant; therefore, each variable has an asymmetric nominal level of measurement. The absence value is 0.

The DISTANCE procedure is used to compute the Jaccard coefficient (Anderberg 1973, pp. 89, 115, and 117) between each pair of states. The Jaccard coefficient is defined as the number of variables that are coded as 1 for both states divided by the number of variables that are coded as 1 for either or both states. Since dissimilarity measures are required by the CLUSTER procedure, the DJACCARD coefficient is selected. Output 26.1.1 displays the distance matrix between the first ten states.

The CENTROID method is used to perform the cluster analysis, and the the resulting tree diagram from PROC CLUSTER is saved into the tree output data set. Output 26.1.2 displays the cluster history.

The TREE procedure generates nine clusters in the output data set out. After being sorted by the state, the out data set is then merged with the input data set divorce. After being sorted by the state, the merged data set is printed to display the cluster membership as shown in Output 26.1.3.

```
options ls=120 ps=60;
data divorce;
   title 'Grounds for Divorce';
   input state $15.
         (incompat cruelty desertn non_supp alcohol
          felony impotenc insanity separate) (1.) @@;
   if mod(_n_,2) then input +4 @@; else input;
   datalines;
Alabama        111111111    Alaska         111011110
Arizona        100000000    Arkansas       011111111
California     100000010    Colorado       100000000
Connecticut    111111011    Delaware       100000001
```

```
Florida        100000010    Georgia        111011110
Hawaii         100000001    Idaho          111111011
Illinois       011011100    Indiana        100001110
Iowa           100000000    Kansas         111011110
Kentucky       100000000    Louisiana      000001001
Maine          111110110    Maryland       011001111
Massachusetts  111111101    Michigan       100000000
Minnesota      100000000    Mississippi    111011110
Missouri       100000000    Montana        100000000
Nebraska       100000000    Nevada         100000011
New Hampshire  111111100    New Jersey     011011011
New Mexico     111000000    New York       011001001
North Carolina 000000111    North Dakota   111111110
Ohio           111011101    Oklahoma       111111110
Oregon         100000000    Pennsylvania   011001110
Rhode Island   111111101    South Carolina 011010001
South Dakota   011111000    Tennessee      111111100
Texas          111001011    Utah           011111110
Vermont        011101011    Virginia       010001001
Washington     100000001    West Virginia  111011011
Wisconsin      100000001    Wyoming        100000011
;

proc distance data=divorce method=djaccard absent=0 out=distjacc;
   var anominal(incompat--separate);
   id state;
run;

proc print data=distjacc(obs=10);
   id state; var alabama--georgia;
   title2 ’First 10 states’;
run;
title2;

proc cluster data=distjacc method=centroid
             pseudo outtree=tree;
   id state;
   var alabama--wyoming;
run;

proc tree data=tree noprint n=9 out=out;
   id state;
run;

proc sort;
   by state;
run;

data clus;
   merge divorce out;
   by state;
run;

proc sort;
```

```
   by cluster;
run;

proc print;
   id state;
   var incompat--separate;
   by cluster;
run;
```

Output 26.1.1. Distance Matrix Based on the Jaccard Coefficient

```
                                              Grounds for Divorce
                                                First 10 states

state        Alabama   Alaska  Arizona  Arkansas  California  Colorado  Connecticut  Delaware  Florida  Georgia

Alabama      0.00000   .        .         .          .          .          .           .        .          .
Alaska       0.22222  0.00000   .         .          .          .          .           .        .          .
Arizona      0.88889  0.85714  0.00000    .          .          .          .           .        .          .
Arkansas     0.11111  0.33333  1.00000   0.00000     .          .          .           .        .          .
California   0.77778  0.71429  0.50000   0.88889    0.00000     .          .           .        .          .
Colorado     0.88889  0.85714  0.00000   1.00000    0.50000    0.00000     .           .        .          .
Connecticut  0.11111  0.33333  0.87500   0.22222    0.75000    0.87500    0.00000      .        .          .
Delaware     0.77778  0.87500  0.50000   0.88889    0.66667    0.50000    0.75000     0.00000   .          .
Florida      0.77778  0.71429  0.50000   0.88889    0.00000    0.50000    0.75000     0.66667  0.00000     .
Georgia      0.22222  0.00000  0.85714   0.33333    0.71429    0.85714    0.33333     0.87500  0.71429     0
```

Output 26.1.2. Clustering History

```
                              The CLUSTER Procedure
                       Centroid Hierarchical Cluster Analysis

               Root-Mean-Square Distance Between Observations   = 0.694873


                                 Cluster History
                                                                     Norm    T
                                                                     Cent    i
 NCL    ---------Clusters Joined----------      FREQ     PSF    PST2      Dist    e

  49    Arizona           Colorado               2       .       .         0    T
  48    California        Florida                2       .       .         0    T
  47    Alaska            Georgia                2       .       .         0    T
  46    Delaware          Hawaii                 2       .       .         0    T
  45    Connecticut       Idaho                  2       .       .         0    T
  44    CL49              Iowa                   3       .       .         0    T
  43    CL47              Kansas                 3       .       .         0    T
  42    CL44              Kentucky               4       .       .         0    T
  41    CL42              Michigan               5       .       .         0    T
  40    CL41              Minnesota              6       .       .         0    T
  39    CL43              Mississippi            4       .       .         0    T
  38    CL40              Missouri               7       .       .         0    T
  37    CL38              Montana                8       .       .         0    T
  36    CL37              Nebraska               9       .       .         0    T
  35    North Dakota      Oklahoma               2       .       .         0    T
  34    CL36              Oregon                10       .       .         0    T
  33    Massachusetts     Rhode Island           2       .       .         0    T
  32    New Hampshire     Tennessee              2       .       .         0    T
  31    CL46              Washington             3       .       .         0    T
  30    CL31              Wisconsin              4       .       .         0    T
  29    Nevada            Wyoming                2       .       .         0
  28    Alabama           Arkansas               2    1561       .    0.1599    T
  27    CL33              CL32                   4     479       .    0.1799    T
  26    CL39              CL35                   6     265       .    0.1799    T
  25    CL45              West Virginia          3     231       .    0.1799
  24    Maryland          Pennsylvania           2     199       .    0.2399
  23    CL28              Utah                   3     167     3.2    0.2468
  22    CL27              Ohio                   5     136     5.4    0.2698
  21    CL26              Maine                  7     111     8.9    0.2998
  20    CL23              CL21                  10    75.2     8.7    0.3004
  19    CL25              New Jersey             4    71.8     6.5    0.3053    T
  18    CL19              Texas                  5    69.1     2.5    0.3077
  17    CL20              CL22                  15    48.7     9.9    0.3219
  16    New York          Virginia               2    50.1       .    0.3598
  15    CL18              Vermont                6    49.4     2.9    0.3797
  14    CL17              Illinois              16    47.0     3.2    0.4425
  13    CL14              CL15                  22    29.2    15.3    0.4722
  12    CL48              CL29                   4    29.5       .    0.4797    T
  11    CL13              CL24                  24    27.6     4.5    0.5042
  10    CL11              South Dakota          25    28.4     2.4    0.5449
   9    Louisiana         CL16                   3    30.3     3.5    0.5844
   8    CL34              CL30                  14    23.3       .    0.7196
   7    CL8               CL12                  18    19.3    15.0    0.7175
   6    CL10              South Carolina        26    21.4     4.2    0.7384
   5    CL6               New Mexico            27    24.0     4.7    0.8303
   4    CL5               Indiana               28    28.9     4.1    0.8343
   3    CL4               CL9                   31    31.7    10.9    0.8472
   2    CL3               North Carolina        32    55.1     4.1    1.0017
   1    CL2               CL7                   50       .    55.1    1.0663
```

Output 26.1.3. Cluster Membership

```
----------------------------------------------------- CLUSTER=1 -----------------------------------------------------

    state        incompat    cruelty    desertn    non_supp    alcohol    felony    impotenc    insanity    separate

    Arizona          1          0          0           0          0          0          0           0           0
    Colorado         1          0          0           0          0          0          0           0           0
    Iowa             1          0          0           0          0          0          0           0           0
    Kentucky         1          0          0           0          0          0          0           0           0
    Michigan         1          0          0           0          0          0          0           0           0
    Minnesota        1          0          0           0          0          0          0           0           0
    Missouri         1          0          0           0          0          0          0           0           0
    Montana          1          0          0           0          0          0          0           0           0
    Nebraska         1          0          0           0          0          0          0           0           0
    Oregon           1          0          0           0          0          0          0           0           0


----------------------------------------------------- CLUSTER=2 -----------------------------------------------------

   state         incompat    cruelty    desertn    non_supp    alcohol    felony    impotenc    insanity    separate

   California        1          0          0           0          0          0          0           1           0
   Florida           1          0          0           0          0          0          0           1           0
   Nevada            1          0          0           0          0          0          0           1           1
   Wyoming           1          0          0           0          0          0          0           1           1


----------------------------------------------------- CLUSTER=3 -----------------------------------------------------

  state            incompat    cruelty    desertn    non_supp    alcohol    felony    impotenc    insanity    separate

  Alabama              1          1          1           1          1          1          1           1           1
  Alaska               1          1          1           0          1          1          1           1           0
  Arkansas             0          1          1           1          1          1          1           1           1
  Connecticut          1          1          1           1          1          1          0           1           1
  Georgia              1          1          1           0          1          1          1           1           0
  Idaho                1          1          1           1          1          1          0           1           1
  Illinois             0          1          1           0          1          1          1           0           0
  Kansas               1          1          1           0          1          1          1           1           0
  Maine                1          1          1           1          1          0          1           1           0
  Maryland             0          1          1           0          0          1          1           1           1
  Massachusetts        1          1          1           1          1          1          1           0           1
  Mississippi          1          1          1           0          1          1          1           1           0
  New Hampshire        1          1          1           1          1          1          1           0           0
  New Jersey           0          1          1           0          1          1          0           1           1
  North Dakota         1          1          1           1          1          1          1           1           0
  Ohio                 1          1          1           0          1          1          1           0           1
  Oklahoma             1          1          1           1          1          1          1           1           0
  Pennsylvania         0          1          1           0          0          1          1           1           0
  Rhode Island         1          1          1           1          1          1          1           0           1
  South Dakota         0          1          1           1          1          1          0           0           0
  Tennessee            1          1          1           1          1          1          1           0           0
  Texas                1          1          1           0          0          1          0           1           1
  Utah                 0          1          1           1          1          1          1           1           0
  Vermont              0          1          1           1          0          1          0           1           1
  West Virginia        1          1          1           0          1          1          0           1           1


----------------------------------------------------- CLUSTER=4 -----------------------------------------------------

   state         incompat    cruelty    desertn    non_supp    alcohol    felony    impotenc    insanity    separate

   Delaware          1          0          0           0          0          0          0           0           1
   Hawaii            1          0          0           0          0          0          0           0           1
   Washington        1          0          0           0          0          0          0           0           1
   Wisconsin         1          0          0           0          0          0          0           0           1


----------------------------------------------------- CLUSTER=5 -----------------------------------------------------

      state      incompat    cruelty    desertn    non_supp    alcohol    felony    impotenc    insanity    separate

    Louisiana        0          0          0           0          0          1          0           0           1
    New York         0          1          1           0          0          1          0           0           1
    Virginia         0          1          0           0          0          1          0           0           1
```

```
------------------------------------------------------ CLUSTER=6 -------------------------------------------------------

     state          incompat     cruelty     desertn     non_supp     alcohol     felony     impotenc     insanity     separate

 South Carolina        0            1           1           0            1           0           0            0            1


------------------------------------------------------ CLUSTER=7 -------------------------------------------------------

     state        incompat     cruelty     desertn     non_supp     alcohol     felony     impotenc     insanity     separate

   New Mexico        1            1           1           0            0           0           0            0            0


------------------------------------------------------ CLUSTER=8 -------------------------------------------------------

      state      incompat     cruelty     desertn     non_supp     alcohol     felony     impotenc     insanity     separate

     Indiana        1            0           0           0            0           1           1            1            0


------------------------------------------------------ CLUSTER=9 -------------------------------------------------------

     state          incompat     cruelty     desertn     non_supp     alcohol     felony     impotenc     insanity     separate

 North Carolina        0            0           0           0            0           0           1            1            1
```

Example 26.2. Financial Data – Stock Dividends

The following data set contains the average dividend yields for 15 utility stocks in the U.S. The observations are names of the companies, and the variables correspond to the annual dividend yields for the period of 1986-1990. The objective is to group similar stocks into clusters.

Before the cluster analysis is performed, the correlation similarity is chosen for measuring the closeness between each observation. Since distance type of measures are required by the CLUSTER procedure, METHOD= DCORR is used in the PROC DISTANCE statement to transform the correlation measures to the distance measures. Notice that in Output 26.2.1, all the values in the distance matrix are between 0. and 2.

The macro function do_cluster performs cluster analysis and presents the results in graphs. The CLUSTER procedure performs hierarchical clustering using agglomerative methods based on the distance data created from the previous PROC DISTANCE statement. The resulting tree diagrams can be saved into an output data set and later be plotted by the TREE procedure. Since the CCC statistics is not suitable for distance type of data, only the Pseudo Statistics is requested to identify the number of clusters.

Two clustering methods are invoked in the do_cluster macro: the WARD's and the average linkage methods. Since the results of the Pseudo T statistics from both the WARD's and the average linkage methods contain many missing values, only the graphs of the Pseudo F statistics versus the number of clusters are plotted.

Both Output 26.2.2 and Output 26.2.3 suggest a possible clusters of 4, and the resulting clusters are agreed by both clustering methods as shown from Output 26.2.4 to Output 26.2.5. The four clusters are:

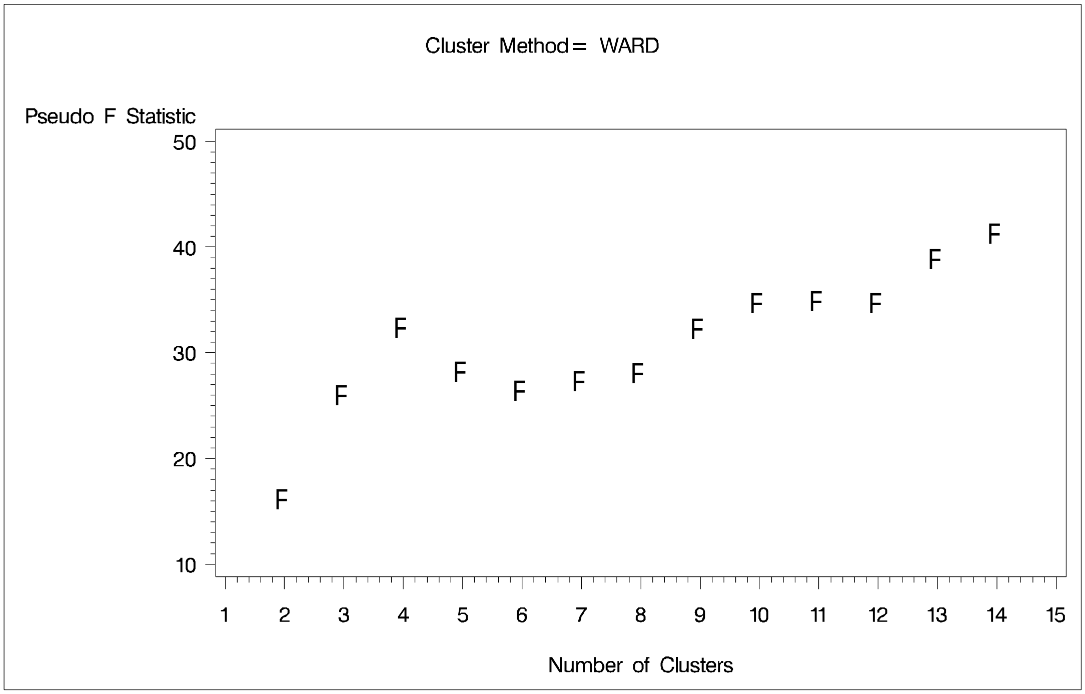
- Cincinnati G&E and Detroit Edison
- Texas Utilities and Pennsylvania Power & Light
- Union Electric, Iowa-Ill Gas & Electric, Oklahoma Gas & Electric, and Wisconsin Energy.
- Orange & Rockland Utilities, Kentucky Utilities, Kansas Power & Light, Allegheny Power, Green Mountain Power, Dominion Resources, and Minnesota Power & Light.

```
data stock;
   title 'Stock Dividends';
   input compname &$26.  div_1986 div_1987 div_1988
                         div_1989 div_1990;
   datalines;
Cincinnati G&E               8.4    8.2    8.4    8.1    8.0
Texas Utilities              7.9    8.9   10.4    8.9    8.3
Detroit Edison               9.7   10.7   11.4    7.8    6.5
Orange & Rockland Utilities  6.5    7.2    7.3    7.7    7.9
Kentucky Utilities           6.5    6.9    7.0    7.2    7.5
Kansas Power & Light         5.9    6.4    6.9    7.4    8.0
Union Electric               7.1    7.5    8.4    7.8    7.7
Dominion Resources           6.7    6.9    7.0    7.0    7.4
Allegheny Power              6.7    7.3    7.8    7.9    8.3
Minnesota Power & Light      5.6    6.1    7.2    7.0    7.5
Iowa-Ill Gas & Electric      7.1    7.5    8.5    7.8    8.0
Pennsylvania Power & Light   7.2    7.6    7.7    7.4    7.1
Oklahoma Gas & Electric      6.1    6.7    7.4    6.7    6.8
Wisconsin Energy             5.1    5.7    6.0    5.7    5.9
Green Mountain Power         7.1    7.4    7.8    7.8    8.3
;

proc distance data=stock method=dcorr out=distdcorr;
   var interval(div_1986 div_1987 div_1988 div_1989 div_1990);
   id compname;
run;

proc print data=distdcorr;
   id compname;
   title2 'Distance Matrix for 15 Utility Stocks';
run;
title2;

%macro do_cluster(clusmtd);

   goptions vsize=5in hsize=5in htitle=2pct htext=1.5pct;
   %let clusmtd = %upcase(&clusmtd);

   proc cluster data=distdcorr method=&clusmtd outtree=Tree pseudo
      id compname;
   run;

   /* plot pseudo statistics vs number of cluster */
   legend1 frame cframe=white cborder=black position=center
   value=(justify=center);
   axis1 label=(angle=90 rotate=0) minor=none;
   axis2 minor=none order=(0 to 15);

   proc gplot;
      title2 "Cluster Method= &clusmtd";
      plot _psf_*_ncl_='F'  /
      frame cframe=white legend=legend1 vaxis=axis1 haxis=axis2;
   run;

   proc tree data=Tree horizontal;
      title2 "Cluster Method= &clusmtd";
      id compname;
   run;
%mend;

%do_cluster(ward);
%do_cluster(average);
```

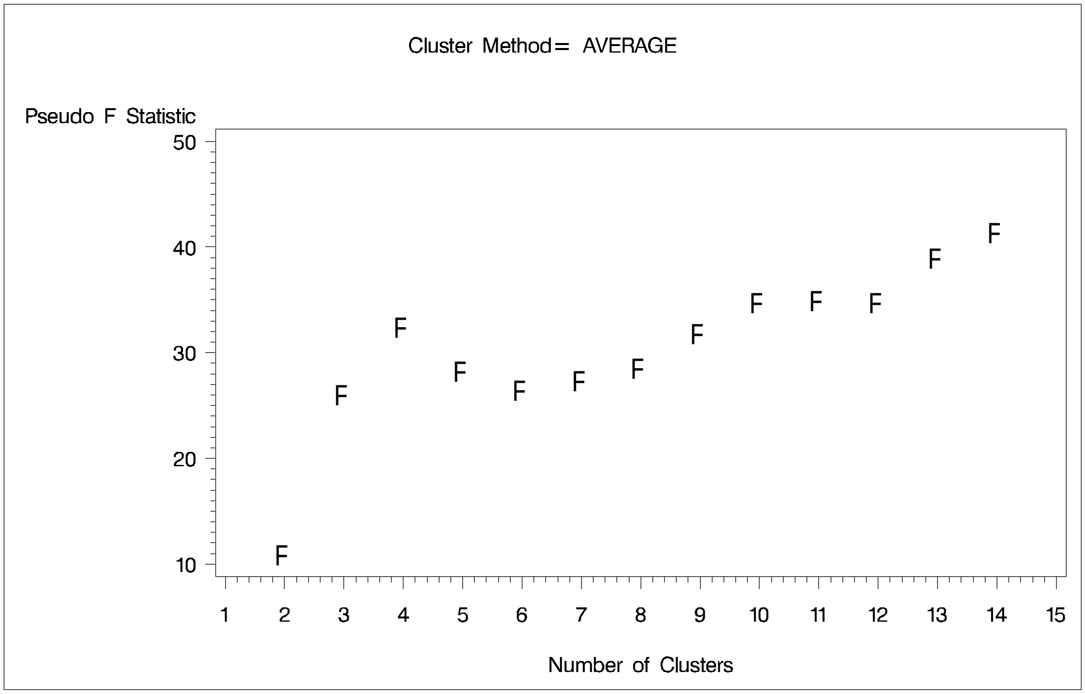
Output 26.2.1. Distance Matrix Based on the DCORR Coefficient

```
                                                  Stock Dividends
                                       Distance Matrix for 15 Utility Stocks

                                                              Orange___              Kansas_
                            Cincinnati_    Texas_   Detroit_  Rockland_  Kentucky_  Power___   Union_   Dominion_
compname                        G_E      Utilities   Edison    Utilitie  Utilities    Light   Electric  Resources

Cincinnati G&E                0.00000       .          .         .          .          .         .         .
Texas Utilities               0.82056     0.00000      .         .          .          .         .         .
Detroit Edison                0.40511     0.65453    0.00000     .          .          .         .         .
Orange & Rockland Utilitie    1.35380     0.88583    1.27306   0.00000      .          .         .         .
Kentucky Utilities            1.35581     0.92539    1.29382   0.12268    0.00000      .         .         .
Kansas Power & Light          1.34227     0.94371    1.31696   0.19905    0.12874    0.00000     .         .
Union Electric                0.98516     0.29043    0.89048   0.68798    0.71824    0.72082   0.00000     .
Dominion Resources            1.32945     0.96853    1.29016   0.33290    0.21510    0.24189   0.76587   0.00000
Allegheny Power               1.30492     0.81666    1.24565   0.17844    0.15759    0.17029   0.58452   0.27819
Minnesota Power & Light       1.24069     0.74082    1.20432   0.32581    0.30462    0.27231   0.48372   0.35733
Iowa-Ill Gas & Electric       1.04924     0.43100    0.97616   0.61166    0.61760    0.61736   0.16923   0.63545
Pennsylvania Power & Light    0.74931     0.37821    0.44256   1.03566    1.08878    1.12876   0.63285   1.14354
Oklahoma Gas & Electric       1.00604     0.30141    0.86200   0.68021    0.70259    0.73158   0.17122   0.72977
Wisconsin Energy              1.17988     0.54830    1.03081   0.45013    0.47184    0.53381   0.37405   0.51969
Green Mountain Power          1.30397     0.88063    1.27176   0.26948    0.17909    0.15377   0.64869   0.17360

                                         Minnesota_   Iowa_Ill_                   Oklahoma_                  Green_
                            Allegheny_    Power___      Gas___    Pennsylvania_    Gas___     Wisconsin_   Mountain_
compname                       Power        Light      Electric   Power___Light    Electric     Energy       Power

Cincinnati G&E                  .            .           .            .              .            .             .
Texas Utilities                 .            .           .            .              .            .             .
Detroit Edison                  .            .           .            .              .            .             .
Orange & Rockland Utilitie      .            .           .            .              .            .             .
Kentucky Utilities              .            .           .            .              .            .             .
Kansas Power & Light            .            .           .            .              .            .             .
Union Electric                  .            .           .            .              .            .             .
Dominion Resources              .            .           .            .              .            .             .
Allegheny Power               0.00000        .           .            .              .            .             .
Minnesota Power & Light       0.15615      0.00000       .            .              .            .             .
Iowa-Ill Gas & Electric       0.47900      0.36368     0.00000        .              .            .             .
Pennsylvania Power & Light    1.02358      0.99384     0.75596      0.00000          .            .             .
Oklahoma Gas & Electric       0.58391      0.50744     0.19673      0.60216        0.00000        .             .
Wisconsin Energy              0.37522      0.36319     0.30259      0.76085        0.28070      0.00000         .
Green Mountain Power          0.13958      0.19370     0.52083      1.09269        0.64175      0.44814         0
```

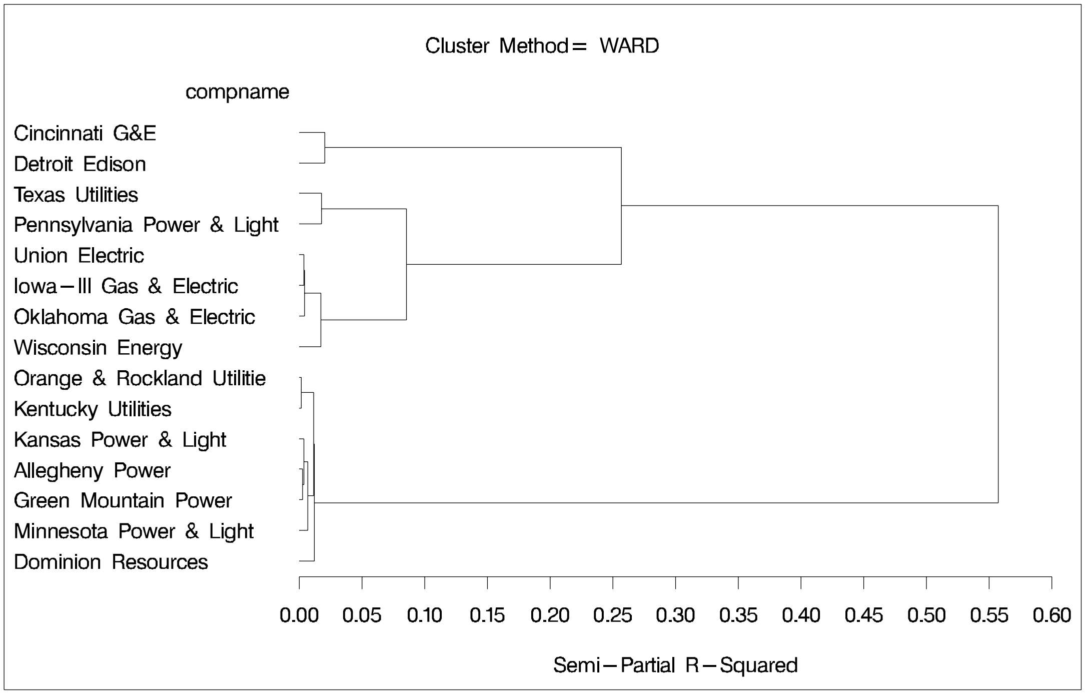
Output 26.2.2. Pseudo F versus Number of Clusters when METHOD= WARD



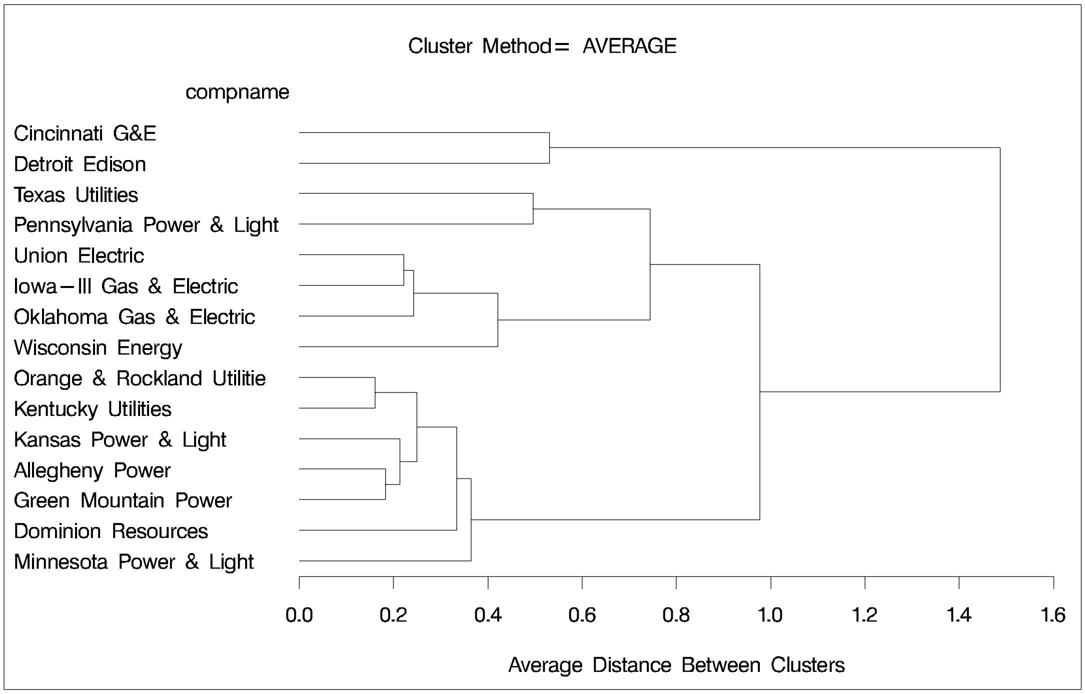
Output 26.2.3. Pseudo F versus Number of Clusters when METHOD= AVERAGE



Output 26.2.4. Tree Diagram of Clusters versus Semi-Partial R-Square Values when METHOD= WARD



Output 26.2.5. Tree Diagram of Clusters versus Average Distance Between Clusters when METHOD= AVERAGE



References

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Hand, D.J., Daly, F., Lunn, A.D., McConway, K.J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman and Hall, 297–298.

Legendre, L. and Legendre, P. (1983), *Numerical Ecology*, New York: Elsevier Scientific Pub. Co.

Kaufman, L. and Rousseeuw, P.J. (1990), *Finding Groups in Data*, New York: John Wiley and Sons, Inc.

Sneath, P.H.A. and Sokal, R.R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

Chapter 27
The FACTOR Procedure

Chapter Contents

Chapter 27
The FACTOR Procedure

Overview

The FACTOR procedure performs a variety of common factor and component analyses and rotations. Input can be multivariate data, a correlation matrix, a covariance matrix, a factor pattern, or a matrix of scoring coefficients. The procedure can factor either the correlation or covariance matrix, and you can save most results in an output data set.

PROC FACTOR can process output from other procedures. For example, it can rotate the canonical coefficients from multivariate analyses in the GLM procedure.

The methods for factor extraction are principal component analysis, principal factor analysis, iterated principal factor analysis, unweighted least-squares factor analysis, maximum likelihood (canonical) factor analysis, alpha factor analysis, image component analysis, and Harris component analysis. A variety of methods for prior communality estimation is also available.

Specific methods for orthogonal rotation are varimax, quartimax, biquartimax, equamax, parsimax, and factor parsimax. Oblique versions of these methods are also available. In addition, quartimin, biquartimin, and covarimin methods for (direct) oblique rotation are available. General methods for orthogonal rotation are orthomax with user-specified gamma, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, and generalized Crawford-Ferguson family with user-specified weights. General methods for oblique rotation are direct oblimin with user-specified tau, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, generalized Crawford-Ferguson family with user-specified weights, promax with user-specified exponent, Harris-Kaiser case II with user-specified exponent, and Procrustean with a user-specified target pattern.

Output includes means, standard deviations, correlations, Kaiser's measure of sampling adequacy, eigenvalues, a scree plot, eigenvectors, prior and final communality estimates, the unrotated factor pattern, residual and partial correlations, the rotated primary factor pattern, the primary factor structure, interfactor correlations, the reference structure, reference axis correlations, the variance explained by each factor both ignoring and eliminating other factors, plots of both rotated and unrotated factors, squared multiple correlation of each factor with the variables, standard error estimates, confidence limits, coverage displays, and scoring coefficients.

Any topics that are not given explicit references are discussed in Mulaik (1972) or Harman (1976).

Background

See Chapter 58, "The PRINCOMP Procedure," for a discussion of principal component analysis. See Chapter 19, "The CALIS Procedure," for a discussion of confirmatory factor analysis.

Common factor analysis was invented by Spearman (1904). Kim and Mueller (1978a, 1978b) provide a very elementary discussion of the common factor model. Gorsuch (1974) contains a broad survey of factor analysis, and Gorsuch (1974) and Cattell (1978) are useful as guides to practical research methodology. Harman (1976) gives a lucid discussion of many of the more technical aspects of factor analysis, especially oblique rotation. Morrison (1976) and Mardia, Kent, and Bibby (1979) provide excellent statistical treatments of common factor analysis. Mulaik (1972) is the most thorough and authoritative general reference on factor analysis and is highly recommended to anyone familiar with matrix algebra. Stewart (1981) gives a nontechnical presentation of some issues to consider when deciding whether or not a factor analysis may be appropriate.

A frequent source of confusion in the field of factor analysis is the term *factor*. It sometimes refers to a hypothetical, unobservable variable, as in the phrase *common factor*. In this sense, *factor analysis* must be distinguished from component analysis since a component is an observable linear combination. *Factor* is also used in the sense of *matrix factor,* in that one matrix is a factor of a second matrix if the first matrix multiplied by its transpose equals the second matrix. In this sense, *factor analysis* refers to all methods of data analysis using matrix factors, including component analysis and common factor analysis.

A *common factor* is an unobservable, hypothetical variable that contributes to the variance of at least two of the observed variables. The unqualified term "factor" often refers to a common factor. A *unique factor* is an unobservable, hypothetical variable that contributes to the variance of only one of the observed variables. The model for common factor analysis posits one unique factor for each observed variable.

The equation for the common factor model is

$$y_{ij} = x_{i1}b_{1j} + x_{i2}b_{2j} + \cdots + x_{iq}b_{qj} + e_{ij}$$

where

y_{ij} is the value of the ith observation on the jth variable

x_{ik} is the value of the ith observation on the kth common factor

b_{kj} is the regression coefficient of the kth common factor for predicting the jth variable

e_{ij} is the value of the ith observation on the jth unique factor

q is the number of common factors

It is assumed, for convenience, that all variables have a mean of 0. In matrix terms, these equations reduce to

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

In the preceding equation, $\mathbf{X}$ is the matrix of factor scores, and $\mathbf{B}'$ is the factor pattern.

There are two critical assumptions:

- The unique factors are uncorrelated with each other.
- The unique factors are uncorrelated with the common factors.

In principal component analysis, the residuals are generally correlated with each other. In common factor analysis, the unique factors play the role of residuals and are defined to be uncorrelated both with each other and with the common factors. Each common factor is assumed to contribute to at least two variables; otherwise, it would be a unique factor.

When the factors are initially extracted, it is also assumed, for convenience, that the common factors are uncorrelated with each other and have unit variance. In this case, the common factor model implies that the covariance s_{jk} between the jth and kth variables, $j \neq k$, is given by

$$s_{jk} = b_{1j}b_{1k} + b_{2j}b_{2k} + \cdots + b_{qj}b_{qk}$$

or

$$\mathbf{S} = \mathbf{B}'\mathbf{B} + \mathbf{U}^2$$

where $\mathbf{S}$ is the covariance matrix of the observed variables, and $\mathbf{U}^2$ is the diagonal covariance matrix of the unique factors.

If the original variables are standardized to unit variance, the preceding formula yields correlations instead of covariances. It is in this sense that common factors explain the correlations among the observed variables. The difference between the correlation predicted by the common factor model and the actual correlation is the *residual correlation*. A good way to assess the goodness-of-fit of the common factor model is to examine the residual correlations.

The common factor model implies that the partial correlations among the variables, removing the effects of the common factors, must all be 0. When the common factors are removed, only unique factors, which are by definition uncorrelated, remain.

The assumptions of common factor analysis imply that the common factors are, in general, not linear combinations of the observed variables. In fact, even if the data contain measurements on the entire population of observations, you cannot compute the scores of the observations on the common factors. Although the common factor scores cannot be computed directly, they can be estimated in a variety of ways.

The problem of factor score indeterminacy has led several factor analysts to propose methods yielding components that can be considered approximations to common factors. Since these components are defined as linear combinations, they are computable. The methods include Harris component analysis and image component analysis. The advantage of producing determinate component scores is offset by the fact that, even if the data fit the common factor model perfectly, component methods do not generally recover the correct factor solution. You should not use any type of component analysis if you really want a common factor analysis (Dziuban and Harris 1973; Lee and Comrey 1979).

After the factors are estimated, it is necessary to interpret them. Interpretation usually means assigning to each common factor a name that reflects the salience of the factor in predicting each of the observed variables, that is, the coefficients in the pattern matrix corresponding to the factor. Factor interpretation is a subjective process. It can sometimes be made less subjective by *rotating* the common factors, that is, by applying a nonsingular linear transformation. A rotated pattern matrix in which all the coefficients are close to 0 or ± 1 is easier to interpret than a pattern with many intermediate elements. Therefore, most rotation methods attempt to optimize a simplicity function of the rotated pattern matrix that measures, in some sense, how close the elements are to 0 or ± 1. Because the loading estimates are subject to sampling variability, it is useful to obtain the standard error estimates for the loadings for assessing the uncertainty due to random sampling. Notice that the salience of a factor loading refers to the magnitude of the loading, while statistical significance refers to the statistical evidence against a particular hypothetical value. A loading significantly different from 0 does not automatically mean it must be salient. For example, if salience is defined as a magnitude bigger than 0.4 while the entire 95% confidence interval for a loading lies between 0.1 and 0.3, the loading is statistically significant larger than 0 but it is not salient. Under the maximum likelihood method, you can obtain standard errors and confidence intervals for judging the salience of factor loadings.

After the initial factor extraction, the common factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation*, the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation*, the rotated factors become correlated. Oblique rotations often produce more useful patterns than do orthogonal rotations. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Rotating a set of factors does not change the statistical explanatory power of the factors. You cannot say that any rotation is better than any other rotation from a statistical point of view; all rotations, orthogonal or oblique, are equally good statistically. Therefore, the choice among different rotations must be based on nonstatistical grounds. For most applications, the preferred rotation is that which is most easily interpretable, or which is most compatible with substantive theories.

If two rotations give rise to different interpretations, those two interpretations must not be regarded as conflicting. Rather, they are two different ways of looking at the same thing, two different points of view in the common-factor space. Any conclusion that depends on one and only one rotation being correct is invalid.

Outline of Use

Principal Component Analysis

One important type of analysis performed by the FACTOR procedure is principal component analysis. The statements

```
proc factor;
run;
```

result in a principal component analysis. The output includes all the eigenvalues and the pattern matrix for eigenvalues greater than one.

Most applications require additional output. For example, you may want to compute principal component scores for use in subsequent analyses or obtain a graphical aid to help decide how many components to keep. You can save the results of the analysis in a permanent SAS data library by using the OUTSTAT= option. (Refer to the *SAS Language Reference: Dictionary* for more information on permanent SAS data libraries and librefs.) Assuming that your SAS data library has the libref save and that the data are in a SAS data set called raw, you could do a principal component analysis as follows:

```
proc factor data=raw method=principal scree mineigen=0 score
     outstat=save.fact_all;
run;
```

The SCREE option produces a plot of the eigenvalues that is helpful in deciding how many components to use. The MINEIGEN=0 option causes all components with variance greater than zero to be retained. The SCORE option requests that scoring coefficients be computed. The OUTSTAT= option saves the results in a specially structured SAS data set. The name of the data set, in this case fact_all, is arbitrary. To compute principal component scores, use the SCORE procedure:

```
proc score data=raw score=save.fact_all out=save.scores;
run;
```

The SCORE procedure uses the data and the scoring coefficients that are saved in save.fact_all to compute principal component scores. The component scores are placed in variables named Factor1, Factor2, . . . , Factor$n$ and are saved in the data set save.scores. If you know ahead of time how many principal components you want to use, you can obtain the scores directly from PROC FACTOR by specifying the NFACTORS= and OUT= options. To get scores from three principal components, specify

```
proc factor data=raw method=principal
     nfactors=3 out=save.scores;
run;
```

To plot the scores for the first three components, use the PLOT procedure:

```
proc plot;
   plot factor2*factor1 factor3*factor1 factor3*factor2;
run;
```

Principal Factor Analysis

The simplest and computationally most efficient method of common factor analysis is principal factor analysis, which is obtained the same way as principal component analysis except for the use of the PRIORS= option. The usual form of the initial analysis is

```
proc factor data=raw method=principal scree
     mineigen=0 priors=smc outstat=save.fact_all;
run;
```

The squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. If your correlation matrix is singular, you should specify PRIORS=MAX instead of PRIORS=SMC. The SCREE and MINEIGEN= options serve the same purpose as in the preceding principal component analysis. Saving the results with the OUTSTAT= option enables you to examine the eigenvalues and scree plot before deciding how many factors to rotate and to try several different rotations without re-extracting the factors. The OUTSTAT= data set is automatically marked TYPE=FACTOR, so the FACTOR procedure realizes that it contains statistics from a previous analysis instead of raw data.

After looking at the eigenvalues to estimate the number of factors, you can try some rotations. Two and three factors can be rotated with the following statements:

```
proc factor data=save.fact_all method=principal n=2
     rotate=promax reorder score outstat=save.fact_2;
proc factor data=save.fact_all method=principal n=3
     rotate=promax reorder score outstat=save.fact_3;
run;
```

The output data set from the previous run is used as input for these analyses. The options N=2 and N=3 specify the number of factors to be rotated. The specification ROTATE=PROMAX requests a promax rotation, which has the advantage of providing both orthogonal and oblique rotations with only one invocation of PROC FACTOR. The REORDER option causes the variables to be reordered in the output so that variables associated with the same factor appear next to each other.

You can now compute and plot factor scores for the two-factor promax-rotated solution as follows:

```
proc score data=raw score=save.fact_2 out=save.scores;
proc plot;
   plot factor2*factor1;
run;
```

Maximum Likelihood Factor Analysis

Although principal factor analysis is perhaps the most commonly used method of common factor analysis, most statisticians prefer maximum likelihood (ML) factor analysis (Lawley and Maxwell 1971). The ML method of estimation has desirable asymptotic properties (Bickel and Doksum 1977) and produces better estimates than principal factor analysis in large samples. You can test hypotheses about the number of common factors using the ML method. You can also obtain standard error and confidence interval estimates for many classes of rotated or unrotated factor loadings, factor correlations, and structure loadings under the ML theory.

The unrotated ML solution is equivalent to Rao's (1955) canonical factor solution and Howe's solution maximizing the determinant of the partial correlation matrix (Morrison 1976). Thus, as a descriptive method, ML factor analysis does not require a multivariate normal distribution. The validity of Bartlett's χ^2 test for the number of factors does require approximate normality plus additional regularity conditions that are usually satisfied in practice (Geweke and Singleton 1980).

Lawley and Maxwell (1971) derive the standard error formulas for unrotated loadings, while Archer and Jennrich (1973) and Jennrich (1973, 1974) derive the standard error formulas for several classes of rotated solutions. Extended results appear in Browne, Cudeck, Tateneni, and Mels (1998), Hayashi and Yung (1999), and Yung and Hayashi (2001). A combination of these methods is used to compute standard errors in an efficient manner. Confidence intervals are computed using the asymptotic normality of the estimates. To ensure that the confidence intervals are range respecting, transformation methods due to Browne (1982) are used. The validity of the standard error estimates and confidence limits requires the assumptions of multivariate normality and a fixed number of factors.

The ML method is more computationally demanding than principal factor analysis for two reasons. First, the communalities are estimated iteratively, and each iteration takes about as much computer time as principal factor analysis. The number of iterations typically ranges from about five to twenty. Second, if you want to extract different numbers of factors, as is often the case, you must run the FACTOR procedure once for each number of factors. Therefore, an ML analysis can take 100 times as long as a principal factor analysis. This does not include the time for computing standard error estimates, which is even more computationally demanding. For analyses with less than 35 variables, the computing time for the ML method, including the computation of standard errors, usually ranges from a few seconds to well under a minute. This seems to be a reasonable performance.

You can use principal factor analysis to get a rough idea of the number of factors before doing an ML analysis. If you think that there are between one and three factors, you can use the following statements for the ML analysis:

```
proc factor data=raw method=ml n=1
     outstat=save.fact1;
run;
proc factor data=raw method=ml n=2 rotate=promax
     outstat=save.fact2;
run;
proc factor data=raw method=ml n=3 rotate=promax
     outstat=save.fact3;
run;
```

The output data sets can be used for trying different rotations, computing scoring coefficients, or restarting the procedure in case it does not converge within the allotted number of iterations.

If you can determine how many factors should be retained before an analysis, you can get the standard errors and confidence limits to aid interpretations for the ML analysis:

```
proc factor data=raw method=ml n=3 rotate=quartimin se
   cover=.4;
run;
```

In the analysis, you define salience as a magnitude greater than 0.4. You can then use the coverage displays to determine the salience. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1327 for more details.

The ML method cannot be used with a singular correlation matrix, and it is especially prone to Heywood cases. (See the section "Heywood Cases and Other Anomalies" on page 1332 for a discussion of Heywood cases.) If you have problems with ML, the best alternative is to use the METHOD=ULS option for unweighted least-squares factor analysis.

Factor Rotation

After the initial factor extraction, the factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation*, the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation*, the rotated factors become correlated. Oblique rotations often produce more useful patterns than do orthogonal rotations. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Nowadays, most rotations are done analytically. There are many choices for orthogonal and oblique rotations. An excellent summary of a wide class of analytic rotations is in Crawford and Ferguson (1970). The Crawford-Ferguson family of orthogonal rotations includes the orthomax rotation as a subclass and the popular varimax rotation as a special case. For example, assuming that there are nine variables in the analysis, the following four specifications for orthogonal rotations give the same results:

```
    /* Orthogonal Crawford-Ferguson Family with
       variable parsimony weight = 8
       factor parsimony weight = 1  */
proc factor data=raw method=ml n=3 rotate=orthcf(8,1);
run;
    /* Orthomax without the GAMMA= option */
proc factor data=raw method=ml n=3 rotate=orthomax(1);
run;
    /* Orthomax without the GAMMA= option */
proc factor data=raw method=ml n=3 rotate=orthomax gamma=1;
run;
    /* Varimax */
proc factor data=raw method=ml n=3 rotate=varimax;
run;
```

You can also get the oblique versions of the varimax in two equivalent ways:

```
    /* Oblique Crawford-Ferguson Family with
       variable parsimony weight = 8
       factor parsimony weight = 1; */
proc factor data=raw method=ml n=3 rotate=oblicf(8,1);
run;
    /* Oblique Varimax */
proc factor data=raw method=ml n=3 rotate=obvarimax;
run;
```

Jennrich (1973) proposes a generalized Crawford-Ferguson family that includes the Crawford-Ferguson family and the (direct) oblimin family (refer to Harman 1976) as subclasses. The more well-known quartimin rotation is a special case of the oblimin class, and hence a special case of the generalized Crawford-Ferguson family. For example, the following four specifications of oblique rotations are equivalent:

```
   /* Oblique generalized Crawford-Ferguson Family
      with weights 0, 1, 0 , -1 */
proc factor data=raw method=ml n=3 rotate=obligencf(0,1,0,-1);
run;
   /* Oblimin family without the TAU= option */
proc factor data=raw method=ml n=3 rotate=oblimin(0);
run;
   /* Oblimin family with the TAU= option */
proc factor data=raw method=ml n=3 rotate=oblimin tau=0;
run;
   /* Quartimin */
proc factor data=raw method=ml n=3 rotate=quartimin;
run;
```

In addition to the generalized Crawford-Ferguson family, the available oblique rotation methods include Harris-Kaiser, promax, and Procrustean. See the section “Simplicity Functions for Rotations” on page 1329 for details about the definitions of various rotations. Refer to Harman (1976) and Mulaik (1972) for further information.

Getting Started

The following example demonstrates how you can use the FACTOR procedure to perform common factor analysis and use a transformation to rotate the extracted factors.

In this example, 103 police officers were rated by their supervisors on 14 scales (variables). You conduct a common factor analysis on these variables to see what latent factors are operating behind these ratings. The overall rating variable is excluded from the factor analysis.

The following DATA step creates the SAS data set jobratings:

```
options validvarname=any;
data jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
   datalines;
26838853879867
74758876857667
56757863775875
67869777988997
99997798878888
89897899888799
89999889899798
87794798468886
35652335143113
89888879576867
76557899446397
97889998898989
76766677598888
77667676779677
63839932588856
25738811284915
88879966797988
87979877959679
87989975878798
99889988898888
78876765687677
88889888899899
88889988878988
```

```
67646577384776
78778788799997
76888866768667
67678665746776
33424476664855
65656765785766
54566676565866
56655566656775
88889988868887
89899999898799
98889999899899
57554776468878
53687777797887
68666716475767
78778889798997
67364767565846
77678865886767
68698955669998
55546866663886
68888999998989
97787888798999
76677899799997
44754687877787
77876678798888
76668778799797
57653634361543
76777745653656
76766665656676
88888888878789
88977888869778
58894888747886
58674565473676
76777767777777
77788878789798
98989987999868
66729911474713
98889976999988
88786856667748
77868887897889
99999986999999
46688587616886
66755778486776
87777788889797
65666656545976
73574488887687
74755556586596
76677778789797
87878746777667
86776955874877
77888767778678
65778787778997
58786887787987
65787766676778
86777875468777
```

```
67788877757777
77778967855867
67887876767777
24786585535866
46532343542533
35566766676784
11231214211211
76886588536887
57784788688589
56667766465666
66787778778898
77687998877997
76668888546676
66477987589998
86788976884597
77868765785477
99988888987888
65948933886457
99999877988898
96636736876587
98676887798968
87878877898979
88897888888788
99997899799799
99899899899899
76656399567486
;
```

The following statements invoke the FACTOR procedure:

```
proc factor data=jobratings(drop='Overall Rating'n) priors=smc
   rotate=varimax;
run;
```

The DATA= option in PROC FACTOR specifies the SAS data set jobratings as the input data set. The DROP= option drops the Overall Rating variable from the analysis. To conduct a common factor analysis, you need to set the prior communality estimate to less than one for each variable. Otherwise, the factor solution would simply be a recast of the principal components solution, in which “factors” are linear combinations of observed variables. However, in the common factor model you always assume that observed variables are functions of underlying factors. In this example, the PRIORS= option specifies that the squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. Note that squared multiple correlations are usually less than one. By default, the principal factor extraction is used if the METHOD= option is not specified. To facilitate interpretations, the ROTATE= option specifies the VARIMAX orthogonal factor rotation to be used.

The output from the factor analysis is displayed in the following figures.

```
                              The FACTOR Procedure
                     Initial Factor Method: Principal Factors

                       Prior Communality Estimates: SMC

                                                       Judgment
Communication         Problem        Learning           Under      Observational
       Skills         Solving         Ability        Pressure             Skills

   0.62981394      0.58657431      0.61009871      0.63766021         0.67187583

      Willingness
     to Confront        Interest      Interpersonal           Desire for
         Problems       in People       Sensitivity     Self-Improvement

       0.64779805      0.75641519         0.75584891           0.57460176

                                                 Physical
         Appearance      Dependability            Ability       Integrity

         0.45505304         0.63449045         0.42245324      0.68195454


                     Eigenvalues of the Reduced Correlation Matrix:
                        Total = 8.06463816  Average = 0.62035678

                      Eigenvalue    Difference    Proportion    Cumulative

                 1    6.17760549    4.71531946        0.7660        0.7660
                 2    1.46228602    0.90183348        0.1813        0.9473
                 3    0.56045254    0.28093933        0.0695        1.0168
                 4    0.27951322    0.04766016        0.0347        1.0515
                 5    0.23185305    0.16113428        0.0287        1.0802
                 6    0.07071877    0.07489624        0.0088        1.0890
                 7    -.00417747    0.03387533       -0.0005        1.0885
                 8    -.03805279    0.04776534       -0.0047        1.0838
                 9    -.08581814    0.02438060       -0.0106        1.0731
                10    -.11019874    0.01452741       -0.0137        1.0595
                11    -.12472615    0.02356465       -0.0155        1.0440
                12    -.14829080    0.05823605       -0.0184        1.0256
                13    -.20652684                      -0.0256        1.0000

                3 factors will be retained by the PROPORTION criterion.
```

Figure 27.1. Table of Eigenvalues from PROC FACTOR

As displayed in Figure 27.1, the prior communality estimates are set to the squared multiple correlations. Figure 27.1 also displays the table of eigenvalues (the variances of the principal factors) of the reduced correlation matrix. Each row of the table pertains to a single eigenvalue. Following the column of eigenvalues are three measures of each eigenvalue's relative size and importance. The first of these displays the difference between the eigenvalue and its successor. The last two columns display the individual and cumulative proportions that the corresponding factor contributes to the total variation. The last line displayed in Figure 27.1 states that three factors are retained, as determined by the PROPORTION criterion.

```
                      Initial Factor Method: Principal Factors

                                  Factor Pattern

                                            Factor1         Factor2         Factor3

Communication Skills                        0.75441         0.07707        -0.25551
Problem Solving                             0.68590         0.08026        -0.34788
Learning Ability                            0.65904         0.34808        -0.25249
Judgment Under Pressure                     0.73391        -0.21405        -0.23513
Observational Skills                        0.69039         0.45292         0.10298
Willingness to Confront Problems            0.66458         0.47460         0.09210
Interest in People                          0.70770        -0.53427         0.10979
Interpersonal Sensitivity                   0.64668        -0.61284        -0.07582
Desire for Self-Improvement                 0.73820         0.12506         0.09062
Appearance                                  0.57188         0.20052         0.16367
Dependability                               0.79475        -0.04516         0.16400
Physical Ability                            0.51285         0.10251         0.34860
Integrity                                   0.74906        -0.35091         0.18656
```

Figure 27.2. Factor Pattern Matrix from PROC FACTOR

Figure 27.2 displays the initial factor pattern matrix. The factor pattern matrix represents standardized regression coefficients for predicting the variables using the extracted factors. Because the initial factors are uncorrelated, the pattern matrix is also equal to the correlations between variables and the common factors.

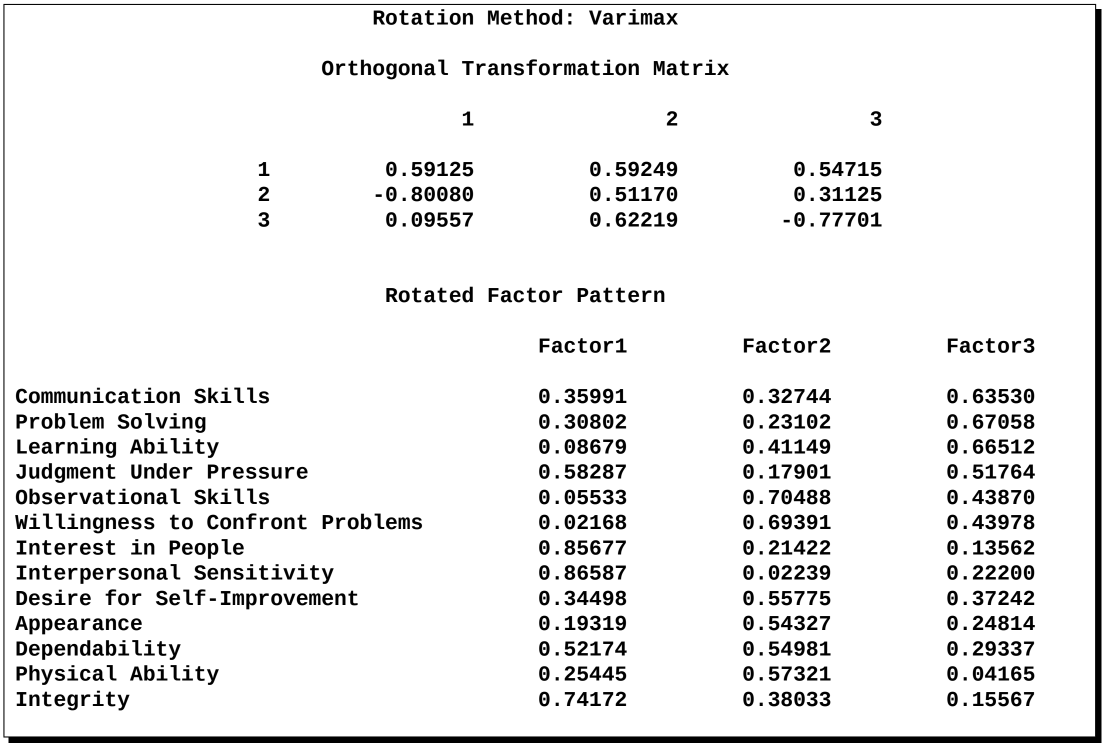
The pattern matrix suggests that Factor1 represents general ability. All loadings for Factor1 in the Factor Pattern are at least 0.5. Factor2 consists of high positive loadings on certain task-related skills (Willingness to Confront Problems, Observational Skills, and Learning Ability) and high negative loadings on some interpersonal skills (Interpersonal Sensitivity, Interest in People, and Integrity). This factor measures individuals' relative strength in these skills. Theoretically, individuals with high positive scores on this factor would exhibit better task-related skills than interpersonal skills. Individuals with high negative scores would exhibit better interpersonal skills than task-related skills. Individuals with scores near zero have those skills balanced. Factor3 does not have a cluster of very high or very low factor loadings. Therefore, interpreting this factor is difficult.

```
                    Initial Factor Method: Principal Factors

                        Variance Explained by Each Factor

                      Factor1         Factor2         Factor3

                    6.1776055       1.4622860       0.5604525


                  Final Communality Estimates: Total = 8.200344

                                                     Judgment
Communication         Problem        Learning           Under      Observational
       Skills         Solving         Ability        Pressure             Skills

   0.64036292      0.59791844      0.61924167      0.63972863         0.69237485

      Willingness
      to Confront         Interest      Interpersonal            Desire for
         Problems        in People        Sensitivity      Self-Improvement

       0.67538695       0.79833968         0.79951357            0.56879171

                                              Physical
         Appearance      Dependability         Ability       Integrity

         0.39403630         0.66056907      0.39504805      0.71903222
```

Figure 27.3. Variance Explained and Final Communality Estimates

Figure 27.3 displays the proportion of variance explained by each factor and the final communality estimates, including the total communality. The final communality estimates are the proportion of variance of the variables accounted for by the common factors. When the factors are orthogonal, the final communalities are calculated by taking the sum of squares of each row of the factor pattern matrix.

Figure 27.4 displays the results of the VARIMAX rotation of the three extracted factors and the corresponding orthogonal transformation matrix. The rotated factor pattern matrix is calculated by postmultiplying the original factor pattern matrix (Figure 27.2) by the transformation matrix.

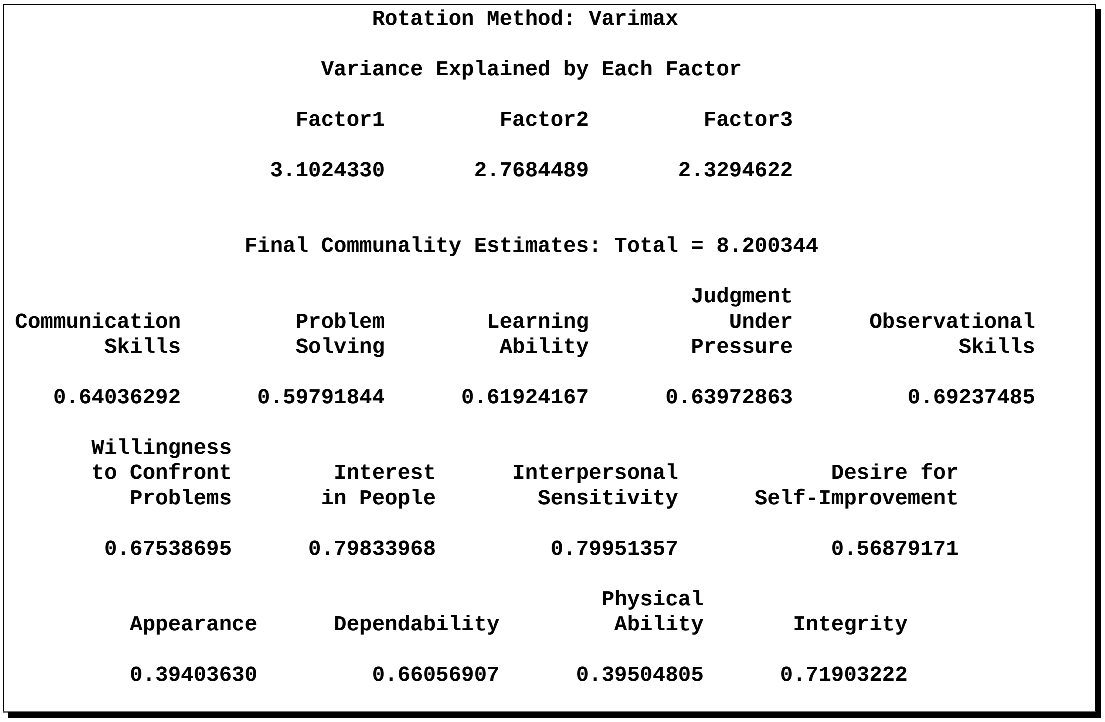


```
                         Rotation Method: Varimax

                     Orthogonal Transformation Matrix

                             1              2              3

                  1        0.59125        0.59249        0.54715
                  2       -0.80080        0.51170        0.31125
                  3        0.09557        0.62219       -0.77701


                          Rotated Factor Pattern

                                      Factor1         Factor2         Factor3

Communication Skills                  0.35991         0.32744         0.63530
Problem Solving                       0.30802         0.23102         0.67058
Learning Ability                      0.08679         0.41149         0.66512
Judgment Under Pressure               0.58287         0.17901         0.51764
Observational Skills                  0.05533         0.70488         0.43870
Willingness to Confront Problems      0.02168         0.69391         0.43978
Interest in People                    0.85677         0.21422         0.13562
Interpersonal Sensitivity             0.86587         0.02239         0.22200
Desire for Self-Improvement           0.34498         0.55775         0.37242
Appearance                            0.19319         0.54327         0.24814
Dependability                         0.52174         0.54981         0.29337
Physical Ability                      0.25445         0.57321         0.04165
Integrity                             0.74172         0.38033         0.15567
```

Figure 27.4. Transformation Matrix and Rotated Factor Pattern

The rotated factor pattern matrix is somewhat simpler to interpret. If a magnitude of at least 0.5 is required to indicate a salient variable-factor relationship, Factor1 now represents interpersonal skills (Interpersonal Sensitivity, Interest in People, Integrity, Judgment Under Pressure, and Dependability). Factor2 measures physical skills and job enthusiasm (Observational Skills, Willingness to Confront Problems, Physical Ability, Desire for Self-Improvement, Dependability, and Appearance). Factor3 measures cognitive skills (Communication Skills, Problem Solving, Learning Ability, and Judgment Under Pressure).

However, using 0.5 for determining a salient variable-factor relationship does not take sampling variability into account. If the underlying assumptions for the maximum likelihood estimation are approximately satisfied, you can output standard error estimates and the confidence intervals with METHOD=ML. You can then determine the salience of the variable-factor relationship using the coverage displays. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1327 for more details.



```
                            Rotation Method: Varimax

                        Variance Explained by Each Factor

                      Factor1         Factor2         Factor3

                    3.1024330       2.7684489       2.3294622


                  Final Communality Estimates: Total = 8.200344

                                                     Judgment
Communication         Problem        Learning           Under      Observational
       Skills         Solving         Ability        Pressure             Skills

   0.64036292      0.59791844      0.61924167      0.63972863         0.69237485

      Willingness
      to Confront         Interest      Interpersonal            Desire for
         Problems        in People        Sensitivity      Self-Improvement

       0.67538695       0.79833968         0.79951357            0.56879171

                                              Physical
         Appearance      Dependability         Ability       Integrity

         0.39403630         0.66056907      0.39504805      0.71903222
```

Figure 27.5. Variance Explained and Final Communality Estimates after Rotation

Figure 27.5 displays the variance explained by each factor and the final communality estimates after the orthogonal rotation. Even though the variances explained by the rotated factors are different from that of the unrotated factor (compare with Figure 27.3), the cumulative variance explained by the common factors remains the same. Note also that the final communalities for variables, as well as the total communality, remain unchanged after rotation. Although rotating a factor solution will not increase or decrease the statistical quality of the factor model, it may simplify the interpretations of the factors and redistribute the variance explained by the factors.

Syntax

You can specify the following statements with the FACTOR procedure:

PROC FACTOR < *options* > **;**
  VAR *variables* **;**
  PRIORS *communalities* **;**
  PARTIAL *variables* **;**
  FREQ *variable* **;**
  WEIGHT *variable* **;**
  BY *variables* **;**

Usually only the VAR statement is needed in addition to the PROC FACTOR statement. The descriptions of the BY, FREQ, PARTIAL, PRIORS, VAR, and WEIGHT statements follow the description of the PROC FACTOR statement in alphabetical order.

PROC FACTOR Statement

PROC FACTOR < *options* > **;**

The options available with the PROC FACTOR statement are listed in the following table and then are described in alphabetical order:

Table 27.1. Options Available in the PROC FACTOR Statement

Task	Option
Data sets	DATA= OUT= OUTSTAT= TARGET=
Extract factors and communalities	HEYWOOD METHOD= PRIORS= RANDOM= ULTRAHEYWOOD
Analyze data	ALPHA= COVARIANCE COVER= NOINT SE VARDEF= WEIGHT
Specify number of factors	MINEIGEN= NFACTORS= PROPORTION=
Specify numerical properties	CONVERGE= MAXITER= SINGULAR=
Specify rotation method and properties	GAMMA= HKPOWER= NORM= NOPROMAXNORM POWER= PREROTATE= RCONVERGE= RITER= ROTATE= TAU=
Control displayed output	ALL CORR EIGENVECTORS

Table 27.1. (continued)

Task	Option
	FLAG= FUZZ= MSA NOPRINT NPLOT= PLOT PREPLOT PRINT REORDER RESIDUALS ROUND SCORE SCREE SIMPLE
Exclude the correlation matrix from the OUTSTAT= data set	NOCORR
Miscellaneous	NOBS=

ALL

displays all optional output except plots. When the input data set is TYPE=CORR, TYPE=UCORR, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR, simple statistics, correlations, and MSA are not displayed.

ALPHA=*p*

specifies the level of confidence $1-p$ for interval construction. By default, p = 0.05, corresponding to $1-p$ = 95% confidence intervals. If p is greater than one, it is interpreted as a percentage and divided by 100. Because the coverage probability is not controlled simultaneously, you may consider supplying a nonconventional p using methods such as Bonferroni adjustment.

CONVERGE=*p*
CONV=*p*

specifies the convergence criterion for the METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, or METHOD=ML option. Iteration stops when the maximum change in the communalities is less than the value of the CONVERGE= option. The default value is 0.001. Negative values are not allowed.

CORR
C

displays the correlation matrix or partial correlation matrix.

COVARIANCE
COV

requests factoring of the covariance matrix instead of the correlation matrix. The COV option can be used only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option.

COVER <=*p*>
CI <=*p*>

computes the confidence intervals and optionally specifies the value of factor loading for coverage detection. By default, $p = 0$. The specified value is represented by an asterisk '*' in the coverage display. This is useful for determining the salience of loadings. For example, if COVER=.4, a display '0*[ ]' indicates that the entire confidence interval is above 0.4, implying strong evidence for the salience of the loading. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1327 for more details.

DATA=*SAS-data-set*

specifies the input data set, which can be an ordinary SAS data set or a specially structured SAS data set as described in the section "Input Data Set" beginning on page 1322. If the DATA= option is omitted, the most recently created SAS data set is used.

EIGENVECTORS
EV

displays the eigenvectors of the reduced correlation matrix, of which the diagonal elements are replaced with the communality estimates. When METHOD=ML, the eigenvectors are for the weighted reduced correlation matrix. PROC FACTOR chooses the solution that makes the sum of the elements of each eigenvector nonnegative. If the sum of the elements is equal to zero, then the sign depends on how the number is rounded off.

FLAG=*p*

flags absolute values larger than *p* with an asterisk in the correlation and loading matrices. Negative values are not allowed for *p*. Values printed in the matrices are multiplied by 100 and rounded to the nearest integer (see the ROUND option). The FLAG= option has no effect when standard errors or confidence intervals are also printed.

FUZZ=*p*

prints correlations and factor loadings with absolute values less than *p* printed as missing. For partial correlations, the FUZZ= value is divided by 2. For residual correlations, the FUZZ= value is divided by 4. The exact values in any matrix can be obtained from the OUTSTAT= and ODS output data sets. Negative values are not allowed. The FUZZ= option has no effect when standard errors or confidence intervals are also printed.

GAMMA=*p*

specifies the orthomax weight used with the option ROTATE=ORTHOMAX or PREROTATE=ORTHOMAX. Alternatively, you can use ROTATE=ORTHOMAX(*p*) with *p* representing the orthomax weight. There is no restriction on valid values for the orthomax weight, although the most common values are between 0 and the number of variables. The default GAMMA= value is one, resulting in the varimax rotation.

HEYWOOD
HEY

sets to 1 any communality greater than 1, allowing iterations to proceed.

HKPOWER=*p*
HKP=*p*

specifies the power of the square roots of the eigenvalues used to rescale the eigenvectors for Harris-Kaiser (ROTATE=HK) rotation, assuming that the factors are extracted by the principal factor method. If the principal factor method is not used for factor extraction, the eigenvectors are replaced by the normalized columns of the unrotated factor matrix, and the eigenvalues replaced by the column normalizing constants. HKPOWER= values between 0.0 and 1.0 are reasonable. The default value is 0.0, yielding the independent cluster solution, in which each variable tends to have a large loading on only one factor. An HKPOWER= value of 1.0 is equivalent to an orthogonal rotation, with the varimax rotation as the default. You can also specify the HKPOWER= option with ROTATE=QUARTIMAX, ROTATE=BIQUARTIMAX, ROTATE=EQUAMAX, or ROTATE=ORTHOMAX, and so on. The only restriction is that the Harris-Kaiser rotation must be associated with an orthogonal rotation.

MAXITER=*n*

specifies the maximum number of iterations for factor extraction. You can use the MAXITER= option with the PRINIT, ULS, ALPHA, or ML methods. The default is 30.

METHOD=*name*
M=*name*

specifies the method for extracting factors. The default is METHOD=PRINCIPAL unless the DATA= data set is TYPE=FACTOR, in which case the default is METHOD=PATTERN. Valid values for *name* are as follows:

ALPHA | A — produces alpha factor analysis.

HARRIS | H — yields Harris component analysis of $\mathbf{S}^{-1}\mathbf{R}\mathbf{S}^{-1}$ (Harris 1962), a noniterative approximation to canonical component analysis.

IMAGE | I — yields principal component analysis of the image covariance matrix, not Kaiser's (1963, 1970) or Kaiser and Rice's (1974) image analysis. A nonsingular correlation matrix is required.

ML | M — performs maximum likelihood factor analysis with an algorithm due, except for minor details, to Fuller (1987). The option METHOD=ML requires a nonsingular correlation matrix.

PATTERN — reads a factor pattern from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV or TYPE=UCOV data set. If you create a TYPE=FACTOR data set in a DATA step, only observations containing the factor pattern (_TYPE_='PATTERN') and, if the factors are correlated, the interfactor correlations (_TYPE_='FCORR') are required.

PRINCIPAL | PRIN | P — yields principal component analysis if no PRIORS option or statement is used or if you specify PRIORS=ONE; if you specify a

	PRIORS statement or a PRIORS= value other than PRIORS=ONE, a principal factor analysis is performed.
PRINIT	yields iterated principal factor analysis.
SCORE	reads scoring coefficients (_TYPE_='SCORE') from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV, or TYPE=UCOV data set. The data set must also contain either a correlation or a covariance matrix. Scoring coefficients are also displayed if you specify the OUT= option.
ULS | U	produces unweighted least squares factor analysis.

MINEIGEN=*p*
MIN=*p*

specifies the smallest eigenvalue for which a factor is retained. If you specify two or more of the MINEIGEN=, NFACTORS=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. The MINEIGEN= option cannot be used with either the METHOD=PATTERN or the METHOD=SCORE option. Negative values are not allowed. The default is 0 unless you omit both the NFACTORS= and the PROPORTION= options and one of the following conditions holds:

- If you specify the METHOD=ALPHA or METHOD=HARRIS option, then MINEIGEN=1.
- If you specify the METHOD=IMAGE option, then

$$\text{MINEIGEN} = \frac{\text{total image variance}}{\text{number of variables}}$$

- For any other METHOD= specification, if prior communality estimates of 1.0 are used, then

$$\text{MINEIGEN} = \frac{\text{total weighted variance}}{\text{number of variables}}$$

When an unweighted correlation matrix is factored, this value is 1.

MSA

produces the partial correlations between each pair of variables controlling for all other variables (the negative anti-image correlations) and Kaiser's measure of sampling adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977).

NFACTORS=*n*
NFACT=*n*
N=*n*

specifies the maximum number of factors to be extracted and determines the amount of memory to be allocated for factor matrices. The default is the number of variables. Specifying a number that is small relative to the number of variables can substantially decrease the amount of memory required to run PROC FACTOR, especially with oblique rotations. If you specify two or more of the NFACTORS=, MINEIGEN=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. If you specify the option NFACTORS=0, eigenvalues are computed, but no factors are extracted. If you specify the option NFACTORS=-1, neither eigenvalues nor factors are computed. You can use the NFACTORS= option with the METHOD=PATTERN or METHOD=SCORE option to specify a smaller number of factors than are present in the data set.

NOBS=*n*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= option overrides this default definition. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, the number of observations can be specified either by using the NOBS= option in the PROC FACTOR statement or by including a _TYPE_='N' observation in the DATA= input data set.

NOCORR

prevents the correlation matrix from being transferred to the OUTSTAT= data set when you specify the METHOD=PATTERN option. The NOCORR option greatly reduces memory requirements when there are many variables but few factors. The NOCORR option is not effective if the correlation matrix is required for other requested output; for example, if the scores or the residual correlations are displayed (using SCORE, RESIDUALS, ALL options).

NOINT

omits the intercept from the analysis; covariances or correlations are not corrected for the mean.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

NOPROMAXNORM | NOPMAXNORM

turns off the default row normalization of the pre-rotated factor pattern, which is used in computing the promax target matrix.

NORM=COV | KAISER | NONE | RAW | WEIGHT

specifies the method for normalizing the rows of the factor pattern for rotation. If you specify the option NORM=KAISER, Kaiser's normalization is used ($\sum_j p_{ij}^2 = 1$). If you specify the option NORM=WEIGHT, the rows are weighted by the Cureton-Mulaik technique (Cureton and Mulaik 1975). If you specify the option NORM=COV, the rows of the pattern matrix are rescaled to represent covariances instead of correlations. If you specify the option NORM=NONE or NORM=RAW, normalization is not performed. The default is NORM=KAISER.

NPLOT=*n*

specifies the number of factors to be plotted. The default is to plot all factors. The smallest allowable value is 2. If you specify the option NPLOT=n, all pairs of the first n factors are plotted, producing a total of $n(n-1)/2$ plots.

OUT=*SAS-data-set*

creates a data set containing all the data from the DATA= data set plus variables called Factor1, Factor2, and so on, containing estimated factor scores. The DATA= data set must contain multivariate data, not correlations or covariances. You must also specify the NFACTORS= option to determine the number of factor score variables. Note that OUT= option is disabled if you specify partial variables in the PARTIAL statement. In order to use the OUT= option with partialed variables, you can first regress the target variables on the partial variables. This can be done using PROC REG or PROC IML. You can then factor analyze the residuals without the PARTIAL statement. In this case, the OUT= option will not be disabled. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTSTAT=*SAS-data-set*

specifies an output data set containing most of the results of the analysis. The output data set is described in detail in the section "Output Data Sets" on page 1325. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

PLOT

plots the factor pattern after rotation.

POWER=*n*

specifies the power to be used in computing the target pattern for the option ROTATE=PROMAX. Valid values must be integers $\geq$ 1. The default value is 3. You can also specify the power= value in the ROTATE= option, e.g., ROTATE=PROMAX(4).

PREPLOT

plots the factor pattern before rotation.

PREROTATE=*name*
PRE=*name*

specifies the prerotation method for the option ROTATE=PROMAX. Any rotation method other than PROMAX or PROCRUSTES can be used. The default is PREROTATE=VARIMAX. If a previously rotated pattern is read using the option METHOD=PATTERN, you should specify the PREROTATE=NONE option.

PRINT

displays the input factor pattern or scoring coefficients and related statistics. In oblique cases, the reference and factor structures are computed and displayed. The PRINT option is effective only with the option METHOD=PATTERN or METHOD=SCORE.

PRIORS=*name*

specifies a method for computing prior communality estimates. You can specify numeric values for the prior communality estimates by using the PRIORS statement. Valid values for *name* are as follows:

ASMC | A	sets the prior communality estimates proportional to the squared multiple correlations but adjusted so that their sum is equal to that of the maximum absolute correlations (Cureton 1968).
INPUT | I	reads the prior communality estimates from the first observation with either _TYPE_='PRIORS' or _TYPE_='COMMUNAL' in the DATA= data set (which must be TYPE=FACTOR).
MAX | M	sets the prior communality estimate for each variable to its maximum absolute correlation with any other variable.
ONE | O	sets all prior communalities to 1.0.
RANDOM | R	sets the prior communality estimates to pseudo-random numbers uniformly distributed between 0 and 1.
SMC | S	sets the prior communality estimate for each variable to its squared multiple correlation with all other variables.

The default prior communality estimates are as follows:

METHOD=	**PRIORS=**
PRINCIPAL	ONE
PRINIT	ONE
ALPHA	SMC
ULS	SMC
ML	SMC
HARRIS	(not applicable)
IMAGE	(not applicable)
PATTERN	(not applicable)
SCORE	(not applicable)

By default, the options METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, and METHOD=ML stop iterating and set the number of factors to 0 if an estimated communality exceeds 1. The options HEYWOOD and ULTRAHEYWOOD allow processing to continue.

PROPORTION=*p*
PERCENT=*p*
P=*p*

specifies the proportion of common variance to be accounted for by the retained factors using the prior communality estimates. If the value is greater than one, it is interpreted as a percentage and divided by 100. The options PROPORTION=0.75 and PERCENT=75 are equivalent. The default value is 1.0 or 100%. You cannot specify the PROPORTION= option with the METHOD=PATTERN or METHOD=SCORE option. If you specify two or more of the PROPORTION=, NFACTORS=, and

MINEIGEN= options, the number of factors retained is the minimum number satisfying any of the criteria.

RANDOM=*n*

specifies a positive integer as a starting value for the pseudo-random number generator for use with the option PRIORS=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence. Valid values must be integers ≥ 1.

RCONVERGE=*p*
RCONV=*p*

specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E-9 by default and is modified by the RCONVERGE= value.

REORDER
RE

causes the rows (variables) of various factor matrices to be reordered on the output. Variables with their highest absolute loading (reference structure loading for oblique rotations) on the first factor are displayed first, from largest to smallest loading, followed by variables with their highest absolute loading on the second factor, and so on. The order of the variables in the output data set is not affected. The factors are not reordered.

RESIDUALS
RES

displays the residual correlation matrix and the associated partial correlation matrix. The diagonal elements of the residual correlation matrix are the unique variances.

RITER=*n*

specifies the maximum number of cycles n for factor rotation. Except for promax and Procrustes, you can use the RITER= option with all rotation methods. The default n is the maximum between 100 and 10 times of the number of variables.

ROTATE=*name*
R=*name*

specifies the rotation method. The default is ROTATE=NONE.

Valid *name*s for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(.5).

EQUAMAX | E specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of factors*/2.

FACTORPARSIMAX | FPA specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of variables*.

NONE | N specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1*,*p2*) | ORCF(*p1*,*p2*) specifies the orthogonal Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1329.

ORTHGENCF(*p1*,*p2*,*p3*,*p4*) | ORGENCF(*p1*,*p2*,*p3*,*p4*) specifies the orthogonal generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1329.

ORTHOMAX<(*p*)> | ORMAX<(*p*)> specifies the orthomax rotation with orthomax weight *p*. If ROTATE=ORTHOMAX is used, the default *p* value is 1 unless specified otherwise in the GAMMA= option. Alternatively, ROTATE=ORTHOMAX(*p*) specifies *p* as the orthomax weight or the GAMMA= value. See the definition of the orthomax weight in the section "Simplicity Functions for Rotations" on page 1329.

PARSIMAX | PA specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

$$\text{GAMMA} = \frac{\mathit{nvar} \times (\mathit{nfact} - 1)}{\mathit{nvar} + \mathit{nfact} - 2}$$

where *nvar* is the number of variables, and *nfact* is the number of factors.

QUARTIMAX | QMAX | Q specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *name*s for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=.5.

COVARIMIN | CVMIN specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

HK<(*p*)> | H<(*p*)> specifies Harris-Kaiser case II orthoblique rotation. When specifying this option, you can use the HKPOWER= option to set the power of the square roots of the eigenvalues by which the eigenvectors are scaled, assuming that the factors are extracted by the principal factor method. For other extraction methods, the unrotated factor pattern is column normalized. The power is then applied to the column normalizing constants, instead of the eigenvalues. You can also use ROTATE=HK(*p*), with *p* representing the HKPOWER= value. The default associated orthogonal rotation with ROTATE=HK is the varimax rotation without Kaiser normalization. You may associate the Harris-Kaiser with other orthogonal rotations using the ROTATE= option together with the HKPOWER= option.

OBBIQUARTIMAX | OBIQMAX specifies oblique biquartimax rotation.

OBEQUAMAX | OE specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*) xspecifies the oblique Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1329.

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*) specifies the oblique generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1329.

OBLIMIN<(*p*)> | OBMIN<(*p*)> specifies the oblimin rotation with oblimin weight *p*. If ROTATE=OBLIMIN is used, the default *p* value is zero unless specified otherwise in the TAU= option. Alternatively, ROTATE=OBLIMIN(*p*) specifies *p* as the oblimin weight or the TAU= value. See the definition of the oblimin weight in the section "Simplicity Functions for Rotations" on page 1329.

OBPARSIMAX | OPA specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX specifies oblique quartimax rotation. This is the same as the QUARTIMIN method.

OBVARIMAX | OV specifies oblique varimax rotation.

PROCRUSTES specifies oblique Procrustes rotation with the target pattern provided by the TARGET= data set. The unrestricted least squares method is used with factors scaled to unit variance after rotation.

PROMAX<(*p*)> | P<(*p*)> specifies oblique promax rotation. You can use the PREROTATE= option to set the desirable pre-rotation method, orthogonal or oblique. When using with ROTATE=PROMAX, the POWER= option lets you specify the power for forming the target. You can also use ROTATE=PROMAX(*p*), where *p* represents the POWER= value.

QUARTIMIN | QMIN specifies quartimin rotation. It is the same as the oblique quartimax method. It also corresponds to the specification ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

ROUND

prints correlation and loading matrices with entries multiplied by 100 and rounded to the nearest integer. The exact values can be obtained from the OUTSTAT= and ODS output data sets. The ROUND option also flags absolute values larger than the FLAG= value with an asterisk in correlation and loading matrices (see the FLAG= option). If the FLAG= option is not specified, the root mean square of all the values in the matrix printed is used as the default FLAG= value. The ROUND option has no effect when standard errors or confidence intervals are also printed.

SCORE

displays the factor scoring coefficients. The squared multiple correlation of each factor with the variables is also displayed except in the case of unrotated principal components. Unless you specify the NOINT option in PROC FACTOR, the scoring coefficients should be applied to standardized variables–variables that are centered by subtracting the original variable means and then divided by the original variable standard deviations. With the NOINT option, the scoring coefficients should be applied to data without centering.

SCREE

displays a scree plot of the eigenvalues (Cattell 1966, 1978; Cattell and Vogelman 1977; Horn and Engstrom 1979).

SE
STDERR

computes standard errors for various classes of unrotated and rotated solutions under the maximum likelihood estimation.

SIMPLE
S

displays means, standard deviations, and the number of observations.

SINGULAR=*p*
SING=*p*

specifies the singularity criterion, where $0 < p < 1$. The default value is 1E-8.

TARGET=*SAS-data-set*

specifies an input data set containing the target pattern for Procrustes rotation (see the description of the ROTATE= option). The TARGET= data set must contain variables with the same names as those being factored. Each observation in the TARGET= data set becomes one column of the target factor pattern. Missing values are treated as zeros. The _NAME_ and _TYPE_ variables are not required and are ignored if present.

TAU=*p*

specifies the oblimin weight used with the option ROTATE=OBLIMIN or PREROTATE=OBLIMIN. Alternatively, you can use ROTATE=OBLIMIN(*p*) with *p* representing the oblimin weight. There is no restriction on valid values for the oblimin weight, although for practical purposes a negative or zero value is recommended. The default TAU= value is 0, resulting in the quartimin rotation.

ULTRAHEYWOOD
ULTRA

allows communalities to exceed 1. The ULTRAHEYWOOD option can cause convergence problems because communalities can become extremely large, and ill-conditioned Hessians may occur.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table where i= 0 if the NOINT option is used and i= 1 otherwise, and where k is the number of partial variables specified in the PARTIAL statement.

Value	Description	Divisor
DF	degrees of freedom	$n - k - i$
N	number of observations	$n - k$
WDF	sum of weights DF	$\sum_i w_i - k - i$
WEIGHT | WGT	sum of weights	$\sum_i w_i - k$

WEIGHT

factors a weighted correlation or covariance matrix. The WEIGHT option can be used only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. The input data set must be of type CORR, UCORR, COV, UCOV, or FACTOR, and the variable weights are obtained from an observation with _TYPE_='WEIGHT'.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FACTOR to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FACTOR procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in Base SAS software). For more information on creating indexes and using the BY statement with indexed datasets, refer to "SAS Files" in *SAS Language Reference: Concepts*.

If you specify the TARGET= option and the TARGET= data set does not contain any of the BY variables, then the entire TARGET= data set is used as a Procrustean target for each BY group in the DATA= data set.

If the TARGET= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the TARGET= data set as in the DATA= data set, then PROC FACTOR displays an error message and stops.

If all the BY variables appear in the TARGET= data set with the same type and length as in the DATA= data set, then each BY group in the TARGET= data set is used as a Procrustean target for the corresponding BY group in the DATA= data set. The BY groups in the TARGET= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be identical BY groups in the same order in both data sets. If you do not specify the NOTSORTED option, some BY groups can appear in one data set but not in the other.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

The WEIGHT and FREQ statements have a similar effect, except in determining the number of observations for significance tests.

PARTIAL Statement

PARTIAL *variables* **;**

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables that are used to partial out the variables in the analysis.

PRIORS Statement

PRIORS *communalities* **;**

The PRIORS statement specifies numeric values between 0.0 and 1.0 for the prior communality estimates for each variable. The first numeric value corresponds to the first variable in the VAR statement, the second value to the second variable, and so on. The number of numeric values must equal the number of variables. For example,

```
proc factor;
   var     x  y  z;
   priors .7 .8 .9;
run;
```

You can specify various methods for computing prior communality estimates with the PRIORS= option of the PROC FACTOR statement. Refer to the description of that option for more information on the default prior communality estimates.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not specified in other statements are analyzed.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use relative weights for each observation in the input data set, specify a variable containing weights in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. If a variable value is negative or is missing, it is excluded from the analysis.

Details

Input Data Set

The FACTOR procedure can read an ordinary SAS data set containing raw data or a special data set specified as a TYPE=CORR, TYPE=UCORR, TYPE=SSCP, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR data set containing previously computed statistics. A TYPE=CORR data set can be created by the CORR procedure or various other procedures such as the PRINCOMP procedure. It contains means, standard deviations, the sample size, the correlation matrix, and possibly other statistics if it is created by some procedure other than PROC CORR. A TYPE=COV data set is similar to a TYPE=CORR data set but contains a covariance matrix. A TYPE=UCORR or TYPE=UCOV data set contains a correlation or covariance matrix that is not corrected for the mean. The default VAR variable list does not include

Intercept if the DATA= data set is TYPE=SSCP. If the Intercept variable is explicitly specified in the VAR statement with a TYPE=SSCP data set, the NOINT option is activated. A TYPE=FACTOR data set can be created by the FACTOR procedure and is described in the section "Output Data Sets" on page 1325.

If your data set has many observations and you plan to run FACTOR several times, you can save computer time by first creating a TYPE=CORR data set and using it as input to PROC FACTOR.

```
proc corr data=raw out=correl;     /* create TYPE=CORR data set */
proc factor data=correl method=ml; /* maximum likelihood        */
proc factor data=correl;           /* principal components      */
```

The data set created by the CORR procedure is automatically given the TYPE=CORR data set option, so you do not have to specify TYPE=CORR. However, if you use a DATA step with a SET statement to modify the correlation data set, you must use the TYPE=CORR attribute in the new data set. You can use a VAR statement with PROC FACTOR when reading a TYPE=CORR data set to select a subset of the variables or change the order of the variables.

Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each correlation from all observations that have values present for the pair of variables involved (pairwise deletion). The resulting correlation matrix may have negative eigenvalues. If you specify the NOMISS option with the CORR procedure, observations with any missing values are completely omitted from the calculations (listwise deletion), and there is no danger of negative eigenvalues.

PROC FACTOR can also create a TYPE=FACTOR data set, which includes all the information in a TYPE=CORR data set, and use it for repeated analyses. For a TYPE=FACTOR data set, the default value of the METHOD= option is PATTERN. The following statements produce the same PROC FACTOR results as the previous example:

```
proc factor data=raw method=ml outstat=fact; /* max. likelihood */
proc factor data=fact method=prin;      /* principal components */
```

You can use a TYPE=FACTOR data set to try several different rotation methods on the same data without repeatedly extracting the factors. In the following example, the second and third PROC FACTOR statements use the data set fact created by the first PROC FACTOR statement:

```
proc factor data=raw outstat=fact; /* principal components */
proc factor rotate=varimax;        /* varimax rotation     */
proc factor rotate=quartimax;      /* quartimax rotation   */
```

You can create a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement and include the _TYPE_ and _NAME_ variables. In a TYPE=CORR data set, only the correlation matrix (_TYPE_='CORR') is necessary. It can contain missing values as long as every pair of variables has at least one nonmissing value.

```
data correl(type=corr);
   _TYPE_='CORR';
   input _NAME_ $ x y z;
   datalines;
x  1.0  .   .
y   .7 1.0  .
z   .5  .4 1.0
;
proc factor;
run;
```

You can create a TYPE=FACTOR data set containing only a factor pattern (_TYPE_='PATTERN') and use the FACTOR procedure to rotate it.

```
data pat(type=factor);
   _TYPE_='PATTERN';
   input _NAME_ $ x y z;
   datalines;
factor1  .5  .7  .3
factor2  .8  .2  .8
;
proc factor rotate=promax prerotate=none;
run;
```

If the input factors are oblique, you must also include the interfactor correlation matrix with _TYPE_='FCORR'.

```
data pat(type=factor);
   input _TYPE_ $ _NAME_ $ x y z;
   datalines;
pattern factor1  .5  .7  .3
pattern factor2  .8  .2  .8
fcorr   factor1 1.0  .2  .
fcorr   factor2  .2 1.0  .
;
proc factor rotate=promax prerotate=none;
run;
```

Some procedures, such as the PRINCOMP and CANDISC procedures, produce TYPE=CORR or TYPE=UCORR data sets containing scoring coefficients (_TYPE_='SCORE' or _TYPE_= 'USCORE'). These coefficients can be input to PROC FACTOR and rotated by using the METHOD=SCORE option. The input data set must contain the correlation matrix as well as the scoring coefficients.

```
proc princomp data=raw n=2 outstat=prin;
run;
proc factor data=prin method=score rotate=varimax;
run;
```

Output Data Sets

The OUT= Data Set

The OUT= data set contains all the data in the DATA= data set plus new variables called Factor1, Factor2, and so on, containing estimated factor scores. Each estimated factor score is computed as a linear combination of the standardized values of the variables that are factored. The coefficients are always displayed if the OUT= option is specified and they are labeled "Standardized Scoring Coefficients."

The OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR or TYPE=UCORR data set produced by the CORR procedure, but it is a TYPE=FACTOR data set and it contains many results in addition to those produced by PROC CORR. The OUTSTAT= data set contains observations with _TYPE_='UCORR' and _TYPE_='USTD' if you specify the NOINT option.

The output data set contains the following variables:

- the BY variables, if any
- two new character variables, _TYPE_ and _NAME_
- the variables analyzed, that is, those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement.

Each observation in the output data set contains some type of statistic as indicated by the _TYPE_ variable. The _NAME_ variable is blank except where otherwise indicated. The values of the _TYPE_ variable are as follows:

TYPE	**Contents**
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations
N	sample size
CORR	correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the correlation matrix.
UCORR	uncorrected correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the uncorrected correlation matrix.
IMAGE	image coefficients. The _NAME_ variable contains the name of the variable corresponding to each row of the image coefficient matrix.
IMAGECOV	image covariance matrix. The _NAME_ variable contains the name of the variable corresponding to each row of the image covariance matrix.

COMMUNAL final communality estimates

PRIORS prior communality estimates, or estimates from the last iteration for iterative methods

WEIGHT variable weights

SUMWGT sum of the variable weights

EIGENVAL eigenvalues

UNROTATE unrotated factor pattern. The _NAME_ variable contains the name of the factor.

SE_UNROT standard error estimates for the unrotated loadings. The _NAME_ variable contains the name of the factor.

RESIDUAL residual correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the residual correlation matrix.

PRETRANS transformation matrix from prerotation. The _NAME_ variable contains the name of the factor.

PREFCORR pre-rotated interfactor correlations. The _NAME_ variable contains the name of the factor.

SE_PREFC standard error estimates for pre-rotated interfactor correlations. The _NAME_ variable contains the name of the factor.

PREROTAT pre-rotated factor pattern. The _NAME_ variable contains the name of the factor.

SE_PREPA standard error estimates for the pre-rotated loadings. The _NAME_ variable contains the name of the factor.

PRERCORR pre-rotated reference axis correlations. The _NAME_ variable contains the name of the factor.

PREREFER pre-rotated reference structure. The _NAME_ variable contains the name of the factor.

PRESTRUC pre-rotated factor structure. The _NAME_ variable contains the name of the factor.

SE_PREST standard error estimates for pre-rotated structure loadings. The _NAME_ variable contains the name of the factor.

PRESCORE pre-rotated scoring coefficients. The _NAME_ variable contains the name of the factor.

TRANSFOR transformation matrix from rotation. The _NAME_ variable contains the name of the factor.

FCORR interfactor correlations. The _NAME_ variable contains the name of the factor.

SE_FCORR standard error estimates for interfactor correlations. The _NAME_ variable contains the name of the factor.

PATTERN factor pattern. The _NAME_ variable contains the name of the factor.

SE_PAT standard error estimates for the rotated loadings. The _NAME_ variable contains the name of the factor.

RCORR reference axis correlations. The _NAME_ variable contains the name of the factor.

REFERENC reference structure. The _NAME_ variable contains the name of the factor.

STRUCTUR factor structure. The _NAME_ variable contains the name of the factor.

SE_STRUC standard error estimates for structure loadings. The _NAME_ variable contains the name of the factor.

SCORE scoring coefficients to be applied to standardized variables. The _NAME_ variable contains the name of the factor.

USCORE scoring coefficients to be applied without subtracting the mean from the raw variables. The _NAME_ variable contains the name of the factor.

Confidence Intervals and the Salience of Factor Loadings

The traditional approach to determining salient loadings (loadings that are considered large in absolute values) employs rules-of-thumb such as 0.3 or 0.4. However, this does not utilize the statistical evidence efficiently. The asymptotic normality of the distribution of factor loadings enables you to construct confidence intervals to gauge the salience of factor loadings. To guarantee the range-respecting properties of confidence intervals, a transformation procedure such as in CEFA (Browne, Cudeck, Tateneni, and Mels 1998) is used. For example, because the orthogonal rotated factor loading θ must be bounded between -1 and $+1$, the Fisher transformation

$$\varphi = \frac{1}{2}\log(\frac{1+\theta}{1-\theta})$$

is employed so that φ is an unbounded parameter. Assuming the asymptotic normality of $\hat{\varphi}$, a symmetric confidence interval for φ is constructed. Then, a back-transformation on the confidence limits yields an asymmetric confidence interval for θ. Applying the results of Browne (1982), a $(1-\alpha)100\%$ confidence interval for the orthogonal factor loading θ is

$$(\hat{\theta}_l = \frac{a/b-1}{a/b+1},\ \hat{\theta}_u = \frac{a\times b-1}{a\times b+1})$$

where

$$a = \frac{1+\hat{\theta}}{1-\hat{\theta}},\ \ b = \exp(z_{\alpha/2} \times \frac{2\hat{\sigma}}{1-\hat{\theta}^2})$$

and $\hat{\theta}$ is the estimated factor loading, $\hat{\sigma}$ is the standard error estimate of the factor loading, and $z_{\alpha/2}$ is the $(1 - \alpha/2)100$ percentile point of a standard normal distribution.

Once the confidence limits are constructed, you can use the corresponding coverage displays for determining the salience of the variable-factor relationship. In a coverage display, the COVER= value is represented by an asterisk '*'. The following table summarizes the various displays and their interpretations:

Table 27.2. Interpretations of the Coverage Displays

Positive Estimate	**Negative Estimate**	**COVER=0 specified**	**Interpretation**
[0]*	*[0]		The estimate is not significantly different from zero and the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the non-salience of the variable-factor relationship.
0[ ]*	*[ ]0		The estimate is significantly different from zero but the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the non-salience of the variable-factor relationship.
[0*]	[*0]	[0]	The estimate is not significantly different from zero or the COVER= value. The population value might have been larger or smaller in magnitude than the COVER= value. There is no statistical evidence for the salience of the variable-factor relationship.
0[*]	[*]0		The estimate is significantly different from zero but not from the COVER= value. This is marginal statistical evidence for the salience of the variable-factor relationship.
0*[ ]	[ ]*0	0[ ] or [ ]0	The estimate is significantly different from zero and the CI covers a region of values that are larger in magnitude than the COVER= value. This is strong statistical evidence for the salience of the variable-factor relationship.

See Example 27.4 on page 1369 for an illustration of the use of confidence intervals for interpreting factors.

Simplicity Functions for Rotations

To rotate a factor pattern is to apply a non-singular linear transformation to the unrotated factor pattern matrix. To arrive at an optimal transformation you must define a so-called simplicity function for assessing the optimal point. For the promax or the Procrustean transformation, the simplicity function is defined as the sum of squared differences between the rotated factor pattern and the target matrix. Thus, the solution of the optimal transformation is easily obtained by the familiar least-squares method.

For the class of the generalized Crawford-Ferguson family (Jennrich 1973), the simplicity function being optimized is

$$f = k_1 Z + k_2 H + k_3 V + k_4 Q$$

where

$$Z = (\sum_j \sum_i b_{ij}^2)^2, \quad H = \sum_i (\sum_j b_{ij}^2)^2$$

$$V = \sum_j (\sum_i b_{ij}^2)^2, \quad Q = \sum_j \sum_i b_{ij}^4$$

k_1, k_2, k_3, and k_4 are constants, and b_{ij} represents an element of the rotated pattern matrix. Except for specialized research purposes, it is rare in practice to use this simplicity function for rotation. However, it reduces to many well-known classes and special cases of rotations. One of these is the Crawford-Ferguson family (Crawford and Ferguson 1970), which minimizes

$$f_{cf} = c_1(H - Q) + c_2(V - Q)$$

where c_1 and c_2 are constants and $(H - Q)$ represents variable (row) parsimony and $(V - Q)$ represents factor (column) parsimony. Therefore, the relative importance of both the variable parsimony and of the factor parsimony is adjusted using the constants c_1 and c_2. The orthomax class (Carroll, see Harman 1976) maximizes the function

$$f_{or} = pQ - \gamma V$$

where γ is the orthomax weight and is usually between 0 and the number of variables p. The oblimin class minimizes the function

$$f_{ob} = p(H - Q) - \tau(Z - V)$$

where τ is the oblimin weight. For practical purposes, a negative or zero value for τ is recommended.

All the above definitions are for rotations without row normalization. For rotations with Kaiser normalization the definition of b_{ij} is replaced by b_{ij}/h_i, where h_i is the communality of variable i.

Missing Values

If the DATA= data set contains data (rather than a matrix or factor pattern), then observations with missing values for any variables in the analysis are omitted from the computations. If a correlation or covariance matrix is read, it can contain missing values as long as every pair of variables has at least one nonmissing entry. Missing values in a pattern or scoring coefficient matrix are treated as zeros.

Cautions

- The amount of time that FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1,000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).
- No computer program is capable of reliably determining the optimal number of factors since the decision is ultimately subjective. You should not blindly accept the number of factors obtained by default; instead, use your own judgment to make a decision.
- Singular correlation matrices cause problems with the options PRIORS=SMC and METHOD=ML. Singularities can result from using a variable that is the sum of other variables, coding too many dummy variables from a classification variable, or having more variables than observations.
- If you use the CORR procedure to compute the correlation matrix and there are missing data and the NOMISS option is not specified, then the correlation matrix may have negative eigenvalues.
- If a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set is copied or modified using a DATA step, the new data set does not automatically have the same TYPE as the old data set. You must specify the TYPE= data set option in the DATA statement. If you try to analyze a data set that has lost its TYPE=CORR attribute, PROC FACTOR displays a warning message saying that the data set contains _NAME_ and _TYPE_ variables but analyzes the data set as an ordinary SAS data set.
- For a TYPE=FACTOR data set, the default is METHOD=PATTERN, not METHOD=PRIN.

Factor Scores

The FACTOR procedure can compute estimated factor scores directly if you specify the NFACTORS= and OUT= options, or indirectly using the SCORE procedure. The latter method is preferable if you use the FACTOR procedure interactively to determine the number of factors, the rotation method, or various other aspects of the analysis. To compute factor scores for each observation using the SCORE procedure,

- use the SCORE option in the PROC FACTOR statement
- create a TYPE=FACTOR output data set with the OUTSTAT= option

- use the SCORE procedure with both the raw data and the TYPE=FACTOR data set
- do not use the TYPE= option in the PROC SCORE statement

For example, the following statements could be used:

```
proc factor data=raw score outstat=fact;
run;
proc score data=raw score=fact out=scores;
run;
```

or

```
proc corr   data=raw out=correl;
run;
proc factor data=correl score outstat=fact;
run;
proc score  data=raw score=fact out=scores;
run;
```

A component analysis (principal, image, or Harris) produces scores with mean zero and variance one. If you have done a common factor analysis, the true factor scores have mean zero and variance one, but the computed factor scores are only estimates of the true factor scores. These estimates have mean zero but variance equal to the squared multiple correlation of the factor with the variables. The estimated factor scores may have small nonzero correlations even if the true factors are uncorrelated.

Variable Weights and Variance Explained

A principal component analysis of a correlation matrix treats all variables as equally important. A principal component analysis of a covariance matrix gives more weight to variables with larger variances. A principal component analysis of a covariance matrix is equivalent to an analysis of a weighted correlation matrix, where the weight of each variable is equal to its variance. Variables with large weights tend to have larger loadings on the first component and smaller residual correlations than variables with small weights.

You may want to give weights to variables using values other than their variances. Mulaik (1972) explains how to obtain a maximally reliable component by means of a weighted principal component analysis. With the FACTOR procedure, you can indirectly give arbitrary weights to the variables by using the COV option and rescaling the variables to have variance equal to the desired weight, or you can give arbitrary weights directly by using the WEIGHT option and including the weights in a TYPE=CORR data set.

Arbitrary variable weights can be used with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. Alpha and ML factor analyses compute variable weights based on the communalities (Harman

1976, pp. 217-218). For alpha factor analysis, the weight of a variable is the reciprocal of its communality. In ML factor analysis, the weight is the reciprocal of the uniqueness. Harris component analysis uses weights equal to the reciprocal of one minus the squared multiple correlation of each variable with the other variables.

For uncorrelated factors, the variance explained by a factor can be computed with or without taking the weights into account. The usual method for computing variance accounted for by a factor is to take the sum of squares of the corresponding column of the factor pattern, yielding an unweighted result. If the square of each loading is multiplied by the weight of the variable before the sum is taken, the result is the weighted variance explained, which is equal to the corresponding eigenvalue except in image analysis. Whether the weighted or unweighted result is more important depends on the purpose of the analysis.

In the case of correlated factors, the variance explained by a factor can be computed with or without taking the other factors into account. If you want to ignore the other factors, the variance explained is given by the weighted or unweighted sum of squares of the appropriate column of the factor structure since the factor structure contains simple correlations. If you want to subtract the variance explained by the other factors from the amount explained by the factor in question (the Type II variance explained), you can take the weighted or unweighted sum of squares of the appropriate column of the reference structure because the reference structure contains semipartial correlations. There are other ways of measuring the variance explained. For example, given a prior ordering of the factors, you can eliminate from each factor the variance explained by previous factors and compute a Type I variance explained. Harman (1976, pp. 268-270) provides another method, which is based on direct and joint contributions.

Heywood Cases and Other Anomalies

Since communalities are squared correlations, you would expect them always to lie between 0 and 1. It is a mathematical peculiarity of the common factor model, however, that final communality estimates may exceed 1. If a communality equals 1, the situation is referred to as a Heywood case, and if a communality exceeds 1, it is an ultra-Heywood case. An ultra-Heywood case implies that some unique factor has negative variance, a clear indication that something is wrong. Possible causes include

- bad prior communality estimates
- too many common factors
- too few common factors
- not enough data to provide stable estimates
- the common factor model is not an appropriate model for the data

An ultra-Heywood case renders a factor solution invalid. Factor analysts disagree about whether or not a factor solution with a Heywood case can be considered legitimate.

Theoretically, the communality of a variable should not exceed its reliability. Violation of this condition is called a quasi-Heywood case and should be regarded with the same suspicion as an ultra-Heywood case.

Elements of the factor structure and reference structure matrices can exceed 1 only in the presence of an ultra-Heywood case. On the other hand, an element of the factor pattern may exceed 1 in an oblique rotation.

The maximum likelihood method is especially susceptible to quasi- or ultra-Heywood cases. During the iteration process, a variable with high communality is given a high weight; this tends to increase its communality, which increases its weight, and so on.

It is often stated that the squared multiple correlation of a variable with the other variables is a lower bound to its communality. This is true if the common factor model fits the data perfectly, but it is not generally the case with real data. A final communality estimate that is less than the squared multiple correlation can, therefore, indicate poor fit, possibly due to not enough factors. It is by no means as serious a problem as an ultra-Heywood case. Factor methods using the Newton-Raphson method can actually produce communalities less than 0, a result even more disastrous than an ultra-Heywood case.

The squared multiple correlation of a factor with the variables may exceed 1, even in the absence of ultra-Heywood cases. This situation is also cause for alarm. Alpha factor analysis seems to be especially prone to this problem, but it does not occur with maximum likelihood. If a squared multiple correlation is negative, there are too many factors retained.

With data that do not fit the common factor model perfectly, you can expect some of the eigenvalues to be negative. If an iterative factor method converges properly, the sum of the eigenvalues corresponding to rejected factors should be 0; hence, some eigenvalues are positive and some negative. If a principal factor analysis fails to yield any negative eigenvalues, the prior communality estimates are probably too large. Negative eigenvalues cause the cumulative proportion of variance explained to exceed 1 for a sufficiently large number of factors. The cumulative proportion of variance explained by the retained factors should be approximately 1 for principal factor analysis and should converge to 1 for iterative methods. Occasionally, a single factor can explain more than 100 percent of the common variance in a principal factor analysis, indicating that the prior communality estimates are too low.

If a squared canonical correlation or a coefficient alpha is negative, there are too many factors retained.

Principal component analysis, unlike common factor analysis, has none of these problems if the covariance or correlation matrix is computed correctly from a data set with no missing values. Various methods for missing value correlation or severe rounding of the correlations can produce negative eigenvalues in principal components.

Time Requirements

n = number of observations
v = number of variables
f = number of factors
i = number of iterations during factor extraction
r = length of iterations during factor rotation

The time required to compute. . .	**is roughly proportional to**
an overall factor analysis	iv^3
the correlation matrix	nv^2
PRIORS=SMC or ASMC	v^3
PRIORS=MAX	v^2
eigenvalues	v^3
final eigenvectors	fv^2
generalized Crawford-Ferguson family of rotations, PROMAX, or HK	rvf^2
ROTATE=PROCRUSTES	vf^2

Each iteration in the PRINIT or ALPHA method requires computation of eigenvalues and f eigenvectors.

Each iteration in the ML or ULS method requires computation of eigenvalues and $v - f$ eigenvectors.

The amount of time that PROC FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).

Displayed Output

PROC FACTOR output includes

- Mean and Std Dev (standard deviation) of each variable and the number of observations, if you specify the SIMPLE option
- Correlations, if you specify the CORR option
- Inverse Correlation Matrix, if you specify the ALL option

- Partial Correlations Controlling all other Variables (negative anti-image correlations), if you specify the MSA option. If the data are appropriate for the common factor model, the partial correlations should be small.
- Kaiser's Measure of Sampling Adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977) both overall and for each variable, if you specify the MSA option. The MSA is a summary of how small the partial correlations are relative to the ordinary correlations. Values greater than 0.8 can be considered good. Values less than 0.5 require remedial action, either by deleting the offending variables or by including other variables related to the offenders.
- Prior Communality Estimates, unless 1.0s are used or unless you specify the METHOD=IMAGE, METHOD=HARRIS, METHOD=PATTERN, or METHOD=SCORE option
- Squared Multiple Correlations of each variable with all the other variables, if you specify the METHOD=IMAGE or METHOD=HARRIS option
- Image Coefficients, if you specify the METHOD=IMAGE option
- Image Covariance Matrix, if you specify the METHOD=IMAGE option
- Preliminary Eigenvalues based on the prior communalities, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced includes the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.
- the number of factors that are retained, unless you specify the METHOD=PATTERN or METHOD=SCORE option
- the Scree Plot of Eigenvalues, if you specify the SCREE option. The preliminary eigenvalues are used if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option.
- the iteration history, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced contains the iteration number (Iter); the Criterion being optimized (Jöreskog 1977); the Ridge value for the iteration if you specify the METHOD=ML or METHOD=ULS option; the maximum Change in any communality estimate; and the Communalities.
- Significance tests, if you specify the option METHOD=ML, including Bartlett's Chi-square, df, and Prob $> \chi^2$ for H_0: No common factors and H_0: factors retained are sufficient to explain the correlations. The variables should have an approximate multivariate normal distribution for the probability levels to be valid. Lawley and Maxwell (1971) suggest that the number of observations should exceed the number of variables by fifty or more, although Geweke and Singleton (1980) claim that as few as ten observations are adequate with five variables and one common factor. Certain regularity conditions must also be satisfied for Bartlett's χ^2 test to be valid (Geweke and Singleton 1980), but in practice these conditions usually are satisfied. The notation Prob>chi**2 means "the probability under the null hypothesis of

obtaining a greater χ^2 statistic than that observed." The Chi-square value is displayed with and without Bartlett's correction.

- Akaike's Information Criterion, if you specify the METHOD=ML option. Akaike's information criterion (AIC) (Akaike 1973, 1974, 1987) is a general criterion for estimating the best number of parameters to include in a model when maximum likelihood estimation is used. The number of factors that yields the smallest value of AIC is considered best. Like the chi-square test, AIC tends to include factors that are statistically significant but inconsequential for practical purposes.
- Schwarz's Bayesian Criterion, if you specify the METHOD=ML option. Schwarz's Bayesian Criterion (SBC) (Schwarz 1978) is another criterion, similar to AIC, for determining the best number of parameters. The number of factors that yields the smallest value of SBC is considered best; SBC seems to be less inclined to include trivial factors than either AIC or the chi-square test.
- Tucker and Lewis's Reliability Coefficient, if you specify the METHOD=ML option (Tucker and Lewis 1973)
- Squared Canonical Correlations, if you specify the METHOD=ML option. These are the same as the squared multiple correlations for predicting each factor from the variables.
- Coefficient Alpha for Each Factor, if you specify the METHOD=ALPHA option
- Eigenvectors, if you specify the EIGENVECTORS or ALL option, unless you also specify the METHOD=PATTERN or METHOD=SCORE option
- Eigenvalues of the (Weighted) (Reduced) (Image) Correlation or Covariance Matrix, unless you specify the METHOD=PATTERN or METHOD=SCORE option. Included are the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.
- the Factor Pattern, which is equal to both the matrix of standardized regression coefficients for predicting variables from common factors and the matrix of correlations between variables and common factors since the extracted factors are uncorrelated. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Variance explained by each factor, both Weighted and Unweighted, if variable weights are used
- Final Communality Estimates, including the Total communality; or Final Communality Estimates and Variable Weights, including the Total communality, both Weighted and Unweighted, if variable weights are used. Final communality estimates are the squared multiple correlations for predicting the variables from the estimated factors, and they can be obtained by taking the sum of squares of each row of the factor pattern, or a weighted sum of squares if variable weights are used.
- Residual Correlations with Uniqueness on the Diagonal, if you specify the RESIDUAL or ALL option

- Root Mean Square Off-diagonal Residuals, both Over-all and for each variable, if you specify the RESIDUAL or ALL option
- Partial Correlations Controlling Factors, if you specify the RESIDUAL or ALL option
- Root Mean Square Off-diagonal Partials, both Over-all and for each variable, if you specify the RESIDUAL or ALL option
- Plots of Factor Pattern for unrotated factors, if you specify the PREPLOT option. The number of plots is determined by the NPLOT= option.
- Variable Weights for Rotation, if you specify the NORM=WEIGHT option
- Factor Weights for Rotation, if you specify the HKPOWER= option
- Orthogonal Transformation Matrix, if you request an orthogonal rotation
- Rotated Factor Pattern, if you request an orthogonal rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Variance explained by each factor after rotation. If you request an orthogonal rotation and if variable weights are used, both weighted and unweighted values are produced.
- Target Matrix for Procrustean Transformation, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option
- the Procrustean Transformation Matrix, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option
- the Normalized Oblique Transformation Matrix, if you request an oblique rotation, which, for the option ROTATE=PROMAX, is the product of the prerotation and the Procrustean rotation
- Inter-factor Correlations, if you specify an oblique rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Rotated Factor Pattern (Std Reg Coefs), if you specify an oblique rotation, giving standardized regression coefficients for predicting the variables from the factors. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Reference Axis Correlations if you specify an oblique rotation. These are the partial correlations between the primary factors when all factors other than the two being correlated are partialed out.
- Reference Structure (Semipartial Correlations), if you request an oblique rotation. The reference structure is the matrix of semipartial correlations (Kerlinger and Pedhazur 1973) between variables and common factors, removing from each common factor the effects of other common factors. If the common factors are uncorrelated, the reference structure is equal to the factor pattern.

- Variance explained by each factor eliminating the effects of all other factors, if you specify an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the reference structure corresponding to each factor.
- Factor Structure (Correlations), if you request an oblique rotation. The (primary) factor structure is the matrix of correlations between variables and common factors. If the common factors are uncorrelated, the factor structure is equal to the factor pattern. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Variance explained by each factor ignoring the effects of all other factors, if you request an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the factor structure corresponding to each factor.
- Final Communality Estimates for the rotated factors if you specify the ROTATE= option. The estimates should equal the unrotated communalities.
- Squared Multiple Correlations of the Variables with Each Factor, if you specify the SCORE or ALL option, except for unrotated principal components
- Standardized Scoring Coefficients, if you specify the SCORE or ALL option
- Plots of the Factor Pattern for rotated factors, if you specify the PLOT option and you request an orthogonal rotation. The number of plots is determined by the NPLOT= option.
- Plots of the Reference Structure for rotated factors, if you specify the PLOT option and you request an oblique rotation. The number of plots is determined by the NPLOT= option. Included are the Reference Axis Correlation and the Angle between the Reference Axes for each pair of factors plotted.

If you specify the ROTATE=PROMAX option, the output includes results for both the prerotation and the Procrustean rotation.

ODS Table Names

PROC FACTOR assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 27.3. ODS Tables Produced in PROC FACTOR

ODS Table Name	Description	Option
AlphaCoef	Coefficient alpha for each factor	METHOD=ALPHA
CanCorr	Squared canonical correlations	METHOD=ML
CondStdDev	Conditional standard deviations	SIMPLE w/PARTIAL

Table 27.3. (continued)

ODS Table Name	Description	Option
ConvergenceStatus	Convergence status	METHOD=PRINIT, =ALPHA, =ML, or =ULS
Corr	Correlations	CORR
Eigenvalues	Eigenvalues	default, SCREE
Eigenvectors	Eigenvectors	EIGENVECTORS
FactorWeightRotate	Factor weights for rotation	HKPOWER=
FactorPattern	Factor pattern	default
FactorStructure	Factor structure	ROTATE= any oblique rotation
FinalCommun	Final communalities	default
FinalCommunWgt	Final communalities with weights	METHOD=ML, METHOD=ALPHA
FitMeasures	Measures of fit	METHOD=ML
ImageCoef	Image coefficients	METHOD=IMAGE
ImageCov	Image covariance matrix	METHOD=IMAGE
ImageFactors	Image factor matrix	METHOD=IMAGE
InputFactorPattern	Input factor pattern	METHOD=PATTERN with PRINT or ALL
InputScoreCoef	Standardized input scoring coefficients	METHOD=SCORE with PRINT or ALL
InterFactorCorr	Inter-factor correlations	ROTATE= any oblique rotation
InvCorr	Inverse correlation matrix	ALL
IterHistory	Iteration history	METHOD=PRINIT, =ALPHA, =ML, or =ULS
MultipleCorr	Squared multiple correlations	METHOD=IMAGE or METHOD=HARRIS
NormObliqueTrans	Normalized oblique transformation matrix	ROTATE= any oblique rotation
ObliqueRotFactPat	Rotated factor pattern	ROTATE= any oblique rotation
ObliqueTrans	Oblique transformation matrix	HKPOWER=
OrthRotFactPat	Rotated factor pattern	ROTATE= any orthogonal rotation
OrthTrans	Orthogonal transformation matrix	ROTATE= any orthogonal rotation
ParCorrControlFactor	Partial correlations controlling factors	RESIDUAL
ParCorrControlVar	Partial correlations controlling other variables	MSA
PartialCorr	Partial correlations	MSA, CORR w/PARTIAL
PriorCommunalEst	Prior communality estimates	PRIORS=, METHOD=ML, METHOD=ALPHA
ProcrustesTarget	Target matrix for Procrustean transformation	ROTATE=PROCRUSTES, ROTATE=PROMAX
ProcrustesTrans	Procrustean transformation matrix	ROTATE=PROCRUSTES, ROTATE=PROMAX
RMSOffDiagPartials	Root mean square off-diagonal partials	RESIDUAL

Table 27.3. (continued)

ODS Table Name	Description	Option
RMSOffDiagResids	Root mean square off-diagonal residuals	RESIDUAL
ReferenceAxisCorr	Reference axis correlations	ROTATE= any oblique rotation
ReferenceStructure	Reference structure	ROTATE= any oblique rotation
ResCorrUniqueDiag	Residual correlations with uniqueness on the diagonal	RESIDUAL
SamplingAdequacy	Kaiser’s measure of sampling adequacy	MSA
SignifTests	Significance tests	METHOD=ML
SimpleStatistics	Simple statistics	SIMPLE
StdScoreCoef	Standardized scoring coefficients	SCORE
VarExplain	Variance explained	default
VarExplainWgt	Variance explained with weights	METHOD=ML, METHOD=ALPHA
VarFactorCorr	Squared multiple correlations of the variables with each factor	SCORE
VarWeightRotate	Variable weights for rotation	NORM=WEIGHT, ROTATE=

Examples

Example 27.1. Principal Component Analysis

The following example analyzes socioeconomic data provided by Harman (1976). The five variables represent total population, median school years, total employment, miscellaneous professional services, and median house value. Each observation represents one of twelve census tracts in the Los Angeles Standard Metropolitan Statistical Area.

The first analysis is a principal component analysis. Simple descriptive statistics and correlations are also displayed. This example produces Output 27.1.1:

```
data SocioEconomics;
   title ’Five Socioeconomic Variables’;
   title2 ’See Page 14 of Harman: Modern Factor Analysis, 3rd Ed’;
   input Population School Employment Services HouseValue;
   datalines;
5700     12.8      2500      270       25000
1000     10.9      600       10        10000
3400     8.8       1000      10        9000
3800     13.6      1700      140       25000
4000     12.8      1600      140       25000
8200     8.3       2600      60        12000
1200     11.4      400       10        16000
9100     11.5      3300      60        14000
```

```
9900     12.5      3400      180       18000
9600     13.7      3600      390       25000
9600     9.6       3300      80        12000
9400     11.4      4000      100       13000
;
```

```
proc factor data=SocioEconomics simple corr;
   title3 'Principal Component Analysis';
run;
```

There are two large eigenvalues, 2.8733 and 1.7967, which together account for 93.4% of the standardized variance. Thus, the first two principal components provide an adequate summary of the data for most purposes. Three components, explaining 97.7% of the variation, should be sufficient for almost any application. PROC FACTOR retains two components on the basis of the eigenvalues-greater-than-one rule since the third eigenvalue is only 0.2148.

The first component has large positive loadings for all five variables. The correlation with Services (0.93239) is especially high. The second component is a contrast of Population (0.80642) and Employment (0.72605) against School (-0.54476) and HouseValue (-0.55818), with a very small loading on Services (-0.10431).

The final communality estimates show that all the variables are well accounted for by two components, with final communality estimates ranging from 0.880236 for Services to 0.987826 for Population.

Output 27.1.1. Principal Component Analysis

```
                           Five Socioeconomic Variables
                 See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
                           Principal Component Analysis

                               The FACTOR Procedure

                  Means and Standard Deviations from 12 Observations

                        Variable            Mean       Std Dev

                        Population      6241.667     3439.9943
                        School            11.442        1.7865
                        Employment      2333.333     1241.2115
                        Services         120.833      114.9275
                        HouseValue     17000.000     6367.5313


                                   Correlations

               Population        School    Employment      Services    HouseValue

Population        1.00000       0.00975       0.97245       0.43887       0.02241
School            0.00975       1.00000       0.15428       0.69141       0.86307
Employment        0.97245       0.15428       1.00000       0.51472       0.12193
Services          0.43887       0.69141       0.51472       1.00000       0.77765
HouseValue        0.02241       0.86307       0.12193       0.77765       1.00000
```

Output 27.1.1. (continued)

```
                          Principal Component Analysis

                    Initial Factor Method: Principal Components

            Eigenvalues of the Correlation Matrix: Total = 5  Average = 1

                    Eigenvalue    Difference    Proportion    Cumulative

               1    2.87331359    1.07665350        0.5747        0.5747
               2    1.79666009    1.58182321        0.3593        0.9340
               3    0.21483689    0.11490283        0.0430        0.9770
               4    0.09993405    0.08467868        0.0200        0.9969
               5    0.01525537                      0.0031        1.0000


                                 Factor Pattern

                                   Factor1         Factor2

                     Population         0.58096         0.80642
                     School             0.76704        -0.54476
                     Employment         0.67243         0.72605
                     Services           0.93239        -0.10431
                     HouseValue         0.79116        -0.55818


                          Variance Explained by Each Factor

                               Factor1         Factor2

                             2.8733136       1.7966601


                    Final Community Estimates: Total = 4.669974

   Population          School      Employment         Services      HouseValue

   0.98782629      0.88510555      0.97930583       0.88023562      0.93750041
```

Example 27.2. Principal Factor Analysis

The following example uses the data presented in Example 27.1, and performs a principal factor analysis with squared multiple correlations for the prior communality estimates (PRIORS=SMC).

To help determine if the common factor model is appropriate, Kaiser's measure of sampling adequacy (MSA) is requested, and the residual correlations and partial correlations are computed (RESIDUAL). To help determine the number of factors, a scree plot (SCREE) of the eigenvalues is displayed, and the PREPLOT option plots the unrotated factor pattern.

The ROTATE= and REORDER options are specified to enhance factor interpretability. The ROTATE=PROMAX option produces an orthogonal varimax prerotation (default) followed by an oblique Procrustean rotation, and the REORDER option reorders the variables according to their largest factor loadings. An OUTSTAT= data set is created by PROC FACTOR and displayed in Output 27.2.16.

```
proc factor data=SocioEconomics
     priors=smc msa scree residual preplot
     rotate=promax reorder plot
     outstat=fact_all;
   title3 'Principal Factor Analysis with Promax Rotation';
run;
```

```
proc print;
   title3 'Factor Output Data Set';
run;
```

Output 27.2.1. Principal Factor Analysis

```
                  Principal Factor Analysis with Promax Rotation

                               The FACTOR Procedure
                     Initial Factor Method: Principal Factors

               Partial Correlations Controlling all other Variables

              Population        School    Employment      Services    HouseValue

Population       1.00000      -0.54465       0.97083       0.09612       0.15871
School          -0.54465       1.00000       0.54373       0.04996       0.64717
Employment       0.97083       0.54373       1.00000       0.06689      -0.25572
Services         0.09612       0.04996       0.06689       1.00000       0.59415
HouseValue       0.15871       0.64717      -0.25572       0.59415       1.00000


         Kaiser's Measure of Sampling Adequacy: Overall MSA = 0.57536759

   Population          School      Employment        Services      HouseValue

   0.47207897      0.55158839      0.48851137      0.80664365      0.61281377
```

Output 27.2.1. (continued)

```
                  Principal Factor Analysis with Promax Rotation

                     Initial Factor Method: Principal Factors

                       Prior Community Estimates: SMC

 Population          School      Employment        Services      HouseValue

 0.96859160      0.82228514      0.96918082      0.78572440      0.84701921


                  Eigenvalues of the Reduced Correlation Matrix:
                     Total = 4.39280116  Average = 0.87856023

                   Eigenvalue    Difference    Proportion    Cumulative

              1    2.73430084    1.01823217        0.6225        0.6225
              2    1.71606867    1.67650586        0.3907        1.0131
              3    0.03956281    0.06408626        0.0090        1.0221
              4    -.02452345    0.04808427       -0.0056        1.0165
              5    -.07260772                     -0.0165        1.0000
```

Output 27.2.1 displays the results of the principal factor extraction.

If the data are appropriate for the common factor model, the partial correlations controlling the other variables should be small compared to the original correlations. The partial correlation between the variables School and HouseValue, for example, is 0.65, slightly less than the original correlation of 0.86. The partial correlation between Population and School is -0.54, which is much larger in absolute value than the original correlation; this is an indication of trouble. Kaiser's MSA is a summary, for each variable and for all variables together, of how much smaller the partial correlations are than the original correlations. Values of 0.8 or 0.9 are considered good, while MSAs below 0.5 are unacceptable. The variables Population, School, and Employment have very poor MSAs. Only the Services variable has a good MSA. The overall MSA of 0.58 is sufficiently poor that additional variables should be included in the analysis to better define the common factors. A commonly used rule is that there should be at least three variables per factor. In the following analysis, there seems to be two common factors in these data, so more variables are needed for a reliable analysis.

The SMCs are all fairly large; hence, the factor loadings do not differ greatly from the principal component analysis.

The eigenvalues show clearly that two common factors are present. The first two largest positive eigenvalues account for 101.31% of the common variance. This is possible because the reduced correlation matrix, in general, needs not be positive definite, and negative eigenvalues for the matrix are possible. The scree plot displays a sharp bend at the third eigenvalue, reinforcing the preceding conclusion.

Output 27.2.2. Scree Plot

```
                 Principal Factor Analysis with Promax Rotation

                    Initial Factor Method: Principal Factors

Scree Plot of Eigenvalues
     |
     |
   3 +
     |
     |                  1
     |
     |
     |
E  2 +
i    |
g    |                              2
e    |
n    |
v    |
a  1 +
l    |
u    |
e    |
s    |
     |
   0 +                                          3           4           5
     |
     |
     |
     |
     |
  -1 +
     |
     -------+-----------+-----------+-----------+-----------+-----------+-------
            0           1           2           3           4           5

                                        Number
```

Output 27.2.3. Factor Pattern Matrix and Communalities

```
                  Principal Factor Analysis with Promax Rotation

                     Initial Factor Method: Principal Factors

                                  Factor Pattern

                                       Factor1         Factor2

                   Services            0.87899        -0.15847
                   HouseValue          0.74215        -0.57806
                   Employment          0.71447         0.67936
                   School              0.71370        -0.55515
                   Population          0.62533         0.76621


                          Variance Explained by Each Factor

                               Factor1         Factor2

                             2.7343008       1.7160687


                   Final Communality Estimates: Total = 4.450370

   Population          School      Employment        Services      HouseValue

   0.97811334      0.81756387      0.97199928      0.79774304      0.88494998
```

As displayed in Output 27.2.3, the principal factor pattern is similar to the principal component pattern seen in Example 27.1. For example, the variable Services has the largest loading on the first factor, and the Population variable has the smallest. The variables Population and Employment have large positive loadings on the second factor, and the HouseValue and School variables have large negative loadings.

The final communality estimates are all fairly close to the priors. Only the communality for the variable HouseValue increased appreciably, from 0.847019 to 0.884950. Nearly 100% of the common variance is accounted for. The residual correlations (off-diagonal elements) are low, the largest being 0.03 (Output 27.2.4). The partial correlations are not quite as impressive, since the uniqueness values are also rather small. These results indicate that the SMCs are good but not quite optimal communality estimates.

Output 27.2.4. Residual and Partial Correlations

```
                  Principal Factor Analysis with Promax Rotation

                     Initial Factor Method: Principal Factors

                Residual Correlations With Uniqueness on the Diagonal

              Population        School    Employment      Services    HouseValue

Population       0.02189      -0.01118       0.00514       0.01063       0.00124
School          -0.01118       0.18244       0.02151      -0.02390       0.01248
Employment       0.00514       0.02151       0.02800      -0.00565      -0.01561
Services         0.01063      -0.02390      -0.00565       0.20226       0.03370
HouseValue       0.00124       0.01248      -0.01561       0.03370       0.11505


           Root Mean Square Off-Diagonal Residuals: Overall = 0.01693282

   Population          School       Employment          Services       HouseValue

   0.00815307      0.01813027       0.01382764        0.02151737       0.01960158


                     Partial Correlations Controlling Factors

              Population        School    Employment      Services    HouseValue

Population       1.00000      -0.17693       0.20752       0.15975       0.02471
School          -0.17693       1.00000       0.30097      -0.12443       0.08614
Employment       0.20752       0.30097       1.00000      -0.07504      -0.27509
Services         0.15975      -0.12443      -0.07504       1.00000       0.22093
HouseValue       0.02471       0.08614      -0.27509       0.22093       1.00000
```

Output 27.2.5. Root Mean Square Off-Diagonal Partials

```
                  Principal Factor Analysis with Promax Rotation

                     Initial Factor Method: Principal Factors

           Root Mean Square Off-Diagonal Partials: Overall = 0.18550132

   Population          School       Employment          Services       HouseValue

   0.15850824      0.19025867       0.23181838        0.15447043       0.18201538
```

Output 27.2.6. Unrotated Factor Pattern Plot

```
                 Principal Factor Analysis with Promax Rotation

                    Initial Factor Method: Principal Factors

Plot of Factor Pattern for Factor1 and Factor2

                                     Factor1
                                        1

                                   D   .9

                                       .8
                      E
                       B               .7                   C
                                                              A
                                       .6

                                       .5

                                       .4

                                       .3

                                       .2
                                                                        F
                                       .1                               a
                                                                        c
        -1 -.9-.8-.7-.6-.5-.4-.3-.2-.1  0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0t
                                                                        o
                                      -.1                               r
                                                                        2
                                      -.2

                                      -.3

                                      -.4

                                      -.5

                                      -.6

                                      -.7

                                      -.8

                                      -.9

                                       -1

       Population=A  School=B      Employment=C  Services=D    HouseValue=E
```

As displayed in Output 27.2.6, the unrotated factor pattern reveals two tight clusters of variables, with the variables HouseValue and School at the negative end of Factor2 axis and the variables Employment and Population at the positive end. The Services variable is in between but closer to the HouseValue and School variables. A good rotation would put the reference axes through the two clusters.

Output 27.2.7. Varimax Rotation: Transform Matrix and Rotated Pattern

```
                  Principal Factor Analysis with Promax Rotation

                           Prerotation Method: Varimax

                         Orthogonal Transformation Matrix

                                         1                2

                            1      0.78895          0.61446
                            2     -0.61446          0.78895


                             Rotated Factor Pattern

                                      Factor1          Factor2

                    HouseValue        0.94072         -0.00004
                    School            0.90419          0.00055
                    Services          0.79085          0.41509
                    Population        0.02255          0.98874
                    Employment        0.14625          0.97499
```

Output 27.2.8. Varimax Rotation: Variance Explained and Communalities

```
                  Principal Factor Analysis with Promax Rotation

                           Prerotation Method: Varimax

                         Variance Explained by Each Factor

                              Factor1          Factor2

                            2.3498567        2.1005128


                   Final Communality Estimates: Total = 4.450370

    Population          School      Employment        Services      HouseValue

    0.97811334      0.81756387      0.97199928      0.79774304      0.88494998
```

Output 27.2.9. Varimax Rotated Factor Pattern Plot

```
                 Principal Factor Analysis with Promax Rotation

                          Prerotation Method: Varimax

Plot of Factor Pattern for Factor1 and Factor2

                                     Factor1
                                        1
                                       E
                                       .B

                                       .8          D

                                       .7

                                       .6

                                       .5

                                       .4

                                       .3

                                       .2
                                                                     C  F
                                       .1                               a
                                                                        c
        -1 -.9-.8-.7-.6-.5-.4-.3-.2-.1  0 .1 .2 .3 .4 .5 .6 .7 .8 .9 A.0t
                                                                        o
                                      -.1                               r
                                                                        2
                                      -.2

                                      -.3

                                      -.4

                                      -.5

                                      -.6

                                      -.7

                                      -.8

                                      -.9

                                       -1

       Population=A  School=B      Employment=C  Services=D    HouseValue=E
```

Output 27.2.7, Output 27.2.8, and Output 27.2.9 display the results of the varimax rotation. This rotation puts one axis through the variables HouseValue and School but misses the Population and Employment variables slightly.

Output 27.2.10. Promax Rotation: Procrustean Target and Transform Matrix

```
                  Principal Factor Analysis with Promax Rotation

                       Rotation Method: Promax (power = 3)

                   Target Matrix for Procrustean Transformation

                                    Factor1         Factor2

                   HouseValue       1.00000        -0.00000
                   School           1.00000         0.00000
                   Services         0.69421         0.10045
                   Population       0.00001         1.00000
                   Employment       0.00326         0.96793


                        Procrustean Transformation Matrix

                                         1               2

                           1      1.04116598      -0.0986534
                           2      -0.1057226      0.96303019
```

Output 27.2.11. Promax Rotation: Oblique Transform Matrix and Correlation

```
                  Principal Factor Analysis with Promax Rotation

                       Rotation Method: Promax (power = 3)

                     Normalized Oblique Transformation Matrix

                                         1               2

                           1         0.73803         0.54202
                           2        -0.70555         0.86528


                            Inter-Factor Correlations

                                  Factor1         Factor2

                    Factor1       1.00000         0.20188
                    Factor2       0.20188         1.00000
```

Output 27.2.12. Promax Rotation: Rotated Factor Pattern and Correlations

```
                  Principal Factor Analysis with Promax Rotation

                       Rotation Method: Promax (power = 3)

           Rotated Factor Pattern (Standardized Regression Coefficients)

                                 Factor1         Factor2

                    HouseValue      0.95558485      -0.0979201
                    School          0.91842142      -0.0935214
                    Services        0.76053238      0.33931804
                    Population      -0.0790832      1.00192402
                    Employment         0.04799      0.97509085


                          Reference Axis Correlations

                               Factor1         Factor2

                     Factor1         1.00000        -0.20188
                     Factor2        -0.20188         1.00000
```

Output 27.2.13. Promax Rotation: Variance Explained and Factor Structure

```
                  Principal Factor Analysis with Promax Rotation

                       Rotation Method: Promax (power = 3)

                  Reference Structure (Semipartial Correlations)

                                 Factor1         Factor2

                    HouseValue         0.93591        -0.09590
                    School             0.89951        -0.09160
                    Services           0.74487         0.33233
                    Population        -0.07745         0.98129
                    Employment         0.04700         0.95501


            Variance Explained by Each Factor Eliminating Other Factors

                             Factor1         Factor2

                           2.2480892       2.0030200


                        Factor Structure (Correlations)

                                 Factor1         Factor2

                    HouseValue         0.93582         0.09500
                    School             0.89954         0.09189
                    Services           0.82903         0.49286
                    Population         0.12319         0.98596
                    Employment         0.24484         0.98478
```

Output 27.2.14. Promax Rotation: Variance Explained and Final Communalities

```
                  Principal Factor Analysis with Promax Rotation

                       Rotation Method: Promax (power = 3)

             Variance Explained by Each Factor Ignoring Other Factors

                                Factor1         Factor2

                              2.4473495       2.2022803


                   Final Community Estimates: Total = 4.450370

   Population          School      Employment        Services      HouseValue

   0.97811334      0.81756387      0.97199928      0.79774304      0.88494998
```

Output 27.2.15. Promax Rotated Factor Pattern Plot

```
                 Principal Factor Analysis with Promax Rotation

                      Rotation Method: Promax (power = 3)

Plot of Reference Structure for Factor1 and Factor2
Reference Axis Correlation = -0.2019  Angle = 101.6471

                                     Factor1
                                        1
                                     E
                                     B .9

                                       .8
                                                 D
                                       .7

                                       .6

                                       .5

                                       .4

                                       .3

                                       .2
                                                                        F
                                       .1                               a
                                                                    C   c
        -1 -.9-.8-.7-.6-.5-.4-.3-.2-.1  0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0t
                                                                        o
                                      -.1                            A  r
                                                                        2
                                      -.2

                                      -.3

                                      -.4

                                      -.5

                                      -.6

                                      -.7

                                      -.8

                                      -.9

                                       -1

       Population=A  School=B      Employment=C  Services=D    HouseValue=E
```

The oblique promax rotation (Output 27.2.10 through Output 27.2.15) places an axis through the variables Population and Employment but misses the HouseValue and School variables. Since an independent-cluster solution would be possible if it were not for the variable Services, a Harris-Kaiser rotation weighted by the Cureton-Mulaik technique should be used.

Output 27.2.16. Output Data Set

```
                              Factor Output Data Set

                                                                           House
Obs   _TYPE_     _NAME_       Population    School  Employment  Services    Value

  1   MEAN                       6241.67   11.4417     2333.33   120.833 17000.00
  2   STD                        3439.99    1.7865     1241.21   114.928  6367.53
  3   N                            12.00   12.0000       12.00    12.000    12.00
  4   CORR       Population         1.00    0.0098        0.97     0.439     0.02
  5   CORR       School             0.01    1.0000        0.15     0.691     0.86
  6   CORR       Employment         0.97    0.1543        1.00     0.515     0.12
  7   CORR       Services           0.44    0.6914        0.51     1.000     0.78
  8   CORR       HouseValue         0.02    0.8631        0.12     0.778     1.00
  9   COMMUNAL                      0.98    0.8176        0.97     0.798     0.88
 10   PRIORS                        0.97    0.8223        0.97     0.786     0.85
 11   EIGENVAL                      2.73    1.7161        0.04    -0.025    -0.07
 12   UNROTATE   Factor1            0.63    0.7137        0.71     0.879     0.74
 13   UNROTATE   Factor2            0.77   -0.5552        0.68    -0.158    -0.58
 14   RESIDUAL   Population         0.02   -0.0112        0.01     0.011     0.00
 15   RESIDUAL   School            -0.01    0.1824        0.02    -0.024     0.01
 16   RESIDUAL   Employment         0.01    0.0215        0.03    -0.006    -0.02
 17   RESIDUAL   Services           0.01   -0.0239       -0.01     0.202     0.03
 18   RESIDUAL   HouseValue         0.00    0.0125       -0.02     0.034     0.12
 19   PRETRANS   Factor1            0.79   -0.6145         .         .         .
 20   PRETRANS   Factor2            0.61    0.7889         .         .         .
 21   PREROTAT   Factor1            0.02    0.9042        0.15     0.791     0.94
 22   PREROTAT   Factor2            0.99    0.0006        0.97     0.415    -0.00
 23   TRANSFOR   Factor1            0.74   -0.7055         .         .         .
 24   TRANSFOR   Factor2            0.54    0.8653         .         .         .
 25   FCORR      Factor1            1.00    0.2019         .         .         .
 26   FCORR      Factor2            0.20    1.0000         .         .         .
 27   PATTERN    Factor1           -0.08    0.9184        0.05     0.761     0.96
 28   PATTERN    Factor2            1.00   -0.0935        0.98     0.339    -0.10
 29   RCORR      Factor1            1.00   -0.2019         .         .         .
 30   RCORR      Factor2           -0.20    1.0000         .         .         .
 31   REFERENC   Factor1           -0.08    0.8995        0.05     0.745     0.94
 32   REFERENC   Factor2            0.98   -0.0916        0.96     0.332    -0.10
 33   STRUCTUR   Factor1            0.12    0.8995        0.24     0.829     0.94
 34   STRUCTUR   Factor2            0.99    0.0919        0.98     0.493     0.09
```

The output data set displayed in Output 27.2.16 can be used for Harris-Kaiser rotation by deleting observations with _TYPE_=’PATTERN’ and _TYPE_=’FCORR’, which are for the promax-rotated factors, and changing _TYPE_=’UNROTATE’ to _TYPE_=’PATTERN’. In this way, the initial orthogonal factor pattern matrix is saved in the observations with _TYPE_=’PATTERN’. The following factor analysis will then read in the factor pattern in the fact2 data set as an initial factor solution, which will then be rotated by the Harris-Kaiser rotation with Cureton-Mulaik weights.

The following statements produce Output 27.2.17:

```
data fact2(type=factor);
   set fact_all;
   if _TYPE_ in('PATTERN' 'FCORR') then delete;
   if _TYPE_='UNROTATE' then _TYPE_='PATTERN';


proc factor rotate=hk norm=weight reorder plot;
   title3 'Harris-Kaiser Rotation with Cureton-Mulaik Weights';
run;
```

The results of the Harris-Kaiser rotation are displayed in Output 27.2.17:

Output 27.2.17. Harris-Kaiser Rotation

```
                Harris-Kaiser Rotation with Cureton-Mulaik Weights

                             The FACTOR Procedure
                   Rotation Method: Harris-Kaiser (hkpower = 0)

                          Variable Weights for Rotation

Population          School      Employment        Services      HouseValue

0.95982747      0.93945424      0.99746396      0.12194766      0.94007263


                          Oblique Transformation Matrix

                                          1               2

                        1           0.73537         0.61899
                        2          -0.68283         0.78987


                            Inter-Factor Correlations

                                 Factor1         Factor2

                    Factor1      1.00000         0.08358
                    Factor2      0.08358         1.00000
```

Output 27.2.17. (continued)

```
                Harris-Kaiser Rotation with Cureton-Mulaik Weights

                   Rotation Method: Harris-Kaiser (hkpower = 0)

           Rotated Factor Pattern (Standardized Regression Coefficients)

                                       Factor1         Factor2

                    HouseValue         0.94048         0.00279
                    School             0.90391         0.00327
                    Services           0.75459         0.41892
                    Population        -0.06335         0.99227
                    Employment         0.06152         0.97885


                           Reference Axis Correlations

                                     Factor1         Factor2

                     Factor1         1.00000        -0.08358
                     Factor2        -0.08358         1.00000


                  Reference Structure (Semipartial Correlations)

                                       Factor1         Factor2

                    HouseValue         0.93719         0.00278
                    School             0.90075         0.00326
                    Services           0.75195         0.41745
                    Population        -0.06312         0.98880
                    Employment         0.06130         0.97543


            Variance Explained by Each Factor Eliminating Other Factors

                               Factor1         Factor2

                             2.2628537       2.1034731
```

Output 27.2.17. (continued)

```
                Harris-Kaiser Rotation with Cureton-Mulaik Weights

                   Rotation Method: Harris-Kaiser (hkpower = 0)

                         Factor Structure (Correlations)

                                       Factor1         Factor2

                    HouseValue         0.94071         0.08139
                    School             0.90419         0.07882
                    Services           0.78960         0.48198
                    Population         0.01958         0.98698
                    Employment         0.14332         0.98399


             Variance Explained by Each Factor Ignoring Other Factors

                               Factor1         Factor2

                             2.3468965       2.1875158


                   Final Communality Estimates: Total = 4.450370

    Population          School      Employment        Services      HouseValue

    0.97811334      0.81756387      0.97199928      0.79774304      0.88494998
```

Output 27.2.17. (continued)

```
               Harris-Kaiser Rotation with Cureton-Mulaik Weights

                  Rotation Method: Harris-Kaiser (hkpower = 0)

Plot of Reference Structure for Factor1 and Factor2
Reference Axis Correlation = -0.0836  Angle = 94.7941

                                     Factor1
                                        1
                                        E
                                       .B

                                       .8
                                                    D
                                       .7

                                       .6

                                       .5

                                       .4

                                       .3

                                       .2
                                                                        F
                                       .1                               a
                                                                     C  c
        -1 -.9-.8-.7-.6-.5-.4-.3-.2-.1  0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0t
                                                                     A  o
                                      -.1                               r
                                                                        2
                                      -.2

                                      -.3

                                      -.4

                                      -.5

                                      -.6

                                      -.7

                                      -.8

                                      -.9

                                       -1

       Population=A  School=B      Employment=C  Services=D    HouseValue=E
```

In the results of the Harris-Kaiser rotation, the variable Services receives a small weight, and the axes are placed as desired.

Example 27.3. Maximum Likelihood Factor Analysis

This example uses maximum likelihood factor analyses for one, two, and three factors. It is already apparent from the principal factor analysis that the best number of common factors is almost certainly two. The one- and three-factor ML solutions reinforce this conclusion and illustrate some of the numerical problems that can occur. The following statements produce Output 27.3.1:

```
proc factor data=SocioEconomics method=ml heywood n=1;
   title3 'Maximum Likelihood Factor Analysis with One Factor';
run;


proc factor data=SocioEconomics method=ml heywood n=2;
   title3 'Maximum Likelihood Factor Analysis with Two Factors';
run;


proc factor data=SocioEconomics method=ml heywood n=3;
   title3 'Maximum Likelihood Factor Analysis with Three Factors';
run;
```

Output 27.3.1. Maximum Likelihood Factor Analysis

```
               Maximum Likelihood Factor Analysis with One Factor

                              The FACTOR Procedure
                   Initial Factor Method: Maximum Likelihood

                      Prior Communality Estimates: SMC

   Population          School      Employment        Services      HouseValue

   0.96859160      0.82228514      0.96918082      0.78572440      0.84701921


        Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                  Eigenvalue    Difference    Proportion    Cumulative

             1    63.7010086    50.6462895        0.8369        0.8369
             2    13.0547191    12.7270798        0.1715        1.0084
             3     0.3276393     0.6749199        0.0043        1.0127
             4    -0.3472805     0.2722202       -0.0046        1.0081
             5    -0.6195007                    -0.0081        1.0000


Iteration    Criterion    Ridge   Change              Communalities

    1        6.5429218   0.0000   0.1033   0.93828   0.72227   1.00000   0.71940
                                           0.74371
    2        3.1232699   0.0000   0.7288   0.94566   0.02380   1.00000   0.26493
                                           0.01487
    3        3.1232699   0.0313   0.0000   0.94566   0.02380   1.00000   0.26493
                                           0.01487


        Convergence criterion satisfied.
```

Output 27.3.1. (continued)

```
                Maximum Likelihood Factor Analysis with One Factor

                    Initial Factor Method: Maximum Likelihood

                   Significance Tests Based on 12 Observations

                                                                    Pr >
                      Test                        DF    Chi-Square     ChiSq

         H0: No common factors                    10       54.2517    <.0001
         HA: At least one common factor
         H0: 1 Factor is sufficient                5       24.4656    0.0002
         HA: More factors are needed


             Chi-Square without Bartlett's Correction       34.355969
             Akaike's Information Criterion                 24.355969
             Schwarz's Bayesian Criterion                   21.931436
             Tucker and Lewis's Reliability Coefficient      0.120231


                           Squared Canonical Correlations

                                      Factor1

                                    1.0000000


                  Eigenvalues of the Weighted Reduced Correlation
                  Matrix: Total = -8.66E-15  Average = -2.165E-15

                               Eigenvalue    Difference

                          1         Infty         Infty
                          2    1.92716032    2.15547340
                          3    -.22831308    0.56464322
                          4    -.79295630    0.11293464
                          5    -.90589094
```

Output 27.3.1. (continued)

```
                Maximum Likelihood Factor Analysis with One Factor

                    Initial Factor Method: Maximum Likelihood

                                 Factor Pattern

                                              Factor1

                            Population      0.97244826
                            School          0.15428378
                            Employment               1
                            Services        0.51471836
                            HouseValue      0.12192599


                      Variance Explained by Each Factor

                     Factor        Weighted    Unweighted

                     Factor1     17.8010629    2.24926004


               Final Communality Estimates and Variable Weights
       Total Communality: Weighted = 17.801063   Unweighted = 2.249260

                   Variable      Communality        Weight

                   Population     0.94565561    18.4011648
                   School         0.02380349     1.0243839
                   Employment     1.00000000         Infty
                   Services       0.26493499     1.3604239
                   HouseValue     0.01486595     1.0150903
```

Output 27.3.1 displays the results of the analysis with one factor. The solution on the second iteration is so close to the optimum that PROC FACTOR cannot find a better solution, hence you receive this message:

```
Convergence criterion satisfied.
```

When this message appears, you should try rerunning PROC FACTOR with different prior communality estimates to make sure that the solution is correct. In this case, other prior estimates lead to the same solution or possibly to worse local optima, as indicated by the information criteria or the Chi-square values.

The variable Employment has a communality of 1.0 and, therefore, an infinite weight that is displayed next to the final communality estimate as a missing/infinite value. The first eigenvalue is also infinite. Infinite values are ignored in computing the total of the eigenvalues and the total final communality.

Output 27.3.2. Maximum Likelihood Factor Analysis: Two Factors

```
               Maximum Likelihood Factor Analysis with Two Factors

                              The FACTOR Procedure
                    Initial Factor Method: Maximum Likelihood

                       Prior Communality Estimates: SMC

   Population          School      Employment        Services      HouseValue

   0.96859160      0.82228514      0.96918082      0.78572440      0.84701921


         Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                    Eigenvalue    Difference    Proportion    Cumulative

               1    63.7010086    50.6462895        0.8369        0.8369
               2    13.0547191    12.7270798        0.1715        1.0084
               3     0.3276393     0.6749199        0.0043        1.0127
               4    -0.3472805     0.2722202       -0.0046        1.0081
               5    -0.6195007                      -0.0081        1.0000


Iteration    Criterion    Ridge    Change             Communalities

    1        0.3431221   0.0000    0.0471   1.00000   0.80672   0.95058   0.79348
                                            0.89412
    2        0.3072178   0.0000    0.0307   1.00000   0.80821   0.96023   0.81048
                                            0.92480
    3        0.3067860   0.0000    0.0063   1.00000   0.81149   0.95948   0.81677
                                            0.92023
    4        0.3067373   0.0000    0.0022   1.00000   0.80985   0.95963   0.81498
                                            0.92241
    5        0.3067321   0.0000    0.0007   1.00000   0.81019   0.95955   0.81569
                                            0.92187


         Convergence criterion satisfied.
```

Output 27.3.2. (continued)

```
               Maximum Likelihood Factor Analysis with Two Factors

                    Initial Factor Method: Maximum Likelihood

                   Significance Tests Based on 12 Observations

                                                                     Pr >
                      Test                       DF    Chi-Square    ChiSq

         H0: No common factors                   10       54.2517    <.0001
         HA: At least one common factor
         H0: 2 Factors are sufficient             1        2.1982    0.1382
         HA: More factors are needed


             Chi-Square without Bartlett's Correction       3.3740530
             Akaike's Information Criterion                 1.3740530
             Schwarz's Bayesian Criterion                   0.8891463
             Tucker and Lewis's Reliability Coefficient     0.7292200


                          Squared Canonical Correlations

                               Factor1         Factor2

                             1.0000000       0.9518891


                    Eigenvalues of the Weighted Reduced Correlation
                   Matrix: Total = 19.7853157  Average = 4.94632893

                     Eigenvalue    Difference    Proportion    Cumulative

              1           Infty         Infty
              2      19.7853143    19.2421292        1.0000        1.0000
              3       0.5431851     0.5829564        0.0275        1.0275
              4      -0.0397713     0.4636411       -0.0020        1.0254
              5      -0.5034124                     -0.0254        1.0000
```

Output 27.3.2. (continued)

```
      Maximum Likelihood Factor Analysis with Two Factors

           Initial Factor Method: Maximum Likelihood

                         Factor Pattern

                              Factor1         Factor2

           Population         1.00000         0.00000
           School             0.00975         0.90003
           Employment         0.97245         0.11797
           Services           0.43887         0.78930
           HouseValue         0.02241         0.95989


               Variance Explained by Each Factor

              Factor        Weighted    Unweighted

              Factor1     24.4329707    2.13886057
              Factor2     19.7853143    2.36835294


        Final Communality Estimates and Variable Weights
Total Communality: Weighted = 44.218285   Unweighted = 4.507214

            Variable      Communality        Weight

            Population     1.00000000         Infty
            School         0.81014489     5.2682940
            Employment     0.95957142    24.7246669
            Services       0.81560348     5.4256462
            HouseValue     0.92189372    12.7996793
```

Output 27.3.2 displays the results of the analysis using two factors. The analysis converges without incident. This time, however, the Population variable is a Heywood case.

Output 27.3.3. Maximum Likelihood Factor Analysis: Three Factors

```
             Maximum Likelihood Factor Analysis with Three Factors

                              The FACTOR Procedure
                    Initial Factor Method: Maximum Likelihood

                       Prior Communality Estimates: SMC

   Population          School      Employment        Services      HouseValue

   0.96859160      0.82228514      0.96918082      0.78572440      0.84701921


        Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                  Eigenvalue    Difference    Proportion    Cumulative

             1    63.7010086    50.6462895        0.8369        0.8369
             2    13.0547191    12.7270798        0.1715        1.0084
             3     0.3276393     0.6749199        0.0043        1.0127
             4    -0.3472805     0.2722202       -0.0046        1.0081
             5    -0.6195007                    -0.0081        1.0000


Iteration    Criterion    Ridge    Change              Communalities

    1        0.1798029   0.0313    0.0501   0.96081   0.84184   1.00000   0.80175
                                            0.89716
    2        0.0016405   0.0313    0.0678   0.98081   0.88713   1.00000   0.79559
                                            0.96500
    3        0.0000041   0.0313    0.0094   0.98195   0.88603   1.00000   0.80498
                                            0.96751
    4        0.0000000   0.0313    0.0006   0.98202   0.88585   1.00000   0.80561
                                            0.96735


        ERROR: Converged, but not to a proper optimum.
```

Output 27.3.3. (continued)

```
     Maximum Likelihood Factor Analysis with Three Factors

           Initial Factor Method: Maximum Likelihood

          Significance Tests Based on 12 Observations

                                                           Pr >
             Test                     DF    Chi-Square     ChiSq

H0: No common factors                 10       54.2517    <.0001
HA: At least one common factor
H0: 3 Factors are sufficient          -2        0.0000     .
HA: More factors are needed


    Chi-Square without Bartlett’s Correction       0.0000003
    Akaike’s Information Criterion                 4.0000003
    Schwarz’s Bayesian Criterion                   4.9698136
    Tucker and Lewis’s Reliability Coefficient     0.0000000


                 Squared Canonical Correlations

              Factor1         Factor2         Factor3

            1.0000000       0.9751895       0.6894465


         Eigenvalues of the Weighted Reduced Correlation
        Matrix: Total = 41.5254193  Average = 10.3813548

          Eigenvalue    Difference    Proportion    Cumulative

     1         Infty         Infty
     2    39.3054826    37.0854258        0.9465        0.9465
     3     2.2200568     2.2199693        0.0535        1.0000
     4     0.0000875     0.0002949        0.0000        1.0000
     5    -0.0002075                     -0.0000        1.0000
```

Output 27.3.3. (continued)

```
              Maximum Likelihood Factor Analysis with Three Factors

                    Initial Factor Method: Maximum Likelihood

                                 Factor Pattern

                           Factor1         Factor2         Factor3

         Population        0.97245        -0.11233        -0.15409
         School            0.15428         0.89108         0.26083
         Employment        1.00000         0.00000         0.00000
         Services          0.51472         0.72416        -0.12766
         HouseValue        0.12193         0.97227        -0.08473


                        Variance Explained by Each Factor

                      Factor        Weighted    Unweighted

                      Factor1     54.6115241    2.24926004
                      Factor2     39.3054826    2.27634375
                      Factor3      2.2200568    0.11525433


                 Final Communality Estimates and Variable Weights
      Total Communality: Weighted = 96.137063   Unweighted = 4.640858

                    Variable      Communality        Weight

                    Population     0.98201660    55.6066901
                    School         0.88585165     8.7607194
                    Employment     1.00000000         Infty
                    Services       0.80564301     5.1444261
                    HouseValue     0.96734687    30.6251078
```

The three-factor analysis displayed in Output 27.3.3 generates this message:

```
WARNING:  Too many factors for a unique solution.
```

The number of parameters in the model exceeds the number of elements in the correlation matrix from which they can be estimated, so an infinite number of different perfect solutions can be obtained. The Criterion approaches zero at an improper optimum, as indicated by this message:

```
Converged, but not to a proper optimum.
```

The degrees of freedom for the chi-square test are -2, so a probability level cannot be computed for three factors. Note also that the variable Employment is a Heywood case again.

The probability levels for the chi-square test are 0.0001 for the hypothesis of no common factors, 0.0002 for one common factor, and 0.1382 for two common factors. Therefore, the two-factor model seems to be an adequate representation. Akaike's information criterion and Schwarz's Bayesian criterion attain their minimum values at two common factors, so there is little doubt that two factors are appropriate for these data.

Example 27.4. Using Confidence Intervals to Locate Salient Factor Loadings

This example illustrates how you can utilize the standard errors and confidence intervals to understand the pattern of factor loadings under the maximum likelihood estimation. There are nine tests and you want a three-factor solution for a correlation matrix based on 200 observations. You apply quartimin rotation with (default) Kaiser normalization. You define loadings with magnitudes greater than 0.45 to be salient and use 90% confidence intervals to judge the salience.

```
data test(type=corr);
   title 'Quartimin-Rotated Factor Solution with Standard Errors';
   input _name_ $ test1-test9;
   _type_ = 'corr';
   datalines;
Test1      1  .561  .602  .290  .404  .328  .367  .179 -.268
Test2   .561     1  .743  .414  .526  .442  .523  .289 -.399
Test3   .602  .743     1  .286  .343  .361  .679  .456 -.532
Test4   .290  .414  .286     1  .677  .446  .412  .400 -.491
Test5   .404  .526  .343  .677     1  .584  .408  .299 -.466
Test6   .328  .442  .361  .446  .584     1  .333  .178 -.306
Test7   .367  .523  .679  .412  .408  .333     1  .711 -.760
Test8   .179  .289  .456  .400  .299  .178  .711     1 -.725
Test9  -.268 -.399 -.532 -.491 -.466 -.306 -.760 -.725     1
;
```

```
proc factor data=test method=ml reorder rotate=quartimin
   nobs=200 n=3 se cover=.45 alpha=.1;
   title2 'A nine-variable-three-factor example';
run;
```

Output 27.4.1. QuartiminRotated Factor Solution with Standard Errors

```
             Quartimin-Rotated Factor Solution with Standard Errors
                      A nine-variable-three-factor example

                              The FACTOR Procedure
                          Rotation Method: Quartimin

                          Inter-Factor Correlations
                         With 90% confidence limits
                       Estimate/StdErr/LowerCL/UpperCL

                             Factor1         Factor2         Factor3

            Factor1          1.00000         0.41283         0.38304
                             0.00000         0.06267         0.06060
                               .             0.30475         0.27919
                               .             0.51041         0.47804

            Factor2          0.41283         1.00000         0.47006
                             0.06267         0.00000         0.05116
                             0.30475           .             0.38177
                             0.51041           .             0.54986

            Factor3          0.38304         0.47006         1.00000
                             0.06060         0.05116         0.00000
                             0.27919         0.38177           .
                             0.47804         0.54986           .
```

After the quartimin rotation, the correlation matrix for factors is shown in Output 27.4.1. The factors are medium to highly correlated. The confidence intervals seem to be very wide, suggesting that the estimation of factor correlations may not be very accurate for this sample size. For example, the 90% confidence interval for the correlation between Factor1 and Factor2 is (0.30, 0.51), a range of 0.21. You may need a larger sample to get a narrower interval, or a better estimation.

Output 27.4.2. Interpretations of Factors Using Rotated Factor Pattern

```
                   A nine-variable-three-factor example

                         Rotation Method: Quartimin

         Rotated Factor Pattern (Standardized Regression Coefficients)
                 With 90% confidence limits; Cover |*| = 0.45?
               Estimate/StdErr/LowerCL/UpperCL/Coverage Display

                         Factor1         Factor2         Factor3

        test8            0.86810        -0.05045         0.00114
                         0.03282         0.03185         0.03087
                         0.80271        -0.10265        -0.04959
                         0.91286         0.00204         0.05187
                            0*[]            *[0]            [0]*

        test7            0.73204         0.27296         0.01098
                         0.04434         0.05292         0.03838
                         0.65040         0.18390        -0.05211
                         0.79697         0.35758         0.07399
                            0*[]            0[]*            [0]*

        test9           -0.79654        -0.01230        -0.17307
                         0.03948         0.04225         0.04420
                        -0.85291        -0.08163        -0.24472
                        -0.72180         0.05715        -0.09955
                            []*0            *[0]            *[]0

        test3            0.27715         0.91156        -0.19727
                         0.05489         0.04877         0.02981
                         0.18464         0.78650        -0.24577
                         0.36478         0.96481        -0.14778
                            0[]*            0*[]            *[]0

        test2            0.01063         0.71540         0.20500
                         0.05060         0.05148         0.05496
                        -0.07248         0.61982         0.11310
                         0.09359         0.79007         0.29342
                            [0]*            0*[]            0[]*

        test1           -0.07356         0.63815         0.13983
                         0.04245         0.05380         0.05597
                        -0.14292         0.54114         0.04682
                        -0.00348         0.71839         0.23044
                            *[]0            0*[]            0[]*

        test5            0.00863         0.03234         0.91282
                         0.04394         0.04387         0.04509
                        -0.06356        -0.03986         0.80030
                         0.08073         0.10421         0.96323
                            [0]*            [0]*            0*[]

        test4            0.22357        -0.07576         0.67925
                         0.05956         0.03640         0.05434
                         0.12366        -0.13528         0.57955
                         0.31900        -0.01569         0.75891
                            0[]*            *[]0            0*[]

        test6           -0.04295         0.21911         0.53183
                         0.05114         0.07481         0.06905
                        -0.12656         0.09319         0.40893
                         0.04127         0.33813         0.63578
                            *[0]            0[]*            0[*]
```

The coverage displays in Output 27.4.2 show that Test8, Test7, and Test9 have salient relationships with Factor1. The coverage displays are either ‘0*[ ]’ or ‘[ ]*0’, indicating that the entire 90% confidence intervals for the corresponding loadings are beyond the salience value at 0.45. On the other hand, the coverage display for Test3 on Factor1 is ‘0[ ]*’. This indicates that even though the loading estimate is significantly larger than zero, it is not large enough to be salient. Similarly, Test3, Test2, and Test1 have salient relationships with Factor2, while Test5 and Test4 have salient relationships with Factor3. For Test6, its relationship with Factor3 is a little bit ambiguous; the 90% confidence interval covers approximately values between 0.40 and 0.64. This means that the population value might have been smaller or larger than 0.45. It is marginal evidence for a salient relationship.

Output 27.4.3. Interpretations of Factors Using Factor Structure

```
                     A nine-variable-three-factor example

                          Rotation Method: Quartimin

                       Factor Structure (Correlations)
                  With 90% confidence limits; Cover |*| = 0.45?
                Estimate/StdErr/LowerCL/UpperCL/Coverage Display

                          Factor1         Factor2         Factor3

        test8             0.84771         0.30847         0.30994
                          0.02871         0.06593         0.06263
                          0.79324         0.19641         0.20363
                          0.88872         0.41257         0.40904
                             0*[]            0[]*            0[]*

        test7             0.84894         0.58033         0.41970
                          0.02688         0.05265         0.06060
                          0.79834         0.48721         0.31523
                          0.88764         0.66041         0.51412
                             0*[]            0*[]            0[*]

        test9            -0.86791        -0.42248        -0.48396
                          0.02522         0.06187         0.05504
                         -0.90381        -0.51873        -0.56921
                         -0.81987        -0.31567        -0.38841
                             []*0            [*]0            [*]0

        test3             0.57790         0.93325         0.33738
                          0.05069         0.02953         0.06779
                          0.48853         0.86340         0.22157
                          0.65528         0.96799         0.44380
                             0*[]            0*[]            0[]*

        test2             0.38449         0.81615         0.54535
                          0.06143         0.03106         0.05456
                          0.27914         0.75829         0.44946
                          0.48070         0.86126         0.62883
                             0[*]            0*[]            0[*]

        test1             0.24345         0.67351         0.41162
                          0.06864         0.04284         0.05995
                          0.12771         0.59680         0.30846
                          0.35264         0.73802         0.50522
                             0[]*            0*[]            0[*]

        test5             0.37163         0.46498         0.93132
                          0.06092         0.04979         0.03277
                          0.26739         0.37923         0.85159
                          0.46727         0.54282         0.96894
                             0[*]            0[*]            0*[]

        test4             0.45248         0.33583         0.72927
                          0.05876         0.06289         0.04061
                          0.35072         0.22867         0.65527
                          0.54367         0.43494         0.78941
                             0[*]            0[]*            0*[]

        test6             0.25122         0.45137         0.61837
                          0.07140         0.05858         0.05051
                          0.13061         0.34997         0.52833
                          0.36450         0.54232         0.69465
                             0[]*            0[*]            0*[]
```

For oblique factor solutions, some researchers prefer to examine the factor structure loadings, which represent correlations, for determining salient relationships. In Output 27.4.3, the factor structure loadings and the associated standard error estimates and coverage displays are shown. The interpretations based on the factor structure matrix do not change much except for Test3 and Test9. Test9 now has a salient correlation with Factor3. For Test3, it has salient correlations with both Factor1 and Factor2. Fortunately, there are still tests that only have salient correlations with either Factor1 or Factor2 (but not both). This would make interpretations of factors less problematic.

References

Akaike, H. (1973), "Information Theory and the Extension of the Maximum Likelihood Principle," in *Second International Symposium on Information Theory*, ed. V.N. Petrov and F. Csaki, Budapest: Akailseoniai-Kiudo, 267–281.

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control*, 19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika* 52, 317–332.

Archer, C.O. and Jennrich, R.I. (1973), "Standard errors for orthogonally rotated factor loadings," *Psychometrika*, 38, 581–592.

Bickel, P.J. and Doksum, K.A. (1977), *Mathematical Statistics*, San Francisco: Holden-Day.

Browne, M.W. (1982), "Covariance structures," in *Topics in Applied Multivariate Analysis*, ed. D.M. Hawkins, Cambridge: Cambridge University Press, 72–141.

Browne, M.W., Cudeck, R., Tateneni, K., and Mels, G. (1998), *CEFA: Comprehensive Exploratory Factor Analysis.*

Cattell, R.B. (1966), "The Scree Test for the Number of Factors," *Multivariate Behavioral Research*, 1, 245–276.

Cattell, R.B. (1978), *The Scientific Use of Factor Analysis*, New York: Plenum.

Cattell, R.B. and Vogelman, S. (1977), "A Comprehensive Trial of the Scree and KG Criteria for Determining the Number of Factors," *Multivariate Behavioral Research*, 12, 289–325.

Cerny, B.A. and Kaiser, H.F. (1977), "A Study of a Measure of Sampling Adequacy for Factor-Analytic Correlation Matrices," *Multivariate Behavioral Research*, 12, 43–47.

Crawford, C.B. and Ferguson, G.A. (1970), "A General Rotation Criterion and Its Use in Orthogonal Rotation," *Psychometrika*, 35, 321–332.

Cureton, E.E. (1968), *A Factor Analysis of Project TALENT Tests and Four Other Test Batteries*, (Interim Report 4 to the U.S. Office of Education, Cooperative Research Project No. 3051.) Palo Alto, CA: Project TALENT Office, American Institutes for Research and University of Pittsburgh.

Cureton, E.E. and Mulaik, S.A. (1975), “The Weighted Varimax Rotation and the Promax Rotation,” *Psychometrika,* 40, 183–195.

Dziuban, C.D. and Harris, C.W. (1973), “On the Extraction of Components and the Applicability of the Factor Model,” *American Educational Research Journal,* 10, 93–99.

Fuller (1987), *Measurement Error Models,* New York: John Wiley & Sons, Inc.

Geweke, J.F. and Singleton, K.J. (1980), “Interpreting the Likelihood Ratio Statistic in Factor Models When Sample Size Is Small,” *Journal of the American Statistical Association,* 75, 133–137.

Gorsuch, R.L. (1974), *Factor Analysis,* Philadelphia: W.B. Saunders Co.

Harman, H.H. (1976), *Modern Factor Analysis,* Third Edition, Chicago: University of Chicago Press.

Harris, C.W. (1962), “Some Rao-Guttman Relationships,” *Psychometrika,* 27, 247–263.

Hayashi, K. and Yung, Y.F. (1999), “Standard Errors for the Class of Orthomax-Rotated Factor Loadings: Some Matrix Results,” *Psychometrika,* 64, 451–460.

Horn, J.L. and Engstrom, R. (1979), “Cattell’s Scree Test in Relation to Bartlett’s Chi-Square Test and Other Observations on the Number of Factors Problem,” *Multivariate Behavioral Research,* 14, 283–300.

Jennrich, R.I. (1973), “Standard Errors for Obliquely Rotated Factor Loadings,” *Psychometrika,* 38, 593–604.

Jennrich, R.I. (1974), “Simplified Formulae for Standard Errors in Maximum-Likelihood Factor Analysis,” *British Journal of Mathematical and Statistical Psychology,* 27, 122–131.

Jöreskog, K.G. (1962), “On the Statistical Treatment of Residuals in Factor Analysis,” *Psychometrika,* 27, 335–354.

Jöreskog, K.G. (1977), “Factor Analysis by Least-Squares and Maximum Likelihood Methods,” in *Statistical Methods for Digital Computers*, ed. K. Enslein, A. Ralston, and H.S. Wilf, New York: John Wiley & Sons, Inc.

Kaiser, H.F. (1963), “Image Analysis,” in *Problems in Measuring Change*, ed. C.W. Harris, Madison, WI: University of Wisconsin Press.

Kaiser, H.F. (1970), “A Second Generation Little Jiffy,” *Psychometrika,* 35, 401–415.

Kaiser, H.F. and Cerny, B.A. (1979), “Factor Analysis of the Image Correlation Matrix,” *Educational and Psychological Measurement,* 39, 711–714.

Kaiser, H.F. and Rice, J. (1974), “Little Jiffy, Mark IV,” *Educational and Psychological Measurement,* 34, 111–117.

Kerlinger, F.N. and Pedhazur, E.J. (1973), *Multiple Regression in Behavioral Research,* New York: Holt, Rinehart & Winston, Inc.

Kim, J.O. and Mueller, C.W. (1978a), *Introduction to Factor Analysis: What It Is and How To Do It*, Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 07-013, Beverly Hills, CA: Sage Publications.

Kim, J.O. and Mueller, C.W. (1978b), *Factor Analysis: Statistical Methods and Practical Issues*, Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 07-014, Beverly Hills, CA: Sage Publications.

Lawley, D.N. and Maxwell, A.E. (1971), *Factor Analysis as a Statistical Method,* New York: Macmillan Publishing Co., Inc.

Lee, H.B. and Comrey, A.L. (1979), "Distortions in a Commonly Used Factor Analytic Procedure," *Multivariate Behavioral Research,* 14, 301–321.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis,* London: Academic Press.

McDonald, R.P. (1975), "A Note on Rippe's Test of Significance in Common Factor Analysis," *Psychometrika,* 40, 117–119.

McDonald, R.P. (1985), *Factor Analysis and Related Methods*, New Jersey: Lawrence Erlbaum Associates, Publishers.

Morrison, D.F. (1976), *Multivariate Statistical Methods,* Second Edition, New York: McGraw-Hill Book Co.

Mulaik, S.A. (1972), *The Foundations of Factor Analysis,* New York: McGraw-Hill Book Co.

Rao, C.R. (1955), "Estimation and Tests of Significance in Factor Analysis," *Psychometrika,* 20, 93–111.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics,* 6, 461–464.

Spearman, C. (1904), "General Intelligence Objectively Determined and Measured," *American Journal of Psychology,* 15, 201–293.

Stewart, D.W. (1981), "The Application and Misapplication of Factor Analysis in Marketing Research," *Journal of Marketing Research,* 18, 51–62.

Tucker, L.R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika,* 38, 1–10.

Yung, Y. F. and Hayashi, K. (2001), "A Computationally Efficient Method for Obtaining Standard Error Estimates for the Promax and Related Solutions," *British Journal of Mathematical and Statistical Psychology,* 54, 125–138.

Chapter 28
The FASTCLUS Procedure

Chapter Contents

Chapter 28
The FASTCLUS Procedure

Overview

The FASTCLUS procedure performs a disjoint cluster analysis on the basis of distances computed from one or more quantitative variables. The observations are divided into clusters such that every observation belongs to one and only one cluster; the clusters do not form a tree structure as they do in the CLUSTER procedure. If you want separate analyses for different numbers of clusters, you can run PROC FASTCLUS once for each analysis. Alternatively, to do hierarchical clustering on a large data set, use PROC FASTCLUS to find initial clusters, then use those initial clusters as input to PROC CLUSTER.

By default, the FASTCLUS procedure uses Euclidean distances, so the cluster centers are based on least-squares estimation. This kind of clustering method is often called a *k-means model*, since the cluster centers are the means of the observations assigned to each cluster when the algorithm is run to complete convergence. Each iteration reduces the least-squares criterion until convergence is achieved.

Often there is no need to run the FASTCLUS procedure to convergence. PROC FASTCLUS is designed to find good clusters (but not necessarily the best possible clusters) with only two or three passes over the data set. The initialization method of PROC FASTCLUS guarantees that, if there exist clusters such that all distances between observations in the same cluster are less than all distances between observations in different clusters, and if you tell PROC FASTCLUS the correct number of clusters to find, it can always find such a clustering without iterating. Even with clusters that are not as well separated, PROC FASTCLUS usually finds initial seeds that are sufficiently good so that few iterations are required. Hence, by default, PROC FASTCLUS performs only one iteration.

The initialization method used by the FASTCLUS procedure makes it sensitive to outliers. PROC FASTCLUS can be an effective procedure for detecting outliers because outliers often appear as clusters with only one member.

The FASTCLUS procedure can use an L_p (least pth powers) clustering criterion (Spath 1985, pp. 62–63) instead of the least-squares (L_2) criterion used in k-means clustering methods. The LEAST=p option specifies the power p to be used. Using the LEAST= option increases execution time since more iterations are usually required, and the default iteration limit is increased when you specify LEAST=p. Values of p less than 2 reduce the effect of outliers on the cluster centers compared with least-squares methods; values of p greater than 2 increase the effect of outliers.

The FASTCLUS procedure is intended for use with large data sets, with 100 or more observations. With small data sets, the results may be highly sensitive to the order of the observations in the data set.

PROC FASTCLUS produces brief summaries of the clusters it finds. For more extensive examination of the clusters, you can request an output data set containing a cluster membership variable.

Background

The FASTCLUS procedure combines an effective method for finding initial clusters with a standard iterative algorithm for minimizing the sum of squared distances from the cluster means. The result is an efficient procedure for disjoint clustering of large data sets. PROC FASTCLUS was directly inspired by Hartigan's (1975) *leader algorithm* and MacQueen's (1967) *k-means algorithm*. PROC FASTCLUS uses a method that Anderberg (1973) calls *nearest centroid sorting*. A set of points called *cluster seeds* is selected as a first guess of the means of the clusters. Each observation is assigned to the nearest seed to form temporary clusters. The seeds are then replaced by the means of the temporary clusters, and the process is repeated until no further changes occur in the clusters. Similar techniques are described in most references on clustering (Anderberg 1973; Hartigan 1975; Everitt 1980; Spath 1980).

The FASTCLUS procedure differs from other nearest centroid sorting methods in the way the initial cluster seeds are selected. The importance of initial seed selection is demonstrated by Milligan (1980).

The clustering is done on the basis of Euclidean distances computed from one or more numeric variables. If there are missing values, PROC FASTCLUS computes an adjusted distance using the nonmissing values. Observations that are very close to each other are usually assigned to the same cluster, while observations that are far apart are in different clusters.

The FASTCLUS procedure operates in four steps:

1. Observations called *cluster seeds* are selected.
2. If you specify the DRIFT option, temporary clusters are formed by assigning each observation to the cluster with the nearest seed. Each time an observation is assigned, the cluster seed is updated as the current mean of the cluster. This method is sometimes called *incremental*, *on-line*, or *adaptive* training.
3. If the maximum number of iterations is greater than zero, clusters are formed by assigning each observation to the nearest seed. After all observations are assigned, the cluster seeds are replaced by either the cluster means or other location estimates (cluster centers) appropriate to the LEAST=p option. This step can be repeated until the changes in the cluster seeds become small or zero (MAXITER=$n \geq 1$).
4. Final clusters are formed by assigning each observation to the nearest seed.

If PROC FASTCLUS runs to complete convergence, the final cluster seeds will equal the cluster means or cluster centers. If PROC FASTCLUS terminates before complete convergence, which often happens with the default settings, the final cluster seeds may not equal the cluster means or cluster centers. If you want complete converegnce, specify CONVERGE=0 and a large value for the MAXITER= option.

The initial cluster seeds must be observations with no missing values. You can specify the maximum number of seeds (and, hence, clusters) using the MAXCLUSTERS= option. You can also specify a minimum distance by which the seeds must be separated using the RADIUS= option.

PROC FASTCLUS always selects the first complete (no missing values) observation as the first seed. The next complete observation that is separated from the first seed by at least the distance specified in the RADIUS= option becomes the second seed. Later observations are selected as new seeds if they are separated from all previous seeds by at least the radius, as long as the maximum number of seeds is not exceeded.

If an observation is complete but fails to qualify as a new seed, PROC FASTCLUS considers using it to replace one of the old seeds. Two tests are made to see if the observation can qualify as a new seed.

First, an old seed is replaced if the distance between the observation and the closest seed is greater than the minimum distance between seeds. The seed that is replaced is selected from the two seeds that are closest to each other. The seed that is replaced is the one of these two with the shortest distance to the closest of the remaining seeds when the other seed is replaced by the current observation.

If the observation fails the first test for seed replacement, a second test is made. The observation replaces the nearest seed if the smallest distance from the observation to all seeds other than the nearest one is greater than the shortest distance from the nearest seed to all other seeds. If the observation fails this test, PROC FASTCLUS goes on to the next observation.

You can specify the REPLACE= option to limit seed replacement. You can omit the second test for seed replacement (REPLACE=PART), causing PROC FASTCLUS to run faster, but the seeds selected may not be as widely separated as those obtained by the default method. You can also suppress seed replacement entirely by specifying REPLACE=NONE. In this case, PROC FASTCLUS runs much faster, but you must choose a good value for the RADIUS= option in order to get good clusters. This method is similar to Hartigan's (1975, pp. 74–78) leader algorithm and the *simple cluster seeking algorithm* described by Tou and Gonzalez (1974, pp. 90–92).

Getting Started

The following example demonstrates how to use the FASTCLUS procedure to compute disjoint clusters of observations in a SAS data set.

The data in this example are measurements taken on 159 freshwater fish caught from the same lake (Laengelmavesi) near Tampere in Finland. This data set is available from the Data Archive of the *Journal of Statistics Education*. The complete data set is displayed in Chapter 67, "The STEPDISC Procedure."

The species (bream, parkki, pike, perch, roach, smelt, and whitefish), weight, three different length measurements (measured from the nose of the fish to the beginning of its tail, the notch of its tail, and the end of its tail), height, and width of each fish are tallied. The height and width are recorded as percentages of the third length variable.

Suppose that you want to group empirically the fish measurements into clusters and that you want to associate the clusters with the species. You can use the FASTCLUS procedure to perform a cluster analysis.

The following DATA step creates the SAS data set Fish.

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
data Fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;
   if Weight <=0 or Weight = . then delete;
   Weight3=Weight**(1/3);
   Height=HtPct*Length3/(Weight3*100);
   Width=WidthPct*Length3/(Weight3*100);
   Length1=Length1/Weight3;
   Length3=Length3/Weight3;
   logLengthRatio=log(Length3/Length1);

   format Species specfmt.;
   symbol = put(Species, specfmt2.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4
1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1
1  363.0 26.3 29.0 33.5 38.0 13.3
 ... [155 more records]
;
run;
```

The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values are read. The variables are rescaled in order to adjust for dimensionality. Because the new variables Weight3–logLengthRatio depend on the variable Weight, observations with missing values for Weight are not added to the data set. Consequently, there are 157 observations in the SAS data set Fish.

Variables with larger variances exert a larger influence in calculating the clusters. In the Fish data set, the variables are not measured in the same units and cannot be assumed to have equal variance. Therefore, it is necessary to standardize the variables before performing the cluster analysis.

The following statements standardize the variables and perform a cluster analysis on the standardized data.

```
proc standard data=Fish out=Stand mean=0 std=1;
   var Length1 logLengthRatio Height Width Weight3;
proc fastclus data=Stand out=Clust
              maxclusters=7 maxiter=100 ;
   var Length1 logLengthRatio Height Width Weight3;
run;
```

The STANDARD procedure is first used to standardize all the analytical variables to a mean of 0 and standard deviation of 1. The procedure creates the output data set Stand to contain the transformed variables.

The FASTCLUS procedure then uses the data set Stand as input and creates the data set Clust. This output data set contains the original variables and two new variables, Cluster and Distance. The variable Cluster contains the cluster number to which each observation has been assigned. The variable Distance gives the distance from the observation to its cluster seed.

It is usually desirable to try several values of the MAXCLUSTERS= option. A reasonable beginning for this example is to use MAXCLUSTERS=7, since there are seven species of fish represented in the data set Fish.

The VAR statement specifies the variables used in the cluster analysis.

The results from this analysis are displayed in the following figures.

```
                                      Fish Measurement Data

                                      The FASTCLUS Procedure
                 Replace=FULL  Radius=0  Maxclusters=7 Maxiter=100  Converge=0.02

                                          Initial Seeds

                                         logLength
Cluster              Length1                 Ratio            Height             Width           Weight3
-------------------------------------------------------------------------------------------------
   1             1.388338414      -0.979577858      -1.594561848      -2.254050655       2.103447062
   2            -1.117178039      -0.877218192      -0.336166276       2.528114070       1.170706464
   3             2.393997461      -0.662642015      -0.930738701      -2.073879107      -1.839325419
   4            -0.495085516      -0.964041012      -0.265106856      -0.028245072       1.536846394
   5            -0.728772773       0.540096664       1.130501398      -1.207930053      -1.107018207
   6            -0.506924177       0.748211648       1.762482687       0.211507596       1.368987826
   7             1.573996573      -0.796593995      -0.824217424       1.561715851      -1.607942726


                            Criterion Based on Final Seeds =   0.3979
```

Figure 28.1. Initial Seeds Used in the FASTCLUS Procedure

Figure 28.1 displays the table of initial seeds used for each variable and cluster. The first line in the figure displays the option settings for REPLACE, RADIUS, MAXCLUSTERS, and MAXITER. These options, with the exception of MAXCLUSTERS and MAXITER, are set at their respective default values (REPLACE=FULL, RADIUS=0). Both the MAXCLUSTERS= and MAXITER= options are set in the PROC FASTCLUS statement.

Next, PROC FASTCLUS produces a table of summary statistics for the clusters. Figure 28.2 displays the number of observations in the cluster (frequency) and the root mean square standard deviation. The next two columns display the largest Euclidean distance from the cluster seed to any observation within the cluster and the number of the nearest cluster.

The last column of the table displays the distance between the centroid of the nearest cluster and the centroid of the current cluster. A centroid is the point having coordinates that are the means of all the observations in the cluster.

```
                                    Fish Measurement Data

                                    The FASTCLUS Procedure
                  Replace=FULL  Radius=0  Maxclusters=7 Maxiter=100  Converge=0.02

                                       Cluster Summary

                                    Maximum Distance
                           RMS Std         from Seed      Radius     Nearest      Distance Between
     Cluster     Frequency    Deviation     to Observation    Exceeded     Cluster     Cluster Centroids
     -----------------------------------------------------------------------------------------------
        1            17       0.5064           1.7781                     4                 2.5106
        2            19       0.3696           1.5007                     4                 1.5510
        3            13       0.3803           1.7135                     1                 2.6704
        4            13       0.4161           1.3976                     7                 1.4266
        5            11       0.2466           0.6966                     6                 1.7301
        6            34       0.3563           1.5443                     5                 1.7301
        7            50       0.4447           2.3915                     4                 1.4266
```

Figure 28.2. Cluster Summary Table from the FASTCLUS Procedure

Figure 28.3 displays the table of statistics for the variables. The table lists for each variable the total standard deviation, the pooled within-cluster standard deviation and the R^2 value for predicting the variable from the cluster. The ratio of between-cluster variance to within-cluster variance (R^2 to $1 - R^2$) appears in the last column.

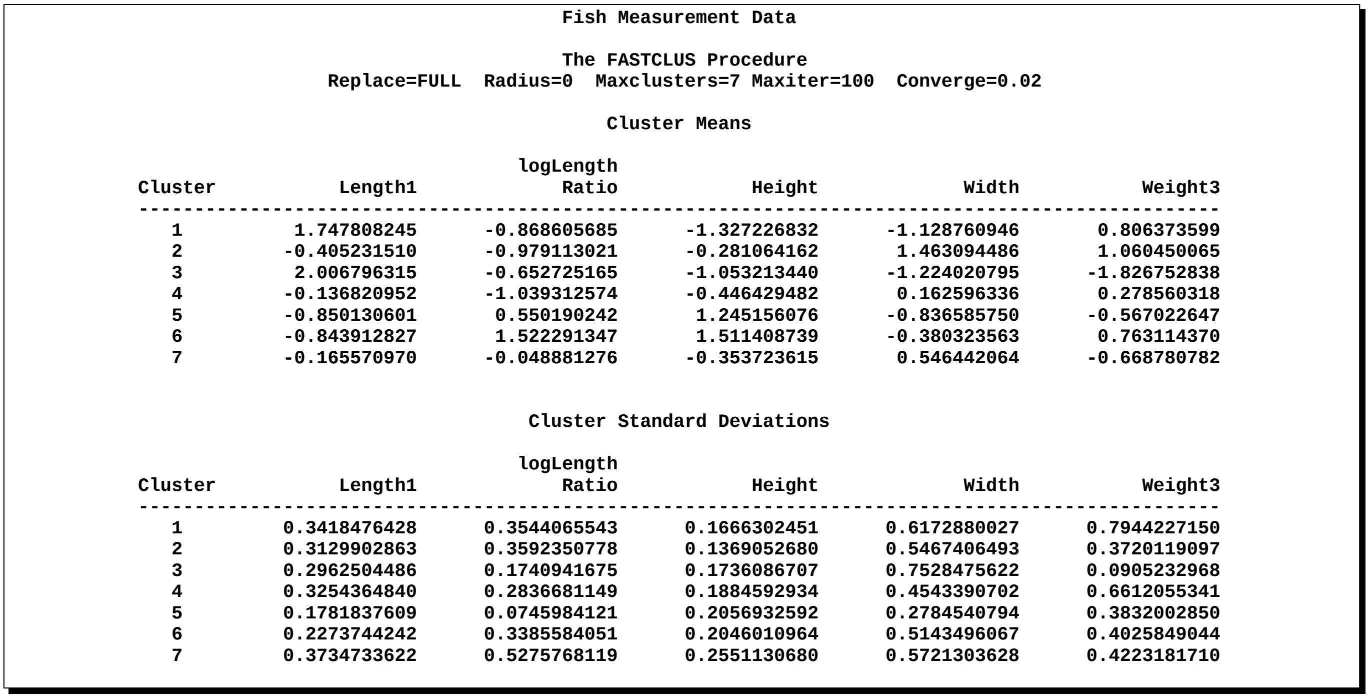
```
                                    Fish Measurement Data

                                    The FASTCLUS Procedure
                  Replace=FULL  Radius=0  Maxclusters=7 Maxiter=100  Converge=0.02

                                   Statistics for Variables

                  Variable           Total STD    Within STD      R-Square     RSQ/(1-RSQ)
                  ------------------------------------------------------------------------
                  Length1              1.00000       0.31428      0.905030        9.529606
                  logLengthRatio       1.00000       0.39276      0.851676        5.741989
                  Height               1.00000       0.20917      0.957929       22.769295
                  Width                1.00000       0.55558      0.703200        2.369270
                  Weight3              1.00000       0.47251      0.785323        3.658162
                  OVER-ALL             1.00000       0.40712      0.840631        5.274764


                                 Pseudo F Statistic =   131.87


                       Approximate Expected Over-All R-Squared =   0.57420


                             Cubic Clustering Criterion =   37.808

                    WARNING: The two values above are invalid for correlated variables.
```

Figure 28.3. Statistics for Variables Used in the FASTCLUS Procedure

The pseudo F statistic, approximate expected overall R^2, and cubic clustering criterion (CCC) are listed at the bottom of the figure. You can compare values of these statistics by running PROC FASTCLUS with different values for the MAXCLUSTERS= option. The R^2 and CCC values are not valid for correlated variables.

Values of the cubic clustering criterion greater than 2 or 3 indicate good clusters. Values between 0 and 2 indicate potential clusters, but they should be taken with caution; large negative values may indicate outliers.

PROC FASTCLUS next produces the within-cluster means and standard deviations of the variables, displayed in Figure 28.4.



```
                                Fish Measurement Data

                                The FASTCLUS Procedure
                   Replace=FULL  Radius=0  Maxclusters=7 Maxiter=100  Converge=0.02

                                    Cluster Means

                               logLength
Cluster          Length1           Ratio          Height           Width         Weight3
------------------------------------------------------------------------------------------
   1         1.747808245    -0.868605685    -1.327226832    -1.128760946     0.806373599
   2        -0.405231510    -0.979113021    -0.281064162     1.463094486     1.060450065
   3         2.006796315    -0.652725165    -1.053213440    -1.224020795    -1.826752838
   4        -0.136820952    -1.039312574    -0.446429482     0.162596336     0.278560318
   5        -0.850130601     0.550190242     1.245156076    -0.836585750    -0.567022647
   6        -0.843912827     1.522291347     1.511408739    -0.380323563     0.763114370
   7        -0.165570970    -0.048881276    -0.353723615     0.546442064    -0.668780782


                             Cluster Standard Deviations

                               logLength
Cluster          Length1           Ratio          Height           Width         Weight3
------------------------------------------------------------------------------------------
   1        0.3418476428    0.3544065543    0.1666302451    0.6172880027    0.7944227150
   2        0.3129902863    0.3592350778    0.1369052680    0.5467406493    0.3720119097
   3        0.2962504486    0.1740941675    0.1736086707    0.7528475622    0.0905232968
   4        0.3254364840    0.2836681149    0.1884592934    0.4543390702    0.6612055341
   5        0.1781837609    0.0745984121    0.2056932592    0.2784540794    0.3832002850
   6        0.2273744242    0.3385584051    0.2046010964    0.5143496067    0.4025849044
   7        0.3734733622    0.5275768119    0.2551130680    0.5721303628    0.4223181710
```

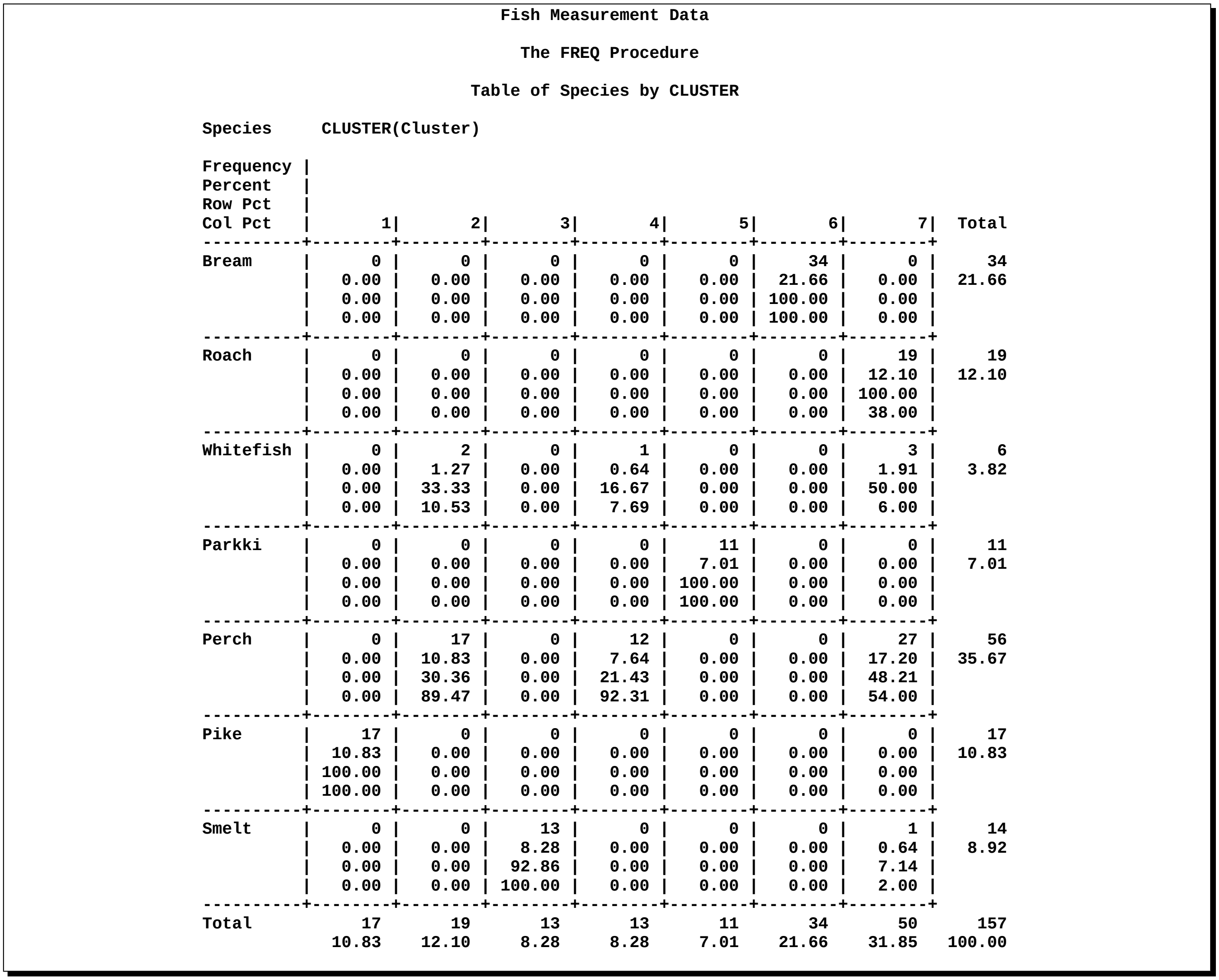
Figure 28.4. Cluster Means and Standard Deviations from the FASTCLUS Procedure

It is useful to study further the clusters calculated by the FASTCLUS procedure. One method is to look at a frequency tabulation of the clusters with other classification variables. The following statements invoke the FREQ procedure to crosstabulate the empirical clusters with the variable Species:

```
proc freq data=Clust;
   tables Species*Cluster;
run;
```

These statements produce a frequency table of the variable Cluster versus the variable Species.

Figure 28.5 displays the marked division between clusters.



```
                                 Fish Measurement Data

                                  The FREQ Procedure

                             Table of Species by CLUSTER

                 Species     CLUSTER(Cluster)

                 Frequency |
                 Percent   |
                 Row Pct   |
                 Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Bream     |      0 |      0 |      0 |      0 |      0 |     34 |      0 |     34
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  21.66 |   0.00 |  21.66
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Roach     |      0 |      0 |      0 |      0 |      0 |      0 |     19 |     19
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  12.10 |  12.10
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |
                           |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  38.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Whitefish |      0 |      2 |      0 |      1 |      0 |      0 |      3 |      6
                           |   0.00 |   1.27 |   0.00 |   0.64 |   0.00 |   0.00 |   1.91 |   3.82
                           |   0.00 |  33.33 |   0.00 |  16.67 |   0.00 |   0.00 |  50.00 |
                           |   0.00 |  10.53 |   0.00 |   7.69 |   0.00 |   0.00 |   6.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Parkki    |      0 |      0 |      0 |      0 |     11 |      0 |      0 |     11
                           |   0.00 |   0.00 |   0.00 |   0.00 |   7.01 |   0.00 |   0.00 |   7.01
                           |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
                           |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Perch     |      0 |     17 |      0 |     12 |      0 |      0 |     27 |     56
                           |   0.00 |  10.83 |   0.00 |   7.64 |   0.00 |   0.00 |  17.20 |  35.67
                           |   0.00 |  30.36 |   0.00 |  21.43 |   0.00 |   0.00 |  48.21 |
                           |   0.00 |  89.47 |   0.00 |  92.31 |   0.00 |   0.00 |  54.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Pike      |     17 |      0 |      0 |      0 |      0 |      0 |      0 |     17
                           |  10.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  10.83
                           | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                           | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Smelt     |      0 |      0 |     13 |      0 |      0 |      0 |      1 |     14
                           |   0.00 |   0.00 |   8.28 |   0.00 |   0.00 |   0.00 |   0.64 |   8.92
                           |   0.00 |   0.00 |  92.86 |   0.00 |   0.00 |   0.00 |   7.14 |
                           |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   2.00 |
                 ----------+--------+--------+--------+--------+--------+--------+--------+
                 Total           17       19       13       13       11       34       50      157
                              10.83    12.10     8.28     8.28     7.01    21.66    31.85   100.00
```

Figure 28.5. Frequency Table of Cluster versus Species

For cases in which you have three or more clusters, you can use the CANDISC and GPLOT procedures to obtain a graphical check on the distribution of the clusters. In the following statements, the CANDISC and GPLOT procedures are used to compute canonical variables and plot the clusters.

```
proc candisc data=Clust out=Can noprint;
   class Cluster;
   var Length1 logLengthRatio Height Width Weight3;

legend1 frame cframe=ligr label=none cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
```

```
proc gplot data=Can;
   plot Can2*Can1=Cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

First, the CANDISC procedure is invoked to perform a canonical discriminant analysis using the data set Clust and creating the output SAS data set Can. The NOPRINT option suppresses display of the output. The CLASS statement specifies the variable Cluster to define groups for the analysis. The VAR statement specifies the variables used in the analysis.

Next, the GPLOT procedure plots the two canonical variables from PROC CANDISC, Can1 and Can2. The PLOT statement specifies the variable Cluster as the identification variable.

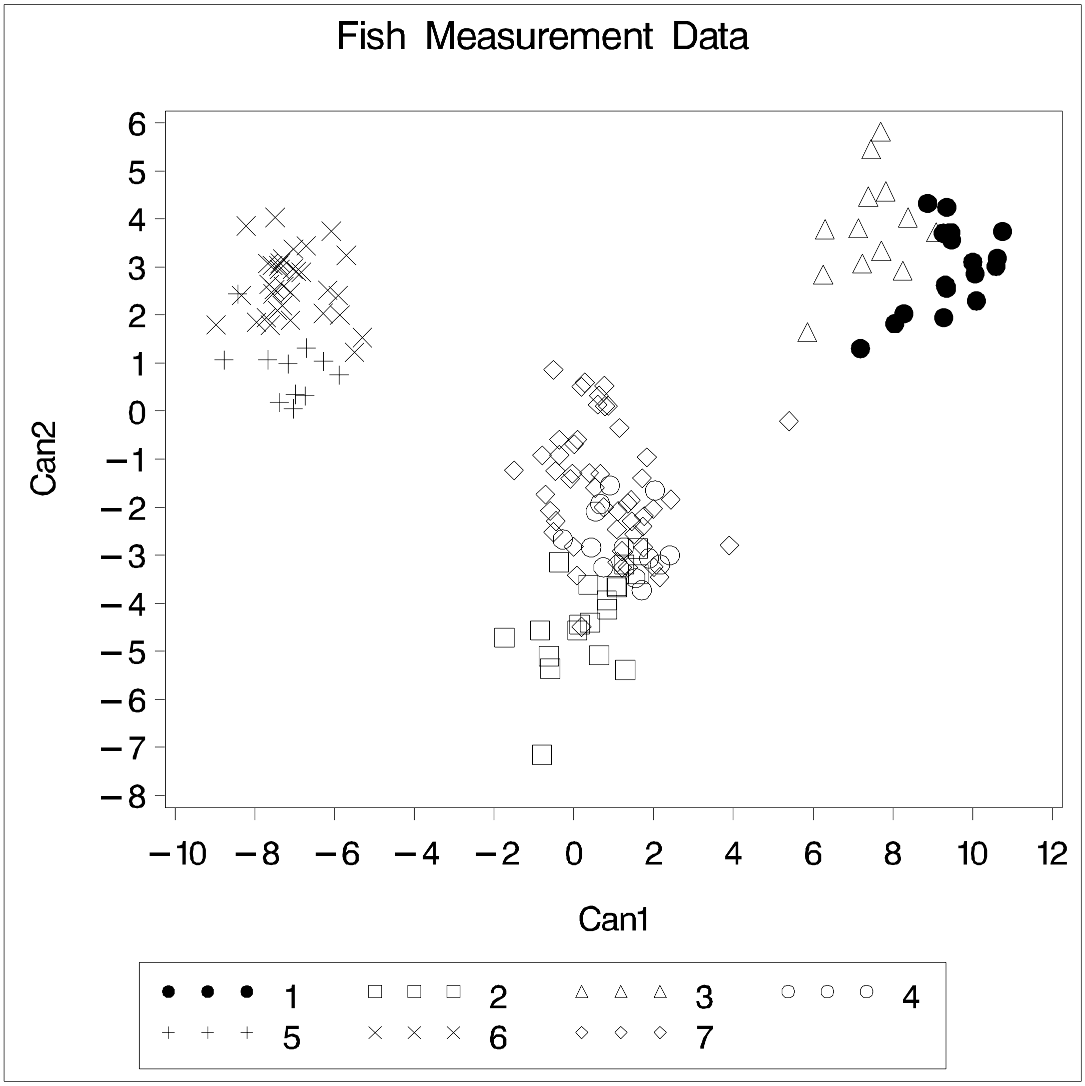


Figure 28.6. Plot of Canonical Variables and Cluster Value

The resulting plot (Figure 28.6) illustrates the spatial separation of the clusters calculated in the FASTCLUS procedure.

Syntax

The following statements are available in the FASTCLUS procedure:

PROC FASTCLUS MAXCLUSTERS=n | **RADIUS=**t < *options* > **;**
  VAR *variables* **;**
  ID *variable* **;**
  FREQ *variable* **;**
  WEIGHT *variable* **;**
  BY *variables* **;**

Usually you need only the VAR statement in addition to the PROC FASTCLUS statement. The BY, FREQ, ID, VAR, and WEIGHT statements are described in alphabetical order after the PROC FASTCLUS statement.

PROC FASTCLUS Statement

PROC FASTCLUS MAXCLUSTERS= n | **RADIUS=**t < *options* > **;**

You must specify either the MAXCLUSTERS= or the RADIUS= argument in the PROC FASTCLUS statement.

MAXCLUSTERS=*n*
MAXC=*n*

specifies the maximum number of clusters allowed. If you omit the MAXCLUSTERS= option, a value of 100 is assumed.

RADIUS=*t*
R=*t*

establishes the minimum distance criterion for selecting new seeds. No observation is considered as a new seed unless its minimum distance to previous seeds exceeds the value given by the RADIUS= option. The default value is 0. If you specify the REPLACE=RANDOM option, the RADIUS= option is ignored.

You can specify the following options in the PROC FASTCLUS statement. Table 28.1 summarizes the options.

Table 28.1. Options Available in the PROC FASTCLUS Statement

Task	**Options**
Specify data set details	CLUSTER= DATA= INSTAT= MEAN= OUT= OUTITER OUTSEED= OUTSTAT= SEED=
Specify distance dimension	BINS= HC= HP= IRLS LEAST=
Select initial cluster seeds	RANDOM= REPLACE=
Compute final cluster seeds	CONVERGE= DELETE= DRIFT MAXCLUSTERS= MAXITER= RADIUS= STRICT
Work with missing values	IMPUTE NOMISS
Specify variance divisor	VARDEF
Control output	DISTANCE LIST NOPRINT SHORT SUMMARY

The following list provides details on these options. The list is in alphabetical order.

BINS=*n*

specifies the number of bins used in the bin-sort algorithm for computing medians for LEAST=1. By default, PROC FASTCLUS uses from 10 to 100 bins, depending on the amount of memory available. Larger values use more memory and make each iteration somewhat slower, but they may reduce the number of iterations. Smaller values have the opposite effect. The minimum value of n is 5.

CLUSTER=*name*

specifies a name for the variable in the OUTSEED= and OUT= data sets that indicates cluster membership. The default name for this variable is CLUSTER.

CONVERGE=*c*
CONV=*c*

specifies the convergence criterion. Any nonnegative value is allowed. The default value is 0.0001 for all values of p if LEAST=p is explicitly specified; otherwise, the default value is 0.02. Iterations stop when the maximum relative change in the cluster seeds is less than or equal to the convergence criterion and additional conditions on the homotopy parameter, if any, are satisfied (see the HP= option). The relative change in a cluster seed is the distance between the old seed and the new seed divided by a scaling factor. If you do not specify the LEAST= option, the scaling factor is the minimum distance between the initial seeds. If you specify the LEAST= option, the scaling factor is an L_1 scale estimate and is recomputed on each iteration. Specify the CONVERGE= option only if you specify a MAXITER= value greater than 1.

DATA=*SAS-data-set*

specifies the input data set containing observations to be clustered. If you omit the DATA= option, the most recently created SAS data set is used. The data must be coordinates, not distances, similarities, or correlations.

DELETE=*n*

deletes cluster seeds to which n or fewer observations are assigned. Deletion occurs after processing for the DRIFT option is completed and after each iteration specified by the MAXITER= option. Cluster seeds are not deleted after the final assignment of observations to clusters, so in rare cases a final cluster may not have more than n members. The DELETE= option is ineffective if you specify MAXITER=0 and do not specify the DRIFT option. By default, no cluster seeds are deleted.

DISTANCE | DIST

computes distances between the cluster means.

DRIFT

executes the second of the four steps described in the section "Background" on page 1380. After initial seed selection, each observation is assigned to the cluster with the nearest seed. After an observation is processed, the seed of the cluster to which it is assigned is recalculated as the mean of the observations currently assigned to the cluster. Thus, the cluster seeds drift about rather than remaining fixed for the duration of the pass.

HC=*c*
HP=*p_1* **<***p_2***>**

pertains to the homotopy parameter for LEAST=p, where $1 < p < 2$. You should specify these options only if you encounter convergence problems using the default values.

For $1 < p < 2$, PROC FASTCLUS tries to optimize a perturbed variant of the L_p clustering criterion (Gonin and Money 1989, pp. 5–6). When the homotopy parameter is 0, the optimization criterion is equivalent to the clustering criterion. For a large homotopy parameter, the optimization criterion approaches the least-squares criterion and is, therefore, easy to optimize. Beginning with a large homotopy parameter, PROC FASTCLUS gradually decreases it by a factor in the range [0.01,0.5] over the course of the iterations. When both the homotopy parameter and the convergence measure are sufficiently small, the optimization process is declared to have converged.

If the initial homotopy parameter is too large or if it is decreased too slowly, the optimization may require many iterations. If the initial homotopy parameter is too small or if it is decreased too quickly, convergence to a local optimum is likely.

HC=c specifies the criterion for updating the homotopy parameter. The homotopy parameter is updated when the maximum relative change in the cluster seeds is less than or equal to c. The default is the minimum of 0.01 and 100 times the value of the CONVERGE= option.

HP=p_1 specifies p_1 as the initial value of the homotopy parameter. The default is 0.05 if the modified Ekblom-Newton method is used; otherwise, it is 0.25.

HP=p_1 p_2 also specifies p_2 as the minimum value for the homotopy parameter, which must be reached for convergence. The default is the minimum of p_1 and 0.01 times the value of the CONVERGE= option.

IMPUTE

requests imputation of missing values after the final assignment of observations to clusters. If an observation has a missing value for a variable used in the cluster analysis, the missing value is replaced by the corresponding value in the cluster seed to which the observation is assigned. If the observation is not assigned to a cluster, missing values are not replaced. If you specify the IMPUTE option, the imputed values are not used in computing cluster statistics.

If you also request an OUT= data set, it contains the imputed values.

INSTAT=*SAS-data-set*

reads a SAS data set previously created by the FASTCLUS procedure using the OUTSTAT= option. If you specify the INSTAT= option, no clustering iterations are performed and no output is displayed. Only cluster assignment and imputation are performed as an OUT= data set is created.

IRLS

causes PROC FASTCLUS to use an iteratively reweighted least-squares method instead of the modified Ekblom-Newton method. If you specify the IRLS option, you must also specify LEAST=p, where $1 < p < 2$. Use the IRLS option only if you encounter convergence problems with the default method.

LEAST=*p* | MAX
L=*p* | MAX

causes PROC FASTCLUS to optimize an L_p criterion, where $1 \leq p \leq \infty$ (Spath 1985, pp. 62–63). Infinity is indicated by LEAST=MAX. The value of this clustering criterion is displayed in the iteration history.

If you do not specify the LEAST= option, PROC FASTCLUS uses the least-squares (L_2) criterion. However, the default number of iterations is only 1 if you omit the LEAST= option, so the optimization of the criterion is generally not completed. If you specify the LEAST= option, the maximum number of iterations is increased to allow the optimization process a chance to converge. See the MAXITER= option on page 1393.

Specifying the LEAST= option also changes the default convergence criterion from 0.02 to 0.0001. See the CONVERGE= option on page 1390.

When LEAST=2, PROC FASTCLUS tries to minimize the root mean square difference between the data and the corresponding cluster means.

When LEAST=1, PROC FASTCLUS tries to minimize the mean absolute difference between the data and the corresponding cluster medians.

When LEAST=MAX, PROC FASTCLUS tries to minimize the maximum absolute difference between the data and the corresponding cluster midranges.

For general values of p, PROC FASTCLUS tries to minimize the pth root of the mean of the pth powers of the absolute differences between the data and the corresponding cluster seeds.

The divisor in the clustering criterion is either the number of nonmissing data used in the analysis or, if there is a WEIGHT statement, the sum of the weights corresponding to all the nonmissing data used in the analysis (that is, an observation with n nonmissing data contributes n times the observation weight to the divisor). The divisor is not adjusted for degrees of freedom.

The method for updating cluster seeds during iteration depends on the LEAST= option, as follows (Gonin and Money 1989).

LEAST=*p*	**Algorithm for Computing Cluster Seeds**
$p = 1$	bin sort for median
$1 < p < 2$	modified Merle-Spath if you specify IRLS, otherwise modified Ekblom-Newton
$p = 2$	arithmetic mean
$2 < p < \infty$	Newton
$p = \infty$	midrange

During the final pass, a modified Merle-Spath step is taken to compute the cluster centers for $1 \leq p < 2$ or $2 < p < \infty$.

If you specify the LEAST=p option with a value other than 2, PROC FASTCLUS computes pooled scale estimates analogous to the root mean square standard deviation but based on pth power deviations instead of squared deviations.

LEAST=*p*	**Scale Estimate**
$p = 1$	mean absolute deviation
$1 < p < \infty$	root mean pth-power absolute deviation
$p = \infty$	maximum absolute deviation

The divisors for computing the mean absolute deviation or the root mean pth-power absolute deviation are adjusted for degrees of freedom just like the divisors for computing standard deviations. This adjustment can be suppressed by the VARDEF= option.

LIST

lists all observations, giving the value of the ID variable (if any), the number of the cluster to which the observation is assigned, and the distance between the observation and the final cluster seed.

MAXITER=*n*

specifies the maximum number of iterations for recomputing cluster seeds. When the value of the MAXITER= option is greater than 0, PROC FASTCLUS executes the third of the four steps described in the "Background" section on page 1380. In each iteration, each observation is assigned to the nearest seed, and the seeds are recomputed as the means of the clusters.

The default value of the MAXITER= option depends on the LEAST=p option.

LEAST=*p*	**MAXITER=**
not specified	1
$p = 1$	20
$1 < p < 1.5$	50
$1.5 \leq p < 2$	20
$p = 2$	10
$2 < p \leq \infty$	20

MEAN=*SAS-data-set*

creates an output data set to contain the cluster means and other statistics for each cluster. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to "SAS Data Files" in *SAS Language Reference: Concepts* for more information on permanent data sets.

NOMISS

excludes observations with missing values from the analysis. However, if you also specify the IMPUTE option, observations with missing values are included in the final cluster assignments.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, “Using the Output Delivery System.”

OUT=*SAS-data-set*

creates an output data set to contain all the original data, plus the new variables CLUSTER and DISTANCE. Refer to “SAS Data Files” in *SAS Language Reference: Concepts* for more information on permanent data sets.

OUTITER

outputs information from the iteration history to the OUTSEED= data set, including the cluster seeds at each iteration.

OUTSEED=*SAS-data-set*
OUTS=*SAS-data-set*

is another name for the MEAN= data set, provided because the data set may contain location estimates other than means. The MEAN= option is still accepted.

OUTSTAT=*SAS-data-set*

creates an output data set to contain various statistics, especially those not included in the OUTSEED= data set. Unlike the OUTSEED= data set, the OUTSTAT= data set is not suitable for use as a SEED= data set in a subsequent PROC FASTCLUS step.

RANDOM=*n*

specifies a positive integer as a starting value for the pseudo-random number generator for use with REPLACE=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence.

REPLACE=FULL | PART | NONE | RANDOM

specifies how seed replacement is performed.

FULL	requests default seed replacement as described in the section “Background” on page 1380.
PART	requests seed replacement only when the distance between the observation and the closest seed is greater than the minimum distance between seeds.
NONE	suppresses seed replacement.
RANDOM	selects a simple pseudo-random sample of complete observations as initial cluster seeds.

SEED=*SAS-data-set*

specifies an input data set from which initial cluster seeds are to be selected. If you do not specify the SEED= option, initial seeds are selected from the DATA= data set. The SEED= data set must contain the same variables that are used in the data analysis.

SHORT

suppresses the display of the initial cluster seeds, cluster means, and standard deviations.

STRICT
STRICT=*s*

prevents an observation from being assigned to a cluster if its distance to the nearest cluster seed exceeds the value of the STRICT= option. If you specify the STRICT option without a numeric value, you must also specify the RADIUS= option, and its value is used instead. In the OUT= data set, observations that are not assigned due to the STRICT= option are given a negative cluster number, the absolute value of which indicates the cluster with the nearest seed.

SUMMARY

suppresses the display of the initial cluster seeds, statistics for variables, cluster means, and standard deviations.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of variances and covariances. The default value is VARDEF=DF. The possible values of the VARDEF= option and associated divisors are as follows.

Value	**Description**	**Divisor**
DF	error degrees of freedom	$n - c$
N	number of observations	n
WDF	sum of weights DF	$(\sum_i w_i) - c$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

In the preceding definitions, c represents the number of clusters.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FASTCLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FASTCLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables using the DATASETS procedure.

If you specify the SEED= option and the SEED= data set does not contain any of the BY variables, then the entire SEED= data set is used to obtain initial cluster seeds for each BY group in the DATA= data set.

If the SEED= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the SEED= data set as in the DATA= data set, then PROC FASTCLUS displays an error message and stops.

If all the BY variables appear in the SEED= data set with the same type and length as in the DATA= data set, then each BY group in the SEED= data set is used to obtain initial cluster seeds for the corresponding BY group in the DATA= data set. All BY groups in the DATA= data set must also appear in the SEED= data set. The BY groups in the SEED= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be exactly the same BY groups in the same order in both data sets. If you do not specify NOTSORTED, some BY groups can appear in the SEED= data set but not in the DATA= data set; such BY groups are not used in the analysis.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation.

If the value of the FREQ variable is missing or ≤ 0, the observation is not used in the analysis. The exact values of the FREQ variable are used in computations: frequency values are not truncated to integers. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

The WEIGHT and FREQ statements have a similar effect, except in determining the number of observations for significance tests.

ID Statement

ID *variable* **;**

The ID variable, which can be character or numeric, identifies observations on the output when you specify the LIST option.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not listed in other statements are used.

WEIGHT Statement

WEIGHT *variable* **;**

The values of the WEIGHT variable are used to compute weighted cluster means. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the degrees of freedom or the number of observations. The WEIGHT variable can take nonintegral values. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

Details

Updates in the FASTCLUS Procedure

Some FASTCLUS procedure options and statements have changed from previous versions. The differences are as follows:

- Values of the FREQ variable are no longer truncated to integers. Noninteger variables specified in the FREQ statement produce results different than in previous releases.
- The IMPUTE option produces different cluster standard deviations and related statistics. When you specify the IMPUTE option, imputed values are no longer used in computing cluster statistics. This change causes the cluster standard deviations and other statistics computed from the standard deviations to be different than in previous releases.
- The INSTAT= option reads a SAS data set previously created by the FASTCLUS procedure using the OUTSTAT= option. If you specify the INSTAT= option, no clustering iterations are performed and no output is produced. Only cluster assignment and imputation are performed as an OUT= data set is created.
- The OUTSTAT= data set contains additional information used for imputation. _TYPE_=SEED corresponds to values that are cluster seeds. Observations previously designated _TYPE_='SCALE' are now _TYPE_='DISPERSION'.

Missing Values

Observations with all missing values are excluded from the analysis. If you specify the NOMISS option, observations with any missing values are excluded. Observations with missing values cannot be cluster seeds.

The distance between an observation with missing values and a cluster seed is obtained by computing the squared distance based on the nonmissing values, multiplying by the ratio of the number of variables, n, to the number of variables having nonmissing values, m, and taking the square root:

$$\sqrt{\left(\frac{n}{m}\right)\sum(x_i - s_i)^2}$$

where

$$\begin{aligned} n &= \text{number of variables} \\ m &= \text{number of variables with nonmissing values} \\ x_i &= \text{value of the } i\text{th variable for the observation} \\ s_i &= \text{value of the } i\text{th variable for the seed} \end{aligned}$$

If you specify the LEAST=p option with a power p other than 2 (the default), the distance is computed using:

$$\left(\left(\frac{n}{m}\right)\sum(x_i - s_i)^p\right)^{\frac{1}{p}}$$

The summation is taken over variables with nonmissing values.

The IMPUTE option fills in missing values in the OUT= output data set.

Output Data Sets

OUT= Data Set

The OUT= data set contains

- the original variables
- a new variable taking values from 1 to the value specified in the MAXCLUSTERS= option, indicating the cluster to which each observation has been assigned. You can specify the variable name with the CLUSTER= option; the default name is CLUSTER.
- a new variable, DISTANCE, giving the distance from the observation to its cluster seed

If you specify the IMPUTE option, the OUT= data set also contains a new variable, _IMPUTE_, giving the number of imputed values in each observation.

OUTSEED= Data Set

The OUTSEED= data set contains one observation for each cluster. The variables are as follows:

- the BY variables, if any
- a new variable giving the cluster number. You can specify the variable name with the CLUSTER= option. The default name is CLUSTER.
- either the FREQ variable or a new variable called _FREQ_ giving the number of observations in the cluster
- the WEIGHT variable, if any
- a new variable, _RMSSTD_, giving the root mean square standard deviation for the cluster. See Chapter 23, "The CLUSTER Procedure," for details.
- a new variable, _RADIUS_, giving the maximum distance between any observation in the cluster and the cluster seed
- a new variable, _GAP_, containing the distance between the current cluster mean and the nearest other cluster mean. The value is the centroid distance given in the output.
- a new variable, _NEAR_, specifying the cluster number of the nearest cluster
- the VAR variables giving the cluster means

If you specify the LEAST=p option with a value other than 2, the _RMSSTD_ variable is replaced by the _SCALE_ variable, which contains the pooled scale estimate analogous to the root mean square standard deviation but based on pth power deviations instead of squared deviations:

LEAST=1	mean absolute deviation
LEAST=p	root mean pth-power absolute deviation
LEAST=MAX	maximum absolute deviation

If you specify the OUTITER option, there is one set of observations in the OUTSEED= data set for each pass through the data set (that is, one set for initial seeds, one for each iteration, and one for the final clusters). Also, several additional variables appear:

ITER is the iteration number. For the initial seeds, the value is 0. For the final cluster means or centers, the _ITER_ variable is one greater than the last iteration reported in the iteration history.

CRIT is the clustering criterion as described under the LEAST= option.

CHANGE is the maximum over clusters of the relative change in the cluster seed from the previous iteration. The relative change in a cluster seed is the distance between the old seed and the new seed divided by a scaling factor. If you do not specify the LEAST= option, the scaling factor is the minimum distance between the initial seeds. If you specify the LEAST= option, the scaling factor is an L_1 scale estimate and is recomputed on each iteration.

HOMPAR is the value of the homotopy parameter. This variable appears only for LEAST=p with $1 < p < 2$.

BINSIZ is the maximum bin size used for estimating medians. This variable appears only for LEAST=1.

If you specify the OUTITER option, the variables _SCALE_ or _RMSSTD_, _RADIUS_, _NEAR_, and _GAP_ have missing values except for the last pass.

You can use the OUTSEED= data set as a SEED= input data set for a subsequent analysis.

OUTSTAT= Data Set

The variables in the OUTSTAT= data set are as follows:

- BY variables, if any
- a new character variable, _TYPE_, specifying the type of statistic given by other variables (see Table 28.2 and Table 28.3)
- a new numeric variable giving the cluster number. You can specify the variable name with the CLUSTER= option. The default name is CLUSTER.
- a new numeric variable, OVER_ALL, containing statistics that apply over all of the VAR variables
- the VAR variables giving statistics for particular variables

The values of _TYPE_ for all LEAST= options are given in the following table.

Table 28.2. _TYPE_ Values for all LEAST= Options

TYPE	**Contents of VAR variables**	**Contents of OVER_ALL**
INITIAL	Initial seeds	Missing
CRITERION	Missing	Optimization criterion; see the LEAST= option; this value is displayed just before the “Cluster Summary” table
CENTER	Cluster centers; see the LEAST= option	Missing
SEED	Cluster seeds: additional information used for imputation	

Table 28.2. (continued)

TYPE	**Contents of VAR variables**	**Contents of OVER_ALL**
DISPERSION	Dispersion estimates for each cluster; see the LEAST= option; these values are displayed in a separate row with title depending on the LEAST= option	Dispersion estimates pooled over variables; see the LEAST= option; these values are displayed in the “Cluster Summary” table with label depending on the LEAST= option
FREQ	Frequency of each cluster omitting observations with missing values for the VAR variable; these values are not displayed	Frequency of each cluster based on all observations with any nonmissing value; these values are displayed in the “Cluster Summary” table
WEIGHT	Sum of weights for each cluster omitting observations with missing values for the VAR variable; these values are not displayed	Sum of weights for each cluster based on all observations with any nonmissing value; these values are displayed in the “Cluster Summary” table

Observations with _TYPE_=’WEIGHT’ are included only if you specify the WEIGHT statement.

The _TYPE_ values included only for least-squares clustering are given in the following table. Least-squares clustering is obtained by omitting the LEAST= option or by specifying LEAST=2.

Table 28.3. _TYPE_ Values for Least-Squares Clustering

TYPE	**Contents of VAR variables**	**Contents of OVER_ALL**
MEAN	Mean for the total sample; this is not displayed	Missing
STD	Standard deviation for the total sample; this is labeled “Total STD” in the output	Standard deviation pooled over all the VAR variables; this is labeled “Total STD” in the output
WITHIN_STD	Pooled within-cluster standard deviation	Within cluster standard deviation pooled over clusters and all the VAR variables
RSQ	R^2 for predicting the variable from the clusters; this is labeled “R-Squared” in the output	R^2 pooled over all the VAR variables; this is labeled “R-Squared” in the output

Table 28.3. (continued)

TYPE	**Contents of VAR variables**	**Contents of OVER_ALL**
RSQ_RATIO	$\frac{R^2}{1-R^2}$; this is labeled "RSQ/(1-RSQ)" in the output	$\frac{R^2}{1-R^2}$; labeled "RSQ/(1-RSQ)" in the output
PSEUDO_F	Missing	Pseudo F statistic
ESRQ	Missing	Approximate expected value of R^2 under the null hypothesis of a single uniform cluster
CCC	Missing	The cubic clustering criterion

Computational Resources

Let

$$\begin{aligned} n &= \text{number of observations} \\ v &= \text{number of variables} \\ c &= \text{number of clusters} \\ p &= \text{number of passes over the data set} \end{aligned}$$

Memory

The memory required is approximately $4(19v + 12cv + 10c + 2\max(c+1, v))$ bytes.

If you request the DISTANCE option, an additional $4c(c+1)$ bytes of space is needed.

Time

The overall time required by PROC FASTCLUS is roughly proportional to $nvcp$ if c is small with respect to n.

Initial seed selection requires one pass over the data set. If the observations are in random order, the time required is roughly proportional to

$$nvc + vc^2$$

unless you specify REPLACE=NONE. In that case, a complete pass may not be necessary, and the time is roughly proportional to mvc, where $c \leq m \leq n$.

The DRIFT option, each iteration, and the final assignment of cluster seeds each require one pass, with time for each pass roughly proportional to nvc.

For greatest efficiency, you should list the variables in the VAR statement in order of decreasing variance.

Using PROC FASTCLUS

Before using PROC FASTCLUS, decide whether your variables should be standardized in some way, since variables with large variances tend to have more effect on the resulting clusters than those with small variances. If all variables are measured in the same units, standardization may not be necessary. Otherwise, some form of standardization is strongly recommended. The STANDARD procedure can standardize all variables to mean zero and variance one. The FACTOR or PRINCOMP procedures can compute standardized principal component scores. The ACECLUS procedure can transform the variables according to an estimated within-cluster covariance matrix.

Nonlinear transformations of the variables may change the number of population clusters and should, therefore, be approached with caution. For most applications, the variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required. Ordinal or ranked data are generally not appropriate.

PROC FASTCLUS produces relatively little output. In most cases you should create an output data set and use other procedures such as PRINT, PLOT, CHART, MEANS, DISCRIM, or CANDISC to study the clusters. It is usually desirable to try several values of the MAXCLUSTERS= option. Macros are useful for running PROC FASTCLUS repeatedly with other procedures.

A simple application of PROC FASTCLUS with two variables to examine the 2- and 3-cluster solutions may proceed as follows:

```
proc standard mean=0 std=1 out=stan;
   var v1 v2;
run;

proc fastclus data=stan out=clust maxclusters=2;
   var v1 v2;
run;

proc plot;
   plot v2*v1=cluster;
run;

proc fastclus data=stan out=clust maxclusters=3;
   var v1 v2;
run;

proc plot;
   plot v2*v1=cluster;
run;
```

If you have more than two variables, you can use the CANDISC procedure to compute canonical variables for plotting the clusters, for example,

```
proc standard mean=0 std=1 out=stan;
   var v1-v10;
```

```
run;

proc fastclus data=stan out=clust maxclusters=3;
   var v1-v10;
run;

proc candisc out=can;
   var v1-v10;
   class cluster;
run;

proc plot;
   plot can2*can1=cluster;
run;
```

If the data set is not too large, it may also be helpful to use

```
proc sort;
   by cluster distance;
run;
proc print;
   by cluster;
run;
```

to list the clusters. By examining the values of DISTANCE, you can determine if any observations are unusually far from their cluster seeds.

It is often advisable, especially if the data set is large or contains outliers, to make a preliminary PROC FASTCLUS run with a large number of clusters, perhaps 20 to 100. Use MAXITER=0 and OUTSEED=*SAS-data-set*. You can save time on subsequent runs by selecting cluster seeds from this output data set using the SEED= option.

You should check the preliminary clusters for outliers, which often appear as clusters with only one member. Use a DATA step to delete outliers from the data set created by the OUTSEED= option before using it as a SEED= data set in later runs. If there are severe outliers, the subsequent PROC FASTCLUS runs should specify the STRICT option to prevent the outliers from distorting the clusters.

You can use the OUTSEED= data set with the PLOT procedure to plot _GAP_ by _FREQ_. An overlay of _RADIUS_ by _FREQ_ provides a baseline against which to compare the values of _GAP_. Outliers appear in the upper left area of the plot, with large values of _GAP_ and small _FREQ_ values. Good clusters appear in the upper right area, with large values of both _GAP_ and _FREQ_. Good potential cluster seeds appear in the lower right, as well as in the upper right, since large _FREQ_ values indicate high density regions. Small _FREQ_ values in the left part of the plot indicate poor cluster seeds because the points are in low density regions. It often helps to remove all clusters with small frequencies even though the clusters may not be remote enough to be considered outliers. Removing points in low density regions improves cluster separation and provides visually sharper cluster outlines in scatter plots.

Displayed Output

Unless the SHORT or SUMMARY option is specified, PROC FASTCLUS displays

- Initial Seeds, cluster seeds selected after one pass through the data
- Change in Cluster Seeds for each iteration, if you specify MAXITER=$n > 1$

If you specify the LEAST=*p* option, with $(1 < p < 2)$, and you omit the IRLS option, an additional column is displayed in the Iteration History table. The column contains a character to identify the method used in each iteration. PROC FASTCLUS chooses the most efficient method to cluster the data at each iterative step, given the condition of the data. Thus, the method chosen is data dependent. The possible values are described as follows:

Value	**Method**
N	Newton's Method
I or L	iteratively weighted least squares (IRLS)
1	IRLS step, halved once
2	IRLS step, halved twice
3	IRLS step, halved three times

PROC FASTCLUS displays a Cluster Summary, giving the following for each cluster:

- Cluster number
- Frequency, the number of observations in the cluster
- Weight, the sum of the weights of the observations in the cluster, if you specify the WEIGHT statement
- RMS Std Deviation, the root mean square across variables of the cluster standard deviations, which is equal to the root mean square distance between observations in the cluster
- Maximum Distance from Seed to Observation, the maximum distance from the cluster seed to any observation in the cluster
- Nearest Cluster, the number of the cluster with mean closest to the mean of the current cluster
- Centroid Distance, the distance between the centroids (means) of the current cluster and the nearest other cluster

A table of statistics for each variable is displayed unless you specify the SUMMARY option. The table contains

- Total STD, the total standard deviation

- Within STD, the pooled within-cluster standard deviation
- R-Squared, the R^2 for predicting the variable from the cluster
- RSQ/(1 - RSQ), the ratio of between-cluster variance to within-cluster variance $(R^2/(1-R^2))$
- OVER-ALL, all of the previous quantities pooled across variables

PROC FASTCLUS also displays

- Pseudo F Statistic,

  $$\frac{\frac{R^2}{c-1}}{\frac{1-R^2}{n-c}}$$

  where R^2 is the observed overall R^2, c is the number of clusters, and n is the number of observations. The pseudo F statistic was suggested by Calinski and Harabasz (1974). Refer to Milligan and Cooper (1985) and Cooper and Milligan (1988) regarding the use of the pseudo F statistic in estimating the number of clusters. See Example 23.2 in Chapter 23, "The CLUSTER Procedure," for a comparison of pseudo F statistics.
- Observed Overall R-Squared, if you specify the SUMMARY option
- Approximate Expected Overall R-Squared, the approximate expected value of the overall R^2 under the uniform null hypothesis assuming that the variables are uncorrelated. The value is missing if the number of clusters is greater than one-fifth the number of observations.
- Cubic Clustering Criterion, computed under the assumption that the variables are uncorrelated. The value is missing if the number of clusters is greater than one-fifth the number of observations.

  If you are interested in the approximate expected R^2 or the cubic clustering criterion but your variables are correlated, you should cluster principal component scores from the PRINCOMP procedure. Both of these statistics are described by Sarle (1983). The performance of the cubic clustering criterion in estimating the number of clusters is examined by Milligan and Cooper (1985) and Cooper and Milligan (1988).
- Distances Between Cluster Means, if you specify the DISTANCE option

Unless you specify the SHORT or SUMMARY option, PROC FASTCLUS displays

- Cluster Means for each variable
- Cluster Standard Deviations for each variable

ODS Table Names

PROC FASTCLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 28.4. ODS Tables Produced in PROC FASTCLUS

ODS Table Name	Description	Statement	Option
ApproxExpOverAllRSq	Approximate expected over-all R-squared, single number	PROC	default
CCC	CCC, Cubic Clustering Criterion, single number	PROC	default
ClusterList	Cluster listing, obs, id, and distances	PROC	LIST
ClusterSum	Cluster summary, cluster number, distances	PROC	PRINTALL
ClusterCenters	Cluster centers	PROC	default
ClusterDispersion	Cluster dispersion	PROC	default
ConvergenceStatus	Convergence status	PROC	PRINTALL
Criterion	Criterion based on final seeds, single number	PROC	default
DistBetweenClust	Distance between clusters	PROC	default
InitialSeeds	Initial seeds	PROC	default
IterHistory	Iteration history, various statistics for each iter	PROC	PRINTALL
MinDist	Minimum distance between initial seeds, single number	PROC	PRINTALL
NumberOfBins	Number of bins	PROC	default
ObsOverAllRSquare	Observed over-all R-squared, single number	PROC	SUMMARY
PrelScaleEst	Preliminary L(1) scale estimate, single number	PROC	PRINTALL
PseudoFStat	Pseudo F statistic, single number	PROC	default
SimpleStatistics	Simple statistics for input variables	PROC	default
VariableStat	Statistics for variables within clusters	PROC	default

Examples

Example 28.1. Fisher’s Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analyses of the iris data.

In this example, the FASTCLUS procedure is used to find two and, then, three clusters. An output data set is created, and PROC FREQ is invoked to compare the clusters with the species classification. See Output 28.1.1 and Output 28.1.2 for these results. For three clusters, you can use the CANDISC procedure to compute canonical variables for plotting the clusters. See Output 28.1.3 for the results.

```
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
run;

data iris;
   title ’Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
```

```
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;

proc fastclus data=iris maxc=2 maxiter=10 out=clus;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;

proc freq;
   tables cluster*species;
run;

proc fastclus data=iris maxc=3 maxiter=10 out=clus;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;

proc freq;
   tables cluster*Species;
run;

proc candisc anova out=can;
   class cluster;
   var SepalLength SepalWidth PetalLength PetalWidth;
   title2 'Canonical Discriminant Analysis of Iris Clusters';
run;
legend1 frame cframe=ligr label=none cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

proc gplot data=Can;
   plot Can2*Can1=Cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
   title2 'Plot of Canonical Variables Identified by Cluster';
run;
```

Output 28.1.1. Fisher's Iris Data: PROC FASTCLUS with MAXC=2 and PROC FREQ

```
                                  Fisher (1936) Iris Data

                                   The FASTCLUS Procedure
                    Replace=FULL  Radius=0  Maxclusters=2 Maxiter=10  Converge=0.02

                                       Initial Seeds

            Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
            -------------------------------------------------------------------------------
               1          43.00000000       30.00000000       11.00000000        1.00000000
               2          77.00000000       26.00000000       69.00000000       23.00000000


                         Minimum Distance Between Initial Seeds = 70.85196
```

```
                                  Fisher (1936) Iris Data

                                   The FASTCLUS Procedure
                    Replace=FULL  Radius=0  Maxclusters=2 Maxiter=10  Converge=0.02

                                     Iteration History

                                                Relative Change
                                               in Cluster Seeds
                        Iteration    Criterion          1           2
                        ---------------------------------------------
                                1      11.0638      0.1904      0.3163
                                2       5.3780      0.0596      0.0264
                                3       5.0718      0.0174     0.00766


                    Convergence criterion is satisfied.


                               Criterion Based on Final Seeds =   5.0417


                                      Cluster Summary

                                   Maximum Distance
                        RMS Std          from Seed     Radius     Nearest     Distance Between
    Cluster     Frequency    Deviation       to Observation    Exceeded    Cluster    Cluster Centroids
    ----------------------------------------------------------------------------------------------------
       1               53       3.7050              21.1621                         2              39.2879
       2               97       5.6779              24.6430                         1              39.2879


                                  Statistics for Variables

                  Variable        Total STD    Within STD      R-Square     RSQ/(1-RSQ)
                  ---------------------------------------------------------------------
                  SepalLength       8.28066       5.49313      0.562896        1.287784
                  SepalWidth        4.35866       3.70393      0.282710        0.394137
                  PetalLength      17.65298       6.80331      0.852470        5.778291
                  PetalWidth        7.62238       3.57200      0.781868        3.584390
                  OVER-ALL         10.69224       5.07291      0.776410        3.472463


                                 Pseudo F Statistic =   513.92


                       Approximate Expected Over-All R-Squared =   0.51539


                              Cubic Clustering Criterion =   14.806

                   WARNING: The two values above are invalid for correlated variables.
```

```
                                 Fisher (1936) Iris Data

                                  The FASTCLUS Procedure
                   Replace=FULL  Radius=0  Maxclusters=2 Maxiter=10  Converge=0.02

                                      Cluster Means

          Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
          -------------------------------------------------------------------------------
             1          50.05660377       33.69811321       15.60377358        2.90566038
             2          63.01030928       28.86597938       49.58762887       16.95876289


                                Cluster Standard Deviations

          Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
          -------------------------------------------------------------------------------
             1          3.427350930       4.396611045       4.404279486       2.105525249
             2          6.336887455       3.267991438       7.800577673       4.155612484
```

```
                                 Fisher (1936) Iris Data

                                   The FREQ Procedure

                                Table of CLUSTER by Species

                            CLUSTER(Cluster)     Species

                            Frequency|
                            Percent  |
                            Row Pct  |
                            Col Pct  |Setosa  |Versicol|Virginic|  Total
                                     |        |or      |a       |
                            ---------+--------+--------+--------+
                                   1 |     50 |      3 |      0 |     53
                                     |  33.33 |   2.00 |   0.00 |  35.33
                                     |  94.34 |   5.66 |   0.00 |
                                     | 100.00 |   6.00 |   0.00 |
                            ---------+--------+--------+--------+
                                   2 |      0 |     47 |     50 |     97
                                     |   0.00 |  31.33 |  33.33 |  64.67
                                     |   0.00 |  48.45 |  51.55 |
                                     |   0.00 |  94.00 | 100.00 |
                            ---------+--------+--------+--------+
                            Total          50       50       50      150
                                        33.33    33.33    33.33   100.00
```

Output 28.1.2. Fisher's Iris Data: PROC FASTCLUS with MAXC=3 and PROC FREQ

```
                                 Fisher (1936) Iris Data

                                  The FASTCLUS Procedure
                   Replace=FULL  Radius=0  Maxclusters=3 Maxiter=10  Converge=0.02

                                      Initial Seeds

          Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
          -------------------------------------------------------------------------------
             1          58.00000000       40.00000000       12.00000000        2.00000000
             2          77.00000000       38.00000000       67.00000000       22.00000000
             3          49.00000000       25.00000000       45.00000000       17.00000000


                         Minimum Distance Between Initial Seeds = 38.23611
```

```
                                  Fisher (1936) Iris Data

                                  The FASTCLUS Procedure
                     Replace=FULL  Radius=0  Maxclusters=3 Maxiter=10  Converge=0.02

                                     Iteration History

                                                Relative Change in Cluster Seeds
                        Iteration    Criterion            1           2           3
                        ----------------------------------------------------------
                                1       6.7591       0.2652      0.3205      0.2985
                                2       3.7097            0      0.0459      0.0317
                                3       3.6427            0      0.0182      0.0124


                     Convergence criterion is satisfied.


                                Criterion Based on Final Seeds =   3.6289


                                     Cluster Summary

                                        Maximum Distance
                              RMS Std           from Seed     Radius     Nearest     Distance Between
            Cluster     Frequency    Deviation      to Observation    Exceeded    Cluster    Cluster Centroids
            --------------------------------------------------------------------------------------------------
               1             50       2.7803             12.4803                        3              33.5693
               2             38       4.0168             14.9736                        3              17.9718
               3             62       4.0398             16.9272                        2              17.9718


                                 Statistics for Variables

                  Variable        Total STD    Within STD      R-Square     RSQ/(1-RSQ)
                  ---------------------------------------------------------------------
                  SepalLength       8.28066       4.39488      0.722096        2.598359
                  SepalWidth        4.35866       3.24816      0.452102        0.825156
                  PetalLength      17.65298       4.21431      0.943773       16.784895
                  PetalWidth        7.62238       2.45244      0.897872        8.791618
                  OVER-ALL         10.69224       3.66198      0.884275        7.641194


                                   Pseudo F Statistic =   561.63


                         Approximate Expected Over-All R-Squared =   0.62728


                               Cubic Clustering Criterion =   25.021

                   WARNING: The two values above are invalid for correlated variables.
```

```
                           Fisher (1936) Iris Data

                           The FASTCLUS Procedure
       Replace=FULL  Radius=0  Maxclusters=3 Maxiter=10  Converge=0.02

                               Cluster Means

    Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
    -------------------------------------------------------------------------------
       1          50.06000000       34.28000000       14.62000000        2.46000000
       2          68.50000000       30.73684211       57.42105263       20.71052632
       3          59.01612903       27.48387097       43.93548387       14.33870968


                        Cluster Standard Deviations

    Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
    -------------------------------------------------------------------------------
       1          3.524896872       3.790643691       1.736639965       1.053855894
       2          4.941550255       2.900924461       4.885895746       2.798724562
       3          4.664100551       2.962840548       5.088949673       2.974997167
```

```
                           Fisher (1936) Iris Data

                             The FREQ Procedure

                        Table of CLUSTER by Species

                      CLUSTER(Cluster)     Species

                      Frequency|
                      Percent  |
                      Row Pct  |
                      Col Pct  |Setosa  |Versicol|Virginic|  Total
                               |        |or      |a       |
                      ---------+--------+--------+--------+
                             1 |     50 |      0 |      0 |     50
                               |  33.33 |   0.00 |   0.00 |  33.33
                               | 100.00 |   0.00 |   0.00 |
                               | 100.00 |   0.00 |   0.00 |
                      ---------+--------+--------+--------+
                             2 |      0 |      2 |     36 |     38
                               |   0.00 |   1.33 |  24.00 |  25.33
                               |   0.00 |   5.26 |  94.74 |
                               |   0.00 |   4.00 |  72.00 |
                      ---------+--------+--------+--------+
                             3 |      0 |     48 |     14 |     62
                               |   0.00 |  32.00 |   9.33 |  41.33
                               |   0.00 |  77.42 |  22.58 |
                               |   0.00 |  96.00 |  28.00 |
                      ---------+--------+--------+--------+
                      Total          50       50       50      150
                                  33.33    33.33    33.33   100.00
```

Output 28.1.3. Fisher's Iris Data: PROC CANDISC and PROC GPLOT

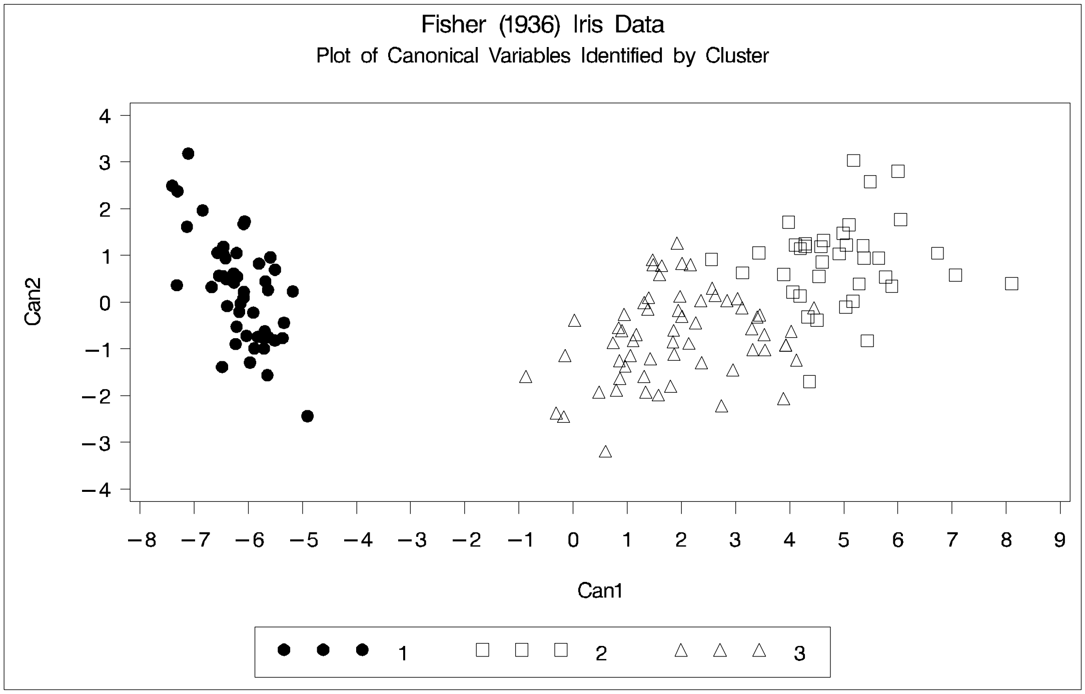
```
                                    Fisher (1936) Iris Data
                            Canonical Discriminant Analysis of Iris Clusters

                                        The CANDISC Procedure

                        Observations     150          DF Total                149
                        Variables          4          DF Within Classes       147
                        Classes            3          DF Between Classes        2


                                      Class Level Information

                                  Variable
                      CLUSTER     Name         Frequency        Weight    Proportion

                            1     _1                  50       50.0000      0.333333
                            2     _2                  38       38.0000      0.253333
                            3     _3                  62       62.0000      0.413333
```

```
                                     Fisher (1936) Iris Data
                             Canonical Discriminant Analysis of Iris Clusters

                                         The CANDISC Procedure

                                      Univariate Test Statistics

                                  F Statistics,    Num DF=2,   Den DF=147

                                        Total       Pooled      Between
                                     Standard     Standard     Standard                 R-Square
Variable       Label                 Deviation    Deviation    Deviation    R-Square    / (1-RSq)     F Value    Pr > F

SepalLength    Sepal Length in mm.      8.2807       4.3949       8.5893      0.7221       2.5984      190.98    <.0001
SepalWidth     Sepal Width in mm.       4.3587       3.2482       3.5774      0.4521       0.8252       60.65    <.0001
PetalLength    Petal Length in mm.     17.6530       4.2143      20.9336      0.9438      16.7849     1233.69    <.0001
PetalWidth     Petal Width in mm.       7.6224       2.4524       8.8164      0.8979       8.7916      646.18    <.0001


                                           Average R-Square

                                      Unweighted              0.7539604
                                      Weighted by Variance    0.8842753


                                Multivariate Statistics and F Approximations

                                        S=2    M=0.5    N=71

                    Statistic                        Value    F Value    Num DF    Den DF    Pr > F

                    Wilks' Lambda               0.03222337     164.55         8       288    <.0001
                    Pillai's Trace              1.25669612      61.29         8       290    <.0001
                    Hotelling-Lawley Trace     21.06722883     377.66         8     203.4    <.0001
                    Roy's Greatest Root        20.63266809     747.93         4       145    <.0001

                             NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                                     NOTE: F Statistic for Wilks' Lambda is exact.
```

```
                                                 Fisher (1936) Iris Data
                                     Canonical Discriminant Analysis of Iris Clusters

                                                  The CANDISC Procedure

                                                        Adjusted    Approximate        Squared
                                        Canonical      Canonical       Standard      Canonical
                                      Correlation    Correlation          Error    Correlation

                                  1      0.976613       0.976123       0.003787       0.953774
                                  2      0.550384       0.543354       0.057107       0.302923

                                                                        Test of H0: The canonical correlations in the
                            Eigenvalues of Inv(E)*H                       current row and all that follow are zero
                              = CanRsq/(1-CanRsq)
                                                                   Likelihood   Approximate
               Eigenvalue   Difference   Proportion   Cumulative        Ratio       F Value   Num DF   Den DF   Pr > F

           1      20.6327      20.1981       0.9794       0.9794   0.03222337        164.55        8      288   <.0001
           2       0.4346                    0.0206       1.0000   0.69707749         21.00        3      145   <.0001
```

```
                                                 Fisher (1936) Iris Data
                                     Canonical Discriminant Analysis of Iris Clusters

                                                  The CANDISC Procedure

                                                Total Canonical Structure

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.          0.831965          0.452137
                         SepalWidth       Sepal Width in mm.          -0.515082          0.810630
                         PetalLength      Petal Length in mm.          0.993520          0.087514
                         PetalWidth       Petal Width in mm.           0.966325          0.154745


                                               Between Canonical Structure

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.          0.956160          0.292846
                         SepalWidth       Sepal Width in mm.          -0.748136          0.663545
                         PetalLength      Petal Length in mm.          0.998770          0.049580
                         PetalWidth       Petal Width in mm.           0.995952          0.089883


                                            Pooled Within Canonical Structure

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.          0.339314          0.716082
                         SepalWidth       Sepal Width in mm.          -0.149614          0.914351
                         PetalLength      Petal Length in mm.          0.900839          0.308136
                         PetalWidth       Petal Width in mm.           0.650123          0.404282
```

```
                                  Fisher (1936) Iris Data
                            Canonical Discriminant Analysis of Iris Clusters

                                   The CANDISC Procedure

                            Total-Sample Standardized Canonical Coefficients

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.       0.047747341       1.021487262
                         SepalWidth       Sepal Width in mm.       -0.577569244       0.864455153
                         PetalLength      Petal Length in mm.       3.341309573      -1.283043758
                         PetalWidth       Petal Width in mm.        0.996451144       0.900476563


                        Pooled Within-Class Standardized Canonical Coefficients

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.      0.0253414487      0.5421446856
                         SepalWidth       Sepal Width in mm.       -.4304161258      0.6442092294
                         PetalLength      Petal Length in mm.      0.7976741592      -.3063023132
                         PetalWidth       Petal Width in mm.       0.3205998034      0.2897207865


                                   Raw Canonical Coefficients

                         Variable         Label                            Can1              Can2

                         SepalLength      Sepal Length in mm.      0.0057661265      0.1233581748
                         SepalWidth       Sepal Width in mm.       -.1325106494      0.1983303556
                         PetalLength      Petal Length in mm.      0.1892773419      -.0726814163
                         PetalWidth       Petal Width in mm.       0.1307270927      0.1181359305


                               Class Means on Canonical Variables

                                   CLUSTER              Can1              Can2

                                         1      -6.131527227       0.244761516
                                         2       4.931414018       0.861972277
                                         3       1.922300462      -0.725693908
```



Example 28.2. Outliers

The second example involves data artificially generated to contain two clusters and several severe outliers. A preliminary analysis specifies twenty clusters and outputs an OUTSEED= data set to be used for a diagnostic plot. The exact number of initial clusters is not important; similar results could be obtained with ten or fifty initial clusters. Examination of the plot suggests that clusters with more than five (again, the exact number is not important) observations may yield good seeds for the main analysis. A DATA step deletes clusters with five or fewer observations, and the remaining cluster means provide seeds for the next PROC FASTCLUS analysis.

Two clusters are requested; the LEAST= option specifies the mean absolute deviation criterion (LEAST=1) . Values of the LEAST= option less than 2 reduce the effect of outliers on cluster centers.

The next analysis also requests two clusters; the STRICT= option is specified to prevent outliers from distorting the results. The STRICT= value is chosen to be close to the _GAP_ and _RADIUS_ values of the larger clusters in the diagnostic plot; the exact value is not critical.

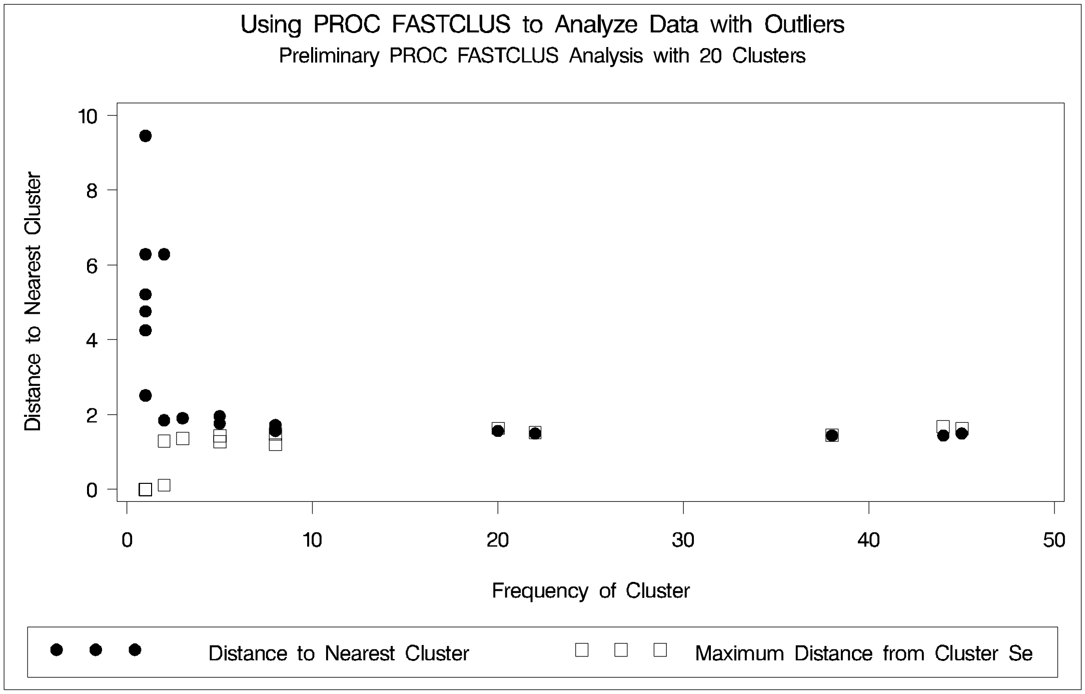
A final PROC FASTCLUS run assigns the outliers to clusters. The results are displayed in Output 28.2.1 through Output 28.2.4.

```
   /*   Create artificial data set with two clusters         */
   /*   and some outliers.                                   */

   data x;
```

```
title ’Using PROC FASTCLUS to Analyze Data with Outliers’;
   drop n;
   do n=1 to 100;
      x=rannor(12345)+2;
      y=rannor(12345);
      output;
   end;
   do n=1 to 100;
      x=rannor(12345)-2;
      y=rannor(12345);
      output;
   end;
   do n=1 to 10;
      x=10*rannor(12345);
      y=10*rannor(12345);
      output;
   end;
run;

/* Run PROC FASTCLUS with many clusters and OUTSEED= output */
/* data set for diagnostic plot.                            */

title2 ’Preliminary PROC FASTCLUS Analysis with 20 Clusters’;
proc fastclus data=x outseed=mean1 maxc=20 maxiter=0 summary;
   var x y;
run;

legend1 frame cframe=ligr label=none cborder=black
        position=center value=(justify=center);

axis1 label=(angle=90 rotate=0) minor=none order=(0 to 10 by 2);
axis2 minor=none ;

proc gplot data=mean1;
     plot  _gap_*_freq_ _radius_*_freq_ /overlay frame
     cframe=ligr vaxis=axis1 haxis=axis2 legend=legend1;
run;
```

Output 28.2.1. Preliminary Analysis of Data with Outliers: PROC FASTCLUS and PROC GPLOT

```
                         Using PROC FASTCLUS to Analyze Data with Outliers
                        Preliminary PROC FASTCLUS Analysis with 20 Clusters

                                     The FASTCLUS Procedure
                          Replace=FULL  Radius=0  Maxclusters=20 Maxiter=0

                               Criterion Based on Final Seeds =   0.6873


                                         Cluster Summary

                                          Maximum Distance
                           RMS Std               from Seed      Radius     Nearest     Distance Between
Cluster     Frequency     Deviation         to Observation    Exceeded     Cluster     Cluster Centroids
-------------------------------------------------------------------------------------------------------
     1              8        0.4753                 1.1924                      19                1.7205
     2              1             .                      0                       6                6.2847
     3             44        0.6252                 1.6774                       5                1.4386
     4              1             .                      0                      20                5.2130
     5             38        0.5603                 1.4528                       3                1.4386
     6              2        0.0542                 0.1085                       2                6.2847
     7              1             .                      0                      14                2.5094
     8              2        0.6480                 1.2961                       1                1.8450
     9              1             .                      0                       7                9.4534
    10              1             .                      0                      18                4.2514
    11              1             .                      0                      16                4.7582
    12             20        0.5911                 1.6291                      16                1.5601
    13              5        0.6682                 1.4244                       3                1.9553
    14              1             .                      0                       7                2.5094
    15              5        0.4074                 1.2678                       3                1.7609
    16             22        0.4168                 1.5139                      19                1.4936
    17              8        0.4031                 1.4794                       5                1.5564
    18              1             .                      0                      10                4.2514
    19             45        0.6475                 1.6285                      16                1.4936
    20              3        0.5719                 1.3642                      15                1.8999


                                  Pseudo F Statistic =   207.58


                        Approximate Expected Over-All R-Squared =   0.96103


                                Cubic Clustering Criterion =   -2.503

                    WARNING: The two values above are invalid for correlated variables.
```



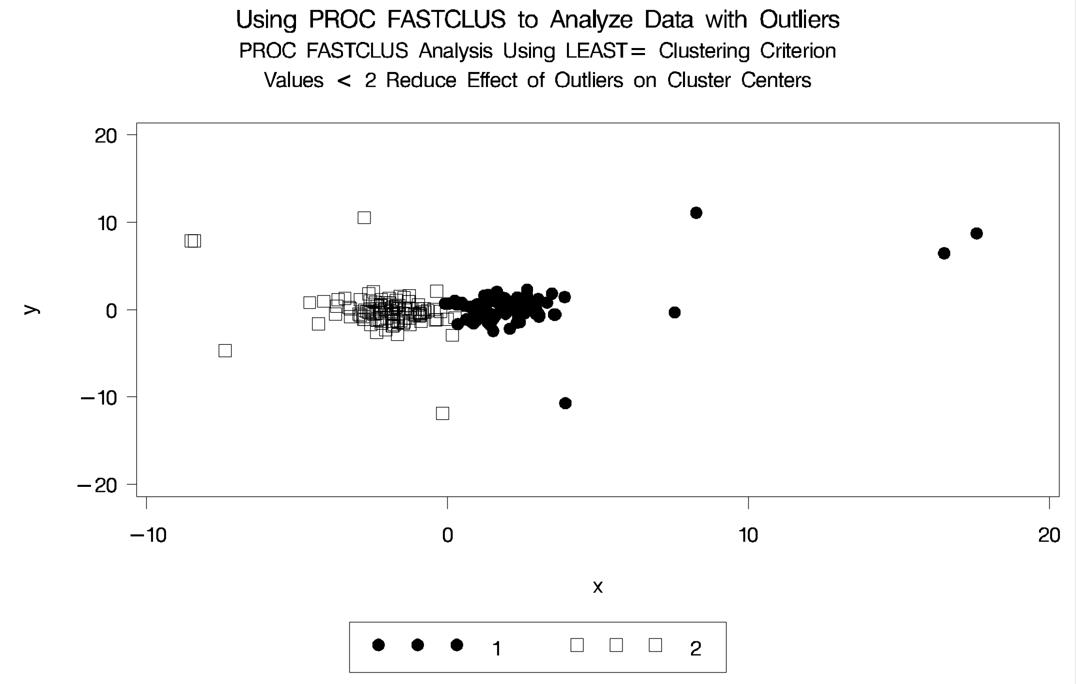
```
   /*    Remove low frequency clusters.  */
data seed;
   set mean1;
   if _freq_>5;
run;

   /*   Run PROC FASTCLUS again, selecting seeds from the          */
   /*   high frequency clusters in the previous analysis           */
   /*   using LEAST=1 Clustering Criterion                         */

title2 'PROC FASTCLUS Analysis Using LEAST= Clustering Criterion';
title3 'Values < 2 Reduce Effect of Outliers on Cluster Centers';
proc fastclus data=x seed=seed maxc=2 least=1 out=out;
   var x y;
run;

legend1 frame cframe=ligr label=none cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

proc gplot data=out;
   plot y*x=cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

Output 28.2.2. Analysis of Data with Outliers using the LEAST= Option

```
                 Using PROC FASTCLUS to Analyze Data with Outliers
                PROC FASTCLUS Analysis Using LEAST= Clustering Criterion
                Values < 2 Reduce Effect of Outliers on Cluster Centers

                              The FASTCLUS Procedure
       Replace=FULL  Radius=0  Maxclusters=2 Maxiter=20  Converge=0.0001  Least=1

                                   Initial Seeds

                       Cluster                 x                 y
                       -------------------------------------------
                          1          2.794174248      -0.065970836
                          2         -2.027300384      -2.051208579


                 Minimum Distance Between Initial Seeds = 6.806712


                 Preliminary L(1) Scale Estimate =        2.796579
```

```
                 Using PROC FASTCLUS to Analyze Data with Outliers
                PROC FASTCLUS Analysis Using LEAST= Clustering Criterion
                Values < 2 Reduce Effect of Outliers on Cluster Centers

                              The FASTCLUS Procedure
       Replace=FULL  Radius=0  Maxclusters=2 Maxiter=20  Converge=0.0001  Least=1

                           Number of Bins =      100

                                Iteration History

                                                       Relative Change
                                          Maximum      in Cluster Seeds
                Iteration    Criterion    Bin Size          1           2
                ----------------------------------------------------------
                        1       1.3983      0.2263      0.4091      0.6696
                        2       1.0776      0.0226     0.00511      0.0452
                        3       1.0771     0.00226     0.00229     0.00234
                        4       1.0771    0.000396    0.000253    0.000144
                        5       1.0771    0.000396           0           0


           Convergence criterion is satisfied.
```

```
                     Using PROC FASTCLUS to Analyze Data with Outliers
                   PROC FASTCLUS Analysis Using LEAST= Clustering Criterion
                   Values < 2 Reduce Effect of Outliers on Cluster Centers

                                  The FASTCLUS Procedure
          Replace=FULL  Radius=0  Maxclusters=2 Maxiter=20  Converge=0.0001  Least=1

                            Criterion Based on Final Seeds =   1.0771


                                      Cluster Summary

                              Mean    Maximum Distance
                          Absolute           from Seed     Radius     Nearest    Distance Between
Cluster     Frequency    Deviation      to Observation   Exceeded     Cluster     Cluster Medians
-------------------------------------------------------------------------------------------------
   1              102       1.1278             24.1622                      2              4.2585
   2              108       1.0494             14.8292                      1              4.2585


                                      Cluster Medians

                           Cluster                  x                 y
                           -------------------------------------------
                              1          1.923023887       0.222482918
                              2         -1.826721743      -0.286253041


                             Mean Absolute Deviations from Final Seeds

                           Cluster                  x                 y
                           -------------------------------------------
                              1          1.113465261       1.142120480
                              2          0.890331835       1.208370913
```


Using PROC FASTCLUS to Analyze Data with Outliers
PROC FASTCLUS Analysis Using LEAST= Clustering Criterion
Values < 2 Reduce Effect of Outliers on Cluster Centers

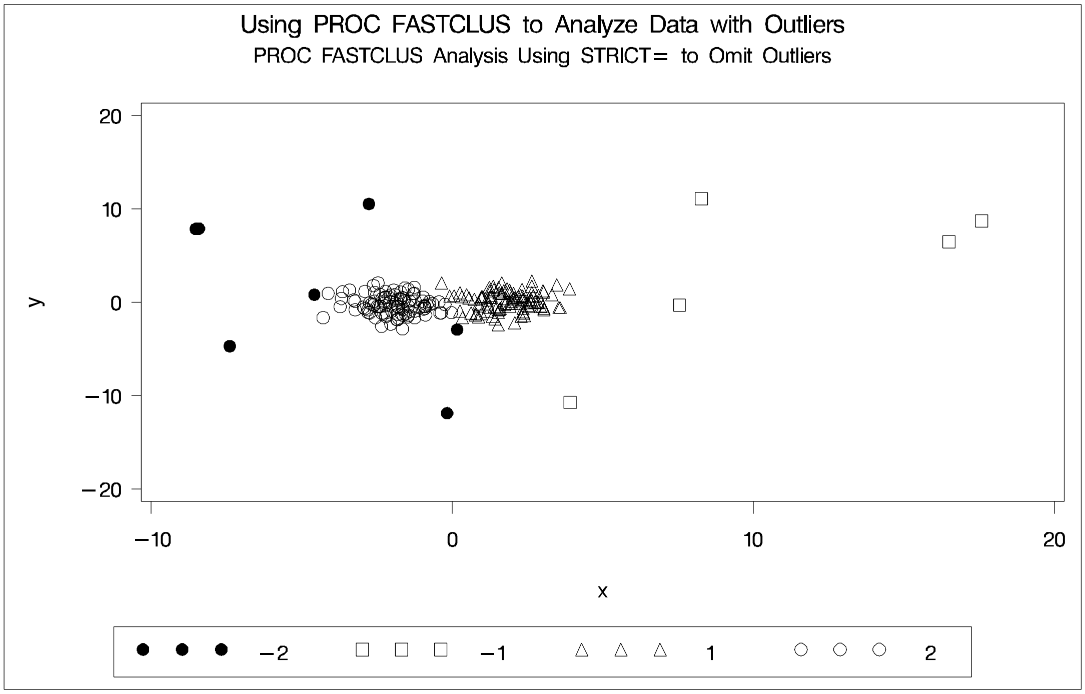
```
/*   Run PROC FASTCLUS again, selecting seeds from the         */
/*   high frequency clusters in the previous analysis          */
/*   STRICT= prevents outliers from distorting the results.    */
```

```
title2 'PROC FASTCLUS Analysis Using STRICT= to Omit Outliers';
proc fastclus data=x seed=seed
     maxc=2 strict=3.0 out=out outseed=mean2;
   var x y;
run;

proc gplot data=out;
   plot y*x=cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

Output 28.2.3. Cluster Analysis with Outliers Omitted: PROC FASTCLUS and PROC GPLOT

```
                           Using PROC FASTCLUS to Analyze Data with Outliers
                          PROC FASTCLUS Analysis Using STRICT= to Omit Outliers

                                         The FASTCLUS Procedure
                          Replace=FULL  Radius=0  Strict=3  Maxclusters=2 Maxiter=1

                                             Initial Seeds

                                 Cluster                 x                 y
                                 -------------------------------------------
                                    1          2.794174248      -0.065970836
                                    2         -2.027300384      -2.051208579


                                  Criterion Based on Final Seeds =   0.9515


                                            Cluster Summary

                                         Maximum Distance
                            RMS Std            from Seed      Radius     Nearest     Distance Between
Cluster      Frequency     Deviation       to Observation    Exceeded    Cluster    Cluster Centroids
-----------------------------------------------------------------------------------------------------
   1               99       0.9501               2.9589                       2               3.7666
   2               99       0.9290               2.8011                       1               3.7666


 12 Observation(s) were not assigned to a cluster because the minimum distance to a cluster seed
                                    exceeded the STRICT= value.


                                       Statistics for Variables

                Variable     Total STD     Within STD      R-Square     RSQ/(1-RSQ)
                -------------------------------------------------------------------
                x              2.06854        0.87098      0.823609        4.669219
                y              1.02113        1.00352      0.039093        0.040683
                OVER-ALL       1.63119        0.93959      0.669891        2.029303


                                   Pseudo F Statistic =   397.74


                        Approximate Expected Over-All R-Squared =   0.60615


                              Cubic Clustering Criterion =    3.197

                      WARNING: The two values above are invalid for correlated variables.
```

```
                  Using PROC FASTCLUS to Analyze Data with Outliers
                  PROC FASTCLUS Analysis Using STRICT= to Omit Outliers

                               The FASTCLUS Procedure
             Replace=FULL  Radius=0  Strict=3  Maxclusters=2 Maxiter=1

                                   Cluster Means

                Cluster                  x                  y
                -------------------------------------------
                   1          1.825111432        0.141211701
                   2         -1.919910712       -0.261558725


                             Cluster Standard Deviations

                Cluster                  x                  y
                -------------------------------------------
                   1          0.889549271        1.006965219
                   2          0.852000588        1.000062579
```

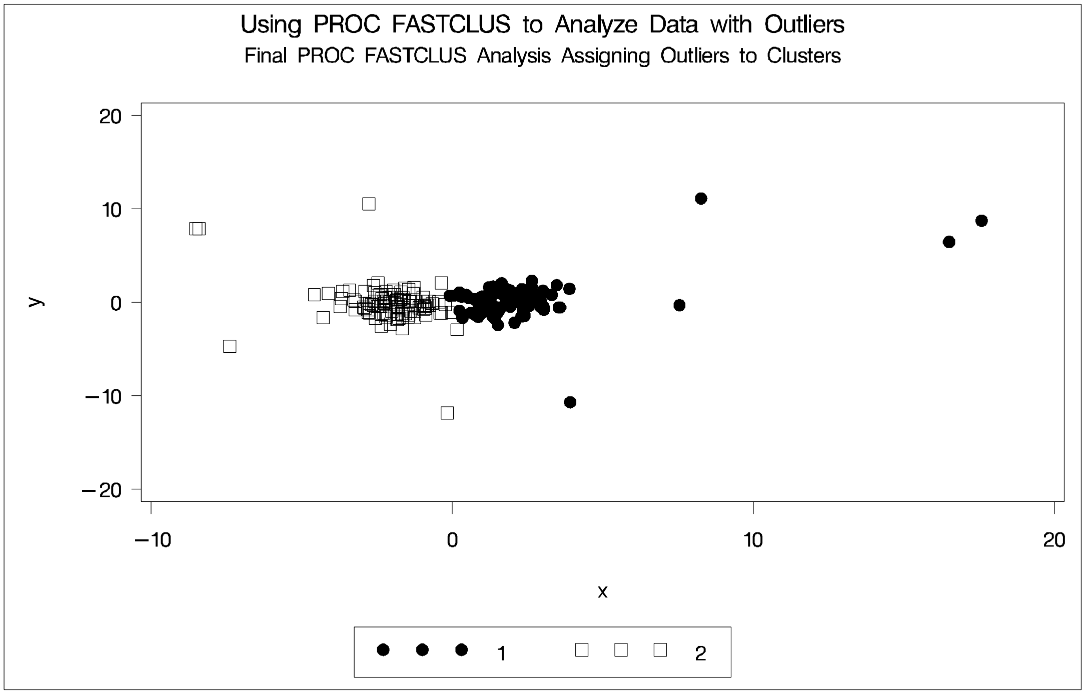


```
   /* Run PROC FASTCLUS one more time with zero iterations */
   /* to assign outliers and tails to clusters.            */
title2 'Final PROC FASTCLUS Analysis Assigning Outliers to '
       'Clusters';
proc fastclus data=x seed=mean2 maxc=2 maxiter=0 out=out;
   var x y;
run;


proc gplot data=out;
   plot y*x=cluster/frame cframe=ligr
                  legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

Output 28.2.4. Final Analysis with Outliers Assigned to Clusters: PROC FASTCLUS and PROC GPLOT

```
                         Using PROC FASTCLUS to Analyze Data with Outliers
                      Final PROC FASTCLUS Analysis Assigning Outliers to Clusters

                                     The FASTCLUS Procedure
                         Replace=FULL  Radius=0  Maxclusters=2 Maxiter=0

                                         Initial Seeds

                            Cluster                 x                 y
                            -------------------------------------------
                               1          1.825111432       0.141211701
                               2         -1.919910712      -0.261558725


                             Criterion Based on Final Seeds =   2.0594


                                        Cluster Summary

                                     Maximum Distance
                       RMS Std              from Seed     Radius     Nearest     Distance Between
Cluster     Frequency    Deviation       to Observation   Exceeded    Cluster    Cluster Centroids
-------------------------------------------------------------------------------------------------
   1              103       2.2569              17.9426                     2               4.3753
   2              107       1.8371              11.7362                     1               4.3753


                                   Statistics for Variables

                Variable     Total STD    Within STD      R-Square     RSQ/(1-RSQ)
                ------------------------------------------------------------------
                x              2.92721       1.95529      0.555950        1.252000
                y              2.15248       2.14754      0.009347        0.009435
                OVER-ALL       2.56922       2.05367      0.364119        0.572621


                                Pseudo F Statistic =   119.11


                      Approximate Expected Over-All R-Squared =   0.49090


                               Cubic Clustering Criterion =   -5.338

                 WARNING: The two values above are invalid for correlated variables.
```

```
                         Using PROC FASTCLUS to Analyze Data with Outliers
                      Final PROC FASTCLUS Analysis Assigning Outliers to Clusters

                                     The FASTCLUS Procedure
                         Replace=FULL  Radius=0  Maxclusters=2 Maxiter=0

                                         Cluster Means

                            Cluster                 x                 y
                            -------------------------------------------
                               1          2.280017469       0.263940765
                               2         -2.075547895      -0.151348765


                                 Cluster Standard Deviations

                            Cluster                 x                 y
                            -------------------------------------------
                               1          2.412264861       2.089922815
                               2          1.379355878       2.201567557
```



References

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, New York: Academic Press, Inc.

Bock, H. H. (1985), “On Some Significance Tests in Cluster Analysis,” *Journal of Classification*, 2, 77–108.

Calinski, T. and Harabasz, J. (1974), “A Dendrite Method for Cluster Analysis,” *Communications in Statistics*, 3, 1–27.

Cooper, M.C. and Milligan, G.W. (1988), “The Effect of Error on Determining the Number of Clusters,” Proceedings of the International Workshop on Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing and Related Areas of Research.

Everitt, B.S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books Ltd.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Gonin, R. and Money, A.H. (1989), *Nonlinear L_p-Norm Estimation*, New York: Marcel Dekker.

Hartigan, J.A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons, Inc.

Hartigan, J.A. (1985), “Statistical Theory in Clustering,” *Journal of Classification*, 2, 63–76.

Journal of Statistics Education, “Fish Catch Data Set,” [http://www.stat.ncsu.edu/info/jse], accessed 4 December 1997.

MacQueen, J.B. (1967), “Some Methods for Classification and Analysis of Multivariate Observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture Models*, New York: Marcel Dekker, Inc.

Mezzich, J.E and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press, Inc.

Milligan, G.W. (1980), “An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms,” *Psychometrika*, 45, 325–342.

Milligan, G.W. and Cooper, M.C. (1985), “An Examination of Procedures for Determining the Number of Clusters in a Data Set,” *Psychometrika, 50,* 159–179.

Pollard, D. (1981), “Strong Consistency of *k* -Means Clustering,” *Annals of Statistics*, 9, 135–140.

Sarle, W.S. (1983), “The Cubic Clustering Criterion,” SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, England: Ellis Horwood.

Spath, H. (1985), *Cluster Dissection and Analysis*, Chichester, England: Ellis Horwood.

Titterington, D.M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, New York: John Wiley & Sons.

Tou, J.T. and Gonzalez, R.C. (1974), *Pattern Recognition Principles*, Reading, MA: The Addison-Wesley Publishing Co.

Chapter 29
The FREQ Procedure

Chapter Contents

Chapter 29
The FREQ Procedure

Overview

The FREQ procedure produces one-way to *n*-way frequency and crosstabulation (contingency) tables. For two-way tables, PROC FREQ computes tests and measures of association. For *n*-way tables, PROC FREQ does stratified analysis, computing statistics within, as well as across, strata. Frequencies and statistics can also be output to SAS data sets.

For one-way frequency tables, PROC FREQ can compute statistics to test for equal proportions, specified proportions, or the binomial proportion. For contingency tables, PROC FREQ can compute various statistics to examine the relationships between two classification variables adjusting for any stratification variables. PROC FREQ automatically displays the output in a report and can also save the output in a SAS data set.

For some pairs of variables, you may want to examine the existence or the strength of any association between the variables. To determine if an association exists, chi-square tests are computed. To estimate the strength of an association, PROC FREQ computes measures of association that tend to be close to zero when there is no association and close to the maximum (or minimum) value when there is perfect association. The statistics for contingency tables include

- chi-square tests and measures
- measures of association
- risks (binomial proportions) and risk differences for 2×2 tables
- odds ratios and relative risks for 2×2 tables
- tests for trend
- tests and measures of agreement
- Cochran-Mantel-Haenszel statistics

PROC FREQ computes asymptotic standard errors, confidence intervals, and tests for measures of association and measures of agreement. Exact *p*-values and confidence intervals are available for various test statistics and measures. PROC FREQ also performs stratified analyses that compute statistics within, as well as across, strata for *n*-way tables. The statistics include Cochran-Mantel-Haenszel statistics and measures of agreement.

In choosing measures of association to use in analyzing a two-way table, you should consider the study design (which indicates whether the row and column variables are dependent or independent), the measurement scale of the variables (nominal, ordinal, or interval), the type of association that each measure is designed to detect, and any assumptions required for valid interpretation of a measure. You should exercise care in selecting measures that are appropriate for your data.

Similar comments apply to the choice and interpretation of the test statistics. For example, the Mantel-Haenszel chi-square statistic requires an ordinal scale for both variables and is designed to detect a linear association. The Pearson chi-square, on the other hand, is appropriate for all variables and can detect any kind of association, but it is less powerful for detecting a linear association because its power is dispersed over a greater number of degrees of freedom (except for 2×2 tables).

Several SAS procedures produce frequency counts; only PROC FREQ computes chi-square tests for one-way to *n*-way tables and measures of association and agreement for contingency tables. Other procedures to consider for counting are TABULATE, CHART, and UNIVARIATE. When you want to fit models to categorical data, use a procedure such as CATMOD, GENMOD, LOGISTIC, PHREG, or PROBIT.

For more information on selecting the appropriate statistical analyses, refer to Agresti (1996) or Stokes, Davis, and Koch (1995).

Getting Started

Frequency Tables and Statistics

The FREQ procedure provides easy access to statistics for testing for association in a crosstabulation table.

In this example, high school students applied for courses in a summer enrichment program: these courses included journalism, art history, statistics, graphic arts, and computer programming. The students accepted were randomly assigned to classes with and without internships in local companies. The following table contains counts of the students who enrolled in the summer program by gender and whether they were assigned an internship slot.

Table 29.1. Summer Enrichment Data

		Enrollment		
Gender	**Internship**	**Yes**	**No**	**Total**
boys	yes	35	29	64
boys	no	14	27	41
girls	yes	32	10	42
girls	no	53	23	76

The SAS data set SummerSchool is created by inputting the summer enrichment data as cell count data, or providing the frequency count for each combination of variable values. The following DATA step statements create the SAS data set SummerSchool.

```
data SummerSchool;
   input Gender $ Internship $ Enrollment $ Count @@;
   datalines;
boys  yes yes 35   boys  yes no 29
boys   no yes 14   boys   no no 27
girls yes yes 32   girls yes no 10
girls  no yes 53   girls  no no 23
;
```

The variable Gender takes the values ‘boys’ or ‘girls’, the variable Internship takes the values ‘yes’ and ‘no’, and the variable Enrollment takes the values ‘yes’ and ‘no’. The variable Count contains the number of students corresponding to each combination of data values. The double at sign (@@) indicates that more than one observation is included on a single data line. In this DATA step, two observations are included on each line.

Researchers are interested in whether there is an association between internship status and summer program enrollment. The Pearson chi-square statistic is an appropriate statistic to assess the association in the corresponding 2 × 2 table. The following PROC FREQ statements specify this analysis.

You specify the table for which you want to compute statistics with the TABLES statement. You specify the statistics you want to compute with options after a slash (/) in the TABLES statement.

```
proc freq data=SummerSchool order=data;
   weight count;
   tables Internship*Enrollment / chisq;
run;
```

The ORDER= option controls the order in which variable values are displayed in the rows and columns of the table. By default, the values are arranged according to the alphanumeric order of their unformatted values. If you specify ORDER=DATA, the data are displayed in the same order as they occur in the input data set. Here, since ‘yes’ appears before ‘no’ in the data, ‘yes’ appears first in any table. Other options for controlling order include ORDER=FORMATTED, which orders according to the formatted values, and ORDER=FREQUENCY, which orders by descending frequency count.

In the TABLES statement, Internship*Enrollment specifies a table where the rows are internship status and the columns are program enrollment. The CHISQ option requests chi-square statistics for assessing association between these two variables. Since the input data are in cell count form, the WEIGHT statement is required. The WEIGHT statement names the variable Count, which provides the frequency of each combination of data values.

Figure 29.1 presents the crosstabulation of Internship and Enrollment. In each cell, the values printed under the cell count are the table percentage, row percentage, and column percentage, respectively. For example, in the first cell, 63.21 percent of those offered courses with internships accepted them and 36.79 percent did not.

```
                      The FREQ Procedure

              Table of Internship by Enrollment

             Internship     Enrollment

             Frequency|
             Percent  |
             Row Pct  |
             Col Pct  |yes     |no      |  Total
             ---------+--------+--------+
             yes      |     67 |     39 |    106
                      |  30.04 |  17.49 |  47.53
                      |  63.21 |  36.79 |
                      |  50.00 |  43.82 |
             ---------+--------+--------+
             no       |     67 |     50 |    117
                      |  30.04 |  22.42 |  52.47
                      |  57.26 |  42.74 |
                      |  50.00 |  56.18 |
             ---------+--------+--------+
             Total         134       89      223
                         60.09    39.91   100.00
```

Figure 29.1. Crosstabulation Table

```
        Statistics for Table of Internship by Enrollment

      Statistic                     DF       Value      Prob
      ------------------------------------------------------
      Chi-Square                     1      0.8189    0.3655
      Likelihood Ratio Chi-Square    1      0.8202    0.3651
      Continuity Adj. Chi-Square     1      0.5899    0.4425
      Mantel-Haenszel Chi-Square     1      0.8153    0.3666
      Phi Coefficient                       0.0606
      Contingency Coefficient               0.0605
      Cramer's V                            0.0606


                       Fisher's Exact Test
                ----------------------------------
                Cell (1,1) Frequency (F)        67
                Left-sided Pr <= F          0.8513
                Right-sided Pr >= F         0.2213

                Table Probability (P)       0.0726
                Two-sided Pr <= P           0.4122

                        Sample Size = 223
```

Figure 29.2. Statistics Produced with the CHISQ Option

Figure 29.2 displays the statistics produced by the CHISQ option. The Pearson chi-square statistic is labeled 'Chi-Square' and has a value of 0.8189 with 1 degree of freedom. The associated p-value is 0.3655, which means that there is no significant evidence of an association between internship status and program enrollment. The other chi-square statistics have similar values and are asymptotically equivalent. The other statistics (Phi Coefficient, Contingency Coefficient, and Cramer's V) are measures of association derived from the Pearson chi-square. For Fisher's exact test, the two-sided p-value is 0.4122, which also shows no association between internship status and program enrollment.

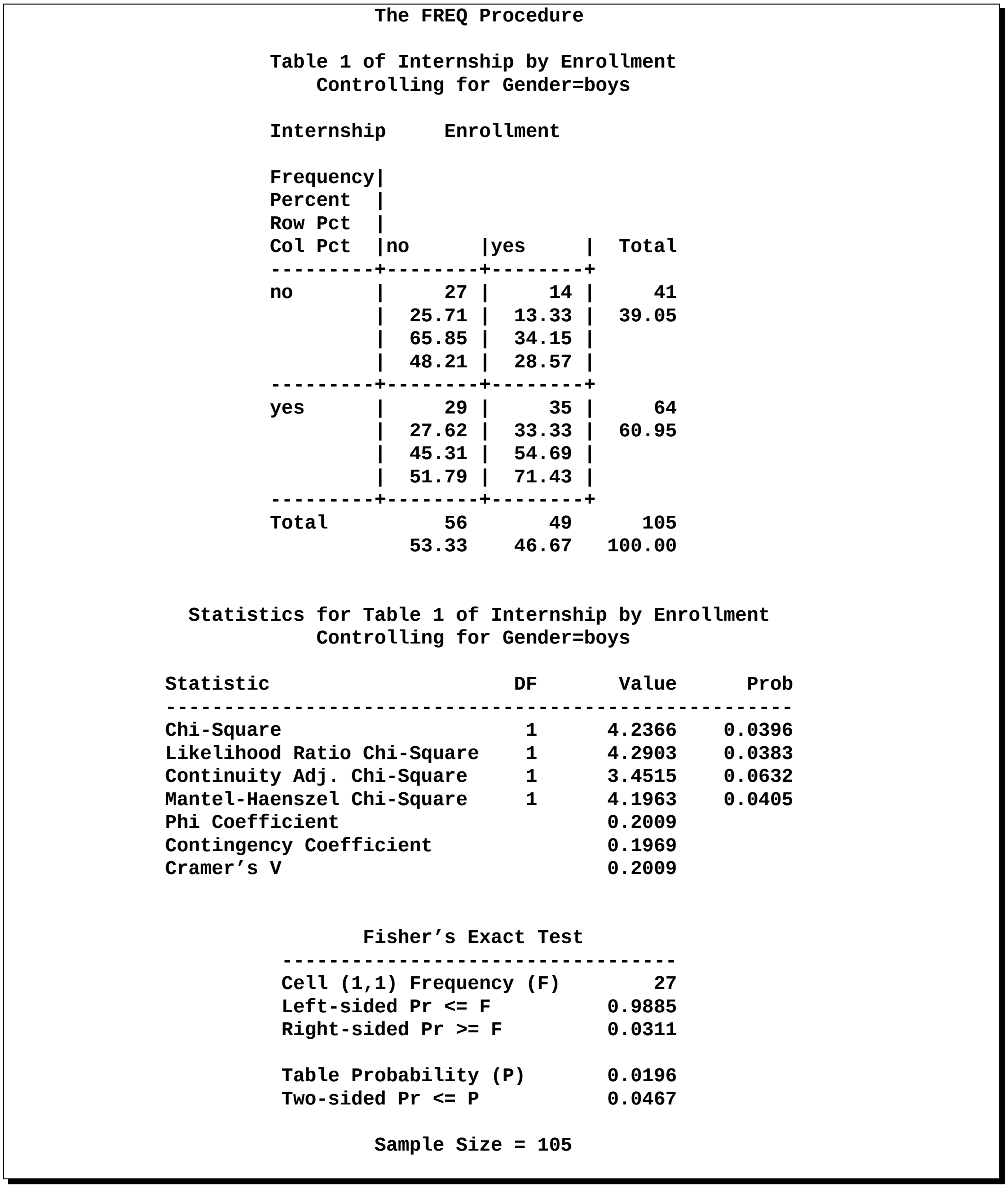
The analysis, so far, has ignored gender. However, it may be of interest to ask whether program enrollment is associated with internship status after adjusting for gender. You can address this question by doing an analysis of a set of tables, in this case, by analyzing the set consisting of one for boys and one for girls. The Cochran-Mantel-Haenszel statistic is appropriate for this situation: it addresses whether rows and columns are associated after controlling for the stratification variable. In this case, you would be stratifying by gender.

The FREQ statements for this analysis are very similar to those for the first analysis, except that there is a third variable, Gender, in the TABLES statement. When you cross more than two variables, the two rightmost variables construct the rows and columns of the table, respectively, and the leftmost variables determine the stratification.

```
proc freq data=SummerSchool;
   weight count;
   tables Gender*Internship*Enrollment / chisq cmh;
run;
```

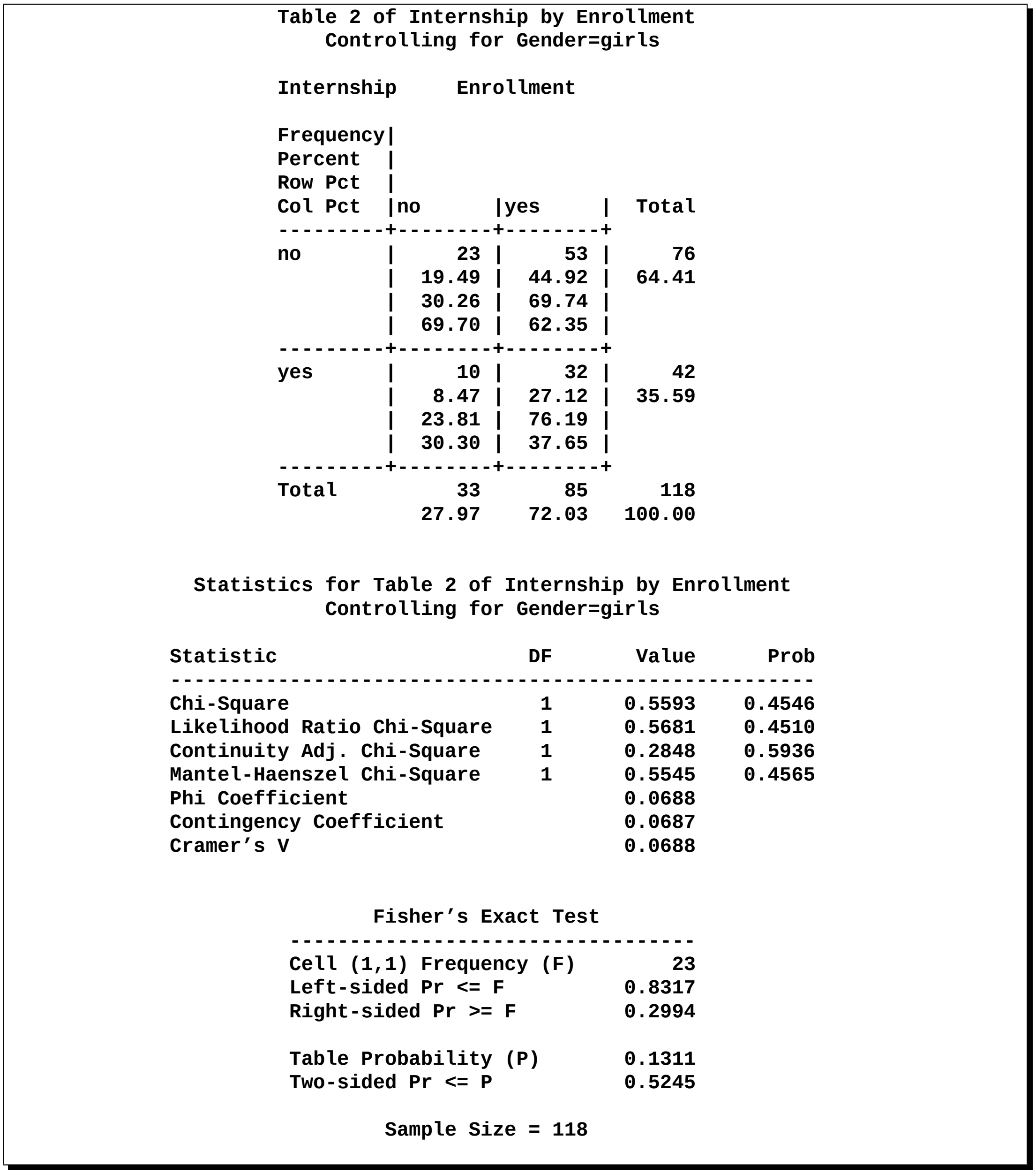
This execution of PROC FREQ first produces two individual crosstabulation tables of Internship*Enrollment, one for boys and one for girls. Chi-square statistics are produced for each individual table. Figure 29.3 shows the results for boys. Note that the chi-square statistic for boys is significant at the $\alpha = 0.05$ level of significance. Boys offered a course with an internship are more likely to enroll than boys who are not.

If you look at the individual table for girls, displayed in Figure 29.4, you see that there is no evidence of association for girls between internship offers and program enrollment.



```
                         The FREQ Procedure

                 Table 1 of Internship by Enrollment
                     Controlling for Gender=boys

                 Internship     Enrollment

                 Frequency|
                 Percent  |
                 Row Pct  |
                 Col Pct  |no      |yes     |  Total
                 ---------+--------+--------+
                 no       |     27 |     14 |     41
                          |  25.71 |  13.33 |  39.05
                          |  65.85 |  34.15 |
                          |  48.21 |  28.57 |
                 ---------+--------+--------+
                 yes      |     29 |     35 |     64
                          |  27.62 |  33.33 |  60.95
                          |  45.31 |  54.69 |
                          |  51.79 |  71.43 |
                 ---------+--------+--------+
                 Total          56       49      105
                             53.33    46.67   100.00


          Statistics for Table 1 of Internship by Enrollment
                     Controlling for Gender=boys

        Statistic                     DF       Value      Prob
        ------------------------------------------------------
        Chi-Square                     1      4.2366    0.0396
        Likelihood Ratio Chi-Square    1      4.2903    0.0383
        Continuity Adj. Chi-Square     1      3.4515    0.0632
        Mantel-Haenszel Chi-Square     1      4.1963    0.0405
        Phi Coefficient                       0.2009
        Contingency Coefficient               0.1969
        Cramer's V                            0.2009


                        Fisher's Exact Test
                  ----------------------------------
                  Cell (1,1) Frequency (F)        27
                  Left-sided Pr <= F          0.9885
                  Right-sided Pr >= F         0.0311

                  Table Probability (P)       0.0196
                  Two-sided Pr <= P           0.0467

                          Sample Size = 105
```

Figure 29.3. Crosstabulation Table and Statistics for Boys

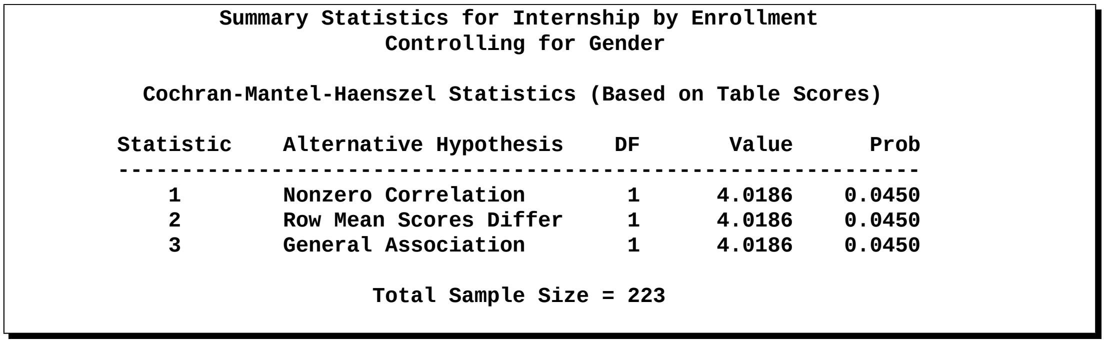


```
                       Table 2 of Internship by Enrollment
                           Controlling for Gender=girls

                       Internship     Enrollment

                       Frequency|
                       Percent  |
                       Row Pct  |
                       Col Pct  |no      |yes     |  Total
                       ---------+--------+--------+
                       no       |     23 |     53 |     76
                                |  19.49 |  44.92 |  64.41
                                |  30.26 |  69.74 |
                                |  69.70 |  62.35 |
                       ---------+--------+--------+
                       yes      |     10 |     32 |     42
                                |   8.47 |  27.12 |  35.59
                                |  23.81 |  76.19 |
                                |  30.30 |  37.65 |
                       ---------+--------+--------+
                       Total          33       85      118
                                   27.97    72.03   100.00


                Statistics for Table 2 of Internship by Enrollment
                           Controlling for Gender=girls

              Statistic                     DF       Value      Prob
              ------------------------------------------------------
              Chi-Square                     1      0.5593    0.4546
              Likelihood Ratio Chi-Square    1      0.5681    0.4510
              Continuity Adj. Chi-Square     1      0.2848    0.5936
              Mantel-Haenszel Chi-Square     1      0.5545    0.4565
              Phi Coefficient                       0.0688
              Contingency Coefficient               0.0687
              Cramer’s V                            0.0688


                              Fisher’s Exact Test
                        ----------------------------------
                        Cell (1,1) Frequency (F)        23
                        Left-sided Pr <= F          0.8317
                        Right-sided Pr >= F         0.2994

                        Table Probability (P)       0.1311
                        Two-sided Pr <= P           0.5245

                                Sample Size = 118
```

Figure 29.4. Crosstabulation Table and Statistics for Girls

These individual table results demonstrate the occasional problems with combining information into one table and not accounting for information in other variables such as Gender. Figure 29.5 contains the CMH results. There are three summary (CMH) statistics; which one you use depends on whether your rows and/or columns have an order in $r \times c$ tables. However, in the case of 2×2 tables, ordering does not matter and all three statistics take the same value. The CMH statistic follows the chi-square distribution under the hypothesis of no association, and here, it takes the value 4.0186 with 1 degree of freedom. The associated p-value is 0.0450, which indicates a significant association at the $\alpha = 0.05$ level.

Thus, when you adjust for the effect of gender in these data, there is an association between internship and program enrollment. But, if you ignore gender, no association is found. Note that the CMH option also produces other statistics, including estimates and confidence limits for relative risk and odds ratios for 2×2 tables and the Breslow-Day Test. These results are not displayed here.



```
                Summary Statistics for Internship by Enrollment
                             Controlling for Gender

          Cochran-Mantel-Haenszel Statistics (Based on Table Scores)

        Statistic    Alternative Hypothesis    DF       Value      Prob
        ---------------------------------------------------------------
            1        Nonzero Correlation        1      4.0186    0.0450
            2        Row Mean Scores Differ     1      4.0186    0.0450
            3        General Association        1      4.0186    0.0450

                             Total Sample Size = 223
```

Figure 29.5. Test for the Hypothesis of No Association

Agreement Study Example

Medical researchers are interested in evaluating the efficacy of a new treatment for a skin condition. Dermatologists from participating clinics were trained to conduct the study and to evaluate the condition. After the training, two dermatologists examined patients with the skin condition from a pilot study and rated the same patients. The possible evaluations are terrible, poor, marginal, and clear. Table 29.2 contains the data.

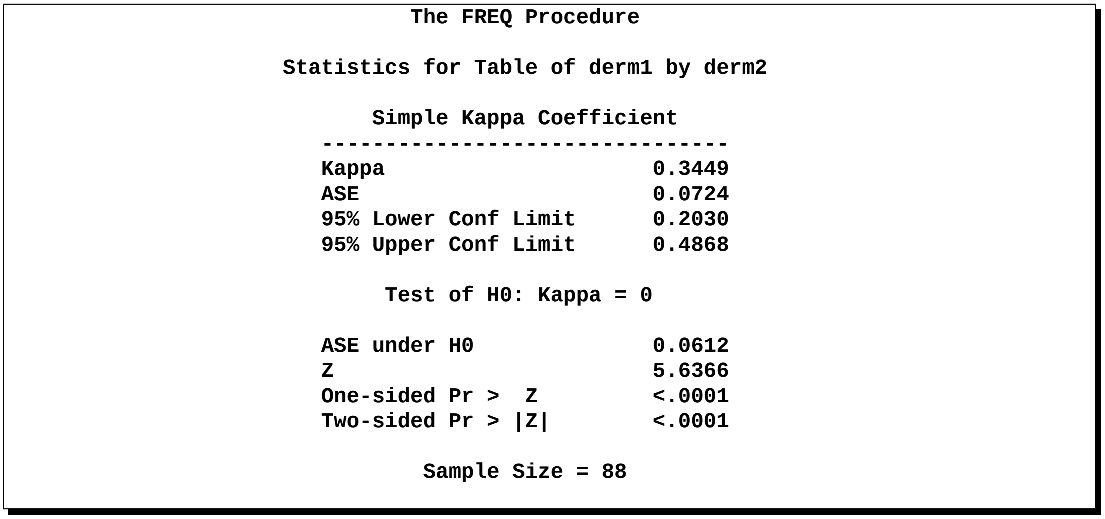
Table 29.2. Skin Condition Data

	Dermatologist 2			
Dermatologist 1	**Terrible**	**Poor**	**Marginal**	**Clear**
Terrible	10	4	1	0
Poor	5	10	12	2
Marginal	2	4	12	5
Clear	0	2	6	13

The dermatologists' evaluations of the patients are contained in the variables derm1 and derm2; the variable count is the number of patients given a particular pair of ratings. In order to evaluate the agreement of the diagnoses (a possible contribution to measurement error in the study), the *kappa coefficient* is computed. You specify the AGREE option in the TABLES statement and use the TEST statement to request a test for the null hypothesis that their agreement is purely by chance. You specify the keyword KAPPA to perform this test for the kappa coefficient. The results are shown in Figure 29.6.

```
      data SkinCondition;
         input derm1 $ derm2 $ count;
         datalines;
      terrible terrible 10
      terrible     poor 4
      terrible marginal 1
      terrible    clear 0
      poor     terrible 5
      poor         poor 10
      poor     marginal 12
      poor        clear 2
      marginal terrible 2
      marginal     poor 4
      marginal marginal 12
      marginal    clear 5
      clear    terrible 0
      clear        poor 2
      clear    marginal 6
      clear       clear 13
      ;
```

```
   proc freq data=SkinCondition order=data;
      weight count;
      tables derm1*derm2 / agree noprint;
      test kappa;
   run;
```



```
                          The FREQ Procedure

                  Statistics for Table of derm1 by derm2

                      Simple Kappa Coefficient
                 --------------------------------
                 Kappa                     0.3449
                 ASE                       0.0724
                 95% Lower Conf Limit      0.2030
                 95% Upper Conf Limit      0.4868

                      Test of H0: Kappa = 0

                 ASE under H0              0.0612
                 Z                         5.6366
                 One-sided Pr >  Z         <.0001
                 Two-sided Pr > |Z|        <.0001

                         Sample Size = 88
```

Figure 29.6. Agreement Study

The kappa coefficient has the value 0.3449, which indicates slight agreement between the dermatologists, and the hypothesis test confirms that you can reject the null hypothesis of no agreement. This conclusion is further supported by the confidence interval of (0.2030, 0.4868), which suggests that the true kappa is greater than zero. The AGREE option also produces Bowker's test for symmetry and the weighted kappa coefficient, but that output is not shown.

Syntax

The following statements are available in PROC FREQ.

PROC FREQ < *options* > ;
  BY *variables* ;
  EXACT *statistic-options* < **/** *computation-options* > ;
  OUTPUT < **OUT=***SAS-data-set* > *options* ;
  TABLES *requests* < **/** *options* > ;
  TEST *options* ;
  WEIGHT *variable* < **/** *option* > ;

The PROC FREQ statement is the only required statement for the FREQ procedure. If you specify the following statements, PROC FREQ produces a one-way frequency table for each variable in the most recently created data set.

```
proc freq;
run;
```

The rest of this section gives detailed syntax information for the BY, EXACT, OUTPUT, TABLES, TEST, and WEIGHT statements in alphabetical order after the description of the PROC FREQ statement. Table 29.3 summarizes the basic functions of each statement.

Table 29.3. Summary of PROC FREQ Statements

Statement	Description
BY	calculates separate frequency or crosstabulation tables for each BY group.
EXACT	requests exact tests for specified statistics.
OUTPUT	creates an output data set that contains specified statistics.
TABLES	specifies frequency or crosstabulation tables and requests tests and measures of association.
TEST	requests asymptotic tests for measures of association and agreement.
WEIGHT	identifies a variable with values that weight each observation.

PROC FREQ Statement

PROC FREQ < *options* > **;**

The PROC FREQ statement invokes the procedure.

The following table lists the options available in the PROC FREQ statement. Descriptions follow in alphabetical order.

Table 29.4. PROC FREQ Statement Options

Option	Description
DATA=	specifies the input data set.
COMPRESS	begins the next one-way table on the current page
FORMCHAR=	specifies the outline and cell divider characters for the cells of the crosstabulation table.
NLEVELS	displays the number of levels for all TABLES variables
NOPRINT	suppresses all displayed output.
ORDER=	specifies the order for listing variable values.
PAGE	displays one table per page.

You can specify the following options in the PROC FREQ statement.

COMPRESS

begins display of the next one-way frequency table on the same page as the preceding one-way table if there is enough space to begin the table. By default, the next one-way table begins on the current page only if the entire table fits on that page. The COMPRESS option is not valid with the PAGE option.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC FREQ. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

FORMCHAR (1,2,7) =’*formchar-string***’**

defines the characters to be used for constructing the outlines and dividers for the cells of contingency tables. The FORMCHAR= option can specify 20 different SAS formatting characters used to display output; however, PROC FREQ uses only the first, second, and seventh formatting characters. Therefore, the proper specification for PROC FREQ is FORMCHAR(1,2,7)= ’*formchar-string*’. The *formchar-string* should be three characters long. The characters are used to denote (1) vertical separator, (2) horizontal separator, and (7) vertical-horizontal intersection. You can use any character in *formchar-string*, including hexadecimal characters. If you use hexadecimal characters, you must put an `x` after the closing quote. For information on which hexadecimal codes to use for which characters, consult the documentation for your hardware.

Specifying all blanks for *formchar-string* produces tables with no outlines or dividers:

```
formchar (1,2,7)='   '
```

If you do not specify the FORMCHAR= option, PROC FREQ uses the default

```
formchar (1,2,7)='|-+'
```

Refer to the CALENDAR, PLOT, and TABULATE procedures in the *Base SAS 9.1 Procedures Guide* for more information on form characters.

Table 29.5. Formatting Characters Used by PROC FREQ

Position	Default	Used to Draw
1	|	vertical separators
2	-	horizontal separators
7	+	intersections of vertical and horizontal separators

NLEVELS

displays the "Number of Variable Levels" table. This table provides the number of levels for each variable named in the TABLES statements. See the section "Number of Variable Levels Table" on page 1517 for more information. PROC FREQ determines the variable levels from the formatted variable values, as described in the section "Grouping with Formats" on page 1465.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System." .

Note: A NOPRINT option is also available in the TABLES statement. It suppresses display of the crosstabulation tables but allows display of the requested statistics.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which the values of the frequency and crosstabulation table variables are to be reported. The following table shows how PROC FREQ interprets values of the ORDER= option.

DATA	orders values according to their order in the input data set.
FORMATTED	orders values by their formatted values. This order is operating-environment dependent. By default, the order is ascending.
FREQ	orders values by descending frequency count.
INTERNAL	orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is operating-environment dependent.

By default, ORDER=INTERNAL. The ORDER= option does not apply to missing values, which are always ordered first.

PAGE

displays only one table per page. Otherwise, PROC FREQ displays multiple tables per page as space permits. The PAGE option is not valid with the COMPRESS option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FREQ to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FREQ procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *Base SAS 9.1 Procedures Guide*.

EXACT Statement

EXACT *statistic-options* < */ computation-options* > **;**

The EXACT statement requests exact tests or confidence limits for the specified statistics. Optionally, PROC FREQ computes Monte Carlo estimates of the exact *p*-values. The *statistic-options* specify the statistics for which to provide exact tests or confidence limits. The *computation-options* specify options for the computation of exact statistics.

CAUTION: PROC FREQ computes exact tests with fast and efficient algorithms that are superior to direct enumeration. Exact tests are appropriate when a data set is small, sparse, skewed, or heavily tied. For some large problems, computation of exact tests may require a large amount of time and memory. Consider using asymptotic tests for such problems. Alternatively, when asymptotic methods may not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values. See the section “Computational Resources” on page 1511 for more information.

Statistic-Options

The *statistic-options* specify the statistics for which exact tests or confidence limits are computed. PROC FREQ can compute exact *p*-values for the following hypothesis tests: chi-square goodness-of-fit test for one-way tables; Pearson chi-square, likelihood-ratio chi-square, Mantel-Haenszel chi-square, Fisher's exact test, Jonckheere-Terpstra test, Cochran-Armitage test for trend, and McNemar's test for two-way tables. PROC FREQ can also compute exact *p*-values for tests of the following statistics: Pearson correlation coefficient, Spearman correlation coefficient, simple kappa coefficient, weighted kappa coefficient, and common odds ratio. PROC FREQ can compute exact *p*-values for the binomial proportion test for one-way tables, as well as exact confidence limits for the binomial proportion. Additionally, PROC FREQ can compute exact confidence limits for the odds ratio for 2×2 tables, as well as exact confidence limits for the common odds ratio for stratified 2×2 tables.

Table 29.6 lists the available *statistic-options* and the exact statistics computed. Most of the option names are identical to the corresponding options in the TABLES statement and the OUTPUT statement. You can request exact computations for groups of statistics by using options that are identical to the following TABLES statement options: CHISQ, MEASURES, and AGREE. For example, when you specify the CHISQ option in the EXACT statement, PROC FREQ computes exact *p*-values for the Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square tests. You request exact *p*-values for an individual test by specifying one of the *statistic-options* shown in Table 29.6 .

Table 29.6. EXACT Statement Statistic-Options

Option	Exact Statistics Computed
AGREE	McNemar's test for 2×2 tables, simple kappa coefficient, and weighted kappa coefficient
BINOMIAL	binomial proportion test for one-way tables
CHISQ	chi-square goodness-of-fit test for one-way tables; Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square tests for two-way tables
COMOR	confidence limits for the common odds ratio for $h \times 2 \times 2$ tables; common odds ratio test
FISHER	Fisher's exact test
JT	Jonckheere-Terpstra test
KAPPA	test for the simple kappa coefficient
LRCHI	likelihood-ratio chi-square test
MCNEM	McNemar's test
MEASURES	tests for the Pearson correlation and the Spearman correlation, and the odds ratio confidence limits for 2×2 tables
MHCHI	Mantel-Haenszel chi-square test
OR	confidence limits for the odds ratio for 2×2 tables
PCHI	Pearson chi-square test
PCORR	test for the Pearson correlation coefficient
SCORR	test for the Spearman correlation coefficient
TREND	Cochran-Armitage test for trend
WTKAP	test for the weighted kappa coefficient

Computation-Options

The *computation-options* specify options for computation of exact statistics. You can specify the following *computation-options* in the EXACT statement:

ALPHA=α

specifies the level of the confidence limits for Monte Carlo *p*-value estimates. The value of the ALPHA= option must be between 0 and 1, and the default is 0.01. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=.01 produces 99% confidence limits for the Monte Carlo estimates. The ALPHA= option invokes the MC option.

MAXTIME=*value*

specifies the maximum clock time (in seconds) that PROC FREQ can use to compute an exact *p*-value. If the procedure does not complete the computation within the specified time, the computation terminates. The value of the MAXTIME= option must be a positive number. The MAXTIME= option is valid for Monte Carlo estimation of exact *p*-values, as well as for direct exact *p*-value computation.

See the section "Computational Resources" on page 1511 for more information.

MC

requests Monte Carlo estimation of exact *p*-values instead of direct exact *p*-value computation. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations may not be sufficient. See the section "Monte Carlo Estimation" on page 1512 for more information.

The MC option is available for all EXACT *statistic-options* except BINOMIAL, COMOR, MCNEM, and OR. PROC FREQ computes only exact tests or confidence limits for those statistics.

The ALPHA=, N=, and SEED= options also invoke the MC option.

N=*n*

specifies the number of samples for Monte Carlo estimation. The value of the N= option must be a positive integer, and the default is 10000 samples. Larger values of *n* produce more precise estimates of exact *p*-values. Because larger values of *n* generate more samples, the computation time increases. The N= option invokes the MC option.

POINT

requests exact point probabilities for the test statistics.

The POINT option is available for all the EXACT statement *statistic-options* except the OR option, which provides exact confidence limits as opposed to an exact test. The POINT option is not available with the MC option.

SEED=*number*

specifies the initial seed for random number generation for Monte Carlo estimation. The value of the SEED= option must be an integer. If you do not specify the SEED= option, or if the SEED= value is negative or zero, PROC FREQ uses the time of day from the computer's clock to obtain the initial seed. The SEED= option invokes the MC option.

Using TABLES Statement Options with the EXACT Statement

If you use only one TABLES statement, you do not need to specify options in the TABLES statement that are identical to options appearing in the EXACT statement. PROC FREQ automatically invokes the corresponding TABLES statement option when you specify the option in the EXACT statement. However, when you use multiple TABLES statements and want exact computations, you must specify options in the TABLES statement to compute the desired statistics. PROC FREQ then performs exact computations for all statistics that are also specified in the EXACT statement.

OUTPUT Statement

OUTPUT < **OUT=** *SAS-data-set* > *options* **;**

The OUTPUT statement creates a SAS data set containing statistics computed by PROC FREQ. The variables contain statistics for each two-way table or stratum, as well as summary statistics across all strata.

Only one OUTPUT statement is allowed for each execution of PROC FREQ. You must specify a TABLES statement with the OUTPUT statement. If you use multiple TABLES statements, the contents of the OUTPUT data set correspond to the last TABLES statement. If you use multiple table requests in a TABLES statement, the contents of the OUTPUT data set correspond to the last table request.

For more information, see the section "Output Data Sets" on page 1514.

Note that you can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC FREQ output. For more information, see Table 29.11 on page 1525 and Chapter 14, "Using the Output Delivery System."

You can specify the following options in an OUTPUT statement.

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

options

specify the statistics that you want in the output data set. Available statistics are those produced by PROC FREQ for each one-way or two-way table, as well as the summary statistics across all strata. When you request a statistic, the OUTPUT data set contains that estimate or test statistic plus any associated standard error, confidence limits, *p*-values, and degrees of freedom. You can output statistics by using group options identical to those specified in the TABLES statement: AGREE, ALL, CHISQ, CMH, and MEASURES. Alternatively, you can request an individual statistic by specifying one of the options shown in the following table.

Table 29.7. OUTPUT Statement Options and Required TABLES Statement Options

Option	**Output Data Set Statistics**	**Required TABLES Statement Option**
AGREE	McNemar's test for 2×2 tables, simple kappa coefficient, and weighted kappa coefficient; for square tables with more than two response categories, Bowker's test of symmetry; for multiple strata, overall simple and weighted kappa statistics, and tests for equal kappas among strata; for multiple strata with two response categories, Cochran's Q test	AGREE
AJCHI	continuity-adjusted chi-square for 2×2 tables	ALL or CHISQ
ALL	all statistics under CHISQ, MEASURES, and CMH, and the number of nonmissing subjects	ALL
BDCHI	Breslow-Day test	ALL or CMH or CMH1 or CMH2
BIN | BINOMIAL	for one-way tables, binomial proportion statistics	BINOMIAL
CHISQ	chi-square goodness-of-fit test for one-way tables; for two-way tables, Pearson chi-square, likelihood-ratio chi-square, continuity-adjusted chi-square for 2×2 tables, Mantel-Haenszel chi-square, Fisher's exact test for 2×2 tables, phi coefficient, contingency coefficient, and Cramer's V	ALL or CHISQ
CMH	Cochran-Mantel-Haenszel correlation, row mean scores (ANOVA), and general association statistics; for 2×2 tables, logit and Mantel-Haenszel adjusted odds ratios, relative risks, and Breslow-Day test	ALL or CMH
CMH1	same as CMH, but excludes general association and row mean scores (ANOVA) statistics	ALL or CMH or CMH1
CMH2	same as CMH, but excludes the general association statistic	ALL or CMH or CMH2
CMHCOR	Cochran-Mantel-Haenszel correlation statistic	ALL or CMH or CMH1 or CMH2
CMHGA	Cochran-Mantel-Haenszel general association statistic	ALL or CMH
CMHRMS	Cochran-Mantel-Haenszel row mean scores (ANOVA) statistic	ALL or CMH or CMH2

Table 29.7. (continued)

Option	**Output Data Set Statistics**	**Required TABLES Statement Option**
COCHQ	Cochran's Q	AGREE
CONTGY	contingency coefficient	ALL or CHISQ
CRAMV	Cramer's V	ALL or CHISQ
EQKAP	test for equal simple kappas	AGREE
EQWKP	test for equal weighted kappas	AGREE
FISHER | EXACT	Fisher's exact test	ALL or CHISQ *
GAMMA	gamma	ALL or MEASURES
JT	Jonckheere-Terpstra test	JT
KAPPA	simple kappa coefficient	AGREE
KENTB	Kendall's tau-b	ALL or MEASURES
LAMCR	lambda asymmetric $(C\|R)$	ALL or MEASURES
LAMDAS	lambda symmetric	ALL or MEASURES
LAMRC	lambda asymmetric $(R\|C)$	ALL or MEASURES
LGOR	adjusted logit odds ratio	ALL or CMH or CMH1 or CMH2
LGRRC1	adjusted column 1 logit relative risk	ALL or CMH or CMH1 or CMH2
LGRRC2	adjusted column 2 logit relative risk	ALL or CMH or CMH1 or CMH2
LRCHI	likelihood-ratio chi-square	ALL or CHISQ
MCNEM	McNemar's test	AGREE
MEASURES	gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C\|R)$, Somers' $D(R\|C)$, Pearson correlation coefficient, Spearman correlation coefficient, lambda asymmetric $(C\|R)$, lambda asymmetric $(R\|C)$, lambda symmetric, uncertainty coefficient $(C\|R)$, uncertainty coefficient $(R\|C)$, and symmetric uncertainty coefficient; for 2 × 2 tables, odds ratio and relative risks	ALL or MEASURES
MHCHI	Mantel-Haenszel chi-square	ALL or CHISQ
MHOR	adjusted Mantel-Haenszel odds ratio	ALL or CMH or CMH1 or CMH2
MHRRC1	adjusted column 1 Mantel-Haenszel relative risk	ALL or CMH or CMH1 or CMH2
MHRRC2	adjusted column 2 Mantel-Haenszel relative risk	ALL or CMH or CMH1 or CMH2
N	number of nonmissing subjects for the stratum	
NMISS	number of missing subjects for the stratum	

*ALL and CHISQ compute Fisher's exact test for 2 × 2 tables. Use the FISHER option to compute Fisher's exact test for general *rxc* tables.

Table 29.7. (continued)

Option	**Output Data Set Statistics**	**Required TABLES Statement Option**
OR	odds ratio	ALL or MEASURES or RELRISK
PCHI	chi-square goodness-of-fit test for one-way tables; for two-way tables, Pearson chi-square	ALL or CHISQ
PCORR	Pearson correlation coefficient	ALL or MEASURES
PHI	phi coefficient	ALL or CHISQ
PLCORR	polychoric correlation coefficient	PLCORR
RDIF1	column 1 risk difference (row 1 - row 2)	RISKDIFF
RDIF2	column 2 risk difference (row 1 - row 2)	RISKDIFF
RELRISK	odds ratio and relative risks for 2 × 2 tables	ALL or MEASURES or RELRISK
RISKDIFF	risks and risk differences	RISKDIFF
RISKDIFF1	column 1 risks and risk difference	RISKDIFF
RISKDIFF2	column 2 risks and risk difference	RISKDIFF
RRC1	column 1 relative risk	ALL or MEASURES or RELRISK
RRC2	column 2 relative risk	ALL or MEASURES or RELRISK
RSK1	column 1 risk (overall)	RISKDIFF
RSK11	column 1 risk, for row 1	RISKDIFF
RSK12	column 2 risk, for row 1	RISKDIFF
RSK2	column 2 risk (overall)	RISKDIFF
RSK21	column 1 risk, for row 2	RISKDIFF
RSK22	column 2 risk, for row 2	RISKDIFF
SCORR	Spearman correlation coefficient	ALL or MEASURES
SMDCR	Somers' $D(C\|R)$	ALL or MEASURES
SMDRC	Somers' $D(R\|C)$	ALL or MEASURES
STUTC	Stuart's tau-c	ALL or MEASURES
TREND	Cochran-Armitage test for trend	TREND
TSYMM	Bowker's test of symmetry	AGREE
U	symmetric uncertainty coefficient	ALL or MEASURES
UCR	uncertainty coefficient $(C\|R)$	ALL or MEASURES
URC	uncertainty coefficient $(R\|C)$	ALL or MEASURES
WTKAP	weighted kappa coefficient	AGREE

Using the TABLES Statement with the OUTPUT Statement

In order to specify that the OUTPUT data set contain a particular statistic, you must have PROC FREQ compute the statistic by using the corresponding option in the TABLES statement or the EXACT statement. For example, you cannot specify the option PCHI (Pearson chi-square) in the OUTPUT statement without also specifying a TABLES statement option or an EXACT statement option to compute the Pearson chi-square. The TABLES statement option ALL or CHISQ computes the Pearson chi-square. Additionally, if you have only one TABLES statement, the EXACT statement option CHISQ or PCHI computes the Pearson chi-square.

TABLES Statement

TABLES *requests* < / *options* > **;**

The TABLES statement requests one-way to *n*-way frequency and crosstabulation tables and statistics for those tables.

If you omit the TABLES statement, PROC FREQ generates one-way frequency tables for all data set variables that are not listed in the other statements.

The following argument is required in the TABLES statement.

requests

specify the frequency and crosstabulation tables to produce. A request is composed of one variable name or several variable names separated by asterisks. To request a one-way frequency table, use a single variable. To request a two-way crosstabulation table, use an asterisk between two variables. To request a multiway table (an *n*-way table, where *n*>2), separate the desired variables with asterisks. The unique values of these variables form the rows, columns, and strata of the table.

For two-way to multiway tables, the values of the last variable form the crosstabulation table columns, while the values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one stratum. PROC FREQ produces a separate crosstabulation table for each stratum. For example, a specification of A*B*C*D in a TABLES statement produces *k* tables, where *k* is the number of different combinations of values for A and B. Each table lists the values for C down the side and the values for D across the top.

You can use multiple TABLES statements in the PROC FREQ step. PROC FREQ builds all the table requests in one pass of the data, so that there is essentially no loss of efficiency. You can also specify any number of table requests in a single TABLES statement. To specify multiple table requests quickly, use a grouping syntax by placing parentheses around several variables and joining other variables or variable combinations. For example, the following statements illustrate grouping syntax.

Table 29.8. Grouping Syntax

Request	**Equivalent to**
tables A*(B C);	tables A*B A*C;
tables (A B)*(C D);	tables A*C B*C A*D B*D;
tables (A B C)*D;	tables A*D B*D C*D;
tables A – – C;	tables A B C;
tables (A – – C)*D;	tables A*D B*D C*D;

Without Options

If you request a one-way frequency table for a variable without specifying options, PROC FREQ produces frequencies, cumulative frequencies, percentages of the total frequency, and cumulative percentages for each value of the variable. If you request a two-way or an *n*-way crosstabulation table without specifying options, PROC FREQ produces crosstabulation tables that include cell frequencies, cell percentages of the total frequency, cell percentages of row frequencies, and cell percentages of column frequencies. The procedure excludes observations with missing values from the table but displays the total frequency of missing observations below each table.

Options

The following table lists the options available with the TABLES statement. Descriptions follow in alphabetical order.

Table 29.9. TABLES Statement Options

Option	**Description**
Control Statistical Analysis	
AGREE	requests tests and measures of classification agreement
ALL	requests tests and measures of association produced by CHISQ, MEASURES, and CMH
ALPHA=	sets the confidence level for confidence limits
BDT	requests Tarone's adjustment for the Breslow-Day test
BINOMIAL	requests binomial proportion, confidence limits and test for one-way tables
BINOMIALC	requests BINOMIAL statistics with a continuity correction
CHISQ	requests chi-square tests and measures of association based on chi-square
CL	requests confidence limits for the MEASURES statistics
CMH	requests all Cochran-Mantel-Haenszel statistics
CMH1	requests the CMH correlation statistic, and adjusted relative risks and odds ratios
CMH2	requests CMH correlation and row mean scores (ANOVA) statistics, and adjusted relative risks and odds ratios
CONVERGE=	specifies convergence criterion to compute polychoric correlation
FISHER	requests Fisher's exact test for tables larger than 2×2
JT	requests Jonckheere-Terpstra test
MAXITER=	specifies maximum number of iterations to compute polychoric correlation
MEASURES	requests measures of association and their asymptotic standard errors
MISSING	treats missing values as nonmissing
PLCORR	requests polychoric correlation
RELRISK	requests relative risk measures for 2×2 tables
RISKDIFF	requests risks and risk differences for 2×2 tables
RISKDIFFC	requests RISKDIFF statistics with a continuity correction
SCORES=	specifies the type of row and column scores
TESTF=	specifies expected frequencies for a one-way table chi-square test
TESTP=	specifies expected proportions for a one-way table chi-square test
TREND	requests Cochran-Armitage test for trend

Table 29.9. (continued)

Option	Description
Control Additional Table Information	
CELLCHI2	displays each cell's contribution to the total Pearson chi-square statistic
CUMCOL	displays the cumulative column percentage in each cell
DEVIATION	displays the deviation of the cell frequency from the expected value for each cell
EXPECTED	displays the expected cell frequency for each cell
MISSPRINT	displays missing value frequencies
SPARSE	lists all possible combinations of variable levels even when a combination does not occur
TOTPCT	displays percentage of total frequency on *n*-way tables when $n > 2$
Control Displayed Output	
CONTENTS=	specifies the HTML contents link for crosstabulation tables
CROSSLIST	displays crosstabulation tables in ODS column format
FORMAT=	formats the frequencies in crosstabulation tables
LIST	displays two-way to *n*-way tables in list format
NOCOL	suppresses display of the column percentage for each cell
NOCUM	suppresses display of cumulative frequencies and cumulative percentages in one-way frequency tables and in list format
NOFREQ	suppresses display of the frequency count for each cell
NOPERCENT	suppresses display of the percentage, row percentage, and column percentage in crosstabulation tables, or percentages and cumulative percentages in one-way frequency tables and in list format
NOPRINT	suppresses display of tables but displays statistics
NOROW	suppresses display of the row percentage for each cell
NOSPARSE	suppresses zero cell frequencies in the list display and in the OUT= data set when ZEROS is specified
NOWARN	suppresses log warning message for the chi-square test
PRINTKWT	displays kappa coefficient weights
SCOROUT	displays the row and the column scores
Create an Output Data Set	
OUT=	specifies an output data set to contain variable values and frequency counts
OUTCUM	includes the cumulative frequency and cumulative percentage in the output data set for one-way tables
OUTEXPECT	includes the expected frequency of each cell in the output data set
OUTPCT	includes the percentage of column frequency, row frequency, and two-way table frequency in the output data set

You can specify the following options in a TABLES statement.

AGREE $<$ **(WT=FC)** $>$

requests tests and measures of classification agreement for square tables. The AGREE option provides McNemar's test for 2×2 tables and Bowker's test of symmetry for tables with more than two response categories. The AGREE option also produces the simple kappa coefficient, the weighted kappa coefficient, the asymptotic standard errors for the simple and weighted kappas, and the corresponding confidence limits. When there are multiple strata, the AGREE option provides overall simple and weighted kappas as well as tests for equal kappas among strata. When there are multiple strata and two response categories, PROC FREQ computes Cochran's Q test. For more information, see the section "Tests and Measures of Agreement" on page 1493.

The (WT=FC) specification requests that PROC FREQ use Fleiss-Cohen weights to compute the weighted kappa coefficient. By default, PROC FREQ uses Cicchetti-Allison weights. See the section "Weighted Kappa Coefficient" on page 1496 for more information. You can specify the PRINTKWT option to display the kappa coefficient weights.

AGREE statistics are computed only for square tables, where the number of rows equals the number of columns. If your table is not square due to observations with zero weights, you can use the ZEROS option in the WEIGHT statement to include these observations. For more details, see the section "Tables with Zero Rows and Columns" on page 1499.

ALL

requests all of the tests and measures that are computed by the CHISQ, MEASURES, and CMH options. The number of CMH statistics computed can be controlled by the CMH1 and CMH2 options.

ALPHA=α

specifies the level of confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

ALPHA= applies to confidence limits requested by TABLES statement options. There is a separate ALPHA= option in the EXACT statement that sets the level of confidence limits for Monte Carlo estimates of exact *p*-values, which are requested in the EXACT statement.

BDT

requests Tarone's adjustment in the Breslow-Day test for homogeneity of odds ratios. (You must specify the CMH option to compute the Breslow-Day test.) See the section "Breslow-Day Test for Homogeneity of the Odds Ratios" on page 1508 for more information.

BINOMIAL $<$ **(P=** *value***) | (LEVEL=** *level-number* **|** *level-value***)** $>$

requests the binomial proportion for one-way tables. The BINOMIAL option also provides the asymptotic standard error, asymptotic and exact confidence intervals, and the asymptotic test for the binomial proportion. To request an exact test for the binomial proportion, use the BINOMIAL option in the EXACT statement.

To specify the null hypothesis proportion for the test, use P=. If you omit P=*value*, PROC FREQ uses 0.5 as the default for the test. By default, BINOMIAL computes the proportion of observations for the first variable level that appears in the output. To specify a different level, use LEVEL=*level-number* or LEVEL=*level-value*, where *level-number* is the variable level's number or order in the output, and *level-value* is the formatted value of the variable level.

To include a continuity correction in the asymptotic confidence interval and test, use the BINOMIALC option instead of the BINOMIAL option.

See the section "Binomial Proportion" on page 1484 for more information.

BINOMIALC < **(P=** *value***) | (LEVEL=** *level-number* **|** *level-value***)** >

requests the BINOMIAL option statistics for one-way tables, and includes a continuity correction in the asymptotic confidence interval and the asymptotic test. The BINOMIAL option statistics include the binomial proportion, the asymptotic standard error, asymptotic and exact confidence intervals, and the asymptotic test for the binomial proportion. To request an exact test for the binomial proportion, use the BINOMIAL option in the EXACT statement.

To specify the null hypothesis proportion for the test, use P=. If you omit P=*value*, PROC FREQ uses 0.5 as the default for the test. By default BINOMIALC computes the proportion of observations for the first variable level that appears in the output. To specify a different level, use LEVEL=*level-number* or LEVEL=*level-value*, where *level-number* is the variable level's number or order in the output, and *level-value* is the formatted value of the variable level.

See the section "Binomial Proportion" on page 1484 for more information.

CELLCHI2

displays each crosstabulation table cell's contribution to the total Pearson chi-square statistic, which is computed as

$$\frac{(\mathit{frequency} - \mathit{expected})^2}{\mathit{expected}}$$

The CELLCHI2 option has no effect for one-way tables or for tables that are displayed with the LIST option.

CHISQ

requests chi-square tests of homogeneity or independence and measures of association based on chi-square. The tests include the Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square. The measures include the phi coefficient, the contingency coefficient, and Cramer's V. For 2×2 tables, the CHISQ option includes Fisher's exact test and the continuity-adjusted chi-square. For one-way tables, the CHISQ option requests a chi-square goodness-of-fit test for equal proportions. If you specify the null hypothesis proportions with the TESTP= option, then PROC FREQ computes a chi-square goodness-of-fit test for the specified proportions. If you specify null hypothesis frequencies with the TESTF= option, PROC

FREQ computes a chi-square goodness-of-fit test for the specified frequencies. See the section "Chi-Square Tests and Statistics" on page 1469 for more information.

CL

requests confidence limits for the MEASURES statistics. If you omit the MEASURES option, the CL option invokes MEASURES. The FREQ procedure determines the confidence coefficient using the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

For more information, see the section "Confidence Limits" on page 1475.

CMH

requests Cochran-Mantel-Haenszel statistics, which test for association between the row and column variables after adjusting for the remaining variables in a multiway table. In addition, for 2×2 tables, PROC FREQ computes the adjusted Mantel-Haenszel and logit estimates of the odds ratios and relative risks and the corresponding confidence limits. For the stratified 2×2 case, PROC FREQ computes the Breslow-Day test for homogeneity of odds ratios. (To request Tarone's adjustment for the Breslow-Day test, use the BDT option.) The CMH1 and CMH2 options control the number of CMH statistics that PROC FREQ computes. For more information, see the section "Cochran-Mantel-Haenszel Statistics" on page 1500.

CMH1

requests the Cochran-Mantel-Haenszel correlation statistic and, for 2×2 tables, the adjusted Mantel-Haenszel and logit estimates of the odds ratios and relative risks and the corresponding confidence limits. For the stratified 2×2 case, PROC FREQ computes the Breslow-Day test for homogeneity of odds ratios. Except for 2×2 tables, the CMH1 option requires less memory than the CMH option, which can require an enormous amount for large tables.

CMH2

requests the Cochran-Mantel-Haenszel correlation statistic, row mean scores (ANOVA) statistic, and, for 2×2 tables, the adjusted Mantel-Haenszel and logit estimates of the odds ratios and relative risks and the corresponding confidence limits. For the stratified 2×2 case, PROC FREQ computes the Breslow-Day test for homogeneity of odds ratios. Except for tables with two columns, the CMH2 option requires less memory than the CMH option, which can require an enormous amount for large tables.

CONTENTS=*link-text*

specifies the text for the HTML contents file links to crosstabulation tables. For information on HTML output, refer to the *SAS Output Delivery System User's Guide*. The CONTENTS= option affects only the HTML contents file, and not the HTML body file.

If you omit the CONTENTS= option, by default, the HTML link text for crosstabulation tables is "Cross-Tabular Freq Table."

Note that links to all crosstabulation tables produced by a single TABLES statement use the same text. To specify different text for different crosstabulation table links, request the tables in separate TABLES statements and use the CONTENTS= option in each TABLES statement.

The CONTENTS= option affects only links to crosstabulation tables. It does not affect links to other PROC FREQ tables. To specify link text for any other PROC FREQ table, you can use PROC TEMPLATE to create a customized table definition. The CONTENTS_LABEL attribute in the DEFINE TABLE statement of PROC TEMPLATE specifies the contents file link for the table. For detailed information, refer to the chapter titled "The TEMPLATE Procedure" in the *SAS Output Delivery System User's Guide*.

CONVERGE=*value*

specifies the convergence criterion for computing the polychoric correlation when you specify the PLCORR option. The value of the CONVERGE= option must be a positive number; by default, CONVERGE=0.0001. Iterative computation of the polychoric correlation stops when the convergence measure falls below the value of the CONVERGE= option or when the number of iterations exceeds the value specified in the MAXITER= option, whichever happens first.

See the section "Polychoric Correlation" on page 1482 for more information.

CROSSLIST

displays crosstabulation tables in ODS column format, instead of the default crosstabulation cell format. In a CROSSLIST table display, the rows correspond to the crosstabulation table cells, and the columns correspond to descriptive statistics such as Frequency, Percent, and so on. See the section "Multiway Tables" on page 1518 for details on the contents of the CROSSLIST table.

The CROSSLIST table displays the same information as the default crosstabulation table, but uses an ODS column format instead of the table cell format. Unlike the default crosstabulation table, the CROSSLIST table has a table definition that you can customize with PROC TEMPLATE. For more information, refer to the chapter titled "The TEMPLATE Procedure" in the *SAS Output Delivery System User's Guide*.

You can control the contents of a CROSSLIST table with the same options available for the default crosstabulation table. These include the NOFREQ, NOPERCENT, NOROW, and NOCOL options. You can request additional information in a CROSSLIST table with the CELLCHI2, DEVIATION, EXPECTED, MISSPRINT, and TOTPCT options.

The FORMAT= option and the CUMCOL option have no effect for CROSSLIST tables. You cannot specify both the LIST option and the CROSSLIST option in the same TABLES statement.

You can use the NOSPARSE option to suppress display of variable levels with zero frequency in CROSSLIST tables. By default for CROSSLIST tables, PROC FREQ displays all levels of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. And for multiway tables displayed with the CROSSLIST option, the procedure displays all levels of the row variable for each stratum of the table by default, including any row variable levels with zero frequency for the stratum.

CUMCOL

displays the cumulative column percentages in the cells of the crosstabulation table.

DEVIATION

displays the deviation of the cell frequency from the expected frequency for each cell of the crosstabulation table. The DEVIATION option is valid for contingency tables but has no effect on tables produced with the LIST option.

EXPECTED

displays the expected table cell frequencies under the hypothesis of independence (or homogeneity). The EXPECTED option is valid for crosstabulation tables but has no effect on tables produced with the LIST option.

FISHER | EXACT

requests Fisher's exact test for tables that are larger than 2×2. This test is also known as the Freeman-Halton test. For more information, see the section "Fisher's Exact Test" on page 1472 and the "EXACT Statement" section on page 1443.

If you omit the CHISQ option in the TABLES statement, the FISHER option invokes CHISQ. You can also request Fisher's exact test by specifying the FISHER option in the EXACT statement.

CAUTION: For tables with many rows or columns or with large total frequency, PROC FREQ may require a large amount of time or memory to compute exact *p*-values. See the section "Computational Resources" on page 1511 for more information.

FORMAT=*format-name*

specifies a format for the following crosstabulation table cell values: frequency, expected frequency, and deviation. PROC FREQ also uses this format to display the total row and column frequencies for crosstabulation tables.

You can specify any standard SAS numeric format or a numeric format defined with the FORMAT procedure. The format length must not exceed 24. If you omit FORMAT=, by default, PROC FREQ uses the BEST6. format to display frequencies less than 1E6, and the BEST7. format otherwise.

To change formats for all other FREQ tables, you can use PROC TEMPLATE. For information on this procedure, refer to the chapter titled "The TEMPLATE Procedure" in the *SAS Output Delivery System User's Guide*.

JT

performs the Jonckheere-Terpstra test. For more information, see the section "Jonckheere-Terpstra Test" on page 1491.

LIST

displays two-way to n-way tables in a list format rather than as crosstabulation tables. PROC FREQ ignores the LIST option when you request statistical tests or measures of association.

MAXITER=*number*

specifies the maximum number of iterations for computing the polychoric correlation when you specify the PLCORR option. The value of the MAXITER= option must be a positive integer; by default, MAXITER=20. Iterative computation of the polychoric correlation stops when the number of iterations exceeds the value of the MAXITER= option, or when the convergence measure falls below the value of the CONVERGE= option, whichever happens first. For more information see the section "Polychoric Correlation" on page 1482.

MEASURES

requests several measures of association and their asymptotic standard errors (ASE). The measures include gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C|R)$, Somers' $D(R|C)$, the Pearson and Spearman correlation coefficients, lambda (symmetric and asymmetric), uncertainty coefficients (symmetric and asymmetric). To request confidence limits for these measures of association, you can specify the CL option.

For 2 × 2 tables, the MEASURES option also provides the odds ratio, column 1 relative risk, column 2 relative risk, and the corresponding confidence limits. Alternatively, you can obtain the odds ratio and relative risks, without the other measures of association, by specifying the RELRISK option.

For more information, see the section "Measures of Association" on page 1474.

MISSING

treats missing values as nonmissing and includes them in calculations of percentages and other statistics.

For more information, see the section "Missing Values" on page 1466.

MISSPRINT

displays missing value frequencies for all tables, even though PROC FREQ does not use the frequencies in the calculation of statistics. For more information, see the section "Missing Values" on page 1466.

NOCOL

suppresses the display of column percentages in cells of the crosstabulation table.

NOCUM

suppresses the display of cumulative frequencies and cumulative percentages for one-way frequency tables and for crosstabulation tables in list format.

NOFREQ

suppresses the display of cell frequencies for crosstabulation tables. This also suppresses frequencies for row totals.

NOPERCENT

suppresses the display of cell percentages, row total percentages, and column total percentages for crosstabulation tables. For one-way frequency tables and crosstabulation tables in list format, the NOPERCENT option suppresses the display of percentages and cumulative percentages.

NOPRINT

suppresses the display of frequency and crosstabulation tables but displays all requested tests and statistics. Use the NOPRINT option in the PROC FREQ statement to suppress the display of all tables.

NOROW

suppresses the display of row percentages in cells of the crosstabulation table.

NOSPARSE

requests that PROC FREQ not invoke the SPARSE option when you specify the ZEROS option in the WEIGHT statement. The NOSPARSE option suppresses the display of cells with a zero frequency count in the list output, and it also omits them from the OUT= data set. By default, the ZEROS option invokes the SPARSE option, which displays table cells with a zero frequency count in the LIST output and includes them in the OUT= data set. For more information, see the description of the ZEROS option.

For CROSSLIST tables, the NOSPARSE option suppresses display of variable levels with zero frequency. By default for CROSSLIST tables, PROC FREQ displays all levels of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. And for multiway tables displayed with the CROSSLIST option, the procedure displays all levels of the row variable for each stratum of the table by default, including any row variable levels with zero frequency for the stratum.

NOWARN

suppresses the log warning message that the asymptotic chi-square test may not be valid. By default, PROC FREQ displays this log message when more than 20 percent of the table cells have expected frequencies less than five.

OUT=*SAS-data-set*

names the output data set that contains variable values and frequency counts. The variable COUNT contains the frequencies and the variable PERCENT contains the percentages. If more than one table request appears in the TABLES statement, the contents of the data set correspond to the last table request in the TABLES statement. For more information, see the section "Output Data Sets" on page 1514 and see the following descriptions for the options OUTCUM, OUTEXPECT, and OUTPCT.

OUTCUM

includes the cumulative frequency and the cumulative percentage for one-way tables in the output data set when you specify the OUT= option in the TABLES statement. The variable CUM_FREQ contains the cumulative frequency for each level of the analysis variable, and the variable CUM_PCT contains the cumulative percentage for each level. The OUTCUM option has no effect for two-way or multiway tables.

For more information, see the section "Output Data Sets" on page 1514.

OUTEXPECT

includes the expected frequency in the output data set for crosstabulation tables when you specify the OUT= option in the TABLES statement. The variable EXPECTED contains the expected frequency for each table cell. The EXPECTED option is valid for two-way or multiway tables, and has no effect for one-way tables.

For more information, see the section "Output Data Sets" on page 1514.

OUTPCT

includes the following additional variables in the output data set when you specify the OUT= option in the TABLES statement for crosstabulation tables:

PCT_COL	the percentage of column frequency
PCT_ROW	the percentage of row frequency
PCT_TABL	the percentage of stratum frequency, for n-way tables where $n > 2$

The OUTPCT option is valid for two-way or multiway tables, and has no effect for one-way tables.

For more information, see the section "Output Data Sets" on page 1514.

PLCORR

requests the polychoric correlation coefficient. For 2×2 tables, this statistic is more commonly known as the tetrachoric correlation coefficient, and it is labeled as such in the displayed output. If you omit the MEASURES option, the PLCORR option invokes MEASURES. For more information, see the section "Polychoric Correlation" on page 1482 and the descriptions for the CONVERGE= and MAXITER= options in this list.

PRINTKWT

displays the weights PROC FREQ uses to compute the weighted kappa coefficient. You must also specify the AGREE option, which requests the weighted kappa coefficient. You can specify (WT=FC) with the AGREE option to request Fleiss-Cohen weights. By default, PROC FREQ uses Cicchetti-Allison weights.

See the section "Weighted Kappa Coefficient" on page 1496 for more information.

RELRISK

requests relative risk measures and their confidence limits for 2×2 tables. These measures include the odds ratio and the column 1 and 2 relative risks. For more information, see the section "Odds Ratio and Relative Risks for 2 x 2 Tables" on page 1488. You can also obtain the RELRISK measures by specifying the MEASURES option, which produces other measures of association in addition to the relative risks.

RISKDIFF

requests column 1 and 2 risks (or binomial proportions), risk differences, and their confidence limits for 2×2 tables. See the section "Risks and Risk Differences" on page 1486 for more information.

RISKDIFFC

requests the RISKDIFF option statistics for 2×2 tables, and includes a continuity correction in the asymptotic confidence limits. The RISKDIFF option statistics include the column 1 and 2 risks (or binomial proportions), risk differences, and their confidence limits. See the section "Risks and Risk Differences" on page 1486 for more information.

SCORES=*type*

specifies the type of row and column scores that PROC FREQ uses with the Mantel-Haenszel chi-square, Pearson correlation, Cochran-Armitage test for trend, weighted kappa coefficient, and Cochran-Mantel-Haenszel statistics, where *type* is one of the following (the default is SCORE=TABLE):

- MODRIDIT
- RANK
- RIDIT
- TABLE

By default, the row or column scores are the integers 1,2,... for character variables and the actual variable values for numeric variables. Using other types of scores yields nonparametric analyses. For more information, see the section "Scores" on page 1468.

To display the row and column scores, you can use the SCOROUT option.

SCOROUT

displays the row and the column scores. You specify the score type with the SCORES= option. PROC FREQ uses the scores when it calculates the Mantel-Haenszel chi-square, Pearson correlation, Cochran-Armitage test for trend, weighted kappa coefficient, or Cochran-Mantel-Haenszel statistics. The SCOROUT option displays the row and column scores only when statistics are computed for two-way tables. To store the scores in an output data set, use the Output Delivery System.

For more information, see the section "Scores" on page 1468.

SPARSE

lists all possible combinations of the variable values for an *n*-way table when $n > 1$, even if a combination does not occur in the data. The SPARSE option applies only to crosstabulation tables displayed in list format and to the OUT= output data set. Otherwise, if you do not use the LIST option or the OUT= option, the SPARSE option has no effect.

When you specify the SPARSE and LIST options, PROC FREQ displays all combinations of variable variables in the table listing, including those values with a frequency count of zero. By default, without the SPARSE option, PROC FREQ does not display zero-frequency values in list output. When you use the SPARSE and OUT= options, PROC FREQ includes empty crosstabulation table cells in the output data set. By default, PROC FREQ does not include zero-frequency table cells in the output data set.

For more information, see the section "Missing Values" on page 1466.

TESTF=(*values*)

specifies the null hypothesis frequencies for a one-way chi-square test for specified frequencies. You can separate *values* with blanks or commas. The sum of the frequency values must equal the total frequency for the one-way table. The number of TESTF= values must equal the number of variable levels in the one-way table. List these values in the order in which the corresponding variable levels appear in the output. If you omit the CHISQ option, the TESTF= option invokes CHISQ.

For more information, see the section "Chi-Square Test for One-Way Tables" on page 1470.

TESTP=(*values*)

specifies the null hypothesis proportions for a one-way chi-square test for specified proportions. You can separate *values* with blanks or commas. Specify *values* in probability form as numbers between 0 and 1, where the proportions sum to 1. Or specify *values* in percentage form as numbers between 0 and 100, where the percentages sum to 100. The number of TESTP= values must equal the number of variable levels in the one-way table. List these values in the order in which the corresponding variable levels appear in the output. If you omit the CHISQ option, the TESTP= option invokes CHISQ.

For more information, see the section "Chi-Square Test for One-Way Tables" on page 1470.

TOTPCT

displays the percentage of total frequency in crosstabulation tables, for n-way tables where $n > 2$. This percentage is also available with the LIST option or as the PERCENT variable in the OUT= output data set.

TREND

performs the Cochran-Armitage test for trend. The table must be $2 \times C$ or $R \times 2$. For more information, see the section "Cochran-Armitage Test for Trend" on page 1490.

TEST Statement

TEST *options* **;**

The TEST statement requests asymptotic tests for the specified measures of association and measures of agreement. You must use a TABLES statement with the TEST statement.

options

specify the statistics for which to provide asymptotic tests. The available statistics are those measures of association and agreement listed in Table 29.10 . The option names are identical to those in the TABLES statement and the OUTPUT statement. You can request all available tests for groups of statistics by using group options MEASURES or AGREE. Or you can request tests individually by using one of the options shown in Table 29.10 .

For each measure of association or agreement that you specify, the TEST statement provides an asymptotic test that the measure equals zero. When you request an asymptotic test, PROC FREQ gives the asymptotic standard error under the null hypothesis, the test statistic, and the *p*-values. Additionally, PROC FREQ reports the confidence limits for that measure. The ALPHA= option in the TABLES statement determines the confidence level, which, by default, equals 0.05 and provides 95% confidence limits. For more information, see the sections "Asymptotic Tests" on page 1475 and "Confidence Limits" on page 1475, and see "Statistical Computations" beginning on page 1468 for sections describing the individual measures.

In addition to these asymptotic tests, exact tests for selected measures of association and agreement are available with the EXACT statement. See the section "EXACT Statement" on page 1443 for more information.

Table 29.10. TEST Statement Options and Required TABLES Statement Options

Option	**Asymptotic Tests Computed**	**Required TABLES Statement Option**
AGREE	simple kappa coefficient and weighted kappa coefficient	AGREE
GAMMA	gamma	ALL or MEASURES
KAPPA	simple kappa coefficient	AGREE
KENTB	Kendall's tau-b	ALL or MEASURES
MEASURES	gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C\|R)$, Somers' $D(R\|C)$, the Pearson correlation, and the Spearman correlation	ALL or MEASURES
PCORR	Pearson correlation coefficient	ALL or MEASURES
SCORR	Spearman correlation coefficient	ALL or MEASURES
SMDCR	Somers' $D(C\|R)$	ALL or MEASURES
SMDRC	Somers' $D(R\|C)$	ALL or MEASURES
STUTC	Stuart's tau-c	ALL or MEASURES
WTKAP	weighted kappa coefficient	AGREE

WEIGHT Statement

WEIGHT *variable* < / *option* > **;**

The WEIGHT statement specifies a numeric *variable* with a value that represents the frequency of the observation. The WEIGHT statement is most commonly used to input cell count data. See the "Inputting Frequency Counts" section on page 1464 for more information. If you use the WEIGHT statement, PROC FREQ assumes that an observation represents *n* observations, where *n* is the value of *variable*. The value of the weight variable need not be an integer. When a weight value is missing, PROC FREQ ignores the corresponding observation. When a weight value is zero, PROC FREQ ignores the corresponding observation unless you specify the ZEROS option, which includes observations with zero weights. If a WEIGHT statement does not appear, each observation has a default weight of 1. The sum of the weight variable values represents the total number of observations.

If any value of the weight variable is negative, PROC FREQ displays the frequencies (as measured by the weighted values) but does not compute percentages and other statistics. If you create an output data set using the OUT= option in the TABLES statement, PROC FREQ creates the PERCENT variable and assigns a missing value for each observation. PROC FREQ also assigns missing values to the variables that the OUTEXPECT and OUTPCT options create. You cannot create an output data set using the OUTPUT statement since statistics are not computed when there are negative weights.

Option

ZEROS

includes observations with zero weight values. By default, PROC FREQ ignores observations with zero weights.

If you specify the ZEROS option, frequency and and crosstabulation tables display any levels corresponding to observations with zero weights. Without the ZEROS option, PROC FREQ does not process observations with zero weights, and so does not display levels that contain only observations with zero weights.

With the ZEROS option, PROC FREQ includes levels with zero weights in the chi-square goodness-of-fit test for one-way tables. Also, PROC FREQ includes any levels with zero weights in binomial computations for one-way tables. This enables computation of binomial estimates and tests when there are no observations with positive weights in the specified level.

For two-way tables, the ZEROS option enables computation of kappa statistics when there are levels containing no observations with positive weight. For more information, see the section "Tables with Zero Rows and Columns" on page 1499.

Note that even with the ZEROS option, PROC FREQ does not compute the CHISQ or MEASURES statistics for two-way tables when the table has a zero row or zero column, because most of these statistics are undefined in this case.

The ZEROS option invokes the SPARSE option in the TABLES statement, which includes table cells with a zero frequency count in the list output and the OUT= data set. By default, without the SPARSE option, PROC FREQ does not include zero frequency cells in the list output or in the OUT= data set. If you specify the ZEROS option in the WEIGHT statement but do not want the SPARSE option, you can specify the NOSPARSE option in the TABLES statement.

Details

Inputting Frequency Counts

PROC FREQ can use either raw data or cell count data to produce frequency and crosstabulation tables. *Raw data*, also known as case-record data, report the data as one record for each subject or sample member. *Cell count data* report the data as a table, listing all possible combinations of data values along with the frequency counts. This way of presenting data often appears in published results.

The following DATA step statements store raw data in a SAS data set:

```
data Raw;
   input Subject $ R C @@;
   datalines;
01 1 1  02 1 1  03 1 1  04 1 1  05 1 1
06 1 2  07 1 2  08 1 2  09 2 1  10 2 1
11 2 1  12 2 1  13 2 2  14 2 2  14 2 2
;
```

You can store the same data as cell counts using the following DATA step statements:

```
data CellCounts;
   input R C Count @@;
   datalines;
1 1 5   1 2 3
2 1 4   2 2 3
;
```

The variable R contains the values for the rows, and the variable C contains the values for the columns. The Count variable contains the cell count for each row and column combination.

Both the Raw data set and the CellCounts data set produce identical frequency counts, two-way tables, and statistics. With the CellCounts data set, you must use a WEIGHT statement to specify that the Count variable contains cell counts. For example, to create a two-way crosstabulation table, submit the following statements:

```
proc freq data=CellCounts;
   weight Count;
   tables R*C;
run;
```

Grouping with Formats

PROC FREQ groups a variable's values according to its formatted values. If you assign a format to a variable with a FORMAT statement, PROC FREQ formats the variable values before dividing observations into the levels of a frequency or crosstabulation table.

For example, suppose that a variable X has the values 1.1, 1.4, 1.7, 2.1, and 2.3. Each of these values appears as a level in the frequency table. If you decide to round each value to a single digit, include the following statement in the PROC FREQ step:

```
format X 1.;
```

Now the table lists the frequency count for formatted level 1 as two and formatted level 2 as three.

PROC FREQ treats formatted character variables in the same way. The formatted values are used to group the observations into the levels of a frequency table or crosstabulation table. PROC FREQ uses the entire value of a character format to classify an observation.

You can also use the FORMAT statement to assign formats that were created with the FORMAT procedure to the variables. User-written formats determine the number of levels for a variable and provide labels for a table. If you use the same data with different formats, then you can produce frequency counts and statistics for different classifications of the variable values.

When you use PROC FORMAT to create a user-written format that combines missing and nonmissing values into one category, PROC FREQ treats the entire category of formatted values as missing. For example, a questionnaire codes 1 as yes, 2 as no, and 8 as a no answer. The following PROC FORMAT step creates a user-written format:

```
proc format;
   value Questfmt 1  ='Yes'
                  2  ='No'
                  8,.='Missing';
run;
```

When you use a FORMAT statement to assign Questfmt. to a variable, the variable's frequency table no longer includes a frequency count for the response of 8. You must use the MISSING or MISSPRINT option in the TABLES statement to list the frequency for no answer. The frequency count for this level includes observations with either a value of 8 or a missing value (.).

The frequency or crosstabulation table lists the values of both character and numeric variables in ascending order based on internal (unformatted) variable values unless you change the order with the ORDER= option. To list the values in ascending order by formatted values, use ORDER=FORMATTED in the PROC FREQ statement.

For more information on the FORMAT statement, refer to *SAS Language Reference: Concepts*.

Missing Values

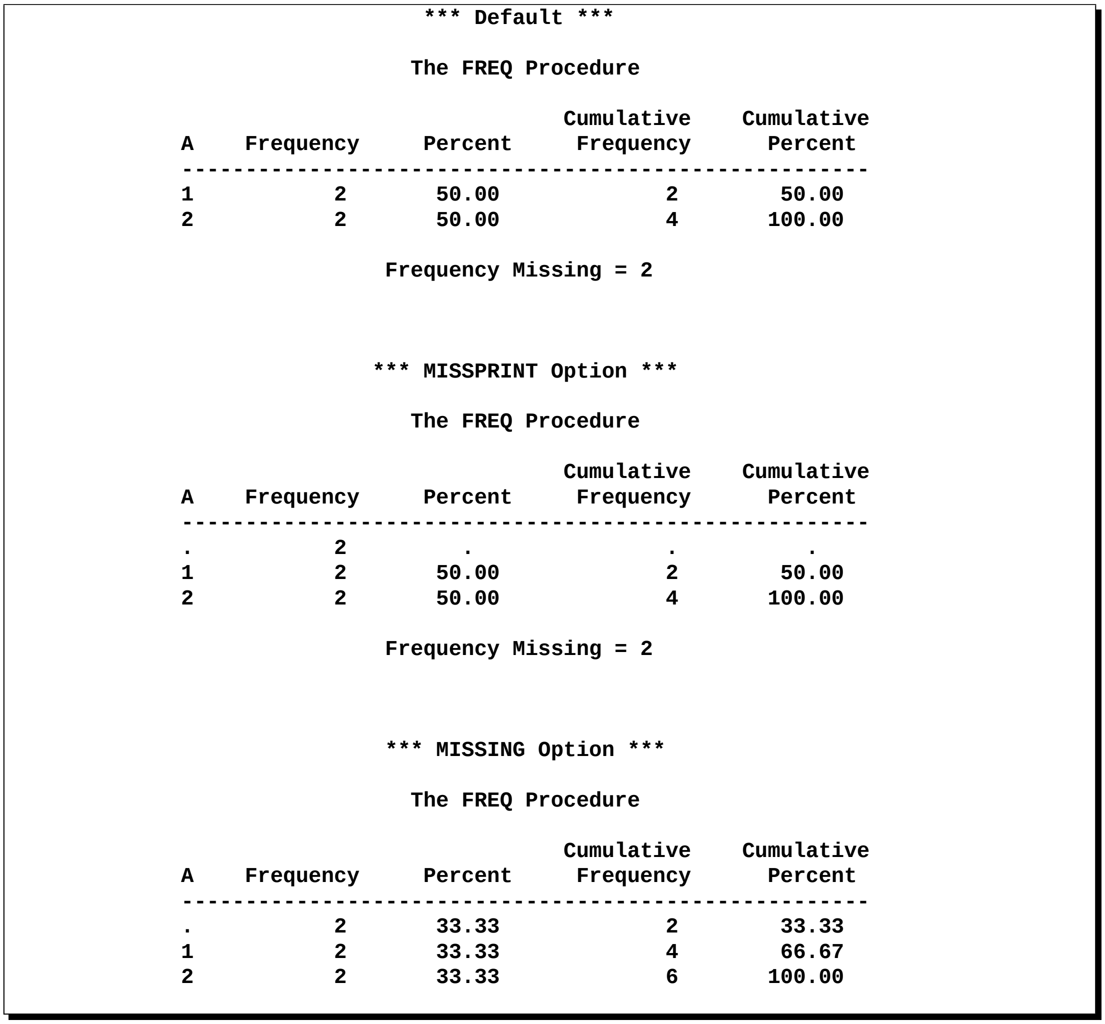
By default, PROC FREQ excludes missing values before it constructs the frequency and crosstabulation tables. PROC FREQ also excludes missing values before computing statistics. However, the total frequency of observations with missing values is displayed below each table. The following options change the way in which PROC FREQ handles missing values:

MISSPRINT includes missing value frequencies in frequency or crosstabulation tables.

MISSING includes missing values in percentage and statistical calculations.

The OUT= option in the TABLES statement includes an observation in the output data set that contains the frequency of missing values. The NMISS option in the OUTPUT statement creates a variable in the output data set that contains the number of missing values.

Figure 29.7 shows three ways in which PROC FREQ handles missing values. The first table uses the default method; the second table uses the MISSPRINT option; and the third table uses the MISSING option.



```
                        *** Default ***

                       The FREQ Procedure

                                        Cumulative    Cumulative
     A    Frequency     Percent     Frequency      Percent
     -------------------------------------------------------
     1           2       50.00             2         50.00
     2           2       50.00             4        100.00

                     Frequency Missing = 2


                    *** MISSPRINT Option ***

                       The FREQ Procedure

                                        Cumulative    Cumulative
     A    Frequency     Percent     Frequency      Percent
     -------------------------------------------------------
     .           2         .               .            .
     1           2       50.00             2         50.00
     2           2       50.00             4        100.00

                     Frequency Missing = 2


                     *** MISSING Option ***

                       The FREQ Procedure

                                        Cumulative    Cumulative
     A    Frequency     Percent     Frequency      Percent
     -------------------------------------------------------
     .           2       33.33             2         33.33
     1           2       33.33             4         66.67
     2           2       33.33             6        100.00
```

Figure 29.7. Missing Values in Frequency Tables

When a combination of variable values for a crosstabulation is missing, PROC FREQ assigns zero to the frequency count for the table cell. By default, PROC FREQ omits missing combinations in list format and in the output data set that is created in a TABLES statement. To include the missing combinations, use the SPARSE option with the LIST or OUT= option in the TABLES statement.

PROC FREQ treats missing BY variable values like any other BY variable value. The missing values form a separate BY group. When the value of a WEIGHT variable is missing, PROC FREQ excludes the observation from the analysis.

Statistical Computations

Definitions and Notation

In this chapter, a two-way table represents the crosstabulation of variables X and Y. Let the rows of the table be labeled by the values X_i, $i = 1, 2, \ldots, R$, and the columns by Y_j, $j = 1, 2, \ldots, C$. Let n_{ij} denote the cell frequency in the ith row and the jth column and define the following:

$$\begin{aligned}
n_{i\cdot} &= \sum_j n_{ij} \quad \text{(row totals)} \\
n_{\cdot j} &= \sum_i n_{ij} \quad \text{(column totals)} \\
n &= \sum_i \sum_j n_{ij} \quad \text{(overall total)} \\
p_{ij} &= n_{ij}/n \quad \text{(cell percentages)} \\
p_{i\cdot} &= n_{i\cdot}/n \quad \text{(row percentages)} \\
p_{\cdot j} &= n_{\cdot j}/n \quad \text{(column percentages)} \\
R_i &= \text{score for row } i \\
C_j &= \text{score for column } j \\
\bar{R} &= \sum_i n_{i\cdot} R_i / n \quad \text{(average row score)} \\
\bar{C} &= \sum_j n_{\cdot j} C_j / n \quad \text{(average column score)} \\
A_{ij} &= \sum_{k>i} \sum_{l>j} n_{kl} + \sum_{k<i} \sum_{l<j} n_{kl} \\
D_{ij} &= \sum_{k>i} \sum_{l<j} n_{kl} + \sum_{k<i} \sum_{l>j} n_{kl} \\
P &= \sum_i \sum_j n_{ij} A_{ij} \quad \text{(twice the number of concordances)} \\
Q &= \sum_i \sum_j n_{ij} D_{ij} \quad \text{(twice the number of discordances)}
\end{aligned}$$

Scores

PROC FREQ uses scores for the variable values when computing the Mantel-Haenszel chi-square, Pearson correlation, Cochran-Armitage test for trend, weighted kappa coefficient, and Cochran-Mantel-Haenszel statistics. The SCORES= option in the TABLES statement specifies the score type that PROC FREQ uses. The available score types are TABLE, RANK, RIDIT, and MODRIDIT scores. The default score type is TABLE.

For numeric variables, table scores are the values of the row and column levels. If the row or column variables are formatted, then the table score is the internal numeric value corresponding to that level. If two or more numeric values are classified into the same formatted level, then the internal numeric value for that level is the smallest of these values. For character variables, table scores are defined as the row numbers and column numbers (that is, 1 for the first row, 2 for the second row, and so on).

Rank scores, which you can use to obtain nonparametric analyses, are defined by

$$\begin{aligned}
\text{Row scores:} \quad R1_i &= \sum_{k<i} n_{k\cdot} + (n_{i\cdot} + 1)/2 \qquad i = 1, 2, \ldots, R \\
\text{Column scores:} \quad C1_j &= \sum_{l<j} n_{\cdot l} + (n_{\cdot j} + 1)/2 \qquad j = 1, 2, \ldots, C
\end{aligned}$$

Note that rank scores yield midranks for tied values.

Ridit scores (Bross 1958; Mack and Skillings 1980) also yield nonparametric analyses, but they are standardized by the sample size. Ridit scores are derived from rank scores as

$$\begin{aligned}
R2_i &= R1_i/n \\
C2_j &= C1_j/n
\end{aligned}$$

Modified ridit (MODRIDIT) scores (van Elteren 1960; Lehmann 1975), which also yield nonparametric analyses, represent the expected values of the order statistics for the uniform distribution on (0,1). Modified ridit scores are derived from rank scores as

$$\begin{aligned}
R3_i &= R1_i/(n+1) \\
C3_j &= C1_j/(n+1)
\end{aligned}$$

Chi-Square Tests and Statistics

When you specify the CHISQ option in the TABLES statement, PROC FREQ performs the following chi-square tests for each two-way table: Pearson chi-square, continuity-adjusted chi-square for 2 × 2 tables, likelihood-ratio chi-square, Mantel-Haenszel chi-square, and Fisher's exact test for 2 × 2 tables. Also, PROC FREQ computes the following statistics derived from the Pearson chi-square: the phi coefficient, the contingency coefficient, and Cramer's V. PROC FREQ computes Fisher's exact test for general $R \times C$ tables when you specify the FISHER (or EXACT) option in the TABLES statement, or, equivalently, when you specify the FISHER option in the EXACT statement.

For one-way frequency tables, PROC FREQ performs a chi-square goodness-of-fit test when you specify the CHISQ option. The other chi-square tests and statistics described in this section are defined only for two-way tables and so are not computed for one-way frequency tables.

All the two-way test statistics described in this section test the null hypothesis of no association between the row variable and the column variable. When the sample size n is large, these test statistics are distributed approximately as chi-square when the null hypothesis is true. When the sample size is not large, exact tests may be useful. PROC FREQ computes exact tests for the following chi-square statistics when you specify the corresponding option in the EXACT statement: Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square. See the section "Exact Statistics" beginning on page 1508 for more information.

Note that the Mantel-Haenszel chi-square statistic is appropriate only when both variables lie on an ordinal scale. The other chi-square tests and statistics in this section are appropriate for either nominal or ordinal variables. The following sections give the formulas that PROC FREQ uses to compute the chi-square tests and statistics. For further information on the formulas and on the applicability of each statistic, refer to Agresti (1996), Stokes, Davis, and Koch (1995), and the other references cited for each statistic.

Chi-Square Test for One-Way Tables

For one-way frequency tables, the CHISQ option in the TABLES statement computes a chi-square goodness-of-fit test. Let C denote the number of classes, or levels, in the one-way table. Let f_i denote the frequency of class i (or the number of observations in class i) for $i = 1, 2, ..., C$. Then PROC FREQ computes the chi-square statistic as

$$Q_P \quad = \quad \sum_{i=1}^{C} \frac{(f_i - e_i)^2}{e_i}$$

where e_i is the expected frequency for class i under the null hypothesis.

In the test for equal proportions, which is the default for the CHISQ option, the null hypothesis specifies equal proportions of the total sample size for each class. Under this null hypothesis, the expected frequency for each class equals the total sample size divided by the number of classes,

$$e_i \quad = \quad n \;/\; C \qquad \text{for } i = 1, 2, \ldots, C$$

In the test for specified frequencies, which PROC FREQ computes when you input null hypothesis frequencies using the TESTF= option, the expected frequencies are those TESTF= values. In the test for specified proportions, which PROC FREQ computes when you input null hypothesis proportions using the TESTP= option, the expected frequencies are determined from the TESTP= proportions p_i, as

$$e_i \quad = \quad p_i \;\times\; n \qquad \text{for } i = 1, 2, \ldots, C$$

Under the null hypothesis (of equal proportions, specified frequencies, or specified proportions), this test statistic has an asymptotic chi-square distribution, with $C - 1$ degrees of freedom. In addition to the asymptotic test, PROC FREQ computes the exact one-way chi-square test when you specify the CHISQ option in the EXACT statement.

Chi-Square Test for Two-Way Tables

The Pearson chi-square statistic for two-way tables involves the differences between the observed and expected frequencies, where the expected frequencies are computed under the null hypothesis of independence. The chi-square statistic is computed as

$$Q_P = \sum_i \sum_j \frac{(n_{ij} - e_{ij})^2}{e_{ij}}$$

where

$$e_{ij} = \frac{n_{i\cdot} \, n_{\cdot j}}{n}$$

When the row and column variables are independent, Q_P has an asymptotic chi-square distribution with $(R-1)(C-1)$ degrees of freedom. For large values of Q_P, this test rejects the null hypothesis in favor of the alternative hypothesis of general association. In addition to the asymptotic test, PROC FREQ computes the exact chi-square test when you specify the PCHI or CHISQ option in the EXACT statement.

For a 2×2 table, the Pearson chi-square is also appropriate for testing the equality of two binomial proportions or, for $R \times 2$ and $2 \times C$ tables, the homogeneity of proportions. Refer to Fienberg (1980).

Likelihood-Ratio Chi-Square Test

The likelihood-ratio chi-square statistic involves the ratios between the observed and expected frequencies. The statistic is computed as

$$G^2 = 2 \sum_i \sum_j n_{ij} \, \ln \left(\frac{n_{ij}}{e_{ij}} \right)$$

When the row and column variables are independent, G^2 has an asymptotic chi-square distribution with $(R - 1)(C - 1)$ degrees of freedom. In addition to the asymptotic test, PROC FREQ computes the exact test when you specify the LRCHI or CHISQ option in the EXACT statement.

Continuity-Adjusted Chi-Square Test

The continuity-adjusted chi-square statistic for 2×2 tables is similar to the Pearson chi-square, except that it is adjusted for the continuity of the chi-square distribution. The continuity-adjusted chi-square is most useful for small sample sizes. The use of the continuity adjustment is controversial; this chi-square test is more conservative, and more like Fisher's exact test, when your sample size is small. As the sample size increases, the statistic becomes more and more like the Pearson chi-square.

The statistic is computed as

$$Q_C = \sum_i \sum_j \frac{[\,\max(0, |n_{ij} - e_{ij}| - 0.5)\,]^2}{e_{ij}}$$

Under the null hypothesis of independence, Q_C has an asymptotic chi-square distribution with $(R-1)(C-1)$ degrees of freedom.

Mantel-Haenszel Chi-Square Test

The Mantel-Haenszel chi-square statistic tests the alternative hypothesis that there is a linear association between the row variable and the column variable. Both variables must lie on an ordinal scale. The statistic is computed as

$$Q_{MH} = (n-1)r^2$$

where r^2 is the Pearson correlation between the row variable and the column variable. For a description of the Pearson correlation, see the "Pearson Correlation Coefficient" section on page 1479. The Pearson correlation and, thus, the Mantel-Haenszel chi-square statistic use the scores that you specify in the SCORES= option in the TABLES statement.

Under the null hypothesis of no association, Q_{MH} has an asymptotic chi-square distribution with one degree of freedom. In addition to the asymptotic test, PROC FREQ computes the exact test when you specify the MHCHI or CHISQ option in the EXACT statement.

Refer to Mantel and Haenszel (1959) and Landis, Heyman, and Koch (1978).

Fisher's Exact Test

Fisher's exact test is another test of association between the row and column variables. This test assumes that the row and column totals are fixed, and then uses the hypergeometric distribution to compute probabilities of possible tables with these observed row and column totals. Fisher's exact test does not depend on any large-sample distribution assumptions, and so it is appropriate even for small sample sizes and for sparse tables.

2 × 2 Tables

For 2×2 tables, PROC FREQ gives the following information for Fisher's exact test: table probability, two-sided p-value, left-sided p-value, and right-sided p-value. The table probability equals the hypergeometric probability of the observed table, and is in fact the value of the test statistic for Fisher's exact test.

Where p is the hypergeometric probability of a specific table with the observed row and column totals, Fisher's exact p-values are computed by summing probabilities p over defined sets of tables,

$$\mathit{PROB} = \sum_A p$$

The two-sided p-value is the sum of all possible table probabilties (for tables having the observed row and column totals) that are less than or equal to the observed table probability. So, for the two-sided p-value, the set A includes all possible tables with hypergeometric probabilities less than or equal to the probability of the observed table. A small two-sided p-value supports the alternative hypothesis of association between the row and column variables.

One-sided tests are defined in terms of the frequency of the cell in the first row and first column of the table, the (1,1) cell. Denoting the observed (1,1) cell frequency by F, the left-sided p-value for Fisher's exact test is probability that the (1,1) cell frequency is less than or equal to F. So, for the left-sided p-value, the set A includes those tables with a (1,1) cell frequency less than or equal to F. A small left-sided p-value supports the alternative hypothesis that the probability of an observation being in the first cell is less than expected under the null hypothesis of independent row and column variables.

Similarly, for a right-sided alternative hypothesis, A is the set of tables where the frequency of the (1,1) cell is greater than or equal to that in the observed table. A small right-sided p-value supports the alternative that the probability of the first cell is greater than that expected under the null hypothesis.

Because the (1,1) cell frequency completely determines the 2×2 table when the marginal row and column sums are fixed, these one-sided alternatives can be equivalently stated in terms of other cell probabilities or ratios of cell probabilities. The left-sided alternative is equivalent to an odds ratio greater than 1, where the odds ratio equals ($n_{11}\, n_{22} \; / \; n_{12}\, n_{21}$). Additionally, the left-sided alternative is equivalent to the column 1 risk for row 1 being less than the column 1 risk for row 2, $p_{1|1} < p_{1|2}$. Similarly, the right-sided alternative is equivalent to the column 1 risk for row 1 being greater than the column 1 risk for row 2, $p_{1|1} > p_{1|2}$. Refer to Agresti (1996).

R × *C* Tables

Fisher's exact test was extended to general $R \times C$ tables by Freeman and Halton (1951), and this test is also known as the Freeman-Halton test. For $R \times C$ tables, the two-sided p-value is defined the same as it is for 2×2 tables. The set A contains all tables with p less than or equal to the probability of the observed table. A small p-value supports the alternative hypothesis of association between the row and column variables. For $R{\times}C$ tables, Fisher's exact test is inherently two-sided. The alternative hypothesis is defined only in terms of general, and not linear, association. Therefore, PROC FREQ does not provide right-sided or left-sided p-values for general $R \times C$ tables.

For $R \times C$ tables, PROC FREQ computes Fisher's exact test using the network algorithm of Mehta and Patel (1983), which provides a faster and more efficient solution than direct enumeration. See the section "Exact Statistics" beginning on page 1508 for more details.

Phi Coefficient

The phi coefficient is a measure of association derived from the Pearson chi-square statistic. It has the range $-1 \leq \phi \leq 1$ for 2×2 tables. Otherwise, the range is $0 \leq \phi \leq min(\sqrt{R-1}, \sqrt{C-1})$ (Liebetrau 1983). The phi coefficient is computed as

$$\phi = \frac{n_{11}\, n_{22} - n_{12}\, n_{21}}{\sqrt{n_{1\cdot}\, n_{2\cdot}\, n_{\cdot 1}\, n_{\cdot 2}}} \qquad \text{for } 2 \times 2 \text{ tables}$$

$$\phi = \sqrt{Q_P / n} \qquad \text{otherwise}$$

Refer to Fleiss (1981, pp. 59–60).

Contingency Coefficient

The contingency coefficient is a measure of association derived from the Pearson chi-square. It has the range $0 \leq P \leq \sqrt{(m-1)/m}$, where $m = min(R, C)$ (Liebetrau 1983). The contingency coefficient is computed as

$$P = \sqrt{\frac{Q_P}{Q_P + n}}$$

Refer to Kendall and Stuart (1979, pp. 587–588).

Cramer's *V*

Cramer's V is a measure of association derived from the Pearson chi-square. It is designed so that the attainable upper bound is always 1. It has the range $-1 \leq V \leq 1$ for 2×2 tables; otherwise, the range is $0 \leq V \leq 1$. Cramer's V is computed as

$$V = \phi \qquad \text{for } 2 \times 2 \text{ tables}$$

$$V = \sqrt{\frac{Q_P / n}{\min(R-1, C-1)}} \qquad \text{otherwise}$$

Refer to Kendall and Stuart (1979, p. 588).

Measures of Association

When you specify the MEASURES option in the TABLES statement, PROC FREQ computes several statistics that describe the association between the two variables of the contingency table. The following are measures of ordinal association that consider whether the variable Y tends to increase as X increases: gamma, Kendall's tau-b, Stuart's tau-c, and Somers' D. These measures are appropriate for ordinal variables, and they classify pairs of observations as *concordant* or *discordant*. A pair is concordant if the observation with the larger value of X also has the larger value of Y. A pair is discordant if the observation with the larger value of X has the smaller value of Y. Refer to Agresti (1996) and the other references cited in the discussion of each measure of association.

The Pearson correlation coefficient and the Spearman rank correlation coefficient are also appropriate for ordinal variables. The Pearson correlation describes the strength of the linear association between the row and column variables, and it is computed using the row and column scores specified by the SCORES= option in the TABLES statement. The Spearman correlation is computed with rank scores. The polychoric correlation (requested by the PLCORR option) also requires ordinal variables and assumes that the variables have an underlying bivariate normal distribution. The following measures of association do not require ordinal variables, but they are appropriate for nominal variables: lambda asymmetric, lambda symmetric, and uncertainty coefficients.

PROC FREQ computes estimates of the measures according to the formulas given in the discussion of each measure of association. For each measure, PROC FREQ computes an asymptotic standard error (ASE), which is the square root of the asymptotic variance denoted by var in the following sections.

Confidence Limits

If you specify the CL option in the TABLES statement, PROC FREQ computes asymptotic confidence limits for all MEASURES statistics. The confidence coefficient is determined according to the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

The confidence limits are computed as

$$est \ \pm \ (\, z_{\alpha/2} \ \times \ \mathrm{ASE} \,)$$

where est is the estimate of the measure, $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution, and ASE is the asymptotic standard error of the estimate.

Asymptotic Tests

For each measure that you specify in the TEST statement, PROC FREQ computes an asymptotic test of the null hypothesis that the measure equals zero. Asymptotic tests are available for the following measures of association: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(R|C)$, Somers' $D(C|R)$, the Pearson correlation coefficient, and the Spearman rank correlation coefficient. To compute an asymptotic test, PROC FREQ uses a standardized test statistic z, which has an asymptotic standard normal distribution under the null hypothesis. The standardized test statistic is computed as

$$z \ = \ \frac{est}{\sqrt{var_0(est)}}$$

where est is the estimate of the measure and $var_0(est)$ is the variance of the estimate under the null hypothesis. Formulas for $var_0(est)$ are given in the discussion of each measure of association.

Note that the ratio of est to $\sqrt{var_0(est)}$ is the same for the following measures: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(R|C)$, and Somers' $D(C|R)$. Therefore, the tests for these measures are identical. For example, the p-values for the test of H_0: gamma $= 0$ equal the p-values for the test of H_0: tau-$b = 0$.

PROC FREQ computes one-sided and two-sided p-values for each of these tests. When the test statistic z is greater than its null hypothesis expected value of zero, PROC FREQ computes the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis that the true value of the measure is greater than zero. When the test statistic is less than or equal to zero, PROC FREQ computes the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. A small left-sided p-value supports the alternative hypothesis that the true value of the measure is less than zero. The one-sided p-value P_1 can be expressed as

$$\begin{aligned} P_1 &= \text{Prob} \; (\; Z > z \;) \qquad \text{if} \;\; z > 0 \\ P_1 &= \text{Prob} \; (\; Z < z \;) \qquad \text{if} \;\; z \leq 0 \end{aligned}$$

where Z has a standard normal distribution. The two-sided p-value P_2 is computed as

$$P_2 = \text{Prob} \; (\; |Z| > |z| \;)$$

Exact Tests

Exact tests are available for two measures of association, the Pearson correlation coefficient and the Spearman rank correlation coefficient. If you specify the PCORR option in the EXACT statement, PROC FREQ computes the exact test of the hypothesis that the Pearson correlation equals zero. If you specify the SCORR option in the EXACT statement, PROC FREQ computes the exact test of the hypothesis that the Spearman correlation equals zero. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

Gamma

The estimator of gamma is based only on the number of concordant and discordant pairs of observations. It ignores tied pairs (that is, pairs of observations that have equal values of X or equal values of Y). Gamma is appropriate only when both variables lie on an ordinal scale. It has the range $-1 \leq \Gamma \leq 1$. If the two variables are independent, then the estimator of gamma tends to be close to zero. Gamma is estimated by

$$G = \frac{P - Q}{P + Q}$$

with asymptotic variance

$$var = \frac{16}{(P + Q)^4} \sum_i \sum_j n_{ij} (QA_{ij} - PD_{ij})^2$$

The variance of the estimator under the null hypothesis that gamma equals zero is computed as

$$var_0(G) = \frac{4}{(P+Q)^2} \left(\sum_i \sum_j n_{ij}(A_{ij} - D_{ij})^2 - (P-Q)^2/n \right)$$

For 2×2 tables, gamma is equivalent to Yule's Q. Refer to Goodman and Kruskal (1979), Agresti (1990), and Brown and Benedetti (1977).

Kendall's Tau-*b*

Kendall's tau-b is similar to gamma except that tau-b uses a correction for ties. Tau-b is appropriate only when both variables lie on an ordinal scale. Tau-b has the range $-1 \le \tau_b \le 1$. It is estimated by

$$t_b = \frac{P-Q}{\sqrt{w_r w_c}}$$

with

$$var = \frac{1}{w^4} \left(\sum_i \sum_j n_{ij}(2wd_{ij} + t_b v_{ij})^2 - n^3 t_b^2 (w_r + w_c)^2 \right)$$

where

$$\begin{aligned} w &= \sqrt{w_r w_c} \\ w_r &= n^2 - \sum_i n_{i\cdot}^2 \\ w_c &= n^2 - \sum_j n_{\cdot j}^2 \\ d_{ij} &= A_{ij} - D_{ij} \\ v_{ij} &= n_{i\cdot} w_c + n_{\cdot j} w_r \end{aligned}$$

The variance of the estimator under the null hypothesis that tau-b equals zero is computed as

$$var_0(t_b) = \frac{4}{w_r w_c} \left(\sum_i \sum_j n_{ij}(A_{ij} - D_{ij})^2 - (P-Q)^2/n \right)$$

Refer to Kendall (1955) and Brown and Benedetti (1977).

Stuart's Tau-*c*

Stuart's tau-c makes an adjustment for table size in addition to a correction for ties. Tau-c is appropriate only when both variables lie on an ordinal scale. Tau-c has the range $-1 \le \tau_c \le 1$. It is estimated by

$$t_c = \frac{m(P-Q)}{n^2(m-1)}$$

with

$$var = \frac{4m^2}{(m-1)^2 n^4} \left(\sum_i \sum_j n_{ij} d_{ij}^2 - (P-Q)^2/n \right)$$

where

$$\begin{aligned} m &= \min(R, C) \\ d_{ij} &= A_{ij} - D_{ij} \end{aligned}$$

The variance of the estimator under the null hypothesis that tau-c equals zero is

$$var_0(t_c) = var$$

Refer to Brown and Benedetti (1977).

Somers' *D* (*C* |*R*) and *D* (*R* |*C*)

Somers' $D(C|R)$ and Somers' $D(R|C)$ are asymmetric modifications of tau-b. $C|R$ denotes that the row variable X is regarded as an independent variable, while the column variable Y is regarded as dependent. Similarly, $R|C$ denotes that the column variable Y is regarded as an independent variable, while the row variable X is regarded as dependent. Somers' D differs from tau-b in that it uses a correction only for pairs that are tied on the independent variable. Somers' D is appropriate only when both variables lie on an ordinal scale. It has the range $-1 \le D \le 1$. Formulas for Somers' $D(R|C)$ are obtained by interchanging the indices.

$$D(C|R) = \frac{P-Q}{w_r}$$

with

$$var = \frac{4}{w_r^4} \sum_i \sum_j n_{ij} \left(w_r d_{ij} - (P-Q)(n - n_{i\cdot}) \right)^2$$

where

$$\begin{aligned} w_r &= n^2 - \sum_i n_{i\cdot}^2 \\ d_{ij} &= A_{ij} - D_{ij} \end{aligned}$$

The variance of the estimator under the null hypothesis that $D(C|R)$ equals zero is computed as

$$var_0(D(C|R)) = \frac{4}{w_r^2}\left(\sum_i \sum_j n_{ij}(A_{ij} - D_{ij})^2 - (P-Q)^2/n\right)$$

Refer to Somers (1962), Goodman and Kruskal (1979), and Liebetrau (1983).

Pearson Correlation Coefficient

PROC FREQ computes the Pearson correlation coefficient using the scores specified in the SCORES= option. The Pearson correlation is appropriate only when both variables lie on an ordinal scale. It has the range $-1 \leq \rho \leq 1$. The Pearson correlation coefficient is computed as

$$r = \frac{v}{w} = \frac{ss_{rc}}{\sqrt{ss_r ss_c}}$$

with

$$var = \frac{1}{w^4} \sum_i \sum_j n_{ij}\left(w(R_i - \bar{R})(C_j - \bar{C}) - \frac{b_{ij} v}{2w}\right)^2$$

The row scores R_i and the column scores C_j are determined by the SCORES= option in the TABLES statement, and

$$\begin{aligned} ss_r &= \sum_i \sum_j n_{ij}(R_i - \bar{R})^2 \\ ss_c &= \sum_i \sum_j n_{ij}(C_j - \bar{C})^2 \\ ss_{rc} &= \sum_i \sum_j n_{ij}(R_i - \bar{R})(C_j - \bar{C}) \\ b_{ij} &= (R_i - \bar{R})^2 ss_c + (C_j - \bar{C})^2 ss_r \\ v &= ss_{rc} \\ w &= \sqrt{ss_r ss_c} \end{aligned}$$

Refer to Snedecor and Cochran (1989) and Brown and Benedetti (1977).

To compute an asymptotic test for the Pearson correlation, PROC FREQ uses a standardized test statistic r^*, which has an asymptotic standard normal distribution under the null hypothesis that the correlation equals zero. The standardized test statistic is computed as

$$r^* = \frac{r}{\sqrt{var_0(r)}}$$

where $var_0(r)$ is the variance of the correlation under the null hypothesis.

$$var_0(r) = \frac{\sum_i \sum_j n_{ij}(R_i - \bar{R})^2 (C_j - \bar{C})^2 - ss_{rc}^2/n}{ss_r ss_c}$$

The asymptotic variance is derived for multinomial sampling in a contingency table framework, and it differs from the form obtained under the assumption that both variables are continuous and normally distributed. Refer to Brown and Benedetti (1977).

PROC FREQ also computes the exact test for the hypothesis that the Pearson correlation equals zero when you specify the PCORR option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

Spearman Rank Correlation Coefficient

The Spearman correlation coefficient is computed using rank scores $R1_i$ and $C1_j$, defined in the section "Scores" beginning on page 1468. It is appropriate only when both variables lie on an ordinal scale. It has the range $-1 \leq \rho_s \leq 1$. The Spearman correlation coefficient is computed as

$$r_s = \frac{v}{w}$$

with

$$var = \frac{1}{n^2 w^4} \sum_i \sum_j n_{ij}(z_{ij} - \bar{z})^2$$

where

$$\begin{aligned} v &= \sum_i \sum_j n_{ij} R(i) C(j) \\ w &= \frac{1}{12}\sqrt{FG} \\ F &= n^3 - \sum_i n_{i\cdot}^3 \end{aligned}$$

$$
\begin{aligned}
G &= n^3 - \sum_j n_{\cdot j}^3 \\
R(i) &= R1_i - n/2 \\
C(j) &= C1_j - n/2 \\
\bar{z} &= \frac{1}{n} \sum_i \sum_j n_{ij} z_{ij} \\
z_{ij} &= w v_{ij} - v w_{ij} \\
v_{ij} &= n \Bigg(R(i)C(j) + \frac{1}{2} \sum_l n_{il} C(l) + \frac{1}{2} \sum_k n_{kj} R(k) + \\
&\qquad \sum_l \sum_{k>i} n_{kl} C(l) + \sum_k \sum_{l>j} n_{kl} R(k) \Bigg) \\
w_{ij} &= \frac{-n}{96w} \left(F n_{\cdot j}^2 + G n_{i \cdot}^2 \right)
\end{aligned}
$$

Refer to Snedecor and Cochran (1989) and Brown and Benedetti (1977).

To compute an asymptotic test for the Spearman correlation, PROC FREQ uses a standardized test statistic r_s^*, which has an asymptotic standard normal distribution under the null hypothesis that the correlation equals zero. The standardized test statistic is computed as

$$r_s^* = \frac{r_s}{\sqrt{var_0(r_s)}}$$

where $var_0(r_s)$ is the variance of the correlation under the null hypothesis.

$$var_0(r_s) = \frac{1}{n^2 w^2} \sum_i \sum_j n_{ij} (v_{ij} - \bar{v})^2$$

where

$$\bar{v} = \sum_i \sum_j n_{ij} v_{ij} / n$$

The asymptotic variance is derived for multinomial sampling in a contingency table framework, and it differs from the form obtained under the assumption that both variables are continuous and normally distributed. Refer to Brown and Benedetti (1977).

PROC FREQ also computes the exact test for the hypothesis that the Spearman rank correlation equals zero when you specify the SCORR option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

Polychoric Correlation

When you specify the PLCORR option in the TABLES statement, PROC FREQ computes the polychoric correlation. This measure of association is based on the assumption that the ordered, categorical variables of the frequency table have an underlying bivariate normal distribution. For 2×2 tables, the polychoric correlation is also known as the tetrachoric correlation. Refer to Drasgow (1986) for an overview of polychoric correlation. The polychoric correlation coefficient is the maximum likelihood estimate of the product-moment correlation between the normal variables, estimating thresholds from the observed table frequencies. The range of the polychoric correlation is from -1 to 1. Olsson (1979) gives the likelihood equations and an asymptotic covariance matrix for the estimates.

To estimate the polychoric correlation, PROC FREQ iteratively solves the likelihood equations by a Newton-Raphson algorithm using the Pearson correlation coefficient as the initial approximation. Iteration stops when the convergence measure falls below the convergence criterion or when the maximum number of iterations is reached, whichever occurs first. The CONVERGE= option sets the convergence criterion, and the default value is 0.0001. The MAXITER= option sets the maximum number of iterations, and the default value is 20.

Lambda Asymmetric

Asymmetric lambda, $\lambda(C|R)$, is interpreted as the probable improvement in predicting the column variable Y given knowledge of the row variable X. Asymmetric lambda has the range $0 \leq \lambda(C|R) \leq 1$. It is computed as

$$\lambda(C|R) = \frac{\sum_i r_i - r}{n - r}$$

with

$$var = \frac{n - \sum_i r_i}{(n-r)^3} \left(\sum_i r_i + r - 2 \sum_i (r_i \mid l_i = l) \right)$$

where

$$\begin{aligned} r_i &= \max_j (n_{ij}) \\ r &= \max_j (n_{\cdot j}) \end{aligned}$$

Also, let l_i be the unique value of j such that $r_i = n_{ij}$, and let l be the unique value of j such that $r = n_{\cdot j}$.

Because of the uniqueness assumptions, ties in the frequencies or in the marginal totals must be broken in an arbitrary but consistent manner. In case of ties, l is defined here as the smallest value of j such that $r = n_{\cdot j}$. For a given i, if there is at least one value j such that $n_{ij} = r_i = c_j$, then l_i is defined here to be the smallest such value of j. Otherwise, if $n_{il} = r_i$, then l_i is defined to be equal to l. If neither condition is true, then l_i is taken to be the smallest value of j such that $n_{ij} = r_i$. The formulas for lambda asymmetric $(R|C)$ can be obtained by interchanging the indices.

Refer to Goodman and Kruskal (1979).

Lambda Symmetric

The nondirectional lambda is the average of the two asymmetric lambdas, $\lambda(C|R)$ and $\lambda(R|C)$. Lambda symmetric has the range $0 \leq \lambda \leq 1$. Lambda symmetric is defined as

$$\lambda = \frac{\sum_i r_i + \sum_j c_j - r - c}{2n - r - c} = \frac{w - v}{w}$$

with

$$var = \frac{1}{w^4}\left(wvy - 2w^2\left[n - \sum_i \sum_j (n_{ij} \mid j = l_i, i = k_j)\right] - 2v^2(n - n_{kl})\right)$$

where

$$\begin{aligned}
c_j &= \max_i(n_{ij}) \\
c &= \max_i(n_{i\cdot}) \\
w &= 2n - r - c \\
v &= 2n - \sum_i r_i - \sum_j c_j \\
x &= \sum_i (r_i \mid l_i = l) \;+\; \sum_j (c_j \mid k_j = k) \;+\; r_k \;+\; c_l \\
y &= 8n - w - v - 2x
\end{aligned}$$

Refer to Goodman and Kruskal (1979).

Uncertainty Coefficients (*C* |*R*) and (*R* |*C*)

The uncertainty coefficient, $U(C|R)$, is the proportion of uncertainty (entropy) in the column variable Y that is explained by the row variable X. It has the range $0 \leq U(C|R) \leq 1$. The formulas for $U(R|C)$ can be obtained by interchanging the indices.

$$U(C|R) = \frac{H(X)+H(Y)-H(XY)}{H(Y)} = \frac{v}{w}$$

with

$$var = \frac{1}{n^2 w^4} \sum_i \sum_j n_{ij} \left(H(Y) \ln \left(\frac{n_{ij}}{n_{i\cdot}} \right) + (H(X) - H(XY)) \ln \left(\frac{n_{\cdot j}}{n} \right) \right)^2$$

where

$$\begin{aligned} v &= H(X) + H(Y) - H(XY) \\ w &= H(Y) \\ H(X) &= -\sum_i \left(\frac{n_{i\cdot}}{n} \right) \ln \left(\frac{n_{i\cdot}}{n} \right) \\ H(Y) &= -\sum_j \left(\frac{n_{\cdot j}}{n} \right) \ln \left(\frac{n_{\cdot j}}{n} \right) \\ H(XY) &= -\sum_i \sum_j \left(\frac{n_{ij}}{n} \right) \ln \left(\frac{n_{ij}}{n} \right) \end{aligned}$$

Refer to Theil (1972, pp. 115–120) and Goodman and Kruskal (1979).

Uncertainty Coefficient (*U*)

The uncertainty coefficient, U, is the symmetric version of the two asymmetric coefficients. It has the range $0 \leq U \leq 1$. It is defined as

$$U = \frac{2(H(X)+H(Y)-H(XY))}{H(X)+H(Y)}$$

with

$$var = 4 \sum_i \sum_j \frac{n_{ij} \left(H(XY) \ln \left(\frac{n_{i\cdot} n_{\cdot j}}{n^2} \right) - (H(X)+H(Y)) \ln \left(\frac{n_{ij}}{n} \right) \right)^2}{n^2 \left(H(X)+H(Y) \right)^4}$$

Refer to Goodman and Kruskal (1979).

Binomial Proportion

When you specify the BINOMIAL option in the TABLES statement, PROC FREQ computes a binomial proportion for one-way tables. By default this is the proportion of observations in the first variable level, or class, that appears in the output. To specify a different level, use the LEVEL= option.

$$\hat{p} \;=\; n_1 \,/\, n$$

where n_1 is the frequency for the first level and n is the total frequency for the one-way table. The standard error for the binomial proportion is computed as

$$se(\hat{p}) \quad = \quad \sqrt{\hat{p}\,(1-\hat{p})\;/\;n}$$

Using the normal approximation to the binomial distribution, PROC FREQ constructs asymptotic confidence limits for p according to

$$\hat{p} \quad \pm \quad (\; z_{\alpha/2} \;\times\; se(\hat{p}) \;)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution. The confidence level α is determined by the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

If you specify the BINOMIALC option, PROC FREQ includes a continuity correction of $1/2n$ in the asymptotic confidence limits for p. The purpose of this correction is to adjust for the difference between the normal approximation and the binomial distribution, which is a discrete distribution. Refer to Fleiss (1981). With the continuity correction, the asymptotic confidence limits for p are

$$\hat{p} \quad \pm \quad (\; z_{\alpha/2} \;\times\; se(\hat{p}) \quad + \quad (1/2n) \;)$$

Additionally, PROC FREQ computes exact confidence limits for the binomial proportion using the *F* distribution method given in Collett (1991) and also described by Leemis and Trivedi (1996).

PROC FREQ computes an asymptotic test of the hypothesis that the binomial proportion equals p_0, where the value of p_0 is specified by the P= option in the TABLES statement. If you do not specify a value for the P= option, PROC FREQ uses $p_0 = 0.5$ by default. The asymptotic test statistic is

$$z \quad = \quad \frac{\hat{p} - p_0}{\sqrt{p_0\;(1-p_0)\;/\;n}}$$

If you specify the BINOMIALC option, PROC FREQ includes a continuity correction in the asymptotic test statistic, towards adjusting for the difference between the normal approximation and the discrete binomial distribution. Refer to Fleiss (1981). The continuity correction of $(1/2n)$ is subtracted from $(\hat{p} - p_0)$ in the numerator of the test statistic z if $(\hat{p} - p_0)$ is positive; otherwise, the continuity correction is added to the numerator.

PROC FREQ computes one-sided and two-sided p-values for this test. When the test statistic z is greater than zero, its expected value under the null hypothesis, PROC FREQ computes the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis that the true value of the proportion is greater than p_0. When the test statistic is less than or equal to zero, PROC FREQ computes the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. A small left-sided p-value supports the alternative hypothesis that the true value of the proportion is less than p_0. The one-sided p-value P_1 can be expressed as

$$\begin{aligned} P_1 &= \text{Prob}\ (\ Z > z\ ) \qquad \text{if} \ \ z > 0 \\ P_1 &= \text{Prob}\ (\ Z < z\ ) \qquad \text{if} \ \ z \le 0 \end{aligned}$$

where Z has a standard normal distribution. The two-sided *p*-value P_2 is computed as

$$P_2 = \text{Prob}\ (\ |Z| > |z|\ )$$

When you specify the BINOMIAL option in the EXACT statement, PROC FREQ also computes an exact test of the null hypothesis $H_0\colon p = p_0$. To compute this exact test, PROC FREQ uses the binomial probability function

$$\text{Prob}\ (X = x \mid p_0) = \binom{n}{x} p_0^{\,x}\ (1 - p_0)^{\ (n-x)} \qquad x = 0, 1, 2, \ldots, n$$

where the variable X has a binomial distribution with parameters n and p_0. To compute $\text{Prob}(X \le n_1)$, PROC FREQ sums these binomial probabilities over x from zero to n_1. To compute $\text{Prob}(X \ge n_1)$, PROC FREQ sums these binomial probabilities over x from n_1 to n. Then the exact one-sided *p*-value is

$$P_1 = \min\ (\ \text{Prob}(X \le n_1 \mid p_0),\ \text{Prob}(X \ge n_1 \mid p_0)\ )$$

and the exact two-sided *p*-value is

$$P_2 = 2 \ \times \ P_1$$

Risks and Risk Differences

The RISKDIFF option in the TABLES statement provides estimates of risks (or binomial proportions) and risk differences for 2×2 tables. This analysis may be appropriate when comparing the proportion of some characteristic for two groups, where row 1 and row 2 correspond to the two groups, and the columns correspond to two possible characteristics or outcomes. For example, the row variable might be a treatment or dose, and the column variable might be the response. Refer to Collett (1991), Fleiss (1981), and Stokes, Davis, and Koch (1995).

Let the frequencies of the 2×2 table be represented as follows.

	Column 1	Column 2	Total
Row 1	n_{11}	n_{12}	$n_{1\cdot}$
Row 2	n_{21}	n_{22}	$n_{2\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	n

The column 1 risk for row 1 is the proportion of row 1 observations classified in column 1,

$$p_{1|1} \quad = \quad n_{11} \;/\; n_{1\cdot}$$

This estimates the conditional probability of the column 1 response, given the first level of the row variable.

The column 1 risk for row 2 is the proportion of row 2 observations classified in column 1,

$$p_{1|2} \quad = \quad n_{21} \;/\; n_{2\cdot}$$

and the overall column 1 risk is the proportion of all observations classified in column 1,

$$p_{\cdot 1} \quad = \quad n_{\cdot 1} \;/\; n$$

The column 1 risk difference compares the risks for the two rows, and it is computed as the column 1 risk for row 1 minus the column 1 risk for row 2,

$$(\mathit{pdiff})_1 \quad = \quad p_{1|1} - p_{1|2}$$

The risks and risk difference are defined similarly for column 2.

The standard error of the column 1 risk estimate for row i is computed as

$$se(p_{1|i}) \quad = \quad \sqrt{p_{1|i} \; (1 - p_{1|i}) \;/\; n_{i\cdot}}$$

The standard error of the overall column 1 risk estimate is computed as

$$se(p_{\cdot 1}) \quad = \quad \sqrt{p_{\cdot 1} \; (1 - p_{\cdot 1}) \;/\; n}$$

If the two rows represent independent binomial samples, the standard error for the column 1 risk difference is computed as

$$se\;(\;(\mathit{pdiff})_1\;) \quad = \quad \sqrt{var(p_{1|1}) + var(p_{1|2})}$$

The standard errors are computed in a similar manner for the column 2 risks and risk difference.

Using the normal approximation to the binomial distribution, PROC FREQ constructs asymptotic confidence limits for the risks and risk differences according to

$$est \quad \pm \quad (\; z_{\alpha/2} \;\; \times \;\; se(est) \;)$$

where est is the estimate, $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution, and $se(est)$ is the standard error of the estimate. The confidence level α is determined from the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

If you specify the RISKDIFFC option, PROC FREQ includes continuity corrections in the asymptotic confidence limits for the risks and risk differences. Continuity corrections adjust for the difference between the normal approximation and the discrete binomial distribution. Refer to Fleiss (1981). Including a continuity correction, the asymptotic confidence limits become

$$est \ \pm \ (\, z_{\alpha/2} \ \times \ se(est) \ + \ cc \,)$$

where cc is the continuity correction. For the column 1 risk for row 1, $cc = (1/2n_{1\cdot})$; for the column 1 risk for row 2, $cc = (1/2n_{2\cdot})$; for the overall column 1 risk, $cc = (1/2n)$; and for the column 1 risk difference, $cc = ((1/n_{1\cdot} + 1/n_{2\cdot})/2)$. Continuity corrections are computed similarly for the column 2 risks and risk difference.

PROC FREQ computes exact confidence limits for the column 1, column 2, and overall risks using the *F* distribution method given in Collett (1991) and also described by Leemis and Trivedi (1996). PROC FREQ does not provide exact confidence limits for the risk differences. Refer to Agresti (1992) for a discussion of issues involved in constructing exact confidence limits for differences of proportions.

Odds Ratio and Relative Risks for 2 x 2 Tables

Odds Ratio (Case-Control Studies)

The odds ratio is a useful measure of association for a variety of study designs. For a retrospective design called a *case-control study*, the odds ratio can be used to estimate the relative risk when the probability of positive response is small (Agresti 1990). In a case-control study, two independent samples are identified based on a binary (yes-no) response variable, and the conditional distribution of a binary explanatory variable is examined, within fixed levels of the response variable. Refer to Stokes, Davis, and Koch (1995) and Agresti (1996).

The odds of a positive response (column 1) in row 1 is n_{11}/n_{12}. Similarly, the odds of a positive response in row 2 is n_{21}/n_{22}. The odds ratio is formed as the ratio of the row 1 odds to the row 2 odds. The odds ratio for 2×2 tables is defined as

$$\mathrm{OR} = \frac{n_{11}/n_{12}}{n_{21}/n_{22}} = \frac{n_{11} \, n_{22}}{n_{12} \, n_{21}}$$

The odds ratio can be any nonnegative number. When the row and column variables are independent, the true value of the odds ratio equals 1. An odds ratio greater than 1 indicates that the odds of a positive response are higher in row 1 than in row 2. Values less than 1 indicate the odds of positive response are higher in row 2. The strength of association increases with the deviation from 1.

The transformation $G = (\mathrm{OR} - 1)/(\mathrm{OR} + 1)$ transforms the odds ratio to the range $(-1, 1)$ with $G = 0$ when $\mathrm{OR} = 1$; $G = -1$ when $\mathrm{OR} = 0$; and G approaches 1 as OR approaches infinity. G is the gamma statistic, which PROC FREQ computes when you specify the MEASURES option.

The asymptotic $100(1-\alpha)\%$ confidence limits for the odds ratio are

$$\left(\mathrm{OR} \cdot \exp(-z\sqrt{v}),\ \mathrm{OR} \cdot \exp(z\sqrt{v}) \right)$$

where

$$v = var(\ln \mathrm{OR}) = \frac{1}{n_{11}} + \frac{1}{n_{12}} + \frac{1}{n_{21}} + \frac{1}{n_{22}}$$

and z is the $100(1-\alpha/2)$ percentile of the standard normal distribution. If any of the four cell frequencies are zero, the estimates are not computed.

When you specify option OR in the EXACT statement, PROC FREQ computes exact confidence limits for the odds ratio. Because this is a discrete problem, the confidence coefficient for these exact confidence limits is not exactly $1-\alpha$ but is at least $1-\alpha$. Thus, these confidence limits are conservative. Refer to Agresti (1992).

PROC FREQ computes exact confidence limits for the odds ratio with an algorithm based on that presented by Thomas (1971). Refer also to Gart (1971). The following two equations are solved iteratively for the lower and upper confidence limits, ϕ_1 and ϕ_2.

$$\sum_{i=n_{11}}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1}-i} \phi_1^i \ \Big/ \ \sum_{i=0}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1}-i} \phi_1^i \quad = \quad \alpha/2$$

$$\sum_{i=0}^{n_{11}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1}-i} \phi_2^i \ \Big/ \ \sum_{i=0}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1}-i} \phi_2^i \quad = \quad \alpha/2$$

When the odds ratio equals zero, which occurs when either $n_{11} = 0$ or $n_{22} = 0$, then PROC FREQ sets the lower exact confidence limit to zero and determines the upper limit with level α. Similarly, when the odds ratio equals infinity, which occurs when either $n_{12} = 0$ or $n_{21} = 0$, then PROC FREQ sets the upper exact confidence limit to infinity and determines the lower limit with level α.

Relative Risks (Cohort Studies)

These measures of relative risk are useful in *cohort* (prospective) study designs, where two samples are identified based on the presence or absence of an explanatory factor. The two samples are observed in future time for the binary (yes-no) response variable under study. Relative risk measures are also useful in cross-sectional studies, where two variable are observed simultaneously. Refer to Stokes, Davis, and Koch (1995) and Agresti (1996).

The column 1 relative risk is the ratio of the column 1 risks for row 1 to row 2. The column 1 risk for row 1 is the proportion of the row 1 observations classified in column 1,

$$p_{1|1} \quad = \quad n_{11} \ / \ n_{1\cdot}$$

Similarly, the column 1 risk for row 2 is

$$p_{1|2} \quad = \quad n_{21} \; / \; n_{2\cdot}$$

The column 1 relative risk is then computed as

$$\mathrm{RR}_1 = \frac{p_{1|1}}{p_{1|2}}$$

A relative risk greater than 1 indicates that the probability of positive response is greater in row 1 than in row 2. Similarly, a relative risk less than 1 indicates that the probability of positive response is less in row 1 than in row 2. The strength of association increases with the deviation from 1.

The asymptotic $100(1 - \alpha)$% confidence limits for the column 1 relative risk are

$$\left(\mathrm{RR}_1 \cdot \exp(-z\sqrt{v}), \; \mathrm{RR}_1 \cdot \exp(z\sqrt{v}) \right)$$

where

$$v = var(\ln \mathrm{RR}_1) = \frac{1 - p_{1|1}}{n_{11}} + \frac{1 - p_{1|2}}{n_{21}}$$

and z is the $100(1-\alpha/2)$ percentile of the standard normal distribution. If either n_{11} or n_{21} is zero, the estimates are not computed.

PROC FREQ computes the column 2 relative risks in a similar manner.

Cochran-Armitage Test for Trend

The TREND option in the TABLES statement requests the Cochran-Armitage test for trend, which tests for trend in binomial proportions across levels of a single factor or covariate. This test is appropriate for a contingency table where one variable has two levels and the other variable is ordinal. The two-level variable represents the response, and the other variable represents an explanatory variable with ordered levels. When the contingency table has two columns and R rows, PROC FREQ tests for trend across the R levels of the row variable, and the binomial proportion is computed as the proportion of observations in the first column. When the table has two rows and C columns, PROC FREQ tests for trend across the C levels of the column variable, and the binomial proportion is computed as the proportion of observations in the first row.

The trend test is based upon the regression coefficient for the weighted linear regression of the binomial proportions on the scores of the levels of the explanatory variable. Refer to Margolin (1988) and Agresti (1990). If the contingency table has two columns and R rows, the trend test statistic is computed as

$$T \quad = \quad \frac{\sum_{i=1}^{R} n_{i1}(R_i - \bar{R})}{\sqrt{p_{\cdot 1}(1 - p_{\cdot 1})s^2}}$$

where

$$s^2 \;=\; \sum_{i=1}^{R} n_{i\cdot}(R_i - \bar{R})^2$$

The row scores R_i are determined by the value of the SCORES= option in the TABLES statement. By default, PROC FREQ uses table scores. For character variables, the table scores for the row variable are the row numbers (for example, 1 for the first row, 2 for the second row, and so on). For numeric variables, the table score for each row is the numeric value of the row level. When you perform the trend test, the explanatory variable may be numeric (for example, dose of a test substance), and these variable values may be appropriate scores. If the explanatory variable has ordinal levels that are not numeric, you can assign meaningful scores to the variable levels. Sometimes equidistant scores, such as the table scores for a character variable, may be appropriate. For more information on choosing scores for the trend test, refer to Margolin (1988).

The null hypothesis for the Cochran-Armitage test is no trend, which means that the binomial proportion $p_{i1} = n_{i1}/n_{i\cdot}$ is the same for all levels of the explanatory variable. Under this null hypothesis, the trend test statistic is asymptotically distributed as a standard normal random variable. In addition to this asymptotic test, PROC FREQ can compute the exact trend test, which you request by specifying the TREND option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

PROC FREQ computes one-sided and two-sided *p*-values for the trend test. When the test statistic is greater than its null hypothesis expected value of zero, PROC FREQ computes the right-sided *p*-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided *p*-value supports the alternative hypothesis of increasing trend in binomial proportions from row 1 to row R. When the test statistic is less than or equal to zero, PROC FREQ outputs the left-sided *p*-value. A small left-sided *p*-value supports the alternative of decreasing trend.

The one-sided *p*-value P_1 can be expressed as

$$P_1 \;=\; \text{Prob (Trend Statistic} > T \text{)} \qquad \text{if} \;\; T > 0$$

$$P_1 \;=\; \text{Prob (Trend Statistic} < T \text{)} \qquad \text{if} \;\; T \le 0$$

The two-sided *p*-value P_2 is computed as

$$P_2 \;=\; \text{Prob (}|\text{Trend Statistic}| > |T| \text{)}$$

Jonckheere-Terpstra Test

The JT option in the TABLES statement requests the Jonckheere-Terpstra test, which is a nonparametric test for ordered differences among classes. It tests the null hypothesis that the distribution of the response variable does not differ among classes. It is designed to detect alternatives of ordered class differences, which can be expressed as $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_R$ (or $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_R$), with at least one of the inequalities being strict, where τ_i denotes the effect of class i. For such ordered alternatives, the Jonckheere-Terpstra test can be preferable to tests of more general class difference alternatives, such as the Kruskal–Wallis test (requested by the option WILCOXON in the NPAR1WAY procedure). Refer to Pirie (1983) and Hollander and Wolfe (1973) for more information about the Jonckheere-Terpstra test.

The Jonckheere-Terpstra test is appropriate for a contingency table in which an ordinal column variable represents the response. The row variable, which can be nominal or ordinal, represents the classification variable. The levels of the row variable should be ordered according to the ordering you want the test to detect. The order of variable levels is determined by the ORDER= option in the PROC FREQ statement. The default is ORDER=INTERNAL, which orders by unformatted values. If you specify ORDER=DATA, PROC FREQ orders values according to their order in the input data set. For more information on how to order variable levels, see the ORDER= option on page 1442.

The Jonckheere-Terpstra test statistic is computed by first forming $R(R-1)/2$ Mann-Whitney counts $M_{i,i'}$, where $i < i'$, for pairs of rows in the contingency table,

$$\begin{aligned} M_{i,i'} \quad = & \quad \{ \quad \text{number of times } X_{i,j} < X_{i',j'}, \\ & \qquad\; j = 1, \ldots, n_{i\cdot}; \quad j' = 1, \ldots, n_{i'\cdot} \quad \} \\ + \tfrac{1}{2} & \quad \{ \quad \text{number of times } X_{i,j} = X_{i',j'}, \\ & \qquad\; j = 1, \ldots, n_{i\cdot}; \quad j' = 1, \ldots, n_{i'\cdot} \quad \} \end{aligned}$$

where $X_{i,j}$ is response j in row i. Then the Jonckheere-Terpstra test statistic is computed as

$$J \quad = \quad \sum_{1 \leq i <} \sum_{i' \leq R} M_{i,i'}$$

This test rejects the null hypothesis of no difference among classes for large values of J. Asymptotic *p*-values for the Jonckheere-Terpstra test are obtained by using the normal approximation for the distribution of the standardized test statistic. The standardized test statistic is computed as

$$J^* \quad = \quad \frac{J - E_0(J)}{\sqrt{var_0(J)}}$$

where $E_0(J)$ and $var_0(J)$ are the expected value and variance of the test statistic under the null hypothesis.

$$E_0(J) \quad = \quad \left(n^2 - \sum_i n_{i\cdot}^2 \right) / 4$$

$$var_0(J) \quad = \quad A/72 + B/\left[36n(n-1)(n-2)\right] + C/\left[8n(n-1)\right]$$

where

$$A = n(n-1)(2n+5) - \sum_i n_{i\cdot}(n_{i\cdot}-1)(2n_{i\cdot}+5) - \sum_j n_{\cdot j}(n_{\cdot j}-1)(2n_{\cdot j}+5)$$

$$B = \left[\sum_i n_{i\cdot}(n_{i\cdot}-1)(n_{i\cdot}-2) \right] \left[\sum_j n_{\cdot j}(n_{\cdot j}-1)(n_{\cdot j}-2) \right]$$

$$C = \left[\sum_i n_{i\cdot}(n_{i\cdot}-1) \right] \left[\sum_j n_{\cdot j}(n_{\cdot j}-1) \right]$$

In addition to this asymptotic test, PROC FREQ can compute the exact Jonckheere-Terpstra test, which you request by specifying the JT option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

PROC FREQ computes one-sided and two-sided *p*-values for the Jonckheere-Terpstra test. When the standardized test statistic is greater than its null hypothesis expected value of zero, PROC FREQ computes the right-sided *p*-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided *p*-value supports the alternative hypothesis of increasing order from row 1 to row R. When the standardized test statistic is less than or equal to zero, PROC FREQ computes the left-sided *p*-value. A small left-sided *p*-value supports the alternative of decreasing order from row 1 to row R.

The one-sided *p*-value P_1 can be expressed as

$$\begin{aligned} P_1 &= \text{Prob (Std JT Statistic} > J^* \text{)} \qquad \text{if} \quad J^* > 0 \\ P_1 &= \text{Prob (Std JT Statistic} < J^* \text{)} \qquad \text{if} \quad J^* \le 0 \end{aligned}$$

The two-sided *p*-value P_2 is computed as

$$P_2 = \text{Prob (|Std JT Statistic|} > |J^*| \text{)}$$

Tests and Measures of Agreement

When you specify the AGREE option in the TABLES statement, PROC FREQ computes tests and measures of agreement for square tables (that is, for tables where the number of rows equals the number of columns). For two-way tables, these tests and measures include McNemar's test for 2 × 2 tables, Bowker's test of symmetry, the simple kappa coefficient, and the weighted kappa coefficient. For multiple strata (n-way tables, where $n > 2$), PROC FREQ computes the overall simple kappa coefficient and the overall weighted kappa coefficient, as well as tests for equal kappas (simple and weighted) among strata. Cochran's Q is computed for multi-way tables when each variable has two levels, that is, for $2 \times 2 \times \cdots \times 2$ tables.

PROC FREQ computes the kappa coefficients (simple and weighted), their asymptotic standard errors, and their confidence limits when you specify the AGREE option in the TABLES statement. If you also specify the KAPPA option in the TEST statement, then PROC FREQ computes the asymptotic test of the hypothesis that simple kappa equals zero. Similarly, if you specify the WTKAP option in the TEST statement, PROC FREQ computes the asymptotic test for weighted kappa.

In addition to the asymptotic tests described in this section, PROC FREQ computes the exact p-value for McNemar's test when you specify the option MCNEM in the EXACT statement. For the kappa statistics, PROC FREQ computes the exact test of the hypothesis that kappa (or weighted kappa) equals zero when you specify the option KAPPA (or WTKAP) in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

The discussion of each test and measures of agreement provides the formulas that PROC FREQ uses to compute the AGREE statistics. For information on the use and interpretation of these statistics, refer to Agresti (1990), Agresti (1996), Fleiss (1981), and the other references cited for each statistic.

McNemar's Test

PROC FREQ computes McNemar's test for 2×2 tables when you specify the AGREE option. McNemar's test is appropriate when you are analyzing data from matched pairs of subjects with a dichotomous (yes-no) response. It tests the null hypothesis of marginal homogeneity, or $p_{1\cdot} = p_{\cdot 1}$. McNemar's test is computed as

$$Q_M = \frac{(n_{12} - n_{21})^2}{n_{12} + n_{21}}$$

Under the null hypothesis, Q_M has an asymptotic chi-square distribution with one degree of freedom. Refer to McNemar (1947), as well as the references cited in the preceding section. In addition to the asymptotic test, PROC FREQ also computes the exact p-value for McNemar's test when you specify the MCNEM option in the EXACT statement.

Bowker's Test of Symmetry

For Bowker's test of symmetry, the null hypothesis is that the probabilities in the square table satisfy symmetry or that $p_{ij} = p_{ji}$ for all pairs of table cells. When there are more than two categories, Bowker's test of symmetry is calculated as

$$Q_B = \sum_{i<j}\sum \frac{(n_{ij} - n_{ji})^2}{n_{ij} + n_{ji}}$$

For large samples, Q_B has an asymptotic chi-square distribution with $R(R-1)/2$ degrees of freedom under the null hypothesis of symmetry of the expected counts. Refer to Bowker (1948). For two categories, this test of symmetry is identical to McNemar's test.

Simple Kappa Coefficient

The simple kappa coefficient, introduced by Cohen (1960), is a measure of interrater agreement:

$$\hat{\kappa} = \frac{P_o - P_e}{1 - P_e}$$

where $P_o = \sum_i p_{ii}$ and $P_e = \sum_i p_{i.}p_{.i}$. If the two response variables are viewed as two independent ratings of the n subjects, the kappa coefficient equals +1 when there is complete agreement of the raters. When the observed agreement exceeds chance agreement, kappa is positive, with its magnitude reflecting the strength of agreement. Although this is unusual in practice, kappa is negative when the observed agreement is less than chance agreement. The minimum value of kappa is between -1 and 0, depending on the marginal proportions.

The asymptotic variance of the simple kappa coefficient can be estimated by the following, according to Fleiss, Cohen, and Everitt (1969):

$$var = \frac{A + B - C}{(1 - P_e)^2 n}$$

where

$$A = \sum_i p_{ii} \left[1 - (p_{i\cdot} + p_{\cdot i})(1 - \hat{\kappa}) \right]^2$$

$$B = (1 - \hat{\kappa})^2 \sum\sum_{i \neq j} p_{ij} (p_{\cdot i} + p_{j\cdot})^2$$

and

$$C = \left[\hat{\kappa} - P_e(1 - \hat{\kappa}) \right]^2$$

PROC FREQ computes confidence limits for the simple kappa coefficient according to

$$\hat{\kappa} \quad \pm \quad (\ z_{\alpha/2} \ \times \ \sqrt{var}\ )$$

where $z_{\alpha/2}$ is the $100(1 - \alpha/2)$ percentile of the standard normal distribution. The value of α is determined by the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

To compute an asymptotic test for the kappa coefficient, PROC FREQ uses a standardized test statistic $\hat{\kappa}^*$, which has an asymptotic standard normal distribution under the null hypothesis that kappa equals zero. The standardized test statistic is computed as

$$\hat{\kappa}^* = \frac{\hat{\kappa}}{\sqrt{var_0(\hat{\kappa})}}$$

where $var_0(\hat{\kappa})$ is the variance of the kappa coefficient under the null hypothesis.

$$var_0(\hat{\kappa}) = \frac{P_e + P_e^2 - \sum_i p_{i\cdot}p_{\cdot i}(p_{i\cdot} + p_{\cdot i})}{(1 - P_e)^2 n}$$

Refer to Fleiss (1981).

In addition to the asymptotic test for kappa, PROC FREQ computes the exact test when you specify the KAPPA or AGREE option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

Weighted Kappa Coefficient

The weighted kappa coefficient is a generalization of the simple kappa coefficient, using weights to quantify the relative difference between categories. For 2×2 tables, the weighted kappa coefficient equals the simple kappa coefficient. PROC FREQ displays the weighted kappa coefficient only for tables larger than 2×2. PROC FREQ computes the weights from the column scores, using either the Cicchetti-Allison weight type or the Fleiss-Cohen weight type, both of which are described in the following section. The weights w_{ij} are constructed so that $0 \leq w_{ij} < 1$ for all $i \neq j$, $w_{ii} = 1$ for all i, and $w_{ij} = w_{ji}$. The weighted kappa coefficient is defined as

$$\hat{\kappa}_w = \frac{P_{o(w)} - P_{e(w)}}{1 - P_{e(w)}}$$

where

$$P_{o(w)} = \sum_i \sum_j w_{ij} p_{ij}$$

and

$$P_{e(w)} = \sum_i \sum_j w_{ij} p_{i\cdot} p_{\cdot j}$$

The asymptotic variance of the weighted kappa coefficient can be estimated by the following, according to Fleiss, Cohen, and Everitt (1969):

$$var = \frac{\sum_i \sum_j p_{ij} \left[w_{ij} - (\overline{w}_{i\cdot} + \overline{w}_{\cdot j})(1 - \hat{\kappa}_w) \right]^2 - \left[\hat{\kappa}_w - P_{e(w)}(1 - \hat{\kappa}_w) \right]^2}{(1 - P_{e(w)})^2 n}$$

where

$$\overline{w}_{i\cdot} = \sum_j p_{\cdot j} w_{ij}$$

and

$$\overline{w}_{\cdot j} = \sum_i p_{i\cdot} w_{ij}$$

PROC FREQ computes confidence limits for the weighted kappa coefficient according to

$$\hat{\kappa}_w \quad \pm \quad (\; z_{\alpha/2} \;\; \times \;\; \sqrt{var} \;)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution. The value of α is determined by the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

To compute an asymptotic test for the weighted kappa coefficient, PROC FREQ uses a standardized test statistic $\hat{\kappa}_w^*$, which has an asymptotic standard normal distribution under the null hypothesis that weighted kappa equals zero. The standardized test statistic is computed as

$$\hat{\kappa}_w^* = \frac{\hat{\kappa}_w}{\sqrt{var_0(\hat{\kappa}_w)}}$$

where $var_0(\hat{\kappa}_w)$ is the variance of the weighted kappa coefficient under the null hypothesis.

$$var_0(\hat{\kappa}_w) = \frac{\sum_i \sum_j p_{i\cdot} p_{\cdot j} \left[w_{ij} - (\overline{w}_{i\cdot} + \overline{w}_{\cdot j}) \right]^2 - P_{e(w)}^2}{(1 - P_{e(w)})^2 n}$$

Refer to Fleiss (1981).

In addition to the asymptotic test for weighted kappa, PROC FREQ computes the exact test when you specify the WTKAP or AGREE option in the EXACT statement. See the section "Exact Statistics" beginning on page 1508 for information on exact tests.

Weights

PROC FREQ computes kappa coefficient weights using the column scores and one of two available weight types. The column scores are determined by the SCORES= option in the TABLES statement. The two available weight types are Cicchetti-Allison and Fleiss-Cohen, and PROC FREQ uses the Cicchetti-Allison type by default. If you specify (WT=FC) with the AGREE option, then PROC FREQ uses the Fleiss-Cohen weight type to construct kappa weights.

PROC FREQ computes Cicchetti-Allison kappa coefficient weights using a form similar to that given by Cicchetti and Allison (1971).

$$w_{ij} = 1 - \frac{|C_i - C_j|}{C_C - C_1}$$

where C_i is the score for column i, and C is the number of categories or columns. You can specify the score type using the SCORES= option in the TABLES statement; if you do not specify the SCORES= option, PROC FREQ uses table scores. For numeric variables, table scores are the values of the numeric row and column headings. You can assign numeric values to the categories in a way that reflects their level of similarity. For example, suppose you have four categories and order them according to similarity. If you assign them values of 0, 2, 4, and 10, the following weights are used for computing the weighted kappa coefficient: w_{12} = 0.8, w_{13} = 0.6, w_{14} = 0, w_{23} = 0.8, w_{24} = 0.2, and w_{34} = 0.4. Note that when there are only two categories (that is, C = 2), the weighted kappa coefficient is identical to the simple kappa coefficient.

If you specify (WT=FC) with the AGREE option in the TABLES statement, PROC FREQ computes Fleiss-Cohen kappa coefficient weights using a form similar to that given by Fleiss and Cohen (1973).

$$w_{ij} = 1 - \frac{(C_i - C_j)^2}{(C_C - C_1)^2}$$

For the preceding example, the weights used for computing the weighted kappa coefficient are: w_{12} = 0.96, w_{13} = 0.84, w_{14} = 0, w_{23} = 0.96, w_{24} = 0.36, and w_{34} = 0.64.

Overall Kappa Coefficient

When there are multiple strata, PROC FREQ combines the stratum-level estimates of kappa into an overall estimate of the supposed common value of kappa. Assume there are q strata, indexed by $h = 1, 2, \ldots, q$, and let $var(\hat{\kappa}_h)$ denote the squared standard error of $\hat{\kappa}_h$. Then the estimate of the overall kappa, according to Fleiss (1981), is computed as

$$\hat{\kappa}_{overall} = \sum_{h=1}^{q} \frac{\hat{\kappa}_h}{var(\hat{\kappa}_h)} \Big/ \sum_{h=1}^{q} \frac{1}{var(\hat{\kappa}_h)}$$

PROC FREQ computes an estimate of the overall weighted kappa in a similar manner.

Tests for Equal Kappa Coefficients

When there are multiple strata, the following chi-square statistic tests whether the stratum-level values of kappa are equal.

$$Q_K = \sum_{h=1}^{q} \frac{(\hat{\kappa}_h - \hat{\kappa}_{overall})^2}{var(\hat{\kappa}_h)}$$

Under the null hypothesis of equal kappas over the q strata, Q_K has an asymptotic chi-square distribution with $q-1$ degrees of freedom. PROC FREQ computes a test for equal weighted kappa coefficients in a similar manner.

Cochran's *Q* Test

Cochran's Q is computed for multi-way tables when each variable has two levels, that is, for $2 \times 2 \cdots \times 2$ tables. Cochran's Q statistic is used to test the homogeneity of the one-dimensional margins. Let m denote the number of variables and N denote the total number of subjects. Then Cochran's Q statistic is computed as

$$Q_C = (m-1)\frac{m\sum_{j=1}^{m} T_j^2 - T^2}{mT - \sum_{k=1}^{N} S_k^2}$$

where T_j is the number of positive responses for variable j, T is the total number of positive responses over all variables, and S_k is the number of positive responses for subject k. Under the null hypothesis, Cochran's Q is an approximate chi-square statistic with $m-1$ degrees of freedom. Refer to Cochran (1950). When there are only two binary response variables ($m = 2$), Cochran's Q simplifies to McNemar's test. When there are more than two response categories, you can test for marginal homogeneity using the repeated measures capabilities of the CATMOD procedure.

Tables with Zero Rows and Columns

The AGREE statistics are defined only for square tables, where the number of rows equals the number of columns. If the table is not square, PROC FREQ does not compute AGREE statistics. In the kappa statistic framework, where two independent raters assign ratings to each of n subjects, suppose one of the raters does not use all possible r rating levels. If the corresponding table has r rows but only $r-1$ columns, then the table is not square, and PROC FREQ does not compute the AGREE statistics. To create a square table in this situation, use the ZEROS option in the WEIGHT statement, which requests that PROC FREQ include observations with zero weights in the analysis. And input zero-weight observations to represent any rating levels that are not used by a rater, so that the input data set has at least one observation for each possible rater and rating combination. This includes all rating levels in the analysis, whether or not all levels are actually assigned by both raters. The resulting table is a square table, $r \times r$, and so all AGREE statistics can be computed.

For more information, see the description of the ZEROS option. By default, PROC FREQ does not process observations that have zero weights, because these observations do not contribute to the total frequency count, and because any resulting zero-weight row or column causes many of the tests and measures of association to be undefined. However, kappa statistics are defined for tables with a zero-weight row or column, and the ZEROS option allows input of zero-weight observations so you can construct the tables needed to compute kappas.

Cochran-Mantel-Haenszel Statistics

For *n*-way crosstabulation tables, consider the following example:

```
proc freq;
   tables A*B*C*D / cmh;
run;
```

The CMH option in the TABLES statement gives a stratified statistical analysis of the relationship between C and D, after controlling for A and B. The stratified analysis provides a way to adjust for the possible confounding effects of A and B without being forced to estimate parameters for them. The analysis produces Cochran-Mantel-Haenszel statistics, and for 2×2 tables, it includes estimation of the common odds ratio, common relative risks, and the Breslow-Day test for homogeneity of the odds ratios.

Let the number of strata be denoted by q, indexing the strata by $h = 1, 2, \ldots, q$. Each stratum contains a contingency table with X representing the row variable and Y representing the column variable. For table h, denote the cell frequency in row i and column j by n_{hij}, with corresponding row and column marginal totals denoted by $n_{hi.}$ and $n_{h.j}$, and the overall stratum total by n_h.

Because the formulas for the Cochran-Mantel-Haenszel statistics are more easily defined in terms of matrices, the following notation is used. Vectors are presumed to be column vectors unless they are transposed $(')$.

$$\begin{array}{rclr}
\mathbf{n}'_{hi} & = & (n_{hi1}, n_{hi2}, \ldots, n_{hiC}) & (1 \times C) \\
\mathbf{n}'_{h} & = & (\mathbf{n}'_{h1}, \mathbf{n}'_{h2}, \ldots, \mathbf{n}'_{hR}) & (1 \times RC) \\
p_{hi\cdot} & = & \frac{n_{hi\cdot}}{n_h} & (1 \times 1) \\
p_{h\cdot j} & = & \frac{n_{h\cdot j}}{n_h} & (1 \times 1) \\
\mathbf{P}'_{h*\cdot} & = & (p_{h1\cdot}, p_{h2\cdot}, \ldots, p_{hR\cdot}) & (1 \times R) \\
\mathbf{P}'_{h\cdot *} & = & (p_{h\cdot 1}, p_{h\cdot 2}, \ldots, p_{h\cdot C}) & (1 \times C)
\end{array}$$

Assume that the strata are independent and that the marginal totals of each stratum are fixed. The null hypothesis, H_0, is that there is no association between X and Y in any of the strata. The corresponding model is the multiple hypergeometric; this implies that, under H_0, the expected value and covariance matrix of the frequencies are, respectively,

$$\mathbf{m}_h = \mathbf{E}[\mathbf{n}_h \mid H_0] = n_h (\mathbf{P}_{h\cdot *} \otimes \mathbf{P}_{h*\cdot})$$

and

$$\mathbf{var}[\mathbf{n}_h \mid H_0] = c \left((\mathbf{D}_{\mathbf{P}h\cdot *} - \mathbf{P}_{h\cdot *}\mathbf{P}'_{h\cdot *}) \otimes (\mathbf{D}_{\mathbf{P}h*\cdot} - \mathbf{P}_{h*\cdot}\mathbf{P}'_{h*\cdot}) \right)$$

where

$$c = \frac{n_h^2}{n_h - 1}$$

and where $\otimes$ denotes Kronecker product multiplication and $\mathbf{D_a}$ is a diagonal matrix with elements of $\mathbf{a}$ on the main diagonal.

The generalized CMH statistic (Landis, Heyman, and Koch 1978) is defined as

$$Q_{\text{CMH}} = \mathbf{G}'\mathbf{V_G}^{-1}\mathbf{G}$$

where

$$\begin{aligned} \mathbf{G} &= \sum_h \mathbf{B}_h(\mathbf{n}_h - \mathbf{m}_h) \\ \mathbf{V_G} &= \sum_h \mathbf{B}_h \left(\mathbf{Var}(\mathbf{n}_h \mid H_0)\right) \mathbf{B}_h' \end{aligned}$$

and where

$$\mathbf{B}_h = \mathbf{C}_h \otimes \mathbf{R}_h$$

is a matrix of fixed constants based on column scores $\mathbf{C}_h$ and row scores $\mathbf{R}_h$. When the null hypothesis is true, the CMH statistic has an asymptotic chi-square distribution with degrees of freedom equal to the rank of $\mathbf{B}_h$. If $\mathbf{V_G}$ is found to be singular, PROC FREQ prints a message and sets the value of the CMH statistic to missing.

PROC FREQ computes three CMH statistics using this formula for the generalized CMH statistic, with different row and column score definitions for each statistic. The CMH statistics that PROC FREQ computes are the correlation statistic, the ANOVA (row mean scores) statistic, and the general association statistic. These statistics test the null hypothesis of no association against different alternative hypotheses. The following sections describe the computation of these CMH statistics.

CAUTION: The CMH statistics have low power for detecting an association in which the patterns of association for some of the strata are in the opposite direction of the patterns displayed by other strata. Thus, a nonsignificant CMH statistic suggests either that there is no association or that no pattern of association has enough strength or consistency to dominate any other pattern.

Correlation Statistic

The correlation statistic, popularized by Mantel and Haenszel (1959) and Mantel (1963), has one degree of freedom and is known as the Mantel-Haenszel statistic.

The alternative hypothesis for the correlation statistic is that there is a linear association between X and Y in at least one stratum. If either X or Y does not lie on an ordinal (or interval) scale, then this statistic is not meaningful.

To compute the correlation statistic, PROC FREQ uses the formula for the generalized CMH statistic with the row and column scores determined by the SCORES= option in the TABLES statement. See the section "Scores" on page 1468 for more information on the available score types. The matrix of row scores $\mathbf{R}_h$ has dimension $1 \times R$, and the matrix of column scores $\mathbf{C}_h$ has dimension $1 \times C$.

When there is only one stratum, this CMH statistic reduces to $(n-1)r^2$, where r is the Pearson correlation coefficient between X and Y. When nonparametric (RANK or RIDIT) scores are specified, then the statistic reduces to $(n-1)r_s^2$, where r_s is the Spearman rank correlation coefficient between X and Y. When there is more than one stratum, then this CMH statistic becomes a stratum-adjusted correlation statistic.

ANOVA (Row Mean Scores) Statistic

The ANOVA statistic can be used only when the column variable Y lies on an ordinal (or interval) scale so that the mean score of Y is meaningful. For the ANOVA statistic, the mean score is computed for each row of the table, and the alternative hypothesis is that, for at least one stratum, the mean scores of the R rows are unequal. In other words, the statistic is sensitive to location differences among the R distributions of Y.

The matrix of column scores $\mathbf{C}_h$ has dimension $1 \times C$, the column scores are determined by the SCORES= option.

The matrix of row scores $\mathbf{R}_h$ has dimension $(R-1) \times R$ and is created internally by PROC FREQ as

$$\mathbf{R}_h = [\mathbf{I}_{R-1}, -\mathbf{J}_{R-1}]$$

where $\mathbf{I}_{R-1}$ is an identity matrix of rank $R-1$, and $\mathbf{J}_{R-1}$ is an $(R-1) \times 1$ vector of ones. This matrix has the effect of forming $R-1$ independent contrasts of the R mean scores.

When there is only one stratum, this CMH statistic is essentially an analysis of variance (ANOVA) statistic in the sense that it is a function of the variance ratio F statistic that would be obtained from a one-way ANOVA on the dependent variable Y. If nonparametric scores are specified in this case, then the ANOVA statistic is a Kruskal-Wallis test.

If there is more than one stratum, then this CMH statistic corresponds to a stratum-adjusted ANOVA or Kruskal-Wallis test. In the special case where there is one subject per row and one subject per column in the contingency table of each stratum, this CMH statistic is identical to Friedman's chi-square. See Example 29.8 on page 1546 for an illustration.

General Association Statistic

The alternative hypothesis for the general association statistic is that, for at least one stratum, there is some kind of association between X and Y. This statistic is always interpretable because it does not require an ordinal scale for either X or Y.

For the general association statistic, the matrix $\mathbf{R}_h$ is the same as the one used for the ANOVA statistic. The matrix $\mathbf{C}_h$ is defined similarly as

$$\mathbf{C}_h = [\mathbf{I}_{C-1}, -\mathbf{J}_{C-1}]$$

PROC FREQ generates both score matrices internally. When there is only one stratum, then the general association CMH statistic reduces to $Q_P(n-1)/n$, where Q_P is the Pearson chi-square statistic. When there is more than one stratum, then the CMH statistic becomes a stratum-adjusted Pearson chi-square statistic. Note that a similar adjustment can be made by summing the Pearson chi-squares across the strata. However, the latter statistic requires a large sample size in each stratum to support the resulting chi-square distribution with $q(R-1)(C-1)$ degrees of freedom. The CMH statistic requires only a large overall sample size since it has only $(R-1)(C-1)$ degrees of freedom.

Refer to Cochran (1954); Mantel and Haenszel (1959); Mantel (1963); Birch (1965); Landis, Heyman, and Koch (1978).

Adjusted Odds Ratio and Relative Risk Estimates

The CMH option provides adjusted odds ratio and relative risk estimates for stratified 2×2 tables. For each of these measures, PROC FREQ computes the Mantel-Haenszel estimate and the logit estimate. These estimates apply to *n*-way table requests in the TABLES statement, when the row and column variables both have only two levels.

For example,

```
proc freq;
   tables A*B*C*D / cmh;
run;
```

In this example, if the row and columns variables C and D both have two levels, PROC FREQ provides odds ratio and relative risk estimates, adjusting for the confounding variables A and B.

The choice of an appropriate measure depends on the study design. For case-control (retrospective) studies, the odds ratio is appropriate. For cohort (prospective) or cross-sectional studies, the relative risk is appropriate. See the section "Odds Ratio and Relative Risks for 2 x 2 Tables" beginning on page 1488 for more information on these measures.

Throughout this section, z denotes the $100(1-\alpha/2)$ percentile of the standard normal distribution.

Odds Ratio, Case-Control Studies

Mantel-Haenszel Estimator

The Mantel-Haenszel estimate of the common odds ratio is computed as

$$\mathrm{OR}_{\mathrm{MH}} = \frac{\sum_h n_{h11}\, n_{h22}/n_h}{\sum_h n_{h12}\, n_{h21}/n_h}$$

It is always computed unless the denominator is zero. Refer to Mantel and Haenszel (1959) and Agresti (1990).

Using the estimated variance for $\log(\mathrm{OR}_{\mathrm{MH}})$ given by Robins, Breslow, and Greenland (1986), PROC FREQ computes the corresponding $100(1-\alpha)\%$ confidence limits for the odds ratio as

$$(\ \mathrm{OR}_{\mathrm{MH}} \cdot \exp(-z\hat{\sigma}),\ \mathrm{OR}_{\mathrm{MH}} \cdot \exp(z\hat{\sigma})\ )$$

where

$$\begin{aligned}\hat{\sigma}^2 &= \hat{var}[\,ln(\mathrm{OR}_{\mathrm{MH}})\,] \\ &= \frac{\sum_h (n_{h11}+n_{h22})(n_{h11}\,n_{h22})/n_h^2}{2\left(\sum_h n_{h11}\,n_{h22}/n_h\right)^2} \\ &\quad + \frac{\sum_h [(n_{h11}+n_{h22})(n_{h12}\,n_{h21})+(n_{h12}+n_{h21})(n_{h11}\,n_{h22})]/n_h^2}{2\left(\sum_h n_{h11}\,n_{h22}/n_h\right)\left(\sum_h n_{h12}\,n_{h21}/n_h\right)} \\ &\quad + \frac{\sum_h (n_{h12}+n_{h21})(n_{h12}\,n_{h21})/n_h^2}{2\left(\sum_h n_{h12}\,n_{h21}/n_h\right)^2}\end{aligned}$$

Note that the Mantel-Haenszel odds ratio estimator is less sensitive to small n_h than the logit estimator.

Logit Estimator

The adjusted logit estimate of the odds ratio (Woolf 1955) is computed as

$$\mathrm{OR}_{\mathrm{L}} = \exp\left(\frac{\sum_h w_h \ln(\mathrm{OR}_h)}{\sum_h w_h}\right)$$

and the corresponding $100(1-\alpha)\%$ confidence limits are

$$\left(\mathrm{OR}_{\mathrm{L}} \cdot \exp\left(\frac{-z}{\sqrt{\sum_h w_h}}\right),\ \mathrm{OR}_{\mathrm{L}} \cdot \exp\left(\frac{z}{\sqrt{\sum_h w_h}}\right)\right)$$

where OR_h is the odds ratio for stratum h, and

$$w_h = \frac{1}{var(\ln \mathrm{OR}_h)}$$

If any cell frequency in a stratum h is zero, then PROC FREQ adds 0.5 to each cell of the stratum before computing OR_h and w_h (Haldane 1955), and prints a warning.

Exact Confidence Limits for the Common Odds Ratio

When you specify the COMOR option in the EXACT statement, PROC FREQ computes exact confidence limits for the common odds ratio for stratified 2×2 tables.

This computation assumes that the odds ratio is constant over all the 2×2 tables. Exact confidence limits are constructed from the distribution of $S = \sum_h n_{h11}$, conditional on the marginal totals of the 2×2 tables.

Because this is a discrete problem, the confidence coefficient for these exact confidence limits is not exactly $1 - \alpha$ but is at least $1 - \alpha$. Thus, these confidence limits are conservative. Refer to Agresti (1992).

PROC FREQ computes exact confidence limits for the common odds ratio with an algorithm based on that presented by Vollset, Hirji, and Elashoff (1991). Refer also to Mehta, Patel, and Gray (1985).

Conditional on the marginal totals of 2×2 table h, let the random variable S_h denote the frequency of table cell $(1,1)$. Given the row totals $n_{h1\cdot}$ and $n_{h2\cdot}$ and column totals $n_{h\cdot 1}$ and $n_{h\cdot 2}$, the lower and upper bounds for S_h are l_h and u_h,

$$\begin{aligned} l_h &= \max\left(\, 0,\ \ n_{h1\cdot} - n_{h\cdot 2}\,\right) \\ u_h &= \min\left(\, n_{h1\cdot},\ \ n_{h\cdot 1}\,\right) \end{aligned}$$

Let C_{s_h} denote the hypergeometric coefficient,

$$C_{s_h} = \binom{n_{h\cdot 1}}{s_h} \binom{n_{h\cdot 2}}{n_{h1\cdot} - s_h}$$

and let ϕ denote the common odds ratio. Then the conditional distribution of S_h is

$$P(\, S_h = s_h \ \mid\ n_{1\cdot},\ n_{\cdot 1},\ n_{\cdot 2}\,) \ = \ C_{s_h}\, \phi^{\,s_h} \ / \sum_{x\,=\,l_h}^{x\,=\,u_h} C_x\, \phi^{\,x}$$

Summing over all the 2×2 tables, $S = \sum_h S_h$, and the lower and upper bounds of S are l and u,

$$l = \sum_h l_h \qquad \text{and} \qquad u = \sum_h u_h$$

The conditional distribution of the sum S is

$$P(\, S = s \ \mid\ n_{h1\cdot},\ n_{h\cdot 1},\ n_{h\cdot 2};\ \ h = 1, \ldots, q\,) \ = \ C_s\, \phi^{\,s} \ / \sum_{x\,=\,l}^{x\,=\,u} C_x\, \phi^{\,x}$$

where

$$C_s = \sum_{s_1 + \ldots. + s_q \,=\, s} \left(\prod_h C_{s_h} \right)$$

Let s_0 denote the observed sum of cell (1,1) frequencies over the q tables. The following two equations are solved iteratively for lower and upper confidence limits for the common odds ratio, ϕ_1 and ϕ_2,

$$\sum_{x\,=\,s_o}^{x\,=\,u} C_x\,\phi_1^{\,x} \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x\,\phi_1^{\,x} \quad = \quad \alpha/2$$

$$\sum_{x\,=\,l}^{x\,=\,s_0} C_x\,\phi_2^{\,x} \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x\,\phi_2^{\,x} \quad = \quad \alpha/2$$

When the observed sum s_0 equals the lower bound l, then PROC FREQ sets the lower exact confidence limit to zero and determines the upper limit with level α. Similarly, when the observed sum s_0 equals the upper bound u, then PROC FREQ sets the upper exact confidence limit to infinity and determines the lower limit with level α.

When you specify the COMOR option in the EXACT statement, PROC FREQ also computes the exact test that the common odds ratio equals one. Setting $\phi = 1$, the conditional distribution of the sum S under the null hypothesis becomes

$$P_0(\; S = s \;\mid\; n_{h1\cdot},\; n_{h\cdot 1},\; n_{h\cdot 2};\;\; h = 1, \ldots, q\;) \quad = \quad C_s \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x$$

The point probability for this exact test is the probability of the observed sum s_0 under the null hypothesis, conditional on the marginals of the stratified 2×2 tables, and is denoted by $P_0(s_0)$. The expected value of S under the null hypothesis is

$$E_0(S) \quad = \quad \sum_{x\,=\,l}^{x\,=\,u} x\,C_x \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x$$

The one-sided exact *p*-value is computed from the conditional distribution as $P_0(S >= s_0)$ or $P_0(S \leq s_0)$, depending on whether the observed sum s_0 is greater or less than $E_0(S)$.

$$P_1 \quad = \quad P_0(\; S >= s_0\;) \quad = \quad \sum_{x\,=\,s_0}^{x\,=\,u} C_x \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x \qquad \text{if } s_0 \;>\; E_0(S)$$
$$P_1 \quad = \quad P_0(\; S <= s_0\;) \quad = \quad \sum_{x\,=\,l}^{x\,=\,s_0} C_x \;/\; \sum_{x\,=\,l}^{x\,=\,u} C_x \qquad \text{if } s_0 \;\leq\; E_0(S)$$

PROC FREQ computes two-sided *p*-values for this test according to three different definitions. A two-sided *p*-value is computed as twice the one-sided *p*-value, setting the result equal to one if it exceeds one.

$$P_2^{\;a} \;=\; 2 \;\times\; P_1$$

Additionally, a two-sided *p*-value is computed as the sum of all probabilities less than or equal to the point probability of the observed sum s_0, summing over all possible values of s, $l \le s \le u$.

$$P_2{}^b \;=\; \sum_{l \le s \le u:\; P_0(s) \le P_0(s_0)} P_0(s)$$

Also, a two-sided *p*-value is computed as the sum of the one-sided *p*-value and the corresponding area in the opposite tail of the distribution, equidistant from the expected value.

$$P_2{}^c \;=\; P_0\left(\, |S - E_0(S)| \;\ge\; |s_0 - E_0(S)| \,\right)$$

Relative Risks, Cohort Studies

Mantel-Haenszel Estimator

The Mantel-Haenszel estimate of the common relative risk for column 1 is computed as

$$\mathrm{RR_{MH}} = \frac{\sum_h n_{h11}\; n_{h2\cdot}/n_h}{\sum_h n_{h21}\; n_{h1\cdot}/n_h}$$

It is always computed unless the denominator is zero. Refer to Mantel and Haenszel (1959) and Agresti (1990).

Using the estimated variance for $\log(\mathrm{RR_{MH}})$ given by Greenland and Robins (1985), PROC FREQ computes the corresponding $100(1-\alpha)$% confidence limits for the relative risk as

$$(\; \mathrm{RR_{MH}} \cdot \exp(-z\hat{\sigma}),\; \mathrm{RR_{MH}} \cdot \exp(z\hat{\sigma}) \;)$$

where

$$\begin{aligned} \hat{\sigma}^2 &= v\hat{a}r[\, ln(\mathrm{RR_{MH}})\,] \\ &= \frac{\sum_h (n_{h1\cdot}\; n_{h2\cdot}\; n_{h\cdot 1} - n_{h11}\; n_{h21}\; n_h)/n_h^2}{(\sum_h n_{h11}\; n_{h2\cdot}/n_h)\,(\sum_h n_{h21}\; n_{h1\cdot}/n_h)} \end{aligned}$$

Logit Estimator

The adjusted logit estimate of the common relative risk for column 1 is computed as

$$\mathrm{RR_L} = \exp\left(\frac{\sum_h w_h \ln \mathrm{RR}_h}{\sum w_h} \right)$$

and the corresponding $100(1-\alpha)$% confidence limits are

$$\left(\mathrm{RR_L} \exp\left(\frac{-z}{\sqrt{\sum_h w_h}} \right), \mathrm{RR_L} \exp\left(\frac{z}{\sqrt{\sum_h w_h}} \right) \right)$$

where RR_h is the column 1 relative risk estimate for stratum h, and

$$w_h = \frac{1}{var(\ln \mathrm{RR}_h)}$$

If n_{h11} or n_{h21} is zero, then PROC FREQ adds 0.5 to each cell of the stratum before computing RR_h and w_h, and prints a warning. Refer to Kleinbaum, Kupper, and Morgenstern (1982, Sections 17.4 and 17.5).

Breslow-Day Test for Homogeneity of the Odds Ratios

When you specify the CMH option, PROC FREQ computes the Breslow-Day test for stratified analysis of 2×2 tables. It tests the null hypothesis that the odds ratios for the q strata are all equal. When the null hypothesis is true, the statistic has approximately a chi-square distribution with $q - 1$ degrees of freedom. Refer to Breslow and Day (1980) and Agresti (1996).

The Breslow-Day statistic is computed as

$$Q_{\mathrm{BD}} = \sum_h \frac{(n_{h11} - E(n_{h11} \mid \mathrm{OR}_{\mathrm{MH}}))^2}{var(n_{h11} \mid \mathrm{OR}_{\mathrm{MH}})}$$

where E and var denote expected value and variance, respectively. The summation does not include any table with a zero row or column. If $\mathrm{OR}_{\mathrm{MH}}$ equals zero or if it is undefined, then PROC FREQ does not compute the statistic and prints a warning message.

For the Breslow-Day test to be valid, the sample size should be relatively large in each stratum, and at least 80% of the expected cell counts should be greater than 5. Note that this is a stricter sample size requirement than the requirement for the Cochran-Mantel-Haenszel test for $q \times 2 \times 2$ tables, in that each stratum sample size (not just the overall sample size) must be relatively large. Even when the Breslow-Day test is valid, it may not be very powerful against certain alternatives, as discussed in Breslow and Day (1980).

If you specify the BDT option, PROC FREQ computes the Breslow-Day test with Tarone's adjustment, which subtracts an adjustment factor from Q_{BD} to make the resulting statistic asymptotically chi-square.

$$Q_{\mathrm{BDT}} = Q_{\mathrm{BD}} - \frac{(\sum_h (n_{h11} - E(n_{h11} \mid \mathrm{OR}_{\mathrm{MH}})))^2}{\sum_h var(n_{h11} \mid \mathrm{OR}_{\mathrm{MH}})}$$

Refer to Tarone (1985), Jones et al. (1989), and Breslow (1996).

Exact Statistics

Exact statistics can be useful in situations where the asymptotic assumptions are not met, and so the asymptotic *p*-values are not close approximations for the true *p*-values. Standard asymptotic methods involve the assumption that the test statistic follows a particular distribution when the sample size is sufficiently large. When the sample size is not large, asymptotic results may not be valid, with the asymptotic *p*-values differing perhaps substantially from the exact *p*-values. Asymptotic results may also be unreliable when the distribution of the data is sparse, skewed, or heavily tied. Refer to Agresti (1996) and Bishop, Fienberg, and Holland (1975). Exact computations are based on the statistical theory of exact conditional inference for contingency tables, reviewed by Agresti (1992).

In addition to computation of exact *p*-values, PROC FREQ provides the option of estimating exact *p*-values by Monte Carlo simulation. This can be useful for problems that are so large that exact computations require a great amount of time and memory, but for which asymptotic approximations may not be sufficient.

PROC FREQ provides exact *p*-values for the following tests for two-way tables: Pearson chi-square, likelihood-ratio chi-square, Mantel-Haenszel chi-square, Fisher's exact test, Jonckheere-Terpstra test, Cochran-Armitage test for trend, and McNemar's test. PROC FREQ also computes exact *p*-values for tests of hypotheses that the following statistics equal zero: Pearson correlation coefficient, Spearman correlation coefficient, simple kappa coefficient, and weighted kappa coefficient. Additionally, PROC FREQ computes exact confidence limits for the odds ratio for 2×2 tables. For stratified 2×2 tables, PROC FREQ computes exact confidence limits for the common odds ratio, as well as an exact test that the common odds ratio equals one. For one-way frequency tables, PROC FREQ provides the exact chi-square goodness-of-fit test (for equal proportions or for proportions or frequencies that you specify). Also for one-way tables, PROC FREQ provides exact confidence limits for the binomial proportion and an exact test for the binomial proportion value.

The following sections summarize the exact computational algorithms, define the exact *p*-values that PROC FREQ computes, discuss the computational resource requirements, and describe the Monte Carlo estimation option.

Computational Algorithms

PROC FREQ computes exact *p*-values for general $R \times C$ tables using the network algorithm developed by Mehta and Patel (1983). This algorithm provides a substantial advantage over direct enumeration, which can be very time-consuming and feasible only for small problems. Refer to Agresti (1992) for a review of algorithms for computation of exact *p*-values, and refer to Mehta, Patel, and Tsiatis (1984) and Mehta, Patel, and Senchaudhuri (1991) for information on the performance of the network algorithm.

The reference set for a given contingency table is the set of all contingency tables with the observed marginal row and column sums. Corresponding to this reference set, the network algorithm forms a directed acyclic network consisting of nodes in a number of stages. A path through the network corresponds to a distinct table in the reference set. The distances between nodes are defined so that the total distance of a path through the network is the corresponding value of the test statistic. At each node, the algorithm computes the shortest and longest path distances for all the paths that pass through that node. For statistics that can be expressed as a linear combination of cell frequencies multiplied by increasing row and column scores, PROC FREQ computes shortest and longest path distances using the algorithm given in Agresti, Mehta, and Patel (1990). For statistics of other forms, PROC FREQ computes an upper bound for the longest path and a lower bound for the shortest path, following the approach of Valz and Thompson (1994).

The longest and shortest path distances or bounds for a node are compared to the value of the test statistic to determine whether all paths through the node contribute to the *p*-value, none of the paths through the node contribute to the *p*-value, or neither of these situations occur. If all paths through the node contribute, the *p*-value is incremented accordingly, and these paths are eliminated from further analysis. If no paths contribute, these paths are eliminated from the analysis. Otherwise, the algorithm continues, still processing this node and the associated paths. The algorithm finishes when all nodes have been accounted for, incrementing the *p*-value accordingly, or eliminated.

In applying the network algorithm, PROC FREQ uses full precision to represent all statistics, row and column scores, and other quantities involved in the computations. Although it is possible to use rounding to improve the speed and memory requirements of the algorithm, PROC FREQ does not do this since it can result in reduced accuracy of the *p*-values.

For one-way tables, PROC FREQ computes the exact chi-square goodness-of-fit test by the method of Radlow and Alf (1975). PROC FREQ generates all possible one-way tables with the observed total sample size and number of categories. For each possible table, PROC FREQ compares its chi-square value with the value for the observed table. If the table's chi-square value is greater than or equal to the observed chi-square, PROC FREQ increments the exact *p*-value by the probability of that table, which is calculated under the null hypothesis using the multinomial frequency distribution. By default, the null hypothesis states that all categories have equal proportions. If you specify null hypothesis proportions or frequencies using the TESTP= or TESTF= option in the TABLES statement, then PROC FREQ calculates the exact chi-square test based on that null hypothesis.

For binomial proportions in one-way tables, PROC FREQ computes exact confidence limits using the F distribution method given in Collett (1991) and also described by Leemis and Trivedi (1996). PROC FREQ computes the exact test for a binomial proportion ($H_0 \colon p = p_0$) by summing binomial probabilities over all alternatives. See the section "Binomial Proportion" on page 1484 for details. By default, PROC FREQ uses $p_0 = 0.5$ as the null hypothesis proportion. Alternatively, you can specify the null hypothesis proportion with the P= option in the TABLES statement.

See the section "Odds Ratio and Relative Risks for 2 x 2 Tables" on page 1488 for details on computation of exact confidence limits for the odds ratio for 2×2 tables. See the section "***Exact Confidence Limits for the Common Odds Ratio***" on page 1504 for details on computation of exact confidence limits for the common odds ratio for stratified 2×2 tables.

Definition of *p*-Values

For several tests in PROC FREQ, the test statistic is nonnegative, and large values of the test statistic indicate a departure from the null hypothesis. Such tests include the Pearson chi-square, the likelihood-ratio chi-square, the Mantel-Haenszel chi-square, Fisher's exact test for tables larger than 2×2 tables, McNemar's test, and the one-way chi-square goodness-of-fit test. The exact *p*-value for these nondirectional tests is the sum of probabilities for those tables having a test statistic greater than or equal to the value of the observed test statistic.

There are other tests where it may be appropriate to test against either a one-sided or a two-sided alternative hypothesis. For example, when you test the null hypothesis that the true parameter value equals 0 ($T = 0$), the alternative of interest may be one-sided ($T \leq 0$, or $T \geq 0$) or two-sided ($T \neq 0$). Such tests include the Pearson correlation coefficient, Spearman correlation coefficient, Jonckheere-Terpstra test, Cochran-Armitage test for trend, simple kappa coefficient, and weighted kappa coefficient. For these tests, PROC FREQ outputs the right-sided *p*-value when the observed value of the test statistic is greater than its expected value. The right-sided *p*-value is the sum of probabilities for those tables having a test statistic greater than or equal to the observed test statistic. Otherwise, when the test statistic is less than or equal to its expected value, PROC FREQ outputs the left-sided *p*-value. The left-sided *p*-value is the sum of probabilities for those tables having a test statistic less than or equal to the one observed. The one-sided *p*-value P_1 can be expressed as

$$\begin{aligned} P_1 &= \text{Prob (Test Statistic} \geq t) \qquad \text{if} \;\; t > E_0(T) \\ P_1 &= \text{Prob (Test Statistic} \leq t) \qquad \text{if} \;\; t \leq E_0(T) \end{aligned}$$

where t is the observed value of the test statistic and $E_0(T)$ is the expected value of the test statistic under the null hypothesis. PROC FREQ computes the two-sided *p*-value as the sum of the one-sided *p*-value and the corresponding area in the opposite tail of the distribution of the statistic, equidistant from the expected value. The two-sided *p*-value P_2 can be expressed as

$$P_2 = \text{Prob } (\mid \text{Test Statistic} - E_0(T) \mid \geq \mid t - E_0(T) \mid)$$

If you specify the POINT option in the EXACT statement, PROC FREQ also displays exact point probabilities for the test statistics. The exact point probability is the exact probability that the test statistic equals the observed value.

Computational Resources

PROC FREQ uses relatively fast and efficient algorithms for exact computations. These recently developed algorithms, together with improvements in computer power, make it feasible now to perform exact computations for data sets where previously only asymptotic methods could be applied. Nevertheless, there are still large problems that may require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For large problems, consider whether exact methods are really needed or whether asymptotic methods might give results quite close to the exact results, while requiring much less computer time and memory. When asymptotic methods may not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values, as described in the section “Monte Carlo Estimation” on page 1512.

A formula does not exist that can predict in advance how much time and memory are needed to compute an exact *p*-value for a certain problem. The time and memory required depend on several factors, including which test is being performed, the total sample size, the number of rows and columns, and the specific arrangement of the observations into table cells. Generally, larger problems (in terms of total sample size, number of rows, and number of columns) tend to require more time and memory. Additionally, for a fixed total sample size, time and memory requirements tend to increase as the number of rows and columns increases, since this corresponds to an increase in the number of tables in the reference set. Also for a fixed sample size, time and memory requirements increase as the marginal row and column totals become more homogeneous. Refer to Agresti, Mehta, and Patel (1990) and Gail and Mantel (1977).

At any time while PROC FREQ is computing exact *p*-values, you can terminate the computations by pressing the system interrupt key sequence (refer to the *SAS Companion* for your system) and choosing to stop computations. After you terminate exact computations, PROC FREQ completes all other remaining tasks. The procedure produces the requested output and reports missing values for any exact *p*-values that were not computed by the time of termination.

You can also use the MAXTIME= option in the EXACT statement to limit the amount of time PROC FREQ uses for exact computations. You specify a MAXTIME= value that is the maximum amount of clock time (in seconds) that PROC FREQ can use to compute an exact *p*-value. If PROC FREQ does not finish computing an exact *p*-value within that time, it terminates the computation and completes all other remaining tasks.

Monte Carlo Estimation

If you specify the option MC in the EXACT statement, PROC FREQ computes Monte Carlo estimates of the exact *p*-values instead of directly computing the exact *p*-values. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations may not be sufficient. To describe the precision of each Monte Carlo estimate, PROC FREQ provides the asymptotic standard error and $100(1-\alpha)$% confidence limits. The confidence level α is determined by the ALPHA= option in the EXACT statement, which, by default, equals 0.01, and produces 99% confidence limits. The N=*n* option in the EXACT statement specifies the number of samples that PROC FREQ uses for Monte Carlo estimation; the default is 10000 samples. You can specify a larger value for n to improve the precision of the Monte Carlo estimates. Because larger values of n generate more samples, the computation time increases. Alternatively, you can specify a smaller value of n to reduce the computation time.

To compute a Monte Carlo estimate of an exact *p*-value, PROC FREQ generates a random sample of tables with the same total sample size, row totals, and column totals as the observed table. PROC FREQ uses the algorithm of Agresti, Wackerly, and Boyett (1979), which generates tables in proportion to their hypergeometric probabilities conditional on the marginal frequencies. For each sample table, PROC FREQ computes the value of the test statistic and compares it to the value for the observed table. When estimating a right-sided *p*-value, PROC FREQ counts all sample tables for which the test statistic is greater than or equal to the observed test statistic. Then the *p*-value estimate equals the number of these tables divided by the total number of tables sampled.

$$\begin{aligned}\hat{P}_{\text{MC}} &= \quad M \,/\, N \\ M &= \text{number of samples with } (\text{Test Statistic} \geq t) \\ N &= \text{total number of samples} \\ t &= \text{observed Test Statistic}\end{aligned}$$

PROC FREQ computes left-sided and two-sided *p*-value estimates in a similar manner. For left-sided *p*-values, PROC FREQ evaluates whether the test statistic for each sampled table is less than or equal to the observed test statistic. For two-sided *p*-values, PROC FREQ examines the sample test statistics according to the expression for P_2 given in the section "Asymptotic Tests" on page 1475. The variable M is a binomially distributed variable with N trials and success probability p. It follows that the asymptotic standard error of the Monte Carlo estimate is

$$se(\hat{P}_{\text{MC}}) \quad = \quad \sqrt{\hat{P}_{\text{MC}}(1-\hat{P}_{\text{MC}})/(N-1)}$$

PROC FREQ constructs asymptotic confidence limits for the *p*-values according to

$$\hat{P}_{\text{MC}} \;\pm\; z_{\alpha/2} \;\cdot\; se(\hat{P}_{\text{MC}})$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution, and the confidence level α is determined by the ALPHA= option in the EXACT statement.

When the Monte Carlo estimate $\hat{P}_{\text{MC}}$ equals 0, then PROC FREQ computes the confidence limits for the *p*-value as

$$(\;0,\;\; 1-\alpha^{(1/N)}\;)$$

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 1, then PROC FREQ computes the confidence limits as

$$(\;\alpha^{(1/N)},\;\; 1\;)$$

Computational Resources

For each variable in a table request, PROC FREQ stores all of the levels in memory. If all variables are numeric and not formatted, this requires about 84 bytes for each variable level. When there are character variables or formatted numeric variables, the memory that is required depends on the formatted variable lengths, with longer formatted lengths requiring more memory. The number of levels for each variable is limited only by the largest integer that your operating environment can store.

For any single crosstabulation table requested, PROC FREQ builds the entire table in memory, regardless of whether the table has zero cell counts. Thus, if the numeric variables A, B, and C each have 10 levels, PROC FREQ requires 2520 bytes to store the variable levels for the table request A*B*C, as follows:

```
3 variables * 10 levels/variable * 84 bytes/level
```

In addition, PROC FREQ requires 8000 bytes to store the table cell frequencies

```
1000 cells * 8 bytes/cell
```

even though there may be only 10 observations.

When the variables have many levels or when there are many multiway tables, your computer may not have enough memory to construct the tables. If PROC FREQ runs out of memory while constructing tables, it stops collecting levels for the variable with the most levels and returns the memory that is used by that variable. The procedure then builds the tables that do not contain the disabled variables.

If there is not enough memory for your table request and if increasing the available memory is impractical, you can reduce the number of multiway tables or variable levels. If you are not using the CMH or AGREE option in the TABLES statement to compute statistics across strata, reduce the number of multiway tables by using PROC SORT to sort the data set by one or more of the variables or by using the DATA step to create an index for the variables. Then remove the sorted or indexed variables from the TABLES statement and include a BY statement that uses these variables. You can also reduce memory requirements by using a FORMAT statement in the PROC FREQ step to reduce the number of levels. Additionally, reducing the formatted variable lengths reduces the amount of memory that is needed to store the variable levels. For more information on using formats, see the "Grouping with Formats" section on page 1465.

Output Data Sets

PROC FREQ produces two types of output data sets that you can use with other statistical and reporting procedures. These data sets are produced as follows:

- Specifying a TABLES statement with an OUT= option creates an output data set that contains frequency or crosstabulation table counts and percentages.
- Specifying an OUTPUT statement creates an output data set that contains statistics.

PROC FREQ does not display the output data sets. Use PROC PRINT, PROC REPORT, or any other SAS reporting tool to display an output data set.

Contents of the TABLES Statement Output Data Set

The OUT= option in the TABLES statement creates an output data set that contains one observation for each combination of the variable values (or table cell) in the last table request. By default, each observation contains the frequency and percentage for the table cell. When the input data set contains missing values, the output data set also contains an observation with the frequency of missing values. The output data set includes the following variables:

- BY variables
- table request variables, such as A, B, C, and D in the table request A*B*C*D
- COUNT, a variable containing the cell frequency
- PERCENT, a variable containing the cell percentage

If you specify the OUTEXPECT and OUTPCT options in the TABLES statement, the output data set also contains expected frequencies and row, column, and table percentages, respectively. The additional variables are

- EXPECTED, a variable containing the expected frequency
- PCT_TABL, a variable containing the percentage of two-way table frequency, for *n*-way tables where $n > 2$
- PCT_ROW, a variable containing the percentage of row frequency
- PCT_COL, a variable containing the percentage of column frequency

If you specify the OUTCUM option in the TABLES statement, the output data set also contains cumulative frequencies and cumulative percentages for one-way tables. The additional variables are

- CUM_FREQ, a variable containing the cumulative frequency
- CUM_PCT, a variable containing the cumulative percentage

The OUTCUM option has no effect for two-way or multiway tables.

When you submit the following statements

```
proc freq;
   tables A A*B / out=D;
run;
```

the output data set D contains frequencies and percentages for the last table request, A*B. If A has two levels (1 and 2), B has three levels (1,2, and 3), and no table cell count is zero or missing, the output data set D includes six observations, one for each combination of A and B. The first observation corresponds to A=1 and B=1; the second observation corresponds to A=1 and B=2; and so on. The data set includes the variables COUNT and PERCENT. The value of COUNT is the number of observations with the given combination of A and B values. The value of PERCENT is the percent of the total number of observations having that A and B combination.

When PROC FREQ combines different variable values into the same formatted level, the output data set contains the smallest internal value for the formatted level. For example, suppose a variable X has the values 1.1., 1.4, 1.7, 2.1, and 2.3. When you submit the statement

```
    format X 1.;
```

in a PROC FREQ step, the formatted levels listed in the frequency table for X are 1 and 2. If you create an output data set with the frequency counts, the internal values of X are 1.1 and 1.7. To report the internal values of X when you display the output data set, use a format of 3.1 with X.

Contents of the OUTPUT Statement Output Data Set

The OUTPUT statement creates a SAS data set containing the statistics that PROC FREQ computes for the last table request. You specify which statistics to store in the output data set. There is an observation with the specified statistics for each stratum or two-way table. If PROC FREQ computes summary statistics for a stratified table, the output data set also contains a summary observation with those statistics.

The OUTPUT data set can include the following variables.

- BY variables
- variables that identify the stratum, such as A and B in the table request A*B*C*D
- variables that contain the specified statistics

The output data set also includes variables with the *p*-values and degrees of freedom, asymptotic standard error (ASE), or confidence limits when PROC FREQ computes these values for a specified statistic.

The variable names for the specified statistics in the output data set are the names of the options enclosed in underscores. PROC FREQ forms variable names for the corresponding *p*-values, degrees of freedom, or confidence limits by combining the name of the option with the appropriate prefix from the following list:

DF_	degrees of freedom
E_	asymptotic standard error (ASE)
L_	lower confidence limit
U_	upper confidence limit
E0_	ASE under the null hypothesis
Z_	standardized value
P_	*p*-value
P2_	two-sided *p*-value
PL_	left-sided *p*-value
PR_	right-sided *p*-value
XP_	exact *p*-value
XP2_	exact two-sided *p*-value
XPL_	exact left-sided *p*-value
XPR_	exact right-sided *p*-value
XPT_	exact point probability
XL_	exact lower confidence limit
XR_	exact upper confidence limit

For example, variable names created for the Pearson chi-square, its degrees of freedom, its p-values are _PCHI_, DF_PCHI, and P_PCHI, respectively.

If the length of the prefix plus the statistic option exceeds eight characters, PROC FREQ truncates the option so that the name of the new variable is eight characters long.

Displayed Output

Number of Variable Levels Table

If you specify the NLEVELS option in the PROC FREQ statement, PROC FREQ displays the "Number of Variable Levels" table. This table provides the number of levels for all variables named in the TABLES statements. PROC FREQ determines the variable levels from the formatted variable values. See "Grouping with Formats" for details. The "Number of Variable Levels" table contains the following information:

- Variable name
- Levels, which is the total number of levels of the variable
- Number of Nonmissing Levels, if there are missing levels for any of the variables
- Number of Missing Levels, if there are missing levels for any of the variables

One-Way Frequency Tables

PROC FREQ displays one-way frequency tables for all one-way table requests in the TABLES statements, unless you specify the NOPRINT option in the PROC statement or the NOPRINT option in the TABLES statement. For a one-way table showing the frequency distribution of a single variable, PROC FREQ displays the following information:

- the name of the variable and its values
- Frequency counts, giving the number of observations that have each value
- specified Test Frequency counts, if you specify the CHISQ and TESTF= options to request a chi-square goodness-of-fit test for specified frequencies
- Percent, giving the percentage of the total number of observations with that value. (The NOPERCENT option suppresses this information.)
- specified Test Percents, if you specify the CHISQ and TESTP= options to request a chi-square goodness-of-fit test for specified percents. (The NOPERCENT option suppresses this information.)
- Cumulative Frequency counts, giving the sum of the frequency counts of that value and all other values listed above it in the table. The last cumulative frequency is the total number of nonmissing observations. (The NOCUM option suppresses this information.)
- Cumulative Percent values, giving the percentage of the total number of observations with that value and all others previously listed in the table. (The NOCUM or the NOPERCENT option suppresses this information.)

The one-way table also displays the Frequency Missing, or the number of observations with missing values.

Statistics for One-Way Frequency Tables

For one-way tables, two statistical options are available in the TABLES statement. The CHISQ option provides a chi-square goodness-of-fit test, and the BINOMIAL option provides binomial proportion statistics. PROC FREQ displays the following information, unless you specify the NOPRINT option in the PROC statement:

- If you specify the CHISQ option for a one-way table, PROC FREQ provides a chi-square goodness-of-fit test, displaying the Chi-Square statistic, the degrees of freedom (DF), and the probability value (Pr > ChiSq). If you specify the CHISQ option in the EXACT statement, PROC FREQ also displays the exact probability value for this test. If you specify the POINT option with the CHISQ option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.
- If you specify the BINOMIAL option for a one-way table, PROC FREQ displays the estimate of the binomial Proportion, which is the proportion of observations in the first class listed in the one-way table. PROC FREQ also displays the asymptotic standard error (ASE) and the asymptotic and exact confidence limits for this estimate. For the binomial proportion test, PROC FREQ displays the asymptotic standard error under the null hypothesis (ASE Under H0), the standardized test statistic (Z), and the one-sided and two-sided probability values. If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test. If you specify the POINT option with the BINOMIAL option in the EXACT statement, PROC FREQ displays the exact point probability for the test.

Multiway Tables

PROC FREQ displays all multiway table requests in the TABLES statements, unless you specify the NOPRINT option in the PROC statement or the NOPRINT option in the TABLES statement.

For two-way to multiway crosstabulation tables, the values of the last variable in the table request form the table columns. The values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one stratum.

There are three ways to display multiway tables in PROC FREQ. By default, PROC FREQ displays multiway tables as separate two-way crosstabulation tables for each stratum of the multiway table. Also by default, PROC FREQ displays these two-way crosstabulation tables in table cell format. Alternatively, if you specify the CROSSLIST option, PROC FREQ displays the two-way crosstabulation tables in ODS column format. If you specify the LIST option, PROC FREQ displays multiway tables in list format.

Crosstabulation Tables

By default, PROC FREQ displays two-way crosstabulation tables in table cell format. The row variable values are listed down the side of the table, the column variable values are listed across the top of the table, and each row and column variable level combination forms a table cell.

Each cell of a crosstabulation table may contain the following information:

- Frequency, giving the number of observations that have the indicated values of the two variables. (The NOFREQ option suppresses this information.)
- the Expected cell frequency under the hypothesis of independence, if you specify the EXPECTED option
- the Deviation of the cell frequency from the expected value, if you specify the DEVIATION option
- Cell Chi-Square, which is the cell's contribution to the total chi-square statistic, if you specify the CELLCHI2 option
- Tot Pct, or the cell's percentage of the total frequency, for *n*-way tables when $n > 2$, if you specify the TOTPCT option
- Percent, the cell's percentage of the total frequency. (The NOPERCENT option suppresses this information.)
- Row Pct, or the row percentage, the cell's percentage of the total frequency count for that cell's row. (The NOROW option suppresses this information.)
- Col Pct, or column percentage, the cell's percentage of the total frequency count for that cell's column. (The NOCOL option suppresses this information.)
- Cumulative Col%, or cumulative column percent, if you specify the CUMCOL option

The table also displays the Frequency Missing, or the number of observations with missing values.

CROSSLIST Tables

If you specify the CROSSLIST option, PROC FREQ displays two-way crosstabulation tables with ODS column format. Using column format, a CROSSLIST table provides the same information (frequencies, percentages, and other statistics) as the default crosstabulation table with cell format. But unlike the default crosstabulation table, a CROSSLIST table has a table definition that you can customize with PROC TEMPLATE. For more information, refer to the chapter titled "The TEMPLATE Procedure" in the *SAS Output Delivery System User's Guide*.

In the CROSSLIST table format, the rows of the display correspond to the crosstabulation table cells, and the columns of the display correspond to descriptive statistics such as frequencies and percentages. Each table cell is identified by the values of its TABLES row and column variable levels, with all column variable levels listed within each row variable level. The CROSSLIST table also provides row totals, column totals, and overall table totals.

For a crosstabulation table in the CROSSLIST format, PROC FREQ displays the following information:

- the row variable name and values
- the column variable name and values
- Frequency, giving the number of observations that have the indicated values of the two variables. (The NOFREQ option suppresses this information.)
- the Expected cell frequency under the hypothesis of independence, if you specify the EXPECTED option
- the Deviation of the cell frequency from the expected value, if you specify the DEVIATION option
- Cell Chi-Square, which is the cell's contribution to the total chi-square statistic, if you specify the CELLCHI2 option
- Total Percent, or the cell's percentage of the total frequency, for *n*-way tables when $n > 2$, if you specify the TOTPCT option
- Percent, the cell's percentage of the total frequency. (The NOPERCENT option suppresses this information.)
- Row Percent, the cell's percentage of the total frequency count for that cell's row. (The NOROW option suppresses this information.)
- Column Percent, the cell's percentage of the total frequency count for that cell's column. (The NOCOL option suppresses this information.)

The table also displays the Frequency Missing, or the number of observations with missing values.

LIST Tables

If you specify the LIST option in the TABLES statement, PROC FREQ displays multiway tables in a list format rather than as crosstabulation tables. The LIST option displays the entire multiway table in one table, instead of displaying a separate two-way table for each stratum. The LIST option is not available when you also request statistical options. Unlike the default crosstabulation output, the LIST output does not display row percentages, column percentages, and optional information such as expected frequencies and cell chi-squares.

For a multiway table in list format, PROC FREQ displays the following information:

- the variable names and values
- Frequency counts, giving the number of observations with the indicated combination of variable values
- Percent, the cell's percentage of the total number of observations. (The NOPERCENT option suppresses this information.)
- Cumulative Frequency counts, giving the sum of the frequency counts of that cell and all other cells listed above it in the table. The last cumulative frequency is the total number of nonmissing observations. (The NOCUM option suppresses this information.)

- Cumulative Percent values, giving the percentage of the total number of observations for that cell and all others previously listed in the table. (The NOCUM or the NOPERCENT option suppresses this information.)

The table also displays the Frequency Missing, or the number of observations with missing values.

Statistics for Multiway Tables

PROC FREQ computes statistical tests and measures for crosstabulation tables, depending on which statements and options you specify. You can suppress the display of all these results by specifying the NOPRINT option in the PROC statement. With any of the following information, PROC FREQ also displays the Sample Size and the Frequency Missing.

- If you specify the SCOROUT option, PROC FREQ displays the Row Scores and Column Scores that it uses for statistical computations. The Row Scores table displays the row variable values and the Score corresponding to each value. The Column Scores table displays the column variable values and the corresponding Scores. PROC FREQ also identifies the score type used to compute the row and column scores. You can specify the score type with the SCORES= option in the TABLES statement.
- If you specify the CHISQ option, PROC FREQ displays the following statistics for each two-way table: Pearson Chi-Square, Likelihood-Ratio Chi-Square, Continuity-Adjusted Chi-Square (for 2×2 tables), Mantel-Haenszel Chi-Square, the Phi Coefficient, the Contingency Coefficient, and Cramer's V. For each test statistic, PROC FREQ also displays the degrees of freedom (DF) and the probability value (Prob).
- If you specify the CHISQ option for 2×2 tables, PROC FREQ also displays Fisher's exact test. The test output includes the cell (1,1) frequency (F), the exact left-sided and right-sided probability values, the table probability (P), and the exact two-sided probability value.
- If you specify the FISHER option in the TABLES statement (or, equivalently, the FISHER option in the EXACT statement), PROC FREQ displays Fisher's exact test for tables larger than 2×2. The test output includes the table probability (P) and the probability value. In addition, PROC FREQ displays the CHISQ output listed earlier, even if you do not also specify the CHISQ option.
- If you specify the PCHI, LRCHI, or MHCHI option in the EXACT statement, PROC FREQ also displays the corresponding exact test: Pearson Chi-Square, Likelihood-Ratio Chi-Square, or Mantel-Haenszel Chi-Square, respectively. The test output includes the test statistic, the degrees of freedom (DF), and the asymptotic and exact probability values. If you also specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested. If you specify the CHISQ option in the EXACT statement, PROC FREQ displays exact probability values for all three of these chi-square tests.

- If you specify the MEASURES option, PROC FREQ displays the following statistics and their asymptotic standard errors (ASE) for each two-way table: Gamma, Kendall's Tau-*b*, Stuart's Tau-*c*, Somers' $D(C|R)$, Somers' $D(R|C)$, Pearson Correlation, Spearman Correlation, Lambda Asymmetric $(C|R)$, Lambda Asymmetric $(R|C)$, Lambda Symmetric, Uncertainty Coefficient $(C|R)$, Uncertainty Coefficient $(R|C)$, and Uncertainty Coefficient Symmetric. If you specify the CL option, PROC FREQ also displays confidence limits for these measures.
- If you specify the PLCORR option, PROC FREQ displays the tetrachoric correlation for 2×2 tables or the polychoric correlation for larger tables. In addition, PROC FREQ displays the MEASURES output listed earlier, even if you do not also specify the MEASURES option.
- If you specify the option GAMMA, KENTB, STUTC, SMDCR, SMDRC, PCORR, or SCORR in the TEST statement, PROC FREQ displays asymptotic tests for Gamma, Kendall's Tau-*b*, Stuart's Tau-*c*, Somers' $D(C|R)$, Somers' $D(R|C)$, the Pearson Correlation, or the Spearman Correlation, respectively. If you specify the MEASURES option in the TEST statement, PROC FREQ displays all these asymptotic tests. The test output includes the statistic, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the one-sided and two-sided probability values.
- If you specify the PCORR or SCORR option in the EXACT statement, PROC FREQ displays asymptotic and exact tests for the Pearson Correlation or the Spearman Correlation, respectively. The test output includes the correlation, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the asymptotic and exact one-sided and two-sided probability values. If you also specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested.
- If you specify the RISKDIFF option for 2×2 tables, PROC FREQ displays the Column 1 and Column 2 Risk Estimates. For each column, PROC FREQ displays Row 1 Risk, Row 2 Risk, Total Risk, and Risk Difference, together with their asymptotic standard errors (ASE), Asymptotic Confidence Limits, and Exact Confidence Limits. Exact confidence limits are not available for the risk difference.
- If you specify the MEASURES option or the RELRISK option for 2×2 tables, PROC FREQ displays Estimates of the Relative Risk for Case-Control and Cohort studies, together with their Confidence Limits. These measures are also known as the Odds Ratio and the Column 1 and 2 Relative Risks. If you specify the OR option in the EXACT statement, PROC FREQ also displays Exact Confidence Limits for the Odds Ratio.
- If you specify the TREND option, PROC FREQ displays the Cochran-Armitage Trend Test for tables that are $2 \times C$ or $R \times 2$. For this test, PROC FREQ gives the Statistic (Z) and the one-sided and two-sided probability values. If you specify the TREND option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test.

If you specify the POINT option with the TREND option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.

- If you specify the JT option, PROC FREQ displays the Jonckheere-Terpstra Test, showing the Statistic (JT), the standardized test statistic (Z), and the one-sided and two-sided probability values. If you specify the JT option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test. If you specify the POINT option with the JT option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.
- If you specify the AGREE option and the PRINTKWT option, PROC FREQ displays the Kappa Coefficient Weights for square tables greater than 2×2.
- If you specify the AGREE option, for two-way tables PROC FREQ displays McNemar's Test and the Simple Kappa Coefficient for 2×2 tables. For square tables larger than 2×2, PROC FREQ displays Bowker's Test of Symmetry, the Simple Kappa Coefficient, and the Weighted Kappa Coefficient. For McNemar's Test and Bowker's Test of Symmetry, PROC FREQ displays the Statistic (S), the degrees of freedom (DF), and the probability value (Pr > S). If you specify the MCNEM option in the EXACT statement, PROC FREQ also displays the exact probability value for McNemar's test. If you specify the POINT option with the MCNEM option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic. For the simple and weighted kappa coefficients, PROC FREQ displays the kappa values, asymptotic standard errors (ASE), and Confidence Limits.
- If you specify the KAPPA or WTKAP option in the TEST statement, PROC FREQ displays asymptotic tests for the simple kappa coefficient or the weighted kappa coefficient, respectively. If you specify the AGREE option in the TEST statement, PROC FREQ displays both these asymptotic tests. The test output includes the kappa coefficient, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the one-sided and two-sided probability values.
- If you specify the KAPPA or WTKAP option in the EXACT statement, PROC FREQ displays asymptotic and exact tests for the simple kappa coefficient or the weighted kappa coefficient, respectively. The test output includes the kappa coefficient, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the asymptotic and exact one-sided and two-sided probability values. If you specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested.
- If you specify the MC option in the EXACT statement, PROC FREQ displays Monte Carlo estimates for all exact *p*-values requested by *statistic-options* in the EXACT statement. The Monte Carlo output includes the *p*-value Estimate, its Confidence Limits, the Number of Samples used to compute the Monte Carlo estimate, and the Initial Seed for random number generation.
- If you specify the AGREE option, for multiple strata PROC FREQ displays Overall Simple and Weighted Kappa Coefficients, with their asymptotic standard errors (ASE) and Confidence Limits. PROC FREQ also displays Tests for

Equal Kappa Coefficients, giving the Chi-Squares, degrees of freedom (DF), and probability values (Pr > ChiSq) for the Simple Kappa and Weighted Kappa. For multiple strata of 2 × 2 tables, PROC FREQ displays Cochran's Q, giving the Statistic (Q), the degrees of freedom (DF), and the probability value (Pr > Q).

- If you specify the CMH option, PROC FREQ displays Cochran-Mantel-Haenszel Statistics for the following three alternative hypotheses: Nonzero Correlation, Row Mean Scores Differ (ANOVA Statistic), and General Association. For each of these statistics, PROC FREQ gives the degrees of freedom (DF) and the probability value (Prob). For 2 × 2 tables, PROC FREQ also displays Estimates of the Common Relative Risk for Case-Control and Cohort studies, together with their confidence limits. These include both Mantel-Haenszel and Logit stratum-adjusted estimates of the common Odds Ratio, Column 1 Relative Risk, and Column 2 Relative Risk. Also for 2 × 2 tables, PROC FREQ displays the Breslow-Day Test for Homogeneity of the Odds Ratios. For this test, PROC FREQ gives the Chi-Square, the degrees of freedom (DF), and the probability value (Pr > ChiSq).
- If you specify the CMH option in the TABLES statement and also specify the COMOR option in the EXACT statement, PROC FREQ displays exact confidence limits for the Common Odds Ratio for multiple strata of 2 × 2 tables. PROC FREQ also displays the Exact Test of H0: Common Odds Ratio = 1. The test output includes the Cell (1,1) Sum (S), Mean of S Under H0, One-sided Pr <= S, and Point Pr = S. PROC FREQ also provides exact two-sided probability values for the test, computed according to the following three methods: 2 * One-sided, Sum of probabilities <= Point probability, and Pr >= |S - Mean|.

ODS Table Names

PROC FREQ assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 29.11 lists the ODS table names together with their descriptions and the options required to produce the tables. Note that the ALL option in the TABLES statement invokes the CHISQ, MEASURES, and CMH options.

Table 29.11. ODS Tables Produced in PROC FREQ

ODS Table Name	Description	Statement	Option
BinomialProp	Binomial proportion	TABLES	BINOMIAL (one-way tables)
BinomialPropTest	Binomial proportion test	TABLES	BINOMIAL (one-way tables)
BreslowDayTest	Breslow-Day test	TABLES	CMH ($h \times 2 \times 2$ tables)
CMH	Cochran-Mantel-Haenszel statistics	TABLES	CMH
ChiSq	Chi-square tests	TABLES	CHISQ
CochransQ	Cochran's Q	TABLES	AGREE ($h \times 2 \times 2$ tables)
ColScores	Column scores	TABLES	SCOROUT
CommonOddsRatioCL	Exact confidence limits for the common odds ratio	EXACT	COMOR
CommonOddsRatioTest	Common odds ratio exact test	EXACT	COMOR
CommonRelRisks	Common relative risks	TABLES	CMH ($h \times 2 \times 2$ tables)
CrossList	Column format crosstabulation table	TABLES	CROSSLIST (n-way table request, $n > 1$)
CrossTabFreqs	Crosstabulation table	TABLES	(n-way table request, $n > 1$)
EqualKappaTest	Test for equal simple kappas	TABLES	AGREE ($h \times 2 \times 2$ tables)
EqualKappaTests	Tests for equal kappas	TABLES	AGREE ($h{\times}r{\times}r$ tables, $r > 2$)
FishersExact	Fisher's exact test	EXACT or TABLES or TABLES	FISHER FISHER or EXACT CHISQ (2×2 tables)
FishersExactMC	Monte Carlo estimates for Fisher's exact test	EXACT	FISHER / MC
Gamma	Gamma	TEST	GAMMA
GammaTest	Gamma test	TEST	GAMMA
JTTest	Jonckheere-Terpstra test	TABLES	JT
JTTestMC	Monte Carlo estimates for the JT exact test	EXACT	JT / MC
KappaStatistics	Kappa statistics	TABLES	AGREE ($r \times r$ tables, $r > 2$, and no TEST or EXACT KAPPA)
KappaWeights	Kappa weights	TABLES	AGREE and PRINTKWT
List	List format multiway table	TABLES	LIST
LRChiSq	Likelihood-ratio chi-square exact test	EXACT	LRCHI
LRChiSqMC	Monte Carlo exact test for likelihood-ratio chi-square	EXACT	LRCHI / MC
McNemarsTest	McNemar's test	TABLES	AGREE (2×2 tables)
Measures	Measures of association	TABLES	MEASURES
MHChiSq	Mantel-Haenszel chi-square exact test	EXACT	MHCHI
MHChiSqMC	Monte Carlo exact test for Mantel-Haenszel chi-square	EXACT	MHCHI / MC
NLevels	Number of variable levels	PROC	NLEVELS
OddsRatioCL	Exact confidence limits for the odds ratio	EXACT	OR
OneWayChiSq	One-way chi-square test	TABLES	CHISQ (one-way tables)

Table 29.11. (continued)

ODS Table Name	Description	Statement	Option
OneWayChiSqMC	Monte Carlo exact test for one-way chi-square	EXACT	CHISQ / MC (one-way tables)
OneWayFreqs	One-way frequencies	PROC or TABLES	(with no TABLES stmt) (one-way table request)
OverallKappa	Overall simple kappa	TABLES	AGREE ($h \times 2 \times 2$ tables)
OverallKappas	Overall kappa coefficients	TABLES	AGREE ($h \times r \times r$ tables, $r > 2$)
PearsonChiSq	Pearson chi-square exact test	EXACT	PCHI
PearsonChiSqMC	Monte Carlo exact test for Pearson chi-square	EXACT	PCHI / MC
PearsonCorr	Pearson correlation	TEST or EXACT	PCORR PCORR
PearsonCorrMC	Monte Carlo exact test for Pearson correlation	EXACT	PCORR / MC
PearsonCorrTest	Pearson correlation test	TEST or EXACT	PCORR PCORR
RelativeRisks	Relative risk estimates	TABLES	RELRISK or MEASURES (2×2 tables)
RiskDiffCol1	Column 1 risk estimates	TABLES	RISKDIFF (2×2 tables)
RiskDiffCol2	Column 2 risk estimates	TABLES	RISKDIFF (2×2 tables)
RowScores	Row scores	TABLES	SCOROUT
SimpleKappa	Simple kappa coefficient	TEST or EXACT	KAPPA KAPPA
SimpleKappaMC	Monte Carlo exact test for simple kappa	EXACT	KAPPA / MC
SimpleKappaTest	Simple kappa test	TEST or EXACT	KAPPA KAPPA
SomersDCR	Somers' $D(C\|R)$	TEST	SMDCR
SomersDCRTest	Somers' $D(C\|R)$ test	TEST	SMDCR
SomersDRC	Somers' $D(R\|C)$	TEST	SMDRC
SomersDRCTest	Somers' $D(R\|C)$ test	TEST	SMDRC
SpearmanCorr	Spearman correlation	TEST or EXACT	SCORR SCORR
SpearmanCorrMC	Monte Carlo exact test for Spearman correlation	EXACT	SCORR / MC
SpearmanCorrTest	Spearman correlation test	TEST or EXACT	SCORR SCORR
SymmetryTest	Test of symmetry	TABLES	AGREE
TauB	Kendall's tau-b	TEST	KENTB
TauBTest	Kendall's tau-b test	TEST	KENTB
TauC	Stuart's tau-c	TEST	STUTC
TauCTest	Stuart's tau-c test	TEST	STUTC
TrendTest	Cochran-Armitage test for trend	TABLES	TREND

Table 29.11. (continued)

ODS Table Name	Description	Statement	Option
TrendTestMC	Monte Carlo exact test for trend	EXACT	TREND / MC
WeightedKappa	Weighted kappa	TEST or EXACT	WTKAP WTKAP
WeightedKappaMC	Monte Carlo exact test for weighted kappa	EXACT	WTKAP / MC
WeightedKappaTest	Weighted kappa test	TEST or EXACT	WTKAP WTKAP

Examples

Example 29.1. Creating an Output Data Set with Table Cell Frequencies

The eye and hair color of children from two different regions of Europe are recorded in the data set Color. Instead of recording one observation per child, the data are recorded as cell counts, where the variable Count contains the number of children exhibiting each of the 15 eye and hair color combinations. The data set does not include missing combinations.

```
data Color;
   input Region Eyes $ Hair $ Count @@;
      label Eyes  ='Eye Color'
            Hair  ='Hair Color'
            Region='Geographic Region';
      datalines;
1 blue  fair   23  1 blue  red     7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red     7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red     5  1 brown medium 41  1 brown dark   40
1 brown black   3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black   6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

The following statements read the Color data set and create an output data set containing the frequencies, percentages, and expected cell frequencies of the Eyes by Hair two-way table. The TABLES statement requests three tables: Eyes and Hair frequency tables and an Eyes by Hair crosstabulation table. The OUT= option creates the FreqCnt data set, which contains the crosstabulation table frequencies. The OUTEXPECT option outputs the expected cell frequencies to FreqCnt, and the SPARSE option includes the zero cell counts. The WEIGHT statement specifies that Count contains the observation weights. These statements create Output 29.1.1 through Output 29.1.3.

```
proc freq data=Color;
   weight Count;
   tables Eyes Hair Eyes*Hair / out=FreqCnt outexpect sparse;
   title1 'Eye and Hair Color of European Children';
run;
proc print data=FreqCnt noobs;
   title2 'Output Data Set from PROC FREQ';
run;
```

Output 29.1.1 displays the two frequency tables produced, one showing the distribution of eye color, and one showing the distribution of hair color. By default, PROC FREQ lists the variables values in alphabetical order. The 'Eyes*Hair' specification produces a crosstabulation table, shown in Output 29.1.2, with eye color defining the table rows and hair color defining the table columns. A zero cell count for green eyes and black hair indicates that this eye and hair color combination does not occur in the data.

The output data set (Output 29.1.3) contains frequency counts and percentages for the last table. The data set also includes an observation for the zero cell count (SPARSE) and a variable with the expected cell frequency for each table cell (OUTEXPECT).

Output 29.1.1. Frequency Tables

```
                   Eye and Hair Color of European Children

                              The FREQ Procedure

                                  Eye Color

                                              Cumulative    Cumulative
        Eyes     Frequency      Percent        Frequency      Percent
        ----------------------------------------------------------
        blue          222        29.13            222          29.13
        brown         341        44.75            563          73.88
        green         199        26.12            762         100.00


                                 Hair Color

                                              Cumulative    Cumulative
       Hair      Frequency      Percent        Frequency      Percent
       -----------------------------------------------------------
       black           22         2.89             22           2.89
       dark           182        23.88            204          26.77
       fair           228        29.92            432          56.69
       medium         217        28.48            649          85.17
       red            113        14.83            762         100.00
```

Output 29.1.2. Crosstabulation Table

```
                   Eye and Hair Color of European Children

                            Table of Eyes by Hair

          Eyes(Eye Color)     Hair(Hair Color)

          Frequency|
          Percent  |
          Row Pct  |
          Col Pct  |black   |dark    |fair    |medium  |red     |  Total
          ---------+--------+--------+--------+--------+--------+
          blue     |      6 |     51 |     69 |     68 |     28 |    222
                   |   0.79 |   6.69 |   9.06 |   8.92 |   3.67 |  29.13
                   |   2.70 |  22.97 |  31.08 |  30.63 |  12.61 |
                   |  27.27 |  28.02 |  30.26 |  31.34 |  24.78 |
          ---------+--------+--------+--------+--------+--------+
          brown    |     16 |     94 |     90 |     94 |     47 |    341
                   |   2.10 |  12.34 |  11.81 |  12.34 |   6.17 |  44.75
                   |   4.69 |  27.57 |  26.39 |  27.57 |  13.78 |
                   |  72.73 |  51.65 |  39.47 |  43.32 |  41.59 |
          ---------+--------+--------+--------+--------+--------+
          green    |      0 |     37 |     69 |     55 |     38 |    199
                   |   0.00 |   4.86 |   9.06 |   7.22 |   4.99 |  26.12
                   |   0.00 |  18.59 |  34.67 |  27.64 |  19.10 |
                   |   0.00 |  20.33 |  30.26 |  25.35 |  33.63 |
          ---------+--------+--------+--------+--------+--------+
          Total          22      182      228      217      113      762
                       2.89    23.88    29.92    28.48    14.83   100.00
```

Output 29.1.3. OUT= Data Set

```
                        Output Data Set from PROC FREQ

                 Eyes     Hair      COUNT    EXPECTED    PERCENT

                 blue     black        6        6.409     0.7874
                 blue     dark        51       53.024     6.6929
                 blue     fair        69       66.425     9.0551
                 blue     medium      68       63.220     8.9239
                 blue     red         28       32.921     3.6745
                 brown    black       16        9.845     2.0997
                 brown    dark        94       81.446    12.3360
                 brown    fair        90      102.031    11.8110
                 brown    medium      94       97.109    12.3360
                 brown    red         47       50.568     6.1680
                 green    black        0        5.745     0.0000
                 green    dark        37       47.530     4.8556
                 green    fair        69       59.543     9.0551
                 green    medium      55       56.671     7.2178
                 green    red         38       29.510     4.9869
```

Example 29.2. Computing Chi-Square Tests for One-Way Frequency Tables

This example examines whether the children's hair color (from Example 29.1 on page 1527) has a specified multinomial distribution for the two regions. The hypothesized distribution for hair color is 30% fair, 12% red, 30% medium, 25% dark, and 3% black.

In order to test the hypothesis for each region, the data are first sorted by Region. Then the FREQ procedure uses a BY statement to produce a separate table for each BY group (Region). The option ORDER=DATA orders the frequency table values (hair color) by their order in the data set. The TABLES statement requests a frequency table for hair color, and the option NOCUM suppresses the display of the cumulative frequencies and percentages. The TESTP= option specifies the hypothesized percentages for the chi-square test; the number of percentages specified equals the number of table levels, and the percentages sum to 100. The following statements produce Output 29.2.1.

```
proc sort data=Color;
   by Region;
run;
proc freq data=Color order=data;
   weight Count;
   tables Hair / nocum testp=(30 12 30 25 3);
   by Region;
   title 'Hair Color of European Children';
run;
```

The frequency tables in Output 29.2.1 list the variable values (hair color) in the order in which they appear in the data set. The “Test Percent” column lists the hypothesized percentages for the chi-square test. Always check that you have ordered the TESTP= percentages to correctly match the order of the variable levels.

PROC FREQ computes a chi-square statistic for each region. The chi-square statistic is significant at the 0.05 level for Region 2 (p=0.0003) but not for Region 1. This indicates a significant departure from the hypothesized percentages in Region 2.

Output 29.2.1. One-Way Frequency Table with BY Groups

```
                        Hair Color of European Children

----------------------------- Geographic Region=1 ------------------------------

                               The FREQ Procedure

                                  Hair Color

                                                        Test
                  Hair      Frequency     Percent     Percent
                  -------------------------------------------
                  fair            76       30.89       30.00
                  red             19        7.72       12.00
                  medium          83       33.74       30.00
                  dark            65       26.42       25.00
                  black            3        1.22        3.00


                                Chi-Square Test
                           for Specified Proportions
                           -------------------------
                           Chi-Square         7.7602
                           DF                      4
                           Pr > ChiSq         0.1008

                               Sample Size = 246
```

```
                        Hair Color of European Children

----------------------------- Geographic Region=2 ------------------------------

                                  Hair Color

                                                        Test
                  Hair      Frequency     Percent     Percent
                  -------------------------------------------
                  fair           152       29.46       30.00
                  red             94       18.22       12.00
                  medium         134       25.97       30.00
                  dark           117       22.67       25.00
                  black           19        3.68        3.00


                                Chi-Square Test
                           for Specified Proportions
                           -------------------------
                           Chi-Square        21.3824
                           DF                      4
                           Pr > ChiSq         0.0003

                               Sample Size = 516
```

Example 29.3. Computing Binomial Proportions for One-Way Frequency Tables

The binomial proportion is computed as the proportion of observations for the first level of the variable that you are studying. The following statements compute the proportion of children with brown eyes (from the data set in Example 29.1 on page 1527) and test this value against the hypothesis that the proportion is 50%. Also, these statements test whether the proportion of children with fair hair is 28%.

```
proc freq data=Color order=freq;
   weight Count;
   tables Eyes / binomial alpha=.1;
   tables Hair / binomial(p=.28);
   title 'Hair and Eye Color of European Children';
run;
```

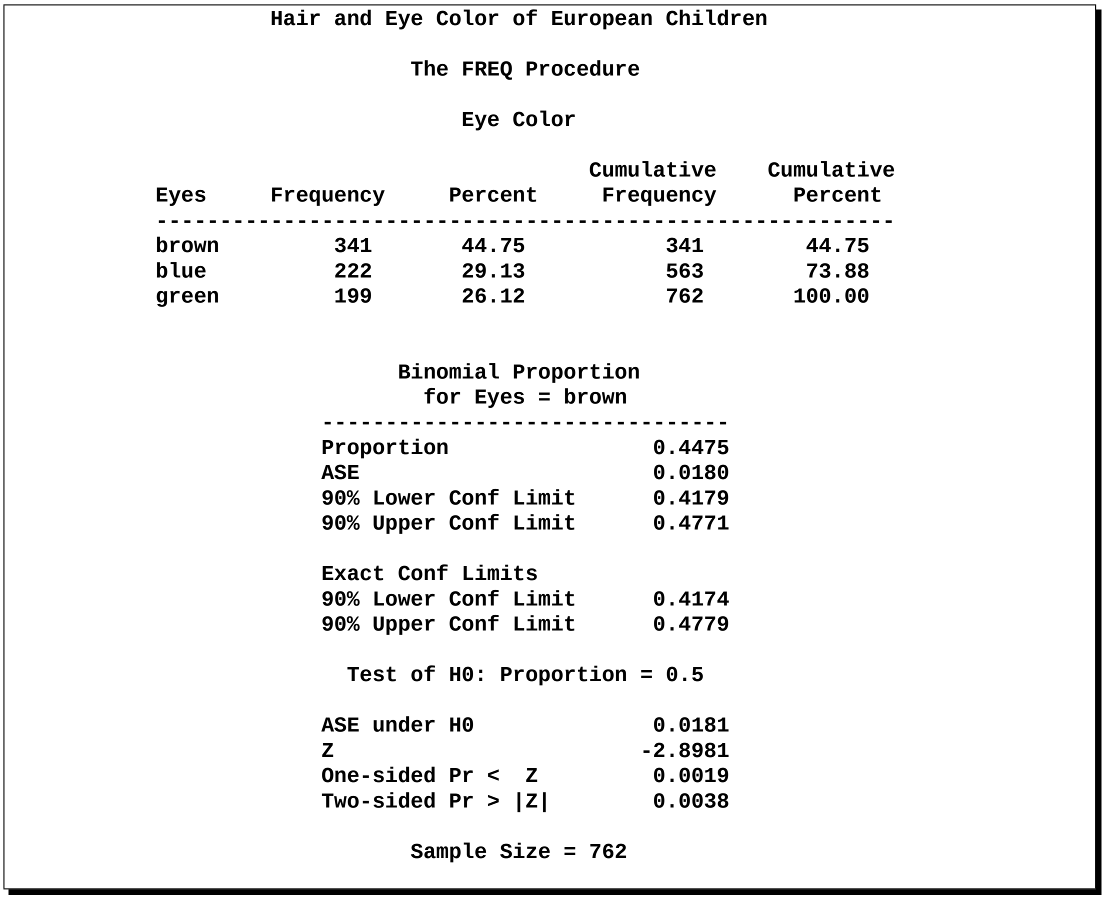
The first TABLES statement produces a frequency table for eye color. The BINOMIAL option computes the binomial proportion and confidence limits, and it tests the hypothesis that the proportion for the first eye color level (brown) is 0.5. The option ALPHA=.1 specifies that 90% confidence limits should be computed. The second TABLES statement creates a frequency table for hair color and computes the binomial proportion and confidence limits, but it tests that the proportion for the first hair color (fair) is 0.28. These statements produce Output 29.3.1 and Output 29.3.2.

The frequency table in Output 29.3.1 displays the variable values in order of descending frequency count. Since the first variable level is 'brown', PROC FREQ computes the binomial proportion of children with brown eyes. PROC FREQ also computes its asymptotic standard error (ASE), and asymptotic and exact 90% confidence limits. If you do not specify the ALPHA= option, then PROC FREQ computes the default 95% confidence limits.

Because the value of Z is less than zero, PROC FREQ computes a left-sided p-value (0.0019). This small p-value supports the alternative hypothesis that the true value of the proportion of children with brown eyes is less than 50%.

Output 29.3.2 displays the results from the second TABLES statement. PROC FREQ computes the default 95% confidence limits since the ALPHA= option is not specified. The value of Z is greater than zero, so PROC FREQ computes a right-sided p-value (0.1188). This large p-value provides insufficient evidence to reject the null hypothesis that the proportion of children with fair hair is 28%.

Output 29.3.1. Binomial Proportion for Eye Color



```
         Hair and Eye Color of European Children

                    The FREQ Procedure

                        Eye Color

                                  Cumulative    Cumulative
Eyes     Frequency     Percent     Frequency      Percent
----------------------------------------------------------
brown         341       44.75           341        44.75
blue          222       29.13           563        73.88
green         199       26.12           762       100.00


                   Binomial Proportion
                     for Eyes = brown
             --------------------------------
             Proportion                0.4475
             ASE                       0.0180
             90% Lower Conf Limit      0.4179
             90% Upper Conf Limit      0.4771

             Exact Conf Limits
             90% Lower Conf Limit      0.4174
             90% Upper Conf Limit      0.4779

               Test of H0: Proportion = 0.5

             ASE under H0              0.0181
             Z                        -2.8981
             One-sided Pr <  Z         0.0019
             Two-sided Pr > |Z|        0.0038

                    Sample Size = 762
```

Output 29.3.2. Binomial Proportion for Hair Color

```
          Hair and Eye Color of European Children

                        Hair Color

                                   Cumulative    Cumulative
Hair      Frequency     Percent     Frequency      Percent
-----------------------------------------------------------
fair           228       29.92           228        29.92
medium         217       28.48           445        58.40
dark           182       23.88           627        82.28
red            113       14.83           740        97.11
black           22        2.89           762       100.00


                    Binomial Proportion
                      for Hair = fair
              --------------------------------
              Proportion                0.2992
              ASE                       0.0166
              95% Lower Conf Limit      0.2667
              95% Upper Conf Limit      0.3317

              Exact Conf Limits
              95% Lower Conf Limit      0.2669
              95% Upper Conf Limit      0.3331

               Test of H0: Proportion = 0.28

              ASE under H0              0.0163
              Z                         1.1812
              One-sided Pr >  Z         0.1188
              Two-sided Pr > |Z|        0.2375

                     Sample Size = 762
```

Example 29.4. Analyzing a 2x2 Contingency Table

This example computes chi-square tests and Fisher's exact test to compare the probability of coronary heart disease for two types of diet. It also estimates the relative risks and computes exact confidence limits for the odds ratio.

The data set FatComp contains hypothetical data for a case-control study of high fat diet and the risk of coronary heart disease. The data are recorded as cell counts, where the variable Count contains the frequencies for each exposure and response combination. The data set is sorted in descending order by the variables Exposure and Response, so that the first cell of the 2×2 table contains the frequency of positive exposure and positive response. The FORMAT procedure creates formats to identify the type of exposure and response with character values.

```
proc format;
   value ExpFmt 1='High Cholesterol Diet'
                0='Low Cholesterol Diet';
   value RspFmt 1='Yes'
                0='No';
run;

data FatComp;
   input Exposure Response Count;
   label Response='Heart Disease';
   datalines;
0 0  6
0 1  2
1 0  4
1 1 11
;

proc sort data=FatComp;
   by descending Exposure descending Response;
run;
```

In the following statements, the TABLES statement creates a two-way table, and the option ORDER=DATA orders the contingency table values by their order in the data set. The CHISQ option produces several chi-square tests, while the RELRISK option produces relative risk measures. The EXACT statement creates the exact Pearson chi-square test and exact confidence limits for the odds ratio. These statements produce Output 29.4.1 through Output 29.4.3.

```
proc freq data=FatComp order=data;
   weight Count;
   tables Exposure*Response / chisq relrisk;
   exact pchi or;
   format Exposure ExpFmt. Response RspFmt.;
   title 'Case-Control Study of High Fat/Cholesterol Diet';
run;
```

Output 29.4.1. Contingency Table

```
                 Case-Control Study of High Fat/Cholesterol Diet

                              The FREQ Procedure

                        Table of Exposure by Response

                   Exposure          Response(Heart Disease)

                   Frequency        |
                   Percent          |
                   Row Pct          |
                   Col Pct          |Yes     |No      |  Total
                   -----------------+--------+--------+
                   High Cholesterol |     11 |      4 |     15
                    Diet            |  47.83 |  17.39 |  65.22
                                    |  73.33 |  26.67 |
                                    |  84.62 |  40.00 |
                   -----------------+--------+--------+
                   Low Cholesterol  |      2 |      6 |      8
                   Diet             |   8.70 |  26.09 |  34.78
                                    |  25.00 |  75.00 |
                                    |  15.38 |  60.00 |
                   -----------------+--------+--------+
                   Total                  13       10       23
                                       56.52    43.48   100.00
```

The contingency table in Output 29.4.1 displays the variable values so that the first table cell contains the frequency for the first cell in the data set, the frequency of positive exposure and positive response.

Output 29.4.2. Chi-Square Statistics

```
                 Case-Control Study of High Fat/Cholesterol Diet

                   Statistics for Table of Exposure by Response

              Statistic                     DF       Value      Prob
              ------------------------------------------------------
              Chi-Square                     1      4.9597    0.0259
              Likelihood Ratio Chi-Square    1      5.0975    0.0240
              Continuity Adj. Chi-Square     1      3.1879    0.0742
              Mantel-Haenszel Chi-Square     1      4.7441    0.0294
              Phi Coefficient                       0.4644
              Contingency Coefficient               0.4212
              Cramer's V                            0.4644

            WARNING: 50% of the cells have expected counts less than 5.
                     (Asymptotic) Chi-Square may not be a valid test.


                             Pearson Chi-Square Test
                        ----------------------------------
                        Chi-Square                  4.9597
                        DF                               1
                        Asymptotic Pr >  ChiSq      0.0259
                        Exact      Pr >= ChiSq      0.0393


                               Fisher's Exact Test
                        ----------------------------------
                        Cell (1,1) Frequency (F)        11
                        Left-sided Pr <= F          0.9967
                        Right-sided Pr >= F         0.0367

                        Table Probability (P)       0.0334
                        Two-sided Pr <= P           0.0393

                                 Sample Size = 23
```

Output 29.4.2 displays the chi-square statistics. Since the expected counts in some of the table cells are small, PROC FREQ gives a warning that the asymptotic chi-square tests may not be appropriate. In this case, the exact tests are appropriate. The alternative hypothesis for this analysis states that coronary heart disease is more likely to be associated with a high fat diet, so a one-sided test is desired. Fisher's exact right-sided test analyzes whether the probability of heart disease in the high fat group exceeds the probability of heart disease in the low fat group; since this p-value is small, the alternative hypothesis is supported.

Output 29.4.3. Relative Risk

```
                Case-Control Study of High Fat/Cholesterol Diet

                  Statistics for Table of Exposure by Response

                  Estimates of the Relative Risk (Row1/Row2)

       Type of Study                   Value       95% Confidence Limits
       -----------------------------------------------------------------
       Case-Control (Odds Ratio)      8.2500        1.1535       59.0029
       Cohort (Col1 Risk)             2.9333        0.8502       10.1204
       Cohort (Col2 Risk)             0.3556        0.1403        0.9009


                        Odds Ratio (Case-Control Study)
                      -----------------------------------
                      Odds Ratio                   8.2500

                      Asymptotic Conf Limits
                      95% Lower Conf Limit         1.1535
                      95% Upper Conf Limit        59.0029

                      Exact Conf Limits
                      95% Lower Conf Limit         0.8677
                      95% Upper Conf Limit       105.5488

                                Sample Size = 23
```

The odds ratio, displayed in Output 29.4.3, provides an estimate of the relative risk when an event is rare. This estimate indicates that the odds of heart disease is 8.25 times higher in the high fat diet group; however, the wide confidence limits indicate that this estimate has low precision.

Example 29.5. Creating an Output Data Set Containing Chi-Square Statistics

This example uses the Color data from Example 29.1 (page 1527) to output the Pearson chi-square and the likelihood-ratio chi-square statistics to a SAS data set. The following statements create a two-way table of eye color versus hair color.

```
proc freq data=Color order=data;
   weight Count;
   tables Eyes*Hair / chisq expected cellchi2 norow nocol;
   output out=ChiSqData pchi lrchi n nmiss;
   title 'Chi-Square Tests for 3 by 5 Table of Eye and Hair Color';
run;
proc print data=ChiSqData noobs;
   title1 'Chi-Square Statistics for Eye and Hair Color';
   title2 'Output Data Set from the FREQ Procedure';
run;
```

The CHISQ option produces chi-square tests, the EXPECTED option displays expected cell frequencies in the table, and the CELLCHI2 option displays the cell contribution to the chi-square. The NOROW and NOCOL options suppress the display of row and column percents in the table.

The OUTPUT statement creates the ChiSqData data set with eight variables: the N option stores the number of nonmissing observations, the NMISS option stores the number of missing observations, and the PCHI and LRCHI options store Pearson and likelihood-ratio chi-square statistics, respectively, together with their degrees of freedom and p-values.

The preceding statements produce Output 29.5.1 and Output 29.5.2.

Output 29.5.1. Contingency Table

```
            Chi-Square Tests for 3 by 5 Table of Eye and Hair Color

                              The FREQ Procedure

                             Table of Eyes by Hair

      Eyes(Eye Color)     Hair(Hair Color)

      Frequency      |
      Expected       |
      Cell Chi-Square|
      Percent        |fair    |red     |medium  |dark    |black   |  Total
      ---------------+--------+--------+--------+--------+--------+
      blue           |     69 |     28 |     68 |     51 |      6 |    222
                     | 66.425 | 32.921 |  63.22 | 53.024 | 6.4094 |
                     | 0.0998 | 0.7357 | 0.3613 | 0.0772 | 0.0262 |
                     |   9.06 |   3.67 |   8.92 |   6.69 |   0.79 |  29.13
      ---------------+--------+--------+--------+--------+--------+
      green          |     69 |     38 |     55 |     37 |      0 |    199
                     | 59.543 |  29.51 | 56.671 |  47.53 | 5.7454 |
                     | 1.5019 | 2.4422 | 0.0492 | 2.3329 | 5.7454 |
                     |   9.06 |   4.99 |   7.22 |   4.86 |   0.00 |  26.12
      ---------------+--------+--------+--------+--------+--------+
      brown          |     90 |     47 |     94 |     94 |     16 |    341
                     | 102.03 | 50.568 | 97.109 | 81.446 | 9.8451 |
                     | 1.4187 | 0.2518 | 0.0995 |  1.935 | 3.8478 |
                     |  11.81 |   6.17 |  12.34 |  12.34 |   2.10 |  44.75
      ---------------+--------+--------+--------+--------+--------+
      Total               228      113      217      182       22      762
                        29.92    14.83    28.48    23.88     2.89   100.00
```

The contingency table in Output 29.5.1 displays eye and hair color in the order in which they appear in the Color data set. The Pearson chi-square statistic in Output 29.5.2 provides evidence of an association between eye and hair color (p=0.0073). The cell chi-square values show that most of the association is due to more green-eyed children with fair or red hair and fewer with dark or black hair. The opposite occurs with the brown-eyed children.

Output 29.5.2. Chi-Square Statistics

```
            Chi-Square Tests for 3 by 5 Table of Eye and Hair Color

                      Statistics for Table of Eyes by Hair

             Statistic                     DF       Value      Prob
             ------------------------------------------------------
             Chi-Square                     8     20.9248    0.0073
             Likelihood Ratio Chi-Square    8     25.9733    0.0011
             Mantel-Haenszel Chi-Square     1      3.7838    0.0518
             Phi Coefficient                       0.1657
             Contingency Coefficient               0.1635
             Cramer's V                            0.1172

                               Sample Size = 762
```

Output 29.5.3. Output Data Set

```
                  Chi-Square Statistics for Eye and Hair Color
                    Output Data Set from the FREQ Procedure

  N    NMISS    _PCHI_    DF_PCHI       P_PCHI    _LRCHI_   DF_LRCHI      P_LRCHI

 762     0     20.9248       8      .007349898   25.9733       8      .001061424
```

The OUT= data set is displayed in Output 29.5.3. It contains one observation with the sample size, the number of missing values, and the chi-square statistics and corresponding degrees of freedom and p-values as in Output 29.5.2.

Example 29.6. Computing Cochran-Mantel-Haenszel Statistics for a Stratified Table

The data set Migraine contains hypothetical data for a clinical trial of migraine treatment. Subjects of both genders receive either a new drug therapy or a placebo. Their response to treatment is coded as ’Better’ or ’Same’. The data are recorded as cell counts, and the number of subjects for each treatment and response combination is recorded in the variable Count.

```
data Migraine;
   input Gender $ Treatment $ Response $ Count @@;
   datalines;
female Active  Better 16   female Active  Same 11
female Placebo Better  5   female Placebo Same 20
male   Active  Better 12   male   Active  Same 16
male   Placebo Better  7   male   Placebo Same 19
;
```

The following statements create a three-way table stratified by Gender, where Treatment forms the rows and Response forms the columns. The CMH option produces the Cochran-Mantel-Haenszel statistics. For this stratified 2×2 table, estimates of the common relative risk and the Breslow-Day test for homogeneity of the odds ratios are also displayed. The NOPRINT option suppresses the display of the contingency tables. These statements produce Output 29.6.1 through Output 29.6.3.

```
proc freq data=Migraine;
   weight Count;
   tables Gender*Treatment*Response / cmh noprint;
   title 'Clinical Trial for Treatment of Migraine Headaches';
run;
```

Output 29.6.1. Cochran-Mantel-Haenszel Statistics

```
               Clinical Trial for Treatment of Migraine Headaches

                              The FREQ Procedure

                Summary Statistics for Treatment by Response
                           Controlling for Gender

          Cochran-Mantel-Haenszel Statistics (Based on Table Scores)

        Statistic    Alternative Hypothesis    DF       Value      Prob
        ---------------------------------------------------------------
            1        Nonzero Correlation        1      8.3052    0.0040
            2        Row Mean Scores Differ     1      8.3052    0.0040
            3        General Association        1      8.3052    0.0040

                           Total Sample Size = 106
```

For a stratified 2×2 table, the three CMH statistics displayed in Output 29.6.1 test the same hypothesis. The significant p-value (0.004) indicates that the association between treatment and response remains strong after adjusting for gender.

Output 29.6.2. CMH Option: Relative Risks

```
            Clinical Trial for Treatment of Migraine Headaches

               Summary Statistics for Treatment by Response
                          Controlling for Gender

            Estimates of the Common Relative Risk (Row1/Row2)

Type of Study     Method                  Value     95% Confidence Limits
-------------------------------------------------------------------------
Case-Control      Mantel-Haenszel        3.3132       1.4456       7.5934
  (Odds Ratio)    Logit                  3.2941       1.4182       7.6515

Cohort            Mantel-Haenszel        2.1636       1.2336       3.7948
  (Col1 Risk)     Logit                  2.1059       1.1951       3.7108

Cohort            Mantel-Haenszel        0.6420       0.4705       0.8761
  (Col2 Risk)     Logit                  0.6613       0.4852       0.9013

                         Total Sample Size = 106
```

The CMH option also produces a table of relative risks, as shown in Output 29.6.2. Because this is a prospective study, the relative risk estimate assesses the effectiveness of the new drug; the “Cohort (Col1 Risk)” values are the appropriate estimates for the first column, or the risk of improvement. The probability of migraine improvement with the new drug is just over two times the probability of improvement with the placebo.

Output 29.6.3. CMH Option: Breslow-Day Test

```
            Clinical Trial for Treatment of Migraine Headaches

               Summary Statistics for Treatment by Response
                          Controlling for Gender

                           Breslow-Day Test for
                      Homogeneity of the Odds Ratios
                      ------------------------------
                      Chi-Square              1.4929
                      DF                           1
                      Pr > ChiSq              0.2218

                         Total Sample Size = 106
```

The large p-value for the Breslow-Day test (0.2218) in Output 29.6.3 indicates no significant gender difference in the odds ratios.

Example 29.7. Computing the Cochran-Armitage Trend Test

The data set Pain contains hypothetical data for a clinical trial of a drug therapy to control pain. The clinical trial investigates whether adverse responses increase with larger drug doses. Subjects receive either a placebo or one of four drug doses. An adverse response is recorded as Adverse='Yes'; otherwise, it is recorded as Adverse='No'. The number of subjects for each drug dose and response combination is contained in the variable Count.

```
data pain;
   input Dose Adverse $ Count @@;
   datalines;
0 No 26   0 Yes  6
1 No 26   1 Yes  7
2 No 23   2 Yes  9
3 No 18   3 Yes 14
4 No  9   4 Yes 23
;
```

The TABLES statement in the following program produces a two-way table. The MEASURES option produces measures of association, and the CL option produces confidence limits for these measures. The TREND option tests for a trend across the ordinal values of the Dose variable with the Cochran-Armitage test. The EXACT statement produces exact p-values for this test, and the MAXTIME= option terminates the exact computations if they do not complete within 60 seconds. The TEST statement computes an asymptotic test for Somer's $D(C|R)$. These statements produce Output 29.7.1 through Output 29.7.3.

```
proc freq data=Pain;
   weight Count;
   tables Dose*Adverse / trend measures cl;
   test smdcr;
   exact trend / maxtime=60;
   title1 'Clinical Trial for Treatment of Pain';
run;
```

Output 29.7.1. Contingency Table

```
                   Clinical Trial for Treatment of Pain

                            The FREQ Procedure

                         Table of Dose by Adverse

                   Dose      Adverse

                   Frequency|
                   Percent  |
                   Row Pct  |
                   Col Pct  |No      |Yes     |  Total
                   ---------+--------+--------+
                          0 |     26 |      6 |     32
                            |  16.15 |   3.73 |  19.88
                            |  81.25 |  18.75 |
                            |  25.49 |  10.17 |
                   ---------+--------+--------+
                          1 |     26 |      7 |     33
                            |  16.15 |   4.35 |  20.50
                            |  78.79 |  21.21 |
                            |  25.49 |  11.86 |
                   ---------+--------+--------+
                          2 |     23 |      9 |     32
                            |  14.29 |   5.59 |  19.88
                            |  71.88 |  28.13 |
                            |  22.55 |  15.25 |
                   ---------+--------+--------+
                          3 |     18 |     14 |     32
                            |  11.18 |   8.70 |  19.88
                            |  56.25 |  43.75 |
                            |  17.65 |  23.73 |
                   ---------+--------+--------+
                          4 |      9 |     23 |     32
                            |   5.59 |  14.29 |  19.88
                            |  28.13 |  71.88 |
                            |   8.82 |  38.98 |
                   ---------+--------+--------+
                   Total         102       59      161
                               63.35    36.65   100.00
```

The “Row Pct” values in Output 29.7.1 show the expected increasing trend in the proportion of adverse effects due to increasing dosage (from 18.75% to 71.88%).

Output 29.7.2. Measures of Association

```
                      Clinical Trial for Treatment of Pain

                    Statistics for Table of Dose by Adverse

                                                                     95%
Statistic                              Value       ASE     Confidence Limits
----------------------------------------------------------------------------
Gamma                                 0.5313    0.0935     0.3480     0.7146
Kendall's Tau-b                       0.3373    0.0642     0.2114     0.4631
Stuart's Tau-c                        0.4111    0.0798     0.2547     0.5675

Somers' D C|R                         0.2569    0.0499     0.1592     0.3547
Somers' D R|C                         0.4427    0.0837     0.2786     0.6068

Pearson Correlation                   0.3776    0.0714     0.2378     0.5175
Spearman Correlation                  0.3771    0.0718     0.2363     0.5178

Lambda Asymmetric C|R                 0.2373    0.0837     0.0732     0.4014
Lambda Asymmetric R|C                 0.1250    0.0662     0.0000     0.2547
Lambda Symmetric                      0.1604    0.0621     0.0388     0.2821

Uncertainty Coefficient C|R           0.1261    0.0467     0.0346     0.2175
Uncertainty Coefficient R|C           0.0515    0.0191     0.0140     0.0890
Uncertainty Coefficient Symmetric     0.0731    0.0271     0.0199     0.1262


                                  Somers' D C|R
                      --------------------------------
                      Somers' D C|R               0.2569
                      ASE                         0.0499
                      95% Lower Conf Limit        0.1592
                      95% Upper Conf Limit        0.3547

                       Test of H0: Somers' D C|R = 0

                      ASE under H0                0.0499
                      Z                           5.1511
                      One-sided Pr >  Z           <.0001
                      Two-sided Pr > |Z|          <.0001

                             Sample Size = 161
```

Output 29.7.2 displays the measures of association produced by the MEASURES option. Somer's $D(C|R)$ measures the association treating the column variable (Adverse) as the response and the row variable (Dose) as a predictor. Because the asymptotic 95% confidence limits do not contain zero, this indicates a strong positive association. Similarly, the Pearson and Spearman correlation coefficients show evidence of a strong positive association, as hypothesized.

Output 29.7.3. Trend Test

```
                 Clinical Trial for Treatment of Pain

               Statistics for Table of Dose by Adverse

                    Cochran-Armitage Trend Test
                  -------------------------------
                  Statistic (Z)           -4.7918

                  Asymptotic Test
                  One-sided Pr <  Z        <.0001
                  Two-sided Pr > |Z|       <.0001

                  Exact Test
                  One-sided Pr <=  Z    7.237E-07
                  Two-sided Pr >= |Z|   1.324E-06

                         Sample Size = 161
```

The Cochran-Armitage test (Output 29.7.3) supports the trend hypothesis. The small left-sided p-values for the Cochran-Armitage test indicate that the probability of the Column 1 level (Adverse='No') decreases as Dose increases or, equivalently, that the probability of the Column 2 level (Adverse='Yes') increases as Dose increases. The two-sided p-value tests against either an increasing or decreasing alternative. This is an appropriate hypothesis when you want to determine whether the drug has progressive effects on the probability of adverse effects but the direction is unknown.

Example 29.8. Computing Friedman's Chi-Square Statistic

Friedman's test is a nonparametric test for treatment differences in a randomized complete block design. Each block of the design may be a subject or a homogeneous group of subjects. If blocks are groups of subjects, the number of subjects in each block must equal the number of treatments. Treatments are randomly assigned to subjects within each block. If there is one subject per block, then the subjects are repeatedly measured once under each treatment. The order of treatments is randomized for each subject.

In this setting, Friedman's test is identical to the ANOVA (row means scores) CMH statistic when the analysis uses rank scores (SCORES=RANK). The three-way table uses subject (or subject group) as the stratifying variable, treatment as the row variable, and response as the column variable. PROC FREQ handles ties by assigning midranks to tied response values. If there are multiple subjects per treatment in each block, the ANOVA CMH statistic is a generalization of Friedman's test.

The data set Hypnosis contains data from a study investigating whether hypnosis has the same effect on skin potential (measured in millivolts) for four emotions (Lehmann 1975, p. 264). Eight subjects are asked to display fear, joy, sadness, and calmness under hypnosis. The data are recorded as one observation per subject for each emotion.

```
data Hypnosis;
   length Emotion $ 10;
   input Subject Emotion $ SkinResponse @@;
   datalines;
1 fear 23.1  1 joy 22.7  1 sadness 22.5  1 calmness 22.6
2 fear 57.6  2 joy 53.2  2 sadness 53.7  2 calmness 53.1
3 fear 10.5  3 joy  9.7  3 sadness 10.8  3 calmness  8.3
4 fear 23.6  4 joy 19.6  4 sadness 21.1  4 calmness 21.6
5 fear 11.9  5 joy 13.8  5 sadness 13.7  5 calmness 13.3
6 fear 54.6  6 joy 47.1  6 sadness 39.2  6 calmness 37.0
7 fear 21.0  7 joy 13.6  7 sadness 13.7  7 calmness 14.8
8 fear 20.3  8 joy 23.6  8 sadness 16.3  8 calmness 14.8
;
```

In the following statements, the TABLES statement creates a three-way table stratified by Subject and a two-way table; the variables Emotion and SkinResponse form the rows and columns of each table. The CMH2 option produces the first two Cochran-Mantel-Haenszel statistics, the option SCORES=RANK specifies that rank scores are used to compute these statistics, and the NOPRINT option suppresses the contingency tables. These statements produce Output 29.8.1 and Output 29.8.2.

```
proc freq data=Hypnosis;
   tables Subject*Emotion*SkinResponse
          / cmh2 scores=rank noprint;
run;
```

Output 29.8.1. CMH Statistics: Stratifying by Subject

```
                          The FREQ Procedure

              Summary Statistics for Emotion by SkinResponse
                          Controlling for Subject

        Cochran-Mantel-Haenszel Statistics (Based on Rank Scores)

     Statistic    Alternative Hypothesis    DF       Value      Prob
     ---------------------------------------------------------------
         1        Nonzero Correlation        1      0.2400    0.6242
         2        Row Mean Scores Differ     3      6.4500    0.0917

                          Total Sample Size = 32
```

Because the CMH statistics in Output 29.8.1 are based on rank scores, the Row Mean Scores Differ statistic is identical to Friedman’s chi-square ($Q = 6.45$). The p-value of 0.0917 indicates that differences in skin potential response for different emotions are significant at the 10% level but not at the 5% level.

Output 29.8.2. CMH Statistics: No Stratification

```
                          The FREQ Procedure

            Summary Statistics for Emotion by SkinResponse

       Cochran-Mantel-Haenszel Statistics (Based on Rank Scores)

    Statistic    Alternative Hypothesis    DF       Value      Prob
    ---------------------------------------------------------------
        1        Nonzero Correlation        1      0.0001    0.9933
        2        Row Mean Scores Differ     3      0.5678    0.9038

                        Total Sample Size = 32
```

When you do not stratify by subject, the Row Mean Scores Differ CMH statistic is identical to a Kruskal-Wallis test and is not significant (p=0.9038 in Output 29.8.2). Thus, adjusting for subject is critical to reducing the background variation due to subject differences.

Example 29.9. Testing Marginal Homogeneity with Cochran's Q

When a binary response is measured several times or under different conditions, Cochran's Q tests that the marginal probability of a positive response is unchanged across the times or conditions. When there are more than two response categories, you can use the CATMOD procedure to fit a repeated-measures model.

The data set Drugs contains data for a study of three drugs to treat a chronic disease (Agresti 1990). Forty-six subjects receive drugs A, B, and C. The response to each drug is either favorable ('F') or unfavorable ('U').

```
proc format;
   value $ResponseFmt 'F'='Favorable'
                      'U'='Unfavorable';

data drugs;
   input Drug_A $ Drug_B $ Drug_C $ Count @@;
   datalines;
F F F  6   U F F  2
F F U 16   U F U  4
F U F  2   U U F  6
F U U  4   U U U  6
;
```

The following statements create one-way frequency tables of the responses to each drug. The AGREE option produces Cochran's Q and other measures of agreement for the three-way table. These statements produce Output 29.9.1 through Output 29.9.3.

```
proc freq data=Drugs;
   weight Count;
   tables Drug_A Drug_B Drug_C / nocum;
   tables Drug_A*Drug_B*Drug_C / agree noprint;
   format Drug_A Drug_B Drug_C $ResponseFmt.;
   title 'Study of Three Drug Treatments for a Chronic Disease';
run;
```

Output 29.9.1. One-Way Frequency Tables

```
             Study of Three Drug Treatments for a Chronic Disease

                             The FREQ Procedure

                     Drug_A         Frequency     Percent
                     ------------------------------------
                     Favorable            28       60.87
                     Unfavorable          18       39.13


                     Drug_B         Frequency     Percent
                     ------------------------------------
                     Favorable            28       60.87
                     Unfavorable          18       39.13


                     Drug_C         Frequency     Percent
                     ------------------------------------
                     Favorable            16       34.78
                     Unfavorable          30       65.22
```

The one-way frequency tables in Output 29.9.1 provide the marginal response for each drug. For drugs A and B, 61% of the subjects reported a favorable response while 35% of the subjects reported a favorable response to drug C.

Output 29.9.2. Measures of Agreement

```
Study of Three Drug Treatments for a Chronic Disease

     Statistics for Table 1 of Drug_B by Drug_C
          Controlling for Drug_A=Favorable

                   McNemar's Test
              ------------------------
              Statistic (S)    10.8889
              DF                     1
              Pr > S            0.0010


              Simple Kappa Coefficient
          --------------------------------
          Kappa                    -0.0328
          ASE                       0.1167
          95% Lower Conf Limit     -0.2615
          95% Upper Conf Limit      0.1960

                  Sample Size = 28


     Statistics for Table 2 of Drug_B by Drug_C
         Controlling for Drug_A=Unfavorable

                  McNemar's Test
              -----------------------
              Statistic (S)    0.4000
              DF                    1
              Pr > S           0.5271


              Simple Kappa Coefficient
          --------------------------------
          Kappa                    -0.1538
          ASE                       0.2230
          95% Lower Conf Limit     -0.5909
          95% Upper Conf Limit      0.2832

                  Sample Size = 18



Study of Three Drug Treatments for a Chronic Disease

      Summary Statistics for Drug_B by Drug_C
               Controlling for Drug_A

             Overall Kappa Coefficient
          --------------------------------
          Kappa                    -0.0588
          ASE                       0.1034
          95% Lower Conf Limit     -0.2615
          95% Upper Conf Limit      0.1439


                Test for Equal Kappa
                    Coefficients
                --------------------
                Chi-Square    0.2314
                DF                 1
                Pr > ChiSq    0.6305


               Total Sample Size = 46
```

McNemar's test (Output 29.9.2) shows strong discordance between drugs B and C when the response to drug A is favorable. The small negative value of the simple kappa indicates no agreement between drug B response and drug C response.

Output 29.9.3. Cochran's Q

```
                Study of Three Drug Treatments for a Chronic Disease

                      Summary Statistics for Drug_B by Drug_C
                               Controlling for Drug_A

                              Cochran's Q, for Drug_A
                                by Drug_B by Drug_C
                              -----------------------
                              Statistic (Q)    8.4706
                              DF                    2
                              Pr > Q           0.0145

                               Total Sample Size = 46
```

Cochran's Q is statistically significant (p=0.0144 in Output 29.9.3), which leads to rejection of the hypothesis that the probability of favorable response is the same for the three drugs.

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A. (1992), "A Survey of Exact Inference for Contingency Tables," *Statistical Science*, 7(1), 131–177.

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A., Mehta, C.R. and Patel, N.R. (1990), "Exact Inference for Contingency Tables with Ordered Categories," *Journal of the American Statistical Association*, 85, 453–458.

Agresti, A., Wackerly, D., and Boyett, J.M. (1979), "Exact Conditional Tests for Cross-Classifications: Approximation of Attained Significance Levels," *Psychometrika*, 44, 75–83.

Birch, M.W. (1965), "The Detection of Partial Association, II: The General Case," *Journal of the Royal Statistical Society, B*, 27, 111–124.

Bishop, Y., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Bowker, A.H. (1948), "Bowker's Test for Symmetry," *Journal of the American Statistical Association*, 43, 572–574.

Breslow, N.E. (1996), "Statistics in Epidemiology: The Case-Control Study," *Journal of the American Statistical Association*, 91, 14–26.

Breslow, N.E. and Day, N.E. (1980), *Statistical Methods in Cancer Research, Volume I: The Analysis of Case-Control Studies*, IARC Scientific Publications, No. 32, Lyon, France: International Agency for Research on Cancer.

Breslow, N.E. and Day, N.E. (1987), *Statistical Methods in Cancer Research, Volume II: The Design and Analysis of Cohort Studies*, IARC Scientific Publications, No. 82, Lyon, France: International Agency for Research on Cancer.

Bross, I.D.J. (1958), "How to Use Ridit Analysis," *Biometrics*, 14, 18–38.

Brown, M.B. and Benedetti, J.K. (1977), "Sampling Behavior of Tests for Correlation in Two-Way Contingency Tables," *Journal of the American Statistical Association*, 72, 309–315.

Cicchetti, D.V. and Allison, T. (1971), "A New Procedure for Assessing Reliability of Scoring EEG Sleep Recordings," *American Journal of EEG Technology*, 11, 101–109.

Cochran, W.G. (1950), "The Comparison of Percentages in Matched Samples," *Biometrika*, 37, 256–266.

Cochran, W.G. (1954), "Some Methods for Strengthening the Common χ^2 Tests," *Biometrics*, 10, 417–451.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman & Hall.

Cohen, J. (1960), "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, 20, 37–46.

Drasgow, F. (1986), "Polychoric and Polyserial Correlations" in *Encyclopedia of Statistical Sciences*, vol. 7, ed. S. Kotz and N. L. Johnson, New York: John Wiley & Sons, Inc., 68–74.

Fienberg, S.E. (1980), *The Analysis of Cross-Classified Data*, Second Edition, Cambridge, MA: MIT Press.

Fleiss, J.L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons, Inc.

Fleiss, J.L. and Cohen, J. (1973), "The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability," *Educational and Psychological Measurement*, 33, 613–619.

Fleiss, J.L., Cohen, J., and Everitt, B.S. (1969), "Large-Sample Standard Errors of Kappa and Weighted Kappa," *Psychological Bulletin*, 72, 323–327.

Freeman, G.H. and Halton, J.H. (1951), "Note on an Exact Treatment of Contingency, Goodness of Fit and Other Problems of Significance," *Biometrika*, 38, 141–149.

Gail, M. and Mantel, N. (1977), "Counting the Number of $r \times c$ Contingency Tables with Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

Gart, J.J. (1971), "The Comparison of Proportions: A Review of Significance Tests, Confidence Intervals and Adjustments for Stratification," *Review of the International Statistical Institute*, 39(2), 148–169.

Goodman, L.A. and Kruskal, W.H. (1979), *Measures of Association for Cross Classification*, New York: Springer-Verlag.

Greenland, S. and Robins, J.M. (1985), “Estimators of the Mantel-Haenszel Variance Consistent in Both Sparse Data and Large-Strata Limiting Models,” *Biometrics*, 42, 311–323.

Haldane, J.B.S. (1955), “The Estimation and Significance of the Logarithm of a Ratio of Frequencies,” *Annals of Human Genetics*, 20, 309–314.

Hollander, M. and Wolfe, D.A. (1973), *Nonparametric Statistical Methods*, New York: John Wiley & Sons, Inc.

Jones, M.P., O’Gorman, T.W., Lemka, J.H., and Woolson, R.F. (1989), “A Monte Carlo Investigation of Homogeneity Tests of the Odds Ratio Under Various Sample Size Configurations,” *Biometrics*, 45, 171–181.

Kendall, M. (1955), *Rank Correlation Methods*, Second Edition, London: Charles Griffin and Co.

Kendall, M. and Stuart, A. (1979), *The Advanced Theory of Statistics*, vol. 2, New York: Macmillan Publishing Company, Inc.

Kleinbaum, D.G., Kupper, L.L., and Morgenstern, H. (1982), *Epidemiologic Research: Principles and Quantitative Methods*, Research Methods Series, New York: Van Nostrand Reinhold.

Landis, R.J., Heyman, E.R., and Koch, G.G. (1978), “Average Partial Association in Three-way Contingency Tables: A Review and Discussion of Alternative Tests,” *International Statistical Review*, 46, 237–254.

Leemis, L.M. and Trivedi, K.S. (1996), “A Comparison of Approximate Interval Estimators for the Bernoulli Parameter,” *The American Statistician*, 50(1), 63–68.

Lehmann, E.L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, San Francisco: Holden-Day, Inc.

Liebetrau, A.M. (1983), *Measures of Association, Quantitative Application in the Social Sciences*, vol. 32, Beverly Hills: Sage Publications, Inc.

Mack, G.A. and Skillings, J.H. (1980), “A Friedman-Type Rank Test for Main Effects in a Two-Factor ANOVA,” *Journal of the American Statistical Association*, 75, 947–951.

Mantel, N. (1963), “Chi-square Tests with One Degree of Freedom: Extensions of the Mantel-Haenszel Procedure,” *Journal of the American Statistical Association*, 58, 690–700.

Mantel, N. and Haenszel, W. (1959), “Statistical Aspects of the Analysis of Data from Retrospective Studies of Disease,” *Journal of the National Cancer Institute*, 22, 719–748.

Margolin, B.H. (1988), “Test for Trend in Proportions,” in *Encyclopedia of Statistical Sciences*, vol. 9, ed. S. Kotz and N.L. Johnson, New York: John Wiley & Sons, Inc., 334–336.

McNemar, Q. (1947), "Note on the Sampling Error of the Difference between Correlated Proportions or Percentages," *Psychometrika*, 12, 153–157.

Mehta, C.R. and Patel, N.R. (1983), "A Network Algorithm for Performing Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of the American Statistical Association*, 78, 427–434.

Mehta, C.R., Patel, N.R., and Gray, R. (1985), "On Computing an Exact Confidence Interval for the Common Odds Ratio in Several 2×2 Contingency Tables," *Journal of the American Statistical Association*, 80, 969–973.

Mehta, C.R., Patel, N.R., and Senchaudhuri, P. (1991), "Exact Stratified Linear Rank Tests for Binary Data," *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, ed. E.M. Keramidas, 200–207.

Mehta, C.R., Patel, N.R., and Tsiatis, A.A. (1984), "Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data," *Biometrics*, 40, 819–825.

Narayanan, A. and Watts, D. (1996), "Exact Methods in the NPAR1WAY Procedure," in *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 1290–1294.

Olsson, U. (1979), "Maximum Likelihood Estimation of the Polychoric Correlation Coefficient," *Psychometrika*, 12, 443–460.

Pirie, W. (1983), "Jonckheere Tests for Ordered Alternatives," in *Encyclopedia of Statistical Sciences*, vol. 4, ed. S. Kotz and N.L. Johnson, New York: John Wiley & Sons, Inc., 315–318.

Radlow, R. and Alf, E.F. (1975), "An Alternate Multinomial Assessment of the Accuracy of the Chi-Square Test of Goodness of Fit," *Journal of the American Statistical Association*, 70, 811–813.

Robins, J.M., Breslow, N., and Greenland, S. (1986), "Estimators of the Mantel-Haenszel Variance Consistent in Both Sparse Data and Large-Strata Limiting Models," *Biometrics*, 42, 311–323.

Snedecor, G.W. and Cochran, W.G. (1989), *Statistical Methods*, Eighth Edition, Ames, IA: Iowa State University Press.

Somers, R.H. (1962), "A New Asymmetric Measure of Association for Ordinal Variables," *American Sociological Review*, 27, 799–811.

Stokes, M.E., Davis, C.S., and Koch, G.G. (1995), *Categorical Data Analysis Using the SAS System*, Cary, NC: SAS Institute Inc.

Tarone, R.E. (1985), "On Heterogeneity Tests Based on Efficient Scores," *Biometrika*, 72, 1, 91–95.

Theil, H. (1972), *Statistical Decomposition Analysis*, Amsterdam: North-Holland Publishing Company.

Thomas, D.G. (1971), "Algorithm AS-36. Exact Confidence Limits for the Odds Ratio in a 2×2 Table," *Applied Statistics*, 20, 105–110.

Valz, P.D. and Thompson, M.E. (1994), “Exact Inference for Kendall’s S and Spearman’s Rho with Extensions to Fisher’s Exact Test in $r \times c$ Contingency Tables,” *Journal of Computational and Graphical Statistics*, 3(4), 459–472.

van Elteren, P.H. (1960), “On the Combination of Independent Two-Sample Tests of Wilcoxon,” *Bulletin of the International Statistical Institute*, 37, 351–361.

Vollset, S.E., Hirji, K.F., and Elashoff, R.M. (1991), “Fast Computation of Exact Confidence Limits for the Common Odds Ratio in a Series of 2×2 Tables,” *Journal of the American Statistical Association*, 86, 404–409.

Woolf, B. (1955), “On Estimating the Relationship between Blood Group and Disease,” *Annals of Human Genetics*, 19, 251–253.

Chapter 30
The GAM Procedure (Experimental)

Chapter Contents

Chapter 30
The GAM Procedure (Experimental)

Overview

The GAM procedure fits generalized additive models as those models are defined by Hastie and Tibshirani (1990). This procedure provides an array of powerful tools for data analysis, based on nonparametric regression and smoothing techniques.

Nonparametric regression relaxes the usual assumption of linearity and enables you to uncover structure in the relationship between the independent variables and the dependent variable that might otherwise be missed. The SAS System provides many procedures for nonparametric regression, such as the LOESS procedure for local regression and the TPSPLINE procedure for thin-plate smoothing splines. The generalized additive models fit by the GAM procedure combine

- an additivity assumption (Stone 1985) that enables relatively many nonparametric relationships to be explored simultaneously with
- the distributional flexibility of generalized linear models (Nelder 1972)

Thus, you can use the GAM procedure when you have multiple independent variables whose effect you want to model nonparametrically, or when the dependent variable is not normally distributed. See the "Nonparametric Regression" section on page 1569 for more details on the form of generalized additive models.

The GAM procedure

- provides nonparametric estimates for additive models
- supports the use of multidimensional data
- supports multiple SCORE statements
- fits both generalized semiparametric additive models and generalized additive models
- enables you to choose a particular model by specifying the model degrees of freedom or smoothing parameter

Experimental graphics are now available with the GAM procedure. For more information, see the "ODS Graphics" section on page 1581.

Getting Started

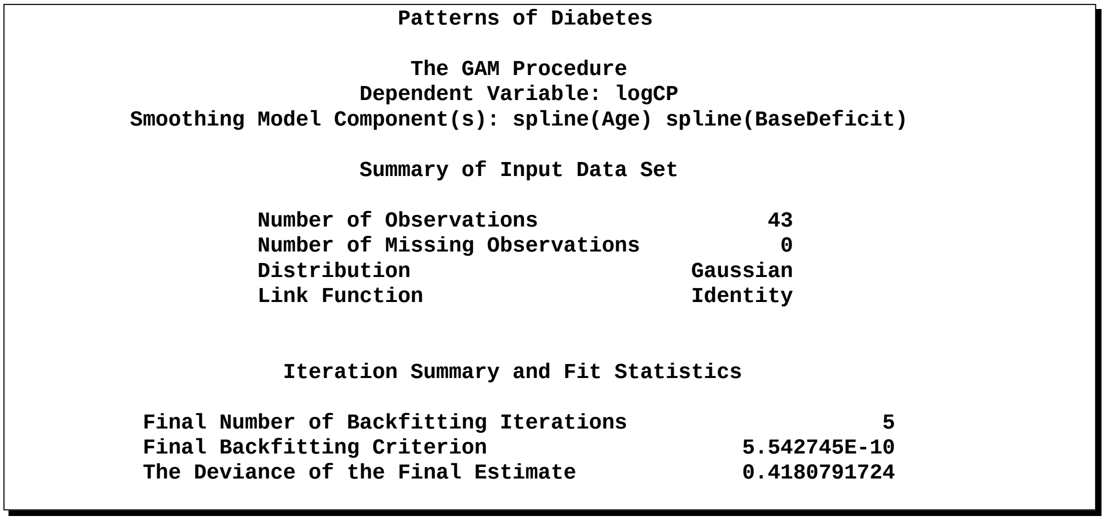
The following example illustrates the use of the GAM procedure to explore in a nonparametric way how two factors affect a response. The data come from a study (Sockett et al. 1987) of the factors affecting patterns of insulin-dependent diabetes mellitus in children. The objective is to investigate the dependence of the level of serum C-peptide on various other factors in order to understand the patterns of residual insulin secretion. The response measurement is the logarithm of C-peptide concentration (pmol/ml) at diagnosis, and the predictor measurements are age and base deficit (a measure of acidity):

```
title 'Patterns of Diabetes';
data diabetes;
   input Age BaseDeficit CPeptide @@;
   logCP = log(CPeptide);
   datalines;
 5.2   -8.1  4.8   8.8  -16.1  4.1  10.5   -0.9  5.2
10.6   -7.8  5.5  10.4  -29.0  5.0   1.8  -19.2  3.4
12.7  -18.9  3.4  15.6  -10.6  4.9   5.8   -2.8  5.6
 1.9  -25.0  3.7   2.2   -3.1  3.9   4.8   -7.8  4.5
 7.9  -13.9  4.8   5.2   -4.5  4.9   0.9  -11.6  3.0
11.8   -2.1  4.6   7.9   -2.0  4.8  11.5   -9.0  5.5
10.6  -11.2  4.5   8.5   -0.2  5.3  11.1   -6.1  4.7
12.8   -1.0  6.6  11.3   -3.6  5.1   1.0   -8.2  3.9
14.5   -0.5  5.7  11.9   -2.0  5.1   8.1   -1.6  5.2
13.8  -11.9  3.7  15.5   -0.7  4.9   9.8   -1.2  4.8
11.0  -14.3  4.4  12.4   -0.8  5.2  11.1  -16.8  5.1
 5.1   -5.1  4.6   4.8   -9.5  3.9   4.2  -17.0  5.1
 6.9   -3.3  5.1  13.2   -0.7  6.0   9.9   -3.3  4.9
12.5  -13.6  4.1  13.2   -1.9  4.6   8.9  -10.0  4.9
10.8  -13.5  5.1
;
run;
```

The following statements perform the desired analysis. The PROC GAM statement invokes the procedure and specifies the diabetes data set as input. The MODEL statement specifies logCP as the response variable and requests that univariate smoothing splines with the default of 4 degrees of freedom be used to model the effect of Age and BaseDeficit.

```
title 'Patterns of Diabetes';
proc gam data=diabetes;
   model logCP = spline(Age) spline(BaseDeficit);
run;
```

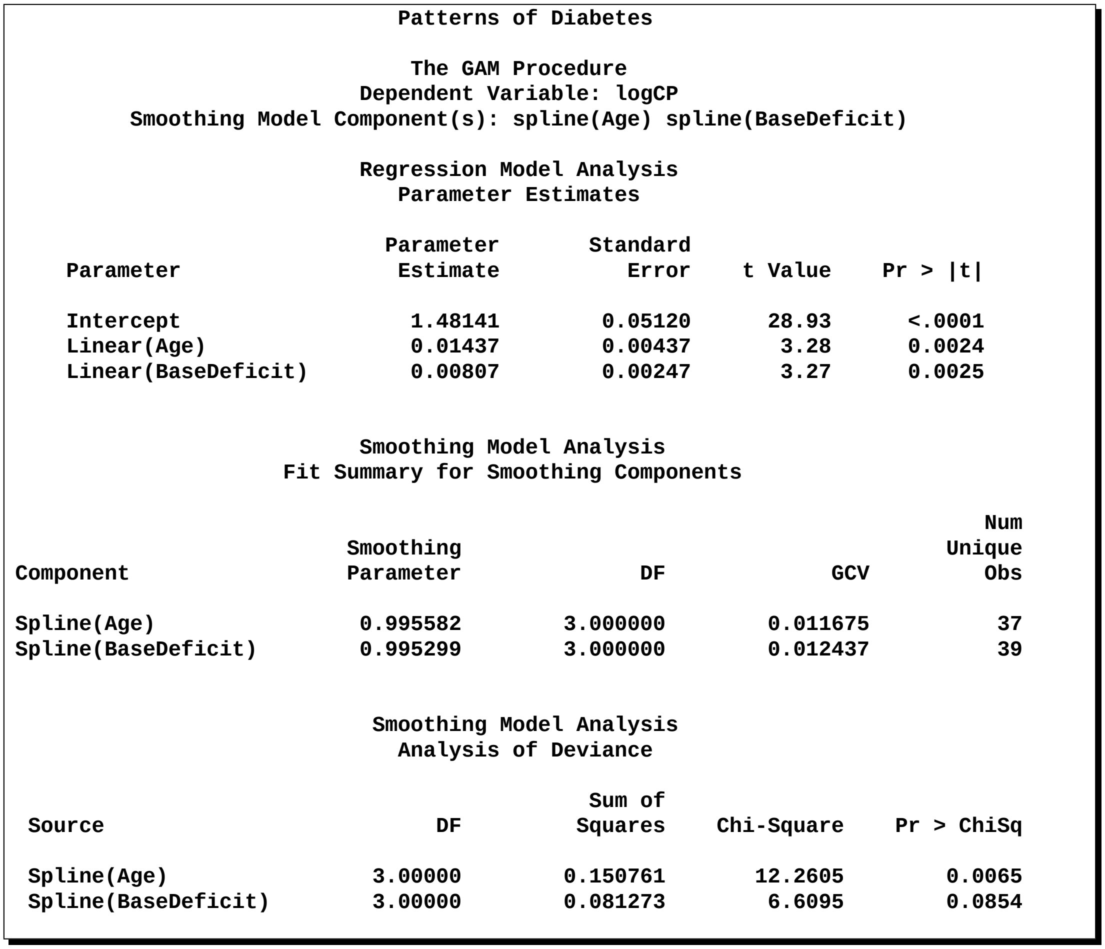
The results are shown in Figure 30.1 and Figure 30.2.



```
                     Patterns of Diabetes

                      The GAM Procedure
                  Dependent Variable: logCP
Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

                  Summary of Input Data Set

          Number of Observations                  43
          Number of Missing Observations           0
          Distribution                      Gaussian
          Link Function                     Identity


            Iteration Summary and Fit Statistics

 Final Number of Backfitting Iterations                     5
 Final Backfitting Criterion                     5.542745E-10
 The Deviance of the Final Estimate              0.4180791724
```

Figure 30.1. Summary Statistics

Figure 30.1 shows two tables. The first table summarizes the input data set and the distributional family used for the model, and the second one summarizes the convergence criterion for backfitting.



```
                            Patterns of Diabetes

                             The GAM Procedure
                          Dependent Variable: logCP
           Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

                          Regression Model Analysis
                             Parameter Estimates

                           Parameter       Standard
    Parameter               Estimate          Error    t Value    Pr > |t|

    Intercept                1.48141        0.05120      28.93      <.0001
    Linear(Age)              0.01437        0.00437       3.28      0.0024
    Linear(BaseDeficit)      0.00807        0.00247       3.27      0.0025


                          Smoothing Model Analysis
                    Fit Summary for Smoothing Components

                                                                       Num
                          Smoothing                                 Unique
Component                 Parameter             DF             GCV     Obs

Spline(Age)                0.995582       3.000000        0.011675      37
Spline(BaseDeficit)        0.995299       3.000000        0.012437      39


                           Smoothing Model Analysis
                             Analysis of Deviance

                                               Sum of
 Source                           DF          Squares    Chi-Square    Pr > ChiSq

 Spline(Age)                 3.00000         0.150761       12.2605        0.0065
 Spline(BaseDeficit)         3.00000         0.081273        6.6095        0.0854
```

Figure 30.2. Analysis of Model

Figure 30.2 displays summary statistics for the model. It consists of three tables. The first is the "Parameter Estimates" table for the parametric part of the model. It indicates that the linear trends for both Age and BaseDeficit are highly significant. The second table is the summary of smoothing components of the nonparametric part of the model. By default, each smoothing component has approximately 4 degrees of freedom (DF). For univariate spline components, one DF is taken up by the (parametric) linear part of the model, so the remaining approximate DF is 3, and the main point of this table is to present the smoothing parameter values that yield this DF for each component. Finally, the third table shows the "Analysis of Deviance" table for the nonparametric component of the model.

In order to explore the overall shape of the relationship between each factor and the response, use the experimental graphics features of PROC GAM to plot the partial predictions.

```
ods html;
ods graphics on;

proc gam data=diabetes;
   model logCP = spline(Age) spline(BaseDeficit);
run;

ods graphics off;
ods html close;
```

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the GAM procedure, see the section “ODS Graphics” on page 1581.

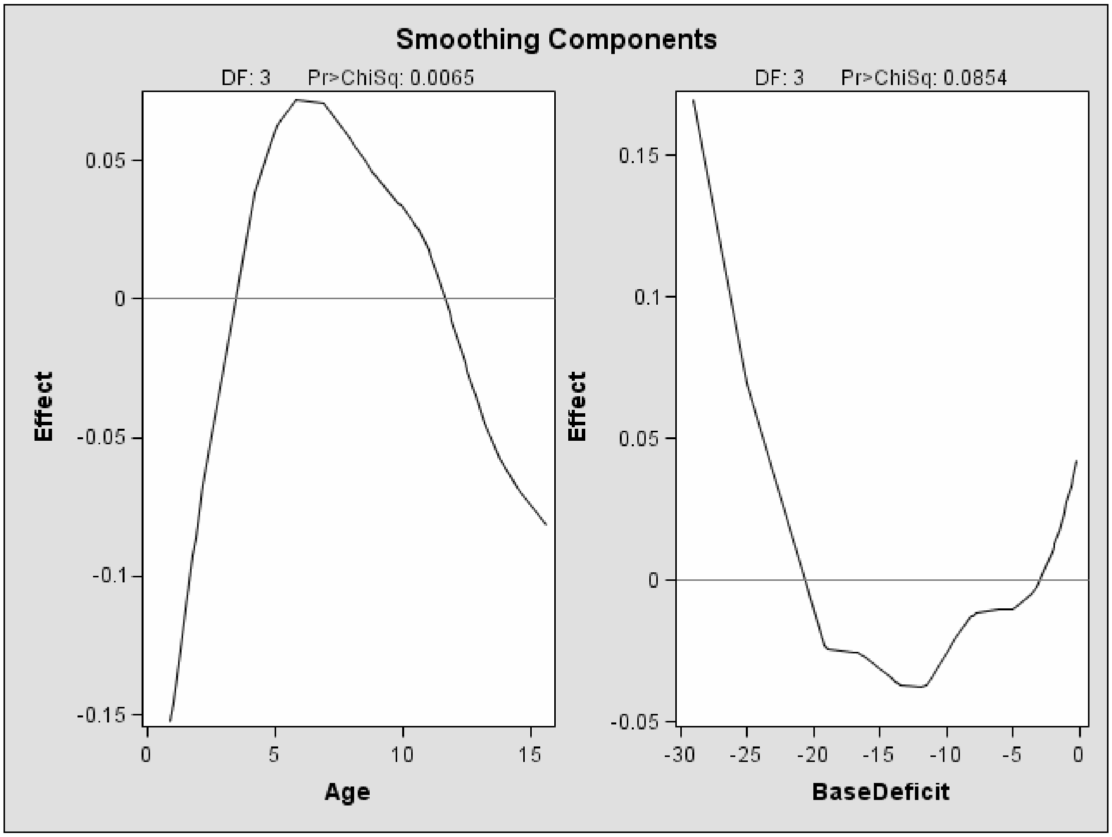


Figure 30.3. Partial Predictions for each Predictor (Experimental)

Both plots show a strong quadratic pattern, with a possible indication of higher-order behavior. Further investigation is required to determine whether these patterns are real or not.

Syntax

PROC GAM < *option* > **;**
  CLASS *variables* **;**
  MODEL *dependent* = < *PARAM(effects)* >
    smoothing effects < */options* > **;**
  SCORE *data=SAS-data-set out=SAS-data-set* **;**
  OUTPUT <*out=SAS-data-set*> *keyword* < · · ·*keyword*> **;**
  BY *variables* **;**
  ID *variables* **;**
  FREQ *variable* **;**

The syntax of the GAM procedure is similar to that of other regression procedures in the SAS System. The PROC GAM and MODEL statements are required. The SCORE statement can appear multiple times; all other statements appear only once.

The syntax for PROC GAM is described in the following sections in alphabetical order after the description of the PROC GAM statement.

PROC GAM Statement

PROC GAM < *option* > **;**

The PROC GAM statement invokes the procedure. You can specify the following option.

DATA=*SAS-data-set*
specifies the SAS data set to be read by PROC GAM. The default value is the most recently created data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GAM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GAM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index for the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Typical class variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide*, and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times.

The analysis produced using a FREQ statement reflects the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first five observations in the new data set are identical. Each observation in the old data set is replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

The FREQ statement is not available when a loess smoother is included in the model.

ID Statement

ID *variables* **;**

The variables in the ID statement are copied from the input data set to the OUT= data set. If you omit the ID statement, only the variables used in the MODEL statement and requested statistics are included in the output data set.

MODEL Statement

MODEL *dependent=<PARAM(effects)><smoothing effects> </options>*;

MODEL *event/trails=<PARAM(effects)> <smoothing effects> </options>* ;

The MODEL statement specifies the dependent variable and the independent effects you want to use to model its values. Specify the independent parametric variables inside the parentheses of PARAM( ). The parametric variables can be either CLASS variables or continuous variables. Class variables must be declared with a CLASS statement. Interactions between variables can also be included as parametric effects. The syntax for the specification of effects is the same as for the GLM procedure.

Any number of smoothing effects can be specified, as follows:

Smoothing Effect	**Meaning**
SPLINE(variable <, df=number>)	fit smoothing spline with the variable and with DF=number
LOESS(variable <, df=number>)	fit local regression with the variable and with DF=number
SPLINE2(variable, variable <,df=number>)	fit bivariate thin-plate smoothing spline with DF=number

If you do not specify the DF=number option with a smoothing effect, DF=4 is used by default, unless you specify the METHOD=GCV model option. Note that for univariate spline components, a degree of freedom is removed by default to account for the linear portion of the model, so the value displayed in the Fit Summary and Analysis of Deviance tables will be one less than the value you specify.

Both parametric effects and smoothing effects are optional, but at least one of them must be present.

If only parametric variables are present, PROC GAM fits a parametric linear model using the terms inside the parentheses of PARAM( ). If only smoothing effects are present, PROC GAM fits a nonparametric additive model. If both types of effect are present, PROC GAM fits a semiparametric model using the parametric effects as the linear part of the model.

The following table shows how to specify various models for a dependent variable y and independent variables x, x1, and x2.

Table 30.1. Syntax for Common GAM Models

Type of Model	Syntax	Mathematical Form
Parametric	model y = param(x);	$E(y) = \beta_0 + \beta_1 x$
Nonparametric	model y = spline(x);	$E(y) = \beta_0 + s(x)$
Nonparametric	model y = loess(x);	$E(y) = \beta_0 + s(x)$
Semiparametric	model y = param(x1) spline(x2);	$E(y) = \beta_0 + \beta_1 x_1 + s(x_2)$
Additive	model y = spline(x1) spline(x2);	$E(y) = \beta_0 + s_1(x_1) + s_2(x_2)$
Thin-plate spline	model y = spline2(x1,x2);	$E(y) = \beta_0 + s(x_1, x_2)$

You can specify the following options in the MODEL statement.

ALPHA=*number*

specifies the significance level α of the confidence limits on the final nonparametric component estimates when you request confidence limits to be included in the output data set. Specify *number* as a value between 0 and 1. The default value is 0.05. See the "OUTPUT Statement" section on page 1568 for more information on the OUTPUT statement.

DIST=*distribution-id*

specifies the distribution family used in the model. The *distribution-id* can be either GAUSSIAN, BINOMIAL, BINARY, GAMMA, IGAUSSIAN, or POISSON. The canonical link is used with those distributions. Although theoretically, alternative links are possible, with nonparametric models the final fit is relatively insensitive to the precise choice of link function. Therefore, only the canonical link for each distribution family is implemented in PROC GAM. The loess smoother is not available for DIST=BINOMIAL when the number of trials is greater than 1.

EPSILON=*number*

specifies the convergence criterion for the backfitting algorithm. The default value is 1E-8.

EPSSCORE=*number*

specifies the convergence criterion for the local score algorithm. The default value is 1E-8.

ITPRINT

produces an iteration summary table for the smoothing effects.

MAXITER=*number*

specifies the maximum number of iterations for the backfitting algorithm. The default value is 50.

MAXITSCORE=*number*

specifies the maximum number of iterations for the local score algorithm. The default value is 100.

METHOD=GCV

specifies that the value of the smoothing parameter should be selected by generalized cross validation. If you specify both METHOD=GCV and the DF= option for the smoothing effects, the user-specified DF= is used, and the METHOD=GCV option is ignored. See the "Selection of Smoothing Parameters" section on page 1575 for more details on the GCV method.

NOTEST

requests that the procedure not produce the "Analysis of Deviance" table. This option reduces the running time of the procedure.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set* < *keyword* · · · *keyword* > **;**

The OUTPUT statement creates a new SAS data set containing diagnostic measures calculated after fitting the model.

You can request a variety of diagnostic measures that are calculated for each observation in the data set. The new data set contains the variables specified in the MODEL statement in addition to the requested variables. If no *keyword* is present, the data set contains only the predicted values.

Details on the specifications in the OUTPUT statement are as follows.

OUT=*SAS-data-set*

specifies the name of the new data set to contain the diagnostic measures. This specification is required.

keyword

specifies the statistics to include in the output data set. The keywords and the statistics they represent are as follows:

PREDICTED	predicted values for each smoothing component and overall predicted values at design points
UCLM	upper confidence limits for each predicted smoothing component
LCLM	lower confidence limits for each predicted smoothing component
ADIAG	diagonal element of the hat matrix associated with the observation for each smoothing spline component
RESIDUAL	residual standardized by its weights
STD	standard deviation of the prediction for each smoothing component
ALL	implies all preceding keywords

The names of the new variables that contain the statistics are formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the variable name.

The prefixes of the new variables are as follows:

Keywords	**Prefix**
PRED	P_
UCLM	UCLM_
LCLM	LCLM_
ADIAG	ADIAG_
RESID	R_
STD	STD_ for spline STDP_ for loess

For example, suppose that you have a dependent variable y and an independent smoothing variable x, and you specify the keywords PRED and ADIAG. In this case, the output SAS data set will contain the variables P_y, P_x, and ADIAG_x.

SCORE Statement

SCORE *DATA=SAS-data-set OUT=SAS-data-set* **;**

The SCORE statement calculates predicted values for a new data set. The variables generated by the SCORE statement use the same naming conventions with prefixes as the OUTPUT statement. If you have multiple data sets to predict, you can specify multiple SCORE statements. You must use a SCORE statement for each data set.

The following options must be specified in the SCORE statement.

DATA=*SAS-data-set*
specifies an input SAS data set containing all the variables included in independent effects in the MODEL statement. The predicted response is computed for each observation in the SCORE DATA= data set.

OUT=*SAS-data-set*
specifies the name of the SAS data set to contain the predictions.

Details

Nonparametric Regression

Nonparametric regression relaxes the usual assumption of linearity and enables you to explore the data more flexibly, uncovering structure in the data that might otherwise be missed.

However, many forms of nonparametric regression do not perform well when the number of independent variables in the model is large. The sparseness of data in this setting causes the variances of the estimates to be unacceptably large unless the sample size is extremely large. The problem of rapidly increasing variance for increasing dimensionality is sometimes referred to as the “curse of dimensionality.” Interpretability is another problem with nonparametric regression based on kernel and smoothing spline estimates. The information these estimates contain about the relationship between the dependent and independent variables is often difficult to comprehend.

To overcome these difficulties, Stone (1985) proposed additive models. These models estimate an additive approximation to the multivariate regression function. The benefits of an additive approximation are at least twofold. First, since each of the individual additive terms is estimated using a univariate smoother, the curse of dimensionality is avoided, at the cost of not being able to approximate universally. Second, estimates of the individual terms explain how the dependent variable changes with the corresponding independent variables.

To extend the additive model to a wide range of distribution families, Hastie and Tibshirani (1990) proposed generalized additive models. These models enable the mean of the dependent variable to depend on an additive predictor through a nonlinear link function. The models permit the response probability distribution to be any member of the exponential family of distributions. Many widely used statistical models belong to this general class; they include additive models for Gaussian data, nonparametric logistic models for binary data, and nonparametric log-linear models for Poisson data.

Additive Models and Generalized Additive Models

This section describes the methodology and the fitting procedure behind generalized additive models.

Let Y be a response random variable and $X_1, X_2, \cdots, X_p$ be a set of predictor variables. A regression procedure can be viewed as a method for estimating the expected value of Y given the values of $X_1, X_2, \cdots, X_p$. The standard linear regression model assumes a linear form for the conditional expectation

$$E(Y|X_1, X_2, \cdots, X_p) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Given a sample, estimates of $\beta_0, \beta_1, \cdots, \beta_p$ are usually obtained by the least squares method.

The additive model generalizes the linear model by modeling the conditional expectation as

$$E(Y|X_1, X_2, \cdots, X_p) = s_0 + s_1(X_1) + s_2(X_2) + \cdots + s_p(X_p)$$

where $s_i(X), i = 1, 2, \ldots, p$ are smooth functions.

In order to be estimable, the smooth functions s_i have to satisfy standardized conditions such as $Es_j(X_j) = 0$. These functions are not given a parametric form but instead are estimated in a nonparametric fashion.

While traditional linear models and additive models can be used in most statistical data analysis, there are types of problems for which they are not appropriate. For example, the normal distribution may not be adequate for modeling discrete responses such as counts or bounded responses such as proportions.

Generalized additive models address these difficulties, extending additive models to many other distributions besides just the normal. Thus, generalized additive models can be applied to a much wider range of data analysis problems.

Similar to generalized linear models, generalized additive models consist of a random component, an additive component, and a link function relating the two components. The response Y, the random component, is assumed to have exponential family density

$$f_Y(y;\theta;\phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

where θ is called the natural parameter and ϕ is the scale parameter. The mean of the response variable μ is related to the set of covariates $X_1, X_2, \cdots, X_p$ by a link function g. The quantity

$$\eta = s_0 + \sum_{i=1}^{p} s_i(X_i)$$

defines the additive component, where $s_1(\cdot), \cdots, s_p(\cdot)$ are smooth functions, and the relationship between μ and η is defined by $g(\mu) = \eta$. The most commonly used link function is the canonical link, for which $\eta = \theta$.

Generalized additive models and generalized linear models can be applied in similar situations, but they serve different analytic purposes. Generalized linear models emphasize estimation and inference for the parameters of the model, while generalized additive models focus on exploring data nonparametrically. Generalized additive models are more suitable for exploring the data and visualizing the relationship between the dependent variable and the independent variables.

Backfitting and Local Scoring Algorithms

Much of the development and notation in this section follows Hastie and Tibshirani (1986). Consider the estimation of the smoothing terms $s_0, s_1(\cdot), \cdots, s_p(\cdot)$ in the additive model

$$\eta(X) = s_0 + \sum_{i=1}^{p} s_j(X_j)$$

where $E\left[s_j(X_j)\right] = 0$ for every j. Since the algorithm for additive models is the basis for fitting generalized additive models, the algorithm for additive models is discussed first.

Many ways are available to approach the formulation and estimation of additive models. The backfitting algorithm is a general algorithm that can fit an additive model using any regression-type fitting mechanisms.

Define the jth set of partial residuals as

$$R_j = Y - s_0 - \sum_{k \neq j} s_k(X_k)$$

then $E(R_j|X_j) = s_j(X_j)$. This observation provides a way for estimating each smoothing function $s_j(\cdot)$ given estimates $\{\hat{s}_i(\cdot), i \neq j\}$ for all the others. The resulting iterative procedure is known as the backfitting algorithm (Friedman and Stuetzle 1981). The following formulation is taken from Hastie and Tibshirani (1986).

The Backfitting Algorithm

The unweighted form of the backfitting algorithm is as follows:

1. **Initialization:**
   $s_0 = E(Y), s_1^1 = s_2^1 = \cdots = s_p^1 = 0, m = 0.$
2. **Iterate:**
   $m = m + 1$
   for $j = 1$ to p do:
   $\quad R_j = Y - s_0 - \sum_{k=1}^{j-1} s_k^m(X_k) - \sum_{k=j+1}^{p} s_k^{m-1}(X_k)$
   $\quad s_j^m = E(R_j|X_j).$
3. **Until:**
   RSS $=$ Avg$(Y - s_0 - \sum_{j=1}^{p} s_j^m(X_j))^2$ fails to decrease, or satisfies the convergence criterion.

In the preceding notation, $s_j^m(\cdot)$ denotes the estimate of $s_j(\cdot)$ at the mth iteration. It can be shown that with many smoothers (including linear regression, univariate and bivariate splines, and combinations of these), RSS never increases at any step. This implies that the algorithm always converges (Hastie and Tibshirani, 1986). Note, however, that for distributions other than Gaussian, numerical instabilities with weights may cause convergence problems. Even when the algorithm converges, the individual functions need not be unique, since dependence among the covariates can lead to more than one representation for the same fitted surface.

A weighted backfitting algorithm has the same form as for the unweighted case, except that the smoothers are weighted. In PROC GAM, weights are used with non-Gaussian data in the local scoring procedure described later in this section.

The GAM procedure uses the following condition as the convergence criterion for the backfitting algorithm:

$$\frac{\sum_{j=1}^{n} \sum_{i=1}^{k} (s_i^{m-1}(x_j) - s_i^m(x_j))^2}{1 + \sum_{j=1}^{n} \sum_{i=1}^{k} (s_i^{m-1}(x_j))^2} \quad \leq \quad \epsilon$$

where $\epsilon = 10^{-8}$ by default; you can change this with the EPSILON= option on the MODEL statement.

The algorithm so far described fits just additive models. The algorithm for generalized additive models is a little more complicated. Generalized additive models extend generalized linear models in the same manner that additive models extend linear regression models, that is, by replacing form $\alpha + \sum_j X_j\beta_j$ with the additive form $\alpha + \sum_j f_j(X_j)$. Refer to "Generalized Linear Models Theory" in Chapter 31, "The GENMOD Procedure," for more information.

PROC GAM fits generalized additive models using a modified form of adjusted dependent variable regression, as described for generalized linear models in McCullagh and Nelder (1989), with the additive predictor taking the role of the linear predictor. Hastie and Tibshirani (1986) call this the *local scoring algorithm*. Important components of this algorithm depend on the link function for each distribution, as shown in the following table.

Distribution	Link	Adjusted Dependent(Z)	Weights(w)
Normal	identity	y	1
Bin(n, μ)	logit	$\eta + (y-\mu)/n\mu(1-\mu)$	$n\mu(1-\mu)$
Gamma	$-1/\mu$	$\eta + (y-\mu)/\mu^2$	μ^2
Poisson	log	$\eta + (y-\mu)/\mu$	μ
Inverse Gaussian	$1/\mu^2$	$\eta - 2(y-\mu)/\mu^3$	$\mu^3/4$

Once the distribution and hence these quantities are defined, the local scoring algorithm proceeds as follows:

The General Local Scoring Algorithm

1. **Initialization:**
   $s_i = g(E(y)), s_1^0 = s_2^0 = \cdots = s_p^0 = 0, m = 0.$
2. **Iterate:**
   $m = m + 1$
   Form the predictor η_i, mean μ_i, weights w_i, and adjusted dependent variable z_i based on the previous iteration

$$\begin{aligned} \eta_i^{m-1} &= s_0 + \sum_{j=1}^{p} s_j^{m-1}(x_{ij}) \\ \mu_i^{m-1} &= g^{-1}(\eta_i^{m-1}) \\ w_i &= (\partial\mu_i/\partial\eta_i)_{m-1}^2 \, V_i^{-1} \\ z_i &= \eta_i^{m-1} + (y_i - \mu_i^{m-1})\,(\partial\eta_i/\partial\mu_i)_{m-1} \end{aligned}$$

   Fit an additive model to Z using the backfitting algorithm with weights W to obtain estimated functions $s_j^m(\cdot)$.
3. **Until:**
   The convergence criterion is satisfied or the deviance fails to decrease. The deviance is an extension to generalized models of the RSS; refer to "Goodness of Fit" in Chapter 31, "The GENMOD Procedure," for a definition.

The GAM procedure uses the following condition as the convergence criterion for local scoring:

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i)(s_i^{m-1}(x_j) - s_i^{m}(x_j))^2}{\sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i) + \sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i)(s_i^{m-1}(x_j))^2} \quad \leq \quad \epsilon^s$$

where $\epsilon^s = 10^{-8}$ by default; you can change this with the EPSSCORE= option on the MODEL statement.

The estimating procedure for generalized additive models consists of two loops. Inside each step of the local scoring algorithm (outer loop), a weighted backfitting algorithm (inner loop) is used until convergence or until the RSS fails to decrease. Then, based on the estimates from this weighted backfitting algorithm, a new set of weights is calculated and the next iteration of the scoring algorithm starts. The scoring algorithm stops when the convergence criterion is satisfied or the deviance of the estimates ceases to decrease.

Smoothers

A smoother is a tool for summarizing the trend of a response measurement Y as a function of one or more predictor measurements $X_1, \cdots, X_p$. It produces an estimate of the trend that is less variable than Y itself. An important property of a smoother is its nonparametric nature. It does not assume a rigid form for the dependence of Y on $X_1, \cdots, X_p$. This section gives a brief overview of the smoothers that can be used with the GAM procedure.

Cubic Smoothing Spline

A smoothing spline is the solution to the following optimization problem: among all functions $\eta(x)$ with two continuous derivatives, find one that minimizes the penalized least square

$$\sum_{i=1}^{n}(y_i - \eta(x_i))^2 + \lambda \int_a^b (\eta''(t))^2 dt$$

where λ is a fixed constant, and $a \leq x_1 \leq \cdots \leq x_n \leq b$. The first term measures closeness to the data while the second term penalizes curvature in the function. It can be shown that there exists an explicit, unique minimizer, and that minimizer is a natural cubic spline with knots at the unique values of x_i.

The value $\lambda/(1+\lambda)$ is the *smoothing parameter*. When λ is large, the smoothing parameter is close to 1, producing a smoother curve; small values of λ, corresponding to smoothing parameters near 0, are apt to produce rougher curves, more nearly interpolating the data.

Local Regression

Local regression was proposed by Cleveland, Devlin, and Grosse (1988). The idea of local regression is that at a predictor x, the regression function $\eta(x)$ can be locally approximated by the value of a function in some specified parametric class. Such a local approximation is obtained by fitting a regression surface to the data points within a chosen neighborhood of the point x. A weighted least squares algorithm is used to fit linear functions of the predictors at the centers of neighborhoods. The radius of each neighborhood is chosen so that the neighborhood contains a specified percentage of the data points. The smoothing parameter for the local regression procedure, which controls the smoothness of the estimated curve, is the fraction of the data in each local neighborhood. Data points in a given local neighborhood are weighted by a smooth decreasing function of their distance from the center of the neighborhood. Refer to Chapter 41, "The LOESS Procedure," for more details.

Thin-Plate Smoothing Spline

The thin-plate smoothing spline is a multivariate version of the cubic smoothing spline. The theoretical foundations for the thin-plate smoothing spline are described in Duchon (1976, 1977) and Meinguet (1979). Further results and applications are given in Wahba and Wendelberger (1980). Refer to Chapter 74, "The TPSPLINE Procedure," for more details.

Selection of Smoothing Parameters

CV and GCV

The smoothers discussed here have a single smoothing parameter. In choosing the smoothing parameter, cross validation can be used. Cross validation works by leaving points (x_i, y_i) out one at a time, estimating the squared residual for smooth function at x_i based on the remaining $n-1$ data points, and choosing the smoother to minimize the sum of those squared residuals. This mimics the use of training and test samples for prediction. The cross validation function is defined as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{\eta}_{\lambda}^{-i}(x_i))^2$$

where $\hat{\eta}_{\lambda}^{-i}(x_i)$ indicates the fit at x_i, computed by leaving out the ith data point. The quantity $nCV(\lambda)$ is sometimes called the prediction sum of squares or $PRESS$ (Allen 1974).

All of the smoothers fit by the GAM procedure can be formulated as a linear combination of the sample responses

$$\hat{\eta}(x) = A(\lambda)Y$$

for some matrix $A(\lambda)$, which depends on λ. (The matrix $A(\lambda)$ depends on x and the sample data, as well, but this dependence is suppressed in the preceding equation.) Let a_{ii} be the diagonal elements of $A(\lambda)$. Then the CV function can be expressed as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \hat{\eta}_\lambda(x_i))}{1 - a_{ii}}\right)^2$$

In most cases, it is very time consuming to compute the quantity a_{ii}. To solve this computational problem, Wahba (1990) has proposed the generalized cross validation function (GCV) that can be used to solve a wide variety of problems involving selection of a parameter to minimize the prediction risk.

The GCV function is defined as

$$GCV(\lambda) = \frac{n\sum_{i=1}^{n}(y_i - \hat{\eta}_\lambda(x_i))^2}{(n - \text{tr}(A(\lambda)))^2}$$

The GCV formula simply replaces the a_{ii} with tr$(A(\lambda))/n$. Therefore, it can be viewed as a weighted version of CV. In most of the cases of interest, GCV is closely related to CV but much easier to compute. Specify the METHOD=GCV option on the MODEL statement in order to use the GCV function to choose the smoothing parameters.

Degrees of Freedom

The estimated GAM model can be expressed as

$$\hat{\eta}(X) = \hat{s}_0 + \sum_{i=1}^{p} A_i(y,\lambda)Y$$

Because the weights are calculated based on previous iteration during the local scoring iteration, the matrices A_i may depend on Y for non-Gaussian data. However, for the final iteration, the A_i matrix for the spline smoothers has the same role as the projection matrix in linear regression and therefore, nonparametric degrees of freedom (DF) for the ith spline smoother can be defined as

$$DF(smoother) = \text{tr}(A_i(y,\lambda))$$

For LOESS smoothers A_i is not symmetric and so is not a projection matrix. In this case PROC GAM uses

$$DF(smoother) = \text{tr}(A_i(y,\lambda)'(A_i(y,\lambda))$$

The GAM procedure gives you the option of specifying the degrees of freedom for each individual smoothing component. If you choose a particular value for the degrees of freedom, then during every local scoring iteration the procedure will search for a corresponding smoothing parameter lambda that yields the specified value or comes as close as possible. The final estimate for the smoother during this local scoring iteration will be based on this lambda. Note that for univariate spline components, an additional degree of freedom is removed by default to account for the linear portion of the model, so the value displayed in the Fit Summary and Analysis of Deviance tables will be one less than the value you specify.

Confidence Intervals for Smoothers

In the GAM procedure, curvewise confidence intervals for smoothing splines and pointwise confidence intervals for loess are provided in the output dataset.

Curvewise Confidence Interval for Smoothing Spline

Viewing the spline model as a Bayesian model, Wahba (1983) proposed Bayesian confidence intervals for smoothing spline estimates as follows:

$$\hat{s}_\lambda(x_i) \pm z_{\alpha/2}\sqrt{\hat{\sigma}^2 a_{ii}(\lambda)}$$

where $a_{ii}(\lambda)$ is the ith diagonal element of the $A(\lambda)$ matrix and $z_{\alpha/2}$ is the $\alpha/2$ point of the normal distribution. The confidence intervals are interpreted as intervals "across the function" as opposed to point-wise intervals.

Suppose that you fit a spline estimate to experimental data that consists of a true function f and a random error term, ϵ_i. In repeated experiments, it is likely that about $100(1-\alpha)\%$ of the confidence intervals cover the corresponding true values, although some values are covered every time and other values are not covered by the confidence intervals most of the time. This effect is more pronounced when the true response curve or surface has small regions of particularly rapid change.

Pointwise Confidence Interval for Loess Smoothers

As defined in Cleveland, Devlin, and Grosse (1988), a standardized residual for a loess smoother follows a t distribution with ρ degrees of freedom, where ρ is called the "lookup degrees of freedom," defined as

$$\rho = \delta_1^2/\delta_2$$

where $\delta_1 = Trace(I - A(\lambda))^T(I - A(\lambda))$ and $\delta_2 = Trace((I - A(\lambda))^T(I - A(\lambda)))^2$. Therefore an approximate pointwise confidence interval at x_i is

$$\hat{s}_\lambda(x_i) \pm t_{\alpha/2;\rho}\hat{\sigma}(x_i)$$

where $\hat{\sigma}(x_i)$ is the estimate of the standard deviation.

Distribution Family and Canonical Link

In general, there is not just one reasonable link function for a given response variable distribution. For parametric models, the choice of link function can lead to substantively different estimates and tests. However, the inherent flexibility of nonparametric models makes them less likely to be sensitive to the precise choice of link function. Thus, for simplicity and computational efficiency, the GAM procedure uses only the canonical link for each distribution, as discussed below.

The Gaussian Model

With this model, the link function is the identity function, and the generalized additive model is the additive model.

The Binomial Model

A binomial response model assumes that the proportion of successes Y is such that Y has a $Bin(n, p(x))$ distribution. The $Bin(n, p(x))$ refers to the binomial distribution with parameters n and $p(x)$. Often the data are binary, in which case $n = 1$. The canonical link is

$$g(p) = log\frac{p}{n-p} = \eta$$

The Poisson Model

The link function for the Poisson model is the log function. Assuming that the mean of the Poisson distribution is $\mu(x)$, the dependence of $\mu(x)$ and independent variable $x_1, \cdots, x_k$ is

$$g(\mu) = log(\mu) = \eta$$

The Gamma Model

Let the mean of the Gamma distribution be $\mu(x)$. The canonical link function for the Gamma distribution is -1/$\mu(x)$. Therefore, the relationship between $\mu(x)$ and the independent variable $x_1, \cdots, x_k$ is

$$g(\mu) = -\frac{1}{\mu} = \eta$$

The Inverse Gaussian Model

Let the mean of the Inverse Gaussian distribution be $\mu(x)$. The canonical link function for inverse Gaussian distribution is $1/\mu^2$. Therefore, the relationship between $\mu(x)$ and the independent variable $x_1, \cdots, x_k$ is

$$g(\mu) = \frac{1}{\mu^2} = \eta$$

Dispersion Parameter

Continuous distributions in the exponential family (Gaussian, Gamma, and Inverse Gaussian) have a dispersion parameter that can be estimated by the scaled deviance. For these continuous response distributions, PROC GAM incorporates this dispersion parameter estimate into standard errors of the parameter estimates, prediction standard errors of spline components, and chi-square statistics. The discrete distributions used in GAM (Binomial and Poisson) do not have a dispersion parameter. For more details on the distributions, dispersion parameter, and deviance, refer to "Generalized Linear Models Theory" in Chapter 31, "The GENMOD Procedure."

Forms of Additive Models

Suppose that y is a continuous variable and x1 and x2 are two explanatory variables of interest. To fit an additive model, you can use a MODEL statement similar to that used in many regression procedures in the SAS System:

```
model y = spline(x1) spline(x2);
```

This model statement requires the procedure to fit the following model:

$$\eta(x1, x2) = \mathit{Intercept} + s_1(x1) + s_2(x2)$$

where the $s_i()$ terms denote nonparametric spline functions of the respective explanatory variables.

The GAM procedure can fit semiparametric models. The following MODEL statement assumes a linear relation with x1 and an unknown functional relation with x2:

```
model y = param(x1) spline(x2);
```

If you want to fit a model containing a functional two-way interaction between x1 and x2, you can use the following MODEL statement:

```
model y = spline2(x1,x2);
```

In this case, the GAM procedure fits a model equivalent to that of PROC TPSPLINE.

Estimates from PROC GAM

PROC GAM provides the ability to fit both nonparametric and semiparametric models. In order to better understand the underlying trend of any given factor, PROC GAM separates the linear trend from any general nonparametric trend during the fitting as well as in the final report. This makes it easy for you to determine whether the significance of a smoothing variable is associated with a simple linear trend or a more complicated pattern.

For example, suppose you want to fit a semiparametric model as

$$y = \alpha_0 + \alpha_1 z + s1(x1) + s2(x2)$$

The GAM estimate for this model is

$$y = \alpha_0 + \alpha_1 z + \beta_1 x1 + \beta_2 x2 + f_1(x1) + f_2(x2)$$

where f_1 and f_2 are linear-adjusted nonparametric estimates of the $s1$ and $s2$ effects. The *p*-values for $\alpha_0, \alpha_1, \beta_1,$ and β_2 are reported in the parameter estimates table. β_1 and β_2 are the estimates labeled Linear(x1) and Linear(x2) in the table. The *p*-values for f_1 and f_2 are reported in the analysis of deviance table.

Only $\hat{f}_1$, $\hat{f}_2$, and $\hat{y}$ are output to the output data set, with corresponding variable names P_x1, P_x2, and P_y, respectively. For Gaussian data, the complete marginal prediction for variable x1 is

$$\beta_1 x1 + P\_x1$$

For non-Gaussian data, an appropriate transformation is required to get back to the original data scale.

ODS Table Names

PROC GAM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, refer to Chapter 14, “Using the Output Delivery System.”

Table 30.2. ODS Tables Produced by PROC GAM

ODS Table Name	Description	Statement	Option
ANODEV	Analysis of Deviance table for smoothing variables	PROC	default
ClassSummary	Summary of class variables	PROC	default
ConvergenceStatus	Convergence status of the local score algorithm	PROC	default
InputSummary	Input data summary	PROC	default
IterHistory	Iteration history table	MODEL	ITPRINT
IterSummary	Iteration summary	PROC	default
FitSummary	Fit parameters and fit summary	PROC	default
ParameterEstimates	Parameter estimation for regression variables	PROC	default

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the GAM procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated below. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

When the ODS GRAPHICS are in effect, the GAM procedure will produce plots of the partial predictions for each nonparametric predictor in the model. Use the PLOTS option on the PROC GAM statement to control aspects of these plots.

PLOTS < **(***general-plot-options***)** > **=** *keywords* < **(***plot-options***)** >

specifies characteristics of the graphics produced when you use the experimental ODS GRAPHICS statement. You can specify the following *general-plot-options* in parentheses after the PLOTS option:

CLM specifies that smoothing component plots should include a 95% confidence band. Note that producing this band can be computationally intensive for large data sets.

COMMONAXES specifies that smoothing component plots within a single graphics panel should all have the same vertical axis limits. This enables you to visually judge relative effect size.

UNPACK|UNPACKPANELS specifies that multiple smoothing component plots that are collected into graphics panels should be additionally displayed separately. Use this option if you want to access individual smoothing component plots within the panel.

You can specify the following keywords as arguments to the PLOTS= option.

COMPONENTS < (*number-list*|ALL) > specifies that only particular smoothing component plots should be produced. Plots for successive smoothing components are named COMPONENT1, COMPONENT2, and so on. For example, specify PLOTS=COMPONENT(1 3) to produce only the first and the third smoothing component plots.

ODS Graph Names

PROC GAM assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 30.3.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Table 30.3. ODS Graphics Produced by PROC GAM

ODS Graph Name	**Plot Description**	**PLOTS= Option**
Component*i*	Partial prediction curve for smoothing component *i*	Component
SmoothingComponentPanel*i*	Panel *i* of multiple partial prediction curves	

By default, partial prediction plots for each component are displayed in panels of multiple plots named SmoothingComponentPanel1, SmoothingComponentPanel2, and so on. Use the PLOTS(UNPANEL) option on the PROC GAM statement to display these plots individually as well. Use the PLOTS(CLM) option to superimpose confidence limits for the partial predictions.

Examples

Example 30.1. Generalized Additive Model with Binary Data

The following example illustrates the capabilities of the GAM procedure and compares it to the GENMOD procedure.

The data used in this example are based on a study by Bell et al. (1994). Bell and his associates studied the result of multiple-level thoracic and lumbar laminectomy, a corrective spinal surgery commonly performed on children. The data in the study consist of retrospective measurements on 83 patients. The specific outcome of interest is the presence (1) or absence (0) of kyphosis, defined as a forward flexion of the spine of at least 40 degrees from vertical. The available predictor variables are Age in months at time of the operation, the starting of vertebrae levels involved in the operation (StartVert), and the number of levels involved (NumVert). The goal of this analysis is to identify risk factors for kyphosis. PROC GENMOD can be used to investigate the relationship among kyphosis and the predictors. The following DATA step creates the data kyphosis:

```
title 'Comparing PROC GAM with PROC GENMOD';
data kyphosis;
   input Age StartVert NumVert Kyphosis @@;
   datalines;
71  5  3  0     158 14 3 0    128 5   4 1
2   1  5  0     1   15 4 0    1   16  2 0
61  17 2  0     37  16 3 0    113 16  2 0
59  12 6  1     82  14 5 1    148 16  3 0
18  2  5  0     1   12 4 0    243 8   8 0
168 18 3  0     1   16 3 0    78  15  6 0
175 13 5  0     80  16 5 0    27  9   4 0
22  16 2  0     105 5  6 1    96  12  3 1
131 3  2  0     15  2  7 1    9   13  5 0
12  2 14  1     8   6  3 0    100 14  3 0
4   16 3  0     151 16 2 0    31  16  3 0
125 11 2  0     130 13 5 0    112 16  3 0
140 11 5  0     93  16 3 0    1   9   3 0
```

```
52  6  5  1     20  9  6 0    91  12  5 1
73  1  5  1     35  13 3 0    143 3   9 0
61  1  4  0     97  16 3 0    139 10  3 1
136 15 4  0     131 13 5 0    121 3   3 1
177 14 2  0     68  10 5 0    9   17  2 0
139 6  10 1     2   17 2 0    140 15  4 0
72  15 5  0     2   13 3 0    120 8   5 1
51  9  7  0     102 13 3 0    130 1   4 1
114 8  7  1     81  1  4 0    118 16  3 0
118 16 4  0     17  10 4 0    195 17  2 0
159 13 4  0     18  11 4 0    15  16  5 0
158 15 4  0     127 12 4 0    87  16  4 0
206 10 4  0     11  15 3 0    178 15  4 0
157 13 3  1     26  13 7 0    120 13  2 0
42  6  7  1     36  13 4 0
;

proc genmod;
   model Kyphosis = Age StartVert NumVert
                    / link=logit dist=binomial;
run;
```

Output 30.1.1. GENMOD Analysis: Partial Output

```
                         Comparing PROC GAM with PROC GENMOD

                                The GENMOD Procedure

PROC GENMOD is modeling the probability that Kyphosis='0'. One way to change
this to model the probability that Kyphosis='1' is to specify the DESCENDING
option in the PROC statement.


                           Analysis Of Parameter Estimates

                               Standard       Wald 95%          Chi-
   Parameter  DF  Estimate      Error    Confidence Limits   Square  Pr > ChiSq

   Intercept   1    1.2497     1.2424   -1.1853    3.6848      1.01      0.3145
   Age         1   -0.0061     0.0055   -0.0170    0.0048      1.21      0.2713
   StartVert   1    0.1972     0.0657    0.0684    0.3260      9.01      0.0027
   NumVert     1   -0.3031     0.1790   -0.6540    0.0477      2.87      0.0904
   Scale       0    1.0000     0.0000    1.0000    1.0000

NOTE: The scale parameter was held fixed.
```

The GENMOD analysis of the independent variable effects is shown in Output 30.1.1. Based on these results, the only significant factor is StartVert with a log odds ratio of -0.1972. The variable NumVert has a *p*-value of 0.0904 with a log odds ratio of 0.3031.

The GENMOD procedure assumes a strict linear relationship between the response and the predictors. The following SAS statements use PROC GAM to investigate a less restrictive model, with moderately flexible spline terms for each of the predictors:

```
title 'Comparing PROC GAM with PROC GENMOD';
proc gam data=kyphosis;
   model Kyphosis = spline(Age      ,df=3)
                    spline(StartVert,df=3)
                    spline(NumVert  ,df=3) / dist=binomial;
run;
```

The MODEL statement requests an additive model using a univariate smoothing spline for each term. The option dist=binomial with binary responses specifies a logistic model. Each term is fit using a univariate smoothing spline with three degrees of freedom. Of these three degrees of freedom, one is taken up by the linear portion of the fit and two are left for the nonlinear spline portion. Although this might seem to be an unduly modest amount of flexibility, it is better to be conservative with a data set this small.

Output 30.1.2 and Output 30.1.3 list the output from PROC GAM.

Output 30.1.2. Summary Statistics

```
                     Comparing PROC GAM with PROC GENMOD

                              The GAM Procedure
                         Dependent Variable: Kyphosis
Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

                           Summary of Input Data Set

                 Number of Observations                 83
                 Number of Missing Observations          0
                 Distribution                     Binomial
                 Link Function                       Logit


                     Iteration Summary and Fit Statistics

        Number of local score iterations                          9
        Local score convergence criterion               2.6635657E-9
        Final Number of Backfitting Iterations                    1
        Final Backfitting Criterion                     5.2326588E-9
        The Deviance of the Final Estimate              46.610922438
```

Output 30.1.3. Model Fit Statistics

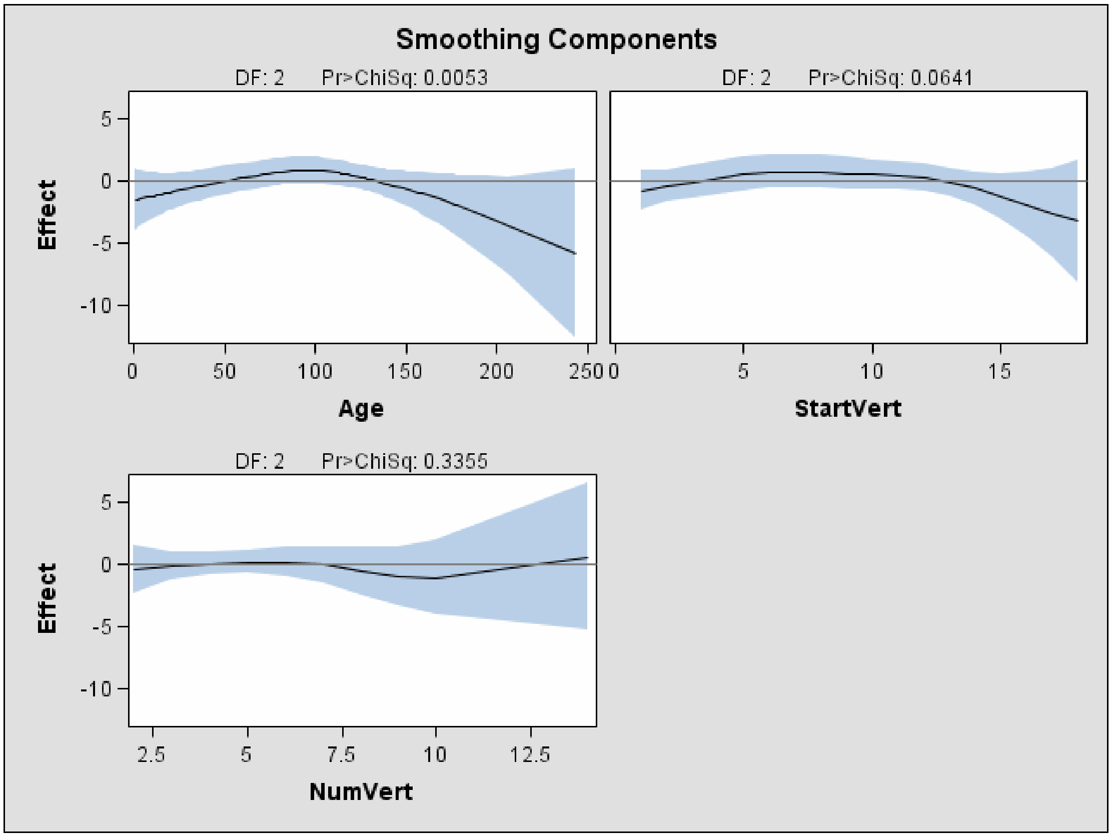
```
                     Comparing PROC GAM with PROC GENMOD

                              The GAM Procedure
                         Dependent Variable: Kyphosis
 Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

                          Regression Model Analysis
                             Parameter Estimates

                           Parameter       Standard
    Parameter               Estimate          Error    t Value    Pr > |t|

    Intercept               -2.01533        1.45620      -1.38      0.1706
    Linear(Age)              0.01213        0.00794       1.53      0.1308
    Linear(StartVert)       -0.18615        0.07628      -2.44      0.0171
    Linear(NumVert)          0.38347        0.19102       2.01      0.0484


                          Smoothing Model Analysis
                    Fit Summary for Smoothing Components

                                                                          Num
                        Smoothing                                      Unique
Component               Parameter            DF             GCV           Obs

Spline(Age)              0.999996      2.000000      328.512831            66
Spline(StartVert)        0.999551      2.000000      317.646685            16
Spline(NumVert)          0.921758      2.000000       20.144056            10


                           Smoothing Model Analysis
                             Analysis of Deviance

                                           Sum of
 Source                        DF         Squares    Chi-Square    Pr > ChiSq

 Spline(Age)              2.00000       10.494369       10.4944        0.0053
 Spline(StartVert)        2.00000        5.494968        5.4950        0.0641
 Spline(NumVert)          2.00000        2.184518        2.1845        0.3355
```

The critical part of the GAM results is the "Analysis of Deviance" table, shown in Output 30.1.3. For each smoothing effect in the model, this table gives a χ^2-test comparing the deviance between the full model and the model without this variable. In this case the analysis of deviance results indicates that the effect of Age is highly significant, the effect of StartVert is nearly significant, and the effect of NumVert is insignificant at the 5% level. Plots of predictions against predictor can be used to investigate why PROC GAM and PROC GENMOD produce different results. The following statements use the experimental ODS GRAPHICS statement to produce plots of the individual smoothing components. The CLM suboption for the PLOTS option adds a curvewise Bayesian confidence band to each smoothing component, while the COMMONAXES suboption forces all three smoothing component plots to share the same vertical axis limits, allowing a visual judgment of nonparametric effect size.

```
ods html;
ods graphics on;

proc gam data=kyphosis plots(clm commonaxes);
   model Kyphosis = spline(Age      ,df=3)
                    spline(StartVert,df=3)
                    spline(NumVert  ,df=3) / dist=binomial;
run;

ods graphics off;
ods html close;
```

For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the GAM procedure, see the section “ODS Graphics” on page 1581. The smoothing component plots are displayed in Output 30.1.4.

Output 30.1.4. Partial Prediction for Each Predictor (Experimental)



The plots show that the partial predictions corresponding to both Age and StartVert have a quadratic pattern, while NumVert has a more complicated but weaker pattern. However, in the plot for NumVert, notice that about half the vertical range of the function is determined by the point at the upper extreme. It would be a good idea, therefore, to rerun the analysis without this point, to explore how much it affects the conclusions. You can do this by simply including a WHERE clause when specifying the data set for the GAM procedure, as in the following code:

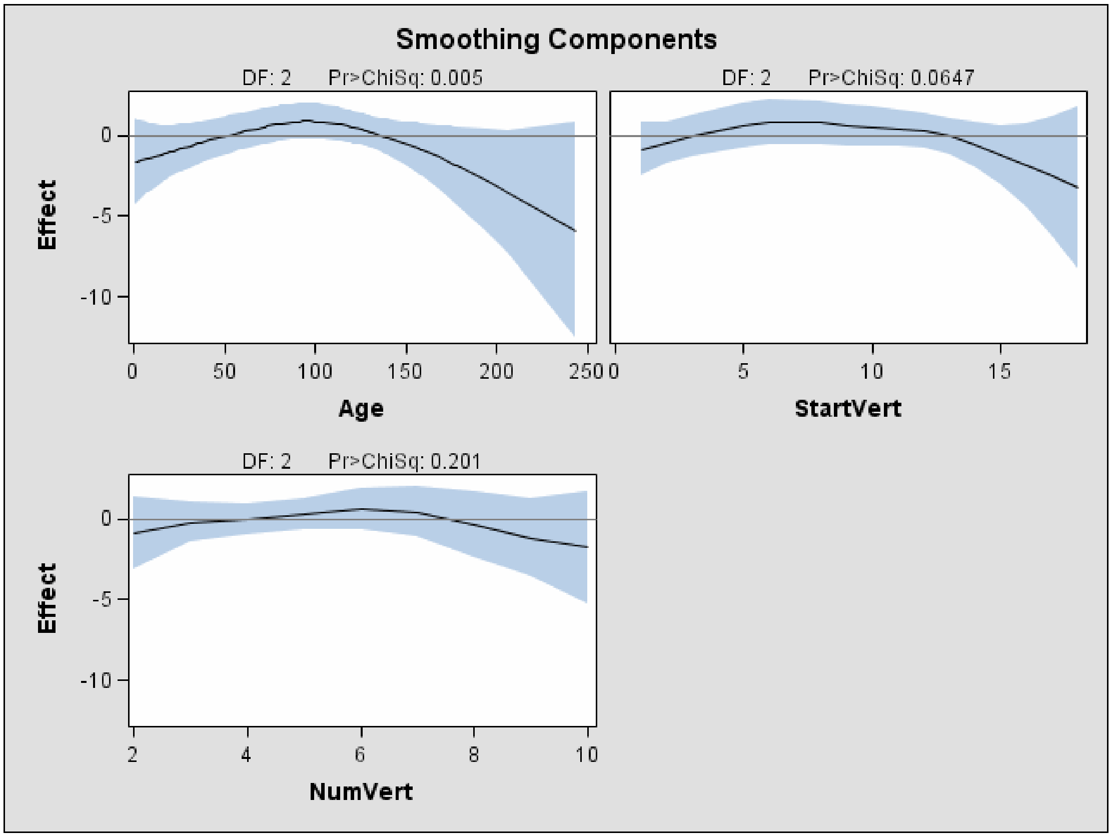
```
ods html;
ods graphics on;

proc gam data=kyphosis(where=(NumVert^=14)) plots(clm commonaxes);
   model Kyphosis = spline(Age      ,df=3)
                    spline(StartVert,df=3)
                    spline(NumVert  ,df=3) / dist=binomial;
run;

ods graphics off;
ods html close;
```

Output 30.1.5. Analysis After Removing NumVert=14

```
                    Comparing PROC GAM with PROC GENMOD

                             The GAM Procedure
                        Dependent Variable: Kyphosis
Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

                          Smoothing Model Analysis
                            Analysis of Deviance

                                           Sum of
Source                          DF        Squares    Chi-Square    Pr > ChiSq

Spline(Age)                2.00000      10.587556       10.5876        0.0050
Spline(StartVert)          2.00000       5.477094        5.4771        0.0647
Spline(NumVert)            2.00000       3.209089        3.2091        0.2010
```

The analysis of deviance table from this reanalysis is shown in Output 30.1.5, and Output 30.1.6 shows the recomputed partial predictor plots.

Output 30.1.6. Partial Prediction After Removing NumVert=14 (Experimental)



Removing data point NumVert=14 has little effect on either the analysis of deviance results or the estimated curves for StartVert and NumVert. But the removal has a noticeable effect on the variable NumVert, whose curve now also seems quadratic, though it is much less pronounced than for the other two variables.

An important difference between the first analysis of this data with GENMOD and the subsequent analysis with GAM is that GAM indicates that age is a significant predictor of kyphosis. The difference is due to the fact that the GENMOD model only includes a linear effect in Age whereas the GAM model allows a more complex relationship, which the plots indicate is nearly quadratic. Having used the GAM procedure to discover an appropriate form of the dependence of Kyphosis on each of the three independent variables, you can use the GENMOD procedure to fit and assess the corresponding parametric model. The following code fits a GENMOD model with quadratic terms for all three variables, including tests for the joint linear and quadratic effects of each variable. The resulting contrast tests are shown in Output 30.1.7.

```
title 'Comparing PROC GAM with PROC GENMOD';
proc genmod data=kyphosis(where=(NumVert^=14));
   model kyphosis = Age       Age      *Age
                    StartVert StartVert*StartVert
                    NumVert   NumVert  *NumVert
                    /link=logit  dist=binomial;
   contrast 'Age'       Age       1, Age*Age             1;
   contrast 'StartVert' StartVert 1, StartVert*StartVert 1;
   contrast 'NumVert'   NumVert   1, NumVert*NumVert     1;
run;
```

Output 30.1.7. Joint Linear and Quadratic Tests

```
                      Comparing PROC GAM with PROC GENMOD

                              The GENMOD Procedure

PROC GENMOD is modeling the probability that Kyphosis='0'. One way to change
this to model the probability that Kyphosis='1' is to specify the DESCENDING
option in the PROC statement.


                                Contrast Results

                                      Chi-
            Contrast         DF     Square    Pr > ChiSq    Type

            Age               2      13.63        0.0011    LR
            StartVert         2      15.41        0.0005    LR
            NumVert           2       3.56        0.1684    LR
```

The results for the quadratic GENMOD model are now quite consistent with the GAM results.

From this example, you can see that PROC GAM is very useful in visualizing the data and detecting the nonlinearity among the variables.

Example 30.2. Poisson Regression Analysis of Component Reliability

In this example, the number of maintenance repairs on a complex system are modeled as realizations of Poisson random variables. The system under investigation has a large number of components, which occasionally break down and are replaced or repaired. During a four-year period, the system was observed to be in a state of steady operation, meaning that the rate of operation remained approximately constant. A monthly maintenance record is available for that period, which tracks the number of components removed for maintenance each month. The data are listed in the following statements that create a SAS data set.

```
title 'Analysis of Component Reliability';
data equip;
   input year month removals @@;
   datalines;
1987   1  2 1987   2  4 1987   3  3
1987   4  3 1987   5  3 1987   6  8
1987   7  2 1987   8  6 1987   9  3
1987  10  9 1987  11  4 1987  12  10
1988   1  4 1988   2  6 1988   3  4
1988   4  4 1988   5  3 1988   6  5
1988   7  3 1988   8  4 1988   9  5
1988  10  3 1988  11  6 1988  12  3
1989   1  2 1989   2  6 1989   3  1
1989   4  5 1989   5  5 1989   6  4
1989   7  2 1989   8  2 1989   9  2
1989  10  5 1989  11  1 1989  12  10
1990   1  3 1990   2  8 1990   3  12
1990   4  7 1990   5  3 1990   6  2
1990   7  4 1990   8  3 1990   9  0
1990  10  6 1990  11  6 1990  12  6
;
run;
```

For planning purposes, it is of interest to understand the long- and short-term trends in the maintenance needs of the system. Over the long term, it is suspected that the quality of new components and repair work improves over time, so the number of component removals would tend to decrease from year to year. It is not known whether the robustness of the system is affected by seasonal variations in the operating environment, but this possibility is also of interest.

Because the maintenance record is in the form of counts, the number of removals are modeled as realizations of Poisson random variables. Denote by λ_{ij} the unobserved component removal rate for year i and month j. Since the data were recorded at regular intervals (from a system operating at a constant rate), each λ_{ij} is assumed to be a function of year and month only.

A preliminary two-way analysis is performed using PROC GENMOD to make broad inferences on repair trends. A log-link is specified for the model

$$\log \lambda_{ij} \quad = \quad \mu + \alpha_i^Y + \alpha_j^M$$

where μ is a grand mean, α_i^Y is the effect of the ith year, and α_j^M is the effect of the jth month. A CLASS statement declares the variables year and month as categorical. Type III sum of squares are requested to test whether there is an overall effect of year and/or month.

```
title2 'Two-way model';
proc genmod data=equip;
   class year month;
   model removals=year month
         / dist=Poisson link=log type3;
run;
```

Output 30.2.1. PROC GENMOD Listing for Type III Analysis

```
                  Analysis of Component Reliability
                           Two-way model

                        The GENMOD Procedure

                  LR Statistics For Type 3 Analysis

                                        Chi-
             Source           DF      Square    Pr > ChiSq

             year              3        2.63        0.4527
             month            11       21.12        0.0321
```

Output 30.2.1 displays the listed Type III statistics for the fitted model. With the test for year effects yielding a *p*-value of 0.4527, there is no evidence of a long-term trend in maintenance rates. Apparently, the quality of new or repaired components did not change between 1987 and 1990. However, the test for monthly trends does yield a small *p*-value of 0.0321, indicating that seasonal trends are just barely significant at the $\alpha = 0.05$ level.

The Type III tests indicate that the year term may be dropped from the model. The focus of the analysis is now on identifying the form of the underlying seasonal trend, which is a task that PROC GAM is especially suited for. PROC GAM will be used to fit both a reduced categorical model, with year eliminated, and a nonparametric spline model. Although PROC GENMOD also has the capability to fit categorical models, as demonstrated above, PROC GAM will be used to fit both models for a better comparison.

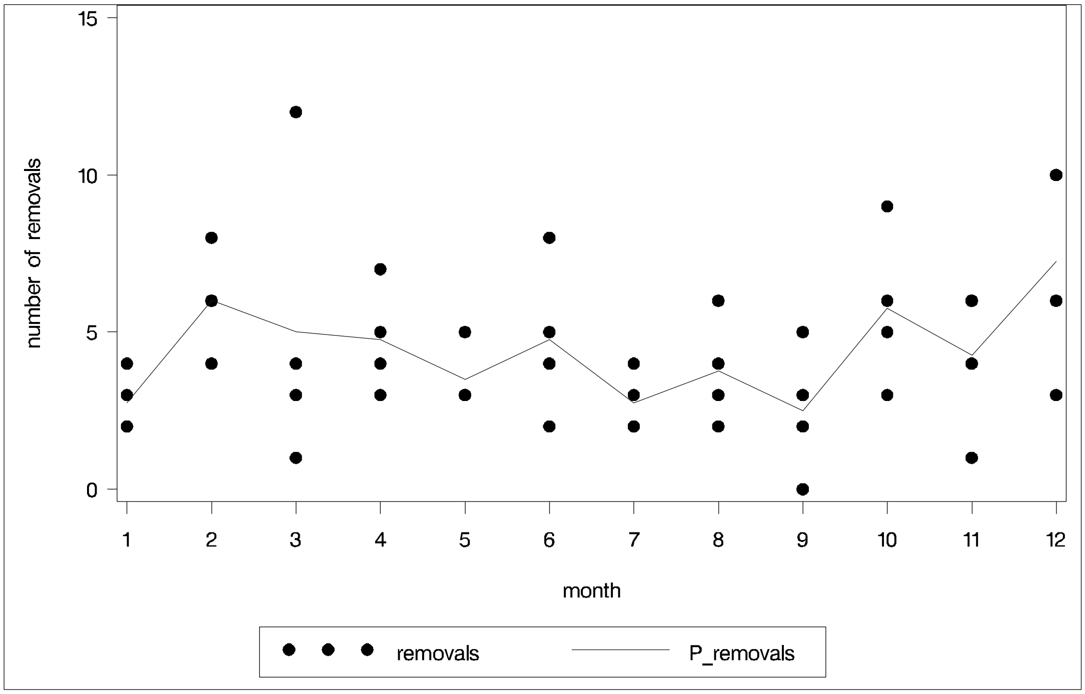
The following PROC GAM statements specify the reduced categorical model. For this part of the analysis, a CLASS statement is again used to specify that month is a categorical variable. In the follow-up, the seasonal effect will be treated as a nonparametric function of month.

```
title2 'One-way model';
proc gam data=equip;
   class month;
   model removals=param(month)
                  / dist=Poisson;
   output out=est predicted;
run;
```

The following statements generate a plot of the estimated seasonal trend. Note that the predicted values in the output data set correspond to the *logarithms* of the λ_{ij}, and so the exponential function is applied to put them on the scale of the original data. The plot is displayed in Output 30.2.2.

```
proc sort data=est out=plot;
   by month;
run;

data plot;
   set plot;
   P_removals = exp(P_removals);
run;

legend1 frame cframe=ligr cborder=black label=none
        position=center;
axis1   minor=none order=(0 to 15 by 5)
        label=(angle=90 rotate=0 "number of removals");
axis2   minor=none label=("month");
symbol1 color=black interpol=none value=dot;
symbol2 color=blue  interpol=join value=none line=1;

title;
proc gplot data=plot;
   plot removals*month=1 P_removals*month=2
          / overlay cframe=ligr legend=legend1 frame
            vaxis=axis1 haxis=axis2;
run; quit;
```

Output 30.2.2. Predicted Seasonal Trend from a Parametric Model Fit Using a CLASS Statement



The predicted repair rates shown in Output 30.2.2 form a jagged seasonal pattern. Ignoring the month-to-month fluctuations, which are difficult to explain and may be artifacts of random noise, the general removal rate trend starts by increasing at the beginning of the year; the trend flattens out in February and then decreases through

August; it flattens out again in September and begins an increasing trend that continues throughout the rest of the year.

One advantage of nonparametric regression is its ability to highlight general trends in the data, such as those described above, and attribute local fluctuations to unexplained random noise. The nonparametric regression model used by PROC GAM specifies that the underlying removal rates λ_j are of the form

$$\log \lambda_j = \beta_0 + \beta_1 \mathit{Month}_j + s(\mathit{Month}_j)$$

where β_1 is a linear coefficient and $s()$ is a nonparametric regression function. β_1 and $s()$ define the linear and nonparametric parts, respectively, of the seasonal trend.

The following statements request that PROC GAM fit a cubic spline model to the monthly repair data. The output listing is displayed in Output 30.2.3.

```
title 'Analysis of Component Reliability';
title2 'Spline model';
proc gam data=equip;
   model removals=spline(month)
                 / dist=Poisson method=gcv;
run;
```

The METHOD=GCV option is used to determine an appropriate level of smoothing. The keywords LCLM and UCLM in the OUTPUT statement request that lower and upper 95% confidence bounds on each $s(\mathit{Month}_j)$ be included in the output data set.

Output 30.2.3. PROC GAM Listing for Cubic Spline Regression Using the METHOD=GCV Option

```
                       Analysis of Component Reliability
                                  Spline model

                               The GAM Procedure
                          Dependent Variable: removals
                  Smoothing Model Component(s): spline(month)

                           Summary of Input Data Set

                   Number of Observations                 48
                   Number of Missing Observations          0
                   Distribution                      Poisson
                   Link Function                         Log


                     Iteration Summary and Fit Statistics

          Number of local score iterations                          5
          Local score convergence criterion              7.241527E-12
          Final Number of Backfitting Iterations                    1
          Final Backfitting Criterion                    1.710339E-11
          The Deviance of the Final Estimate             56.901543546
```

Output 30.2.4. Model Fit Statistics

```
                                 Spline model

                              The GAM Procedure
                        Dependent Variable: removals
                  Smoothing Model Component(s): spline(month)

                           Regression Model Analysis
                              Parameter Estimates

                          Parameter       Standard
   Parameter               Estimate          Error     t Value    Pr > |t|

   Intercept                1.34594        0.14509        9.28      <.0001
   Linear(month)            0.02274        0.01893        1.20      0.2362


                           Smoothing Model Analysis
                      Fit Summary for Smoothing Components

                                                                      Num
                     Smoothing                                     Unique
 Component           Parameter               DF             GCV       Obs

 Spline(month)        0.901512         1.879980        0.115848        12


                            Smoothing Model Analysis
                              Analysis of Deviance

                                          Sum of
  Source                      DF         Squares     Chi-Square    Pr > ChiSq

  Spline(month)          1.87998        8.877764         8.8778        0.0103
```

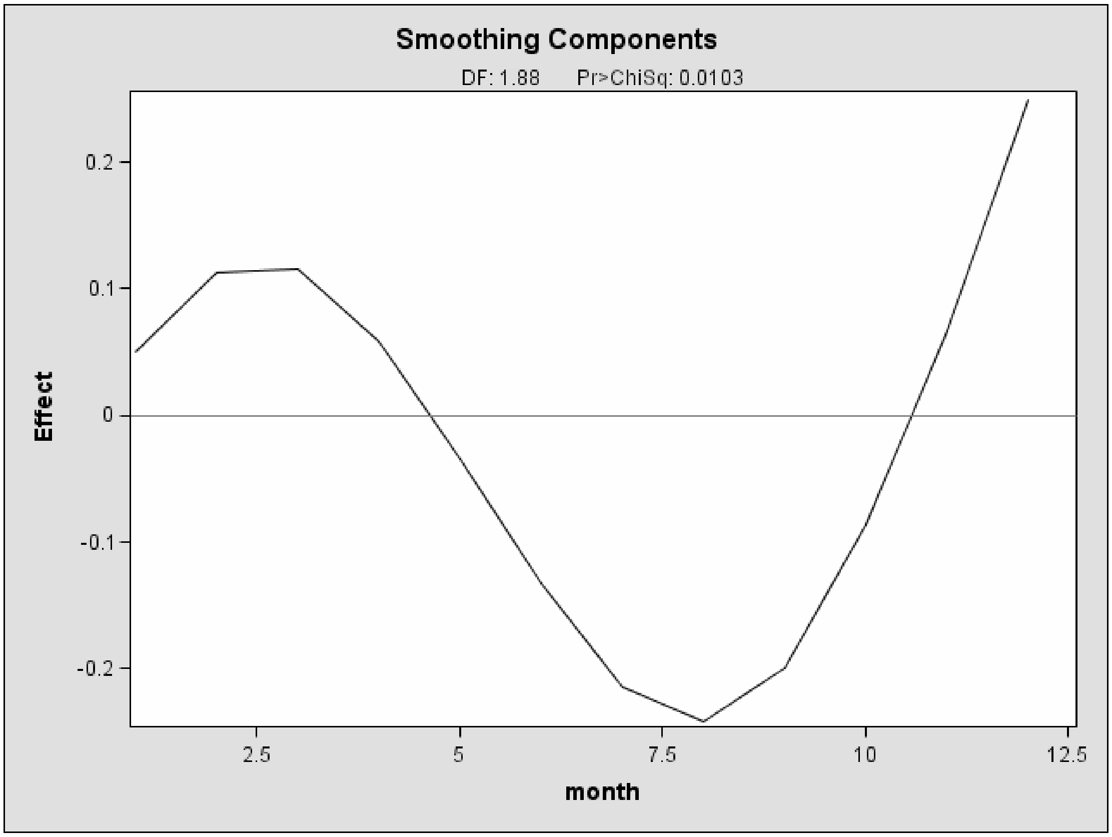
Notice in the listing of Output 30.2.4 that the DF value chosen for the nonlinear portion of the spline by minimizing GCV is about 1.88, which is smaller than the default value of 3. This indicates that the spline model of the seasonal trend is relatively simple. As indicated by the “Analysis of Deviance” table, it is a significant feature of the data: the table lists a *p*-value of 0.0103 for the hypothesis of no seasonal trend. Note also that the “Parameter Estimates” table lists a *p*-value of 0.2362 for the hypothesis of no linear factor in the seasonal trend.

The following statements use the experimental ODS GRAPHICS statement to plot the smoothing component for the effect of Month on predicted repair rates.

```
ods html;
ods graphics on;

proc gam data=equip;
   model removals=spline(month)
                 / dist=Poisson method=gcv;
run;

ods graphics off;
ods html close;
```

For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the GAM procedure, see the section "ODS Graphics" on page 1581. The smoothing component plot is displayed in Output 30.2.5.

Output 30.2.5. Predicted Seasonal Trend from a Cubic Spline Model (Experimental)



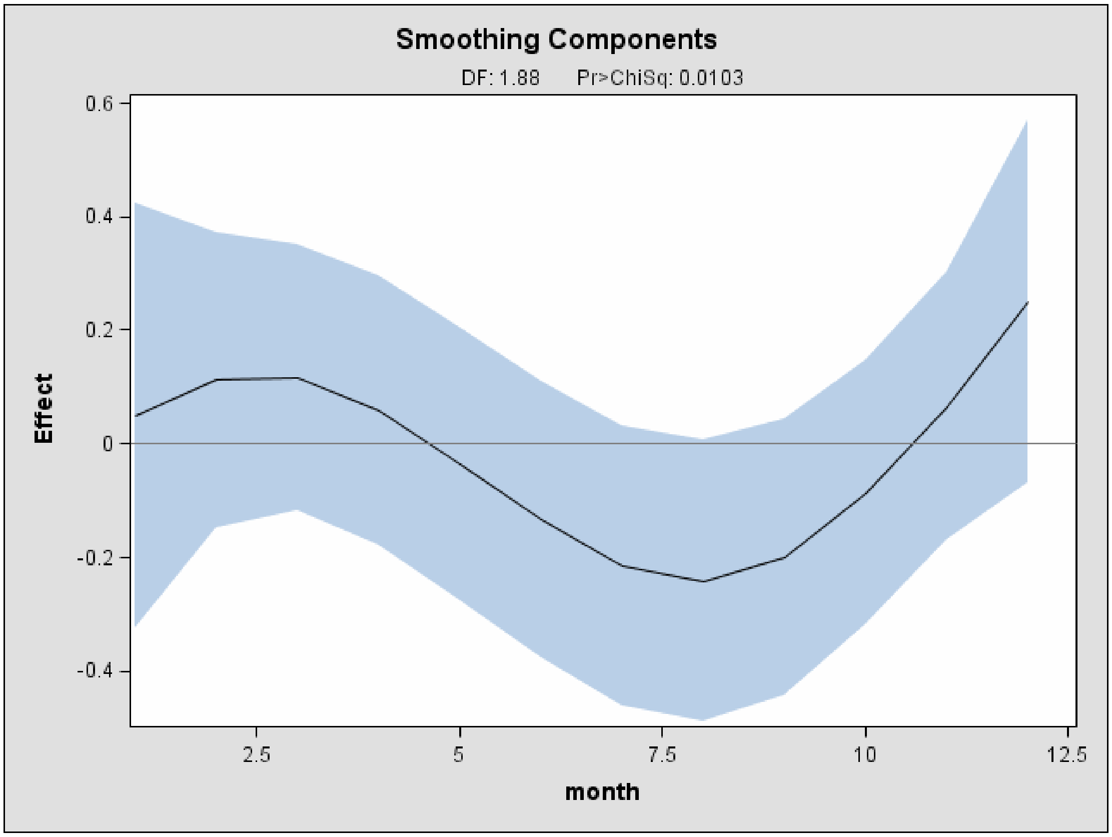
In Output 30.2.5, it is apparent that the pattern of repair rates follows the general pattern observed in Output 30.2.2. However, the plot of Output 30.2.5, is much cleaner as the month-to-month fluctuations are smoothed out to reveal the broader seasonal trend.

You can use the PLOTS(CLM) option to added a 95% confidence band to the plot of $s(\mathit{Month}_j)$, as in the following statements.

```
ods html; ods graphics on;

proc gam data=equip plots(clm);
   model removals=spline(month)
                 / dist=Poisson method=gcv;
run;

ods graphics off; ods html close;
```

The plot is displayed in Output 30.2.6.

Output 30.2.6. Estimated Nonparametric Factor of Seasonal Trend, Along with 95% Confidence Bounds (Experimental)



The small *p*-value in Output 30.2.1 of 0.0321 for the hypothesis of no seasonal trend indicates that the data exhibit significant seasonal structure. However, Output 30.2.6 is a graphical illustration of a degree of indistinctness in that structure. For instance, the horizontal reference line at zero is entirely within the 95% confidence band; that is, the estimated nonlinear part of the trend is relatively flat. Thus, despite evidence of seasonality based on the parametric model, it is difficult to narrow down its significant effects to a specific part of the year.

Example 30.3. Comparing PROC GAM with PROC LOESS

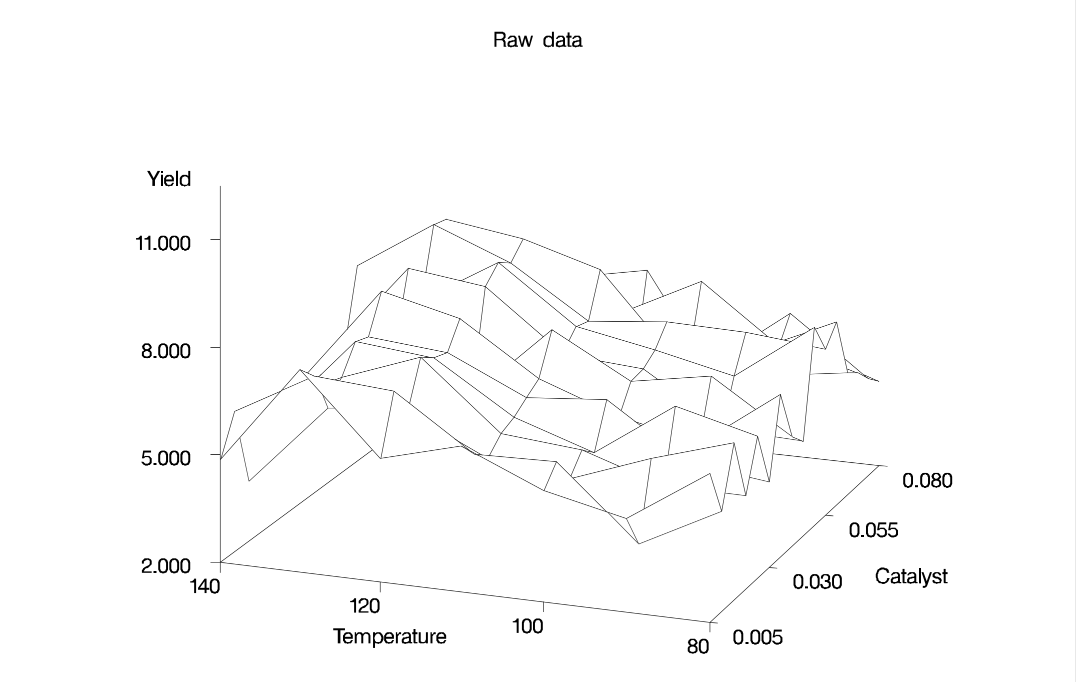
In an analysis of simulated data from a hypothetical chemistry experiment, additive nonparametric regression performed by PROC GAM is compared to the unrestricted multidimensional procedure of PROC LOESS.

In each repetition of the experiment, a catalyst is added to a chemical solution, thereby inducing synthesis of a new material. The data are measurements of the temperature of the solution, the amount of catalyst added, and the yield of the chemical reaction. The following code reads and plots the raw data.

```
data ExperimentA;
   format Temperature f4.0 Catalyst f6.3 Yield f8.3;
   input Temperature Catalyst Yield @@;
   datalines;
 80 0.005 6.039  80 0.010 4.719  80 0.015 6.301
 80 0.020 4.558  80 0.025 5.917  80 0.030 4.365
 80 0.035 6.540  80 0.040 5.063  80 0.045 4.668
 80 0.050 7.641  80 0.055 6.736  80 0.060 7.255
 80 0.065 5.515  80 0.070 5.260  80 0.075 4.813
 80 0.080 4.465  90 0.005 4.540  90 0.010 3.553
 90 0.015 5.611  90 0.020 4.586  90 0.025 6.503
 90 0.030 4.671  90 0.035 4.919  90 0.040 6.536
 90 0.045 4.799  90 0.050 6.002  90 0.055 6.988
 90 0.060 6.206  90 0.065 5.193  90 0.070 5.783
 90 0.075 6.482  90 0.080 5.222 100 0.005 5.042
100 0.010 5.551 100 0.015 4.804 100 0.020 5.313
100 0.025 4.957 100 0.030 6.177 100 0.035 5.433
100 0.040 6.139 100 0.045 6.217 100 0.050 6.498
100 0.055 7.037 100 0.060 5.589 100 0.065 5.593
100 0.070 7.438 100 0.075 4.794 100 0.080 3.692
110 0.005 6.005 110 0.010 5.493 110 0.015 5.107
110 0.020 5.511 110 0.025 5.692 110 0.030 5.969
110 0.035 6.244 110 0.040 7.364 110 0.045 6.412
110 0.050 6.928 110 0.055 6.814 110 0.060 8.071
110 0.065 6.038 110 0.070 6.295 110 0.075 4.308
110 0.080 7.020 120 0.005 5.409 120 0.010 7.009
120 0.015 6.160 120 0.020 7.408 120 0.025 7.123
120 0.030 7.009 120 0.035 7.708 120 0.040 5.278
120 0.045 8.111 120 0.050 8.547 120 0.055 8.279
120 0.060 8.736 120 0.065 6.988 120 0.070 6.283
120 0.075 7.367 120 0.080 6.579 130 0.005 7.629
130 0.010 7.171 130 0.015 5.997 130 0.020 6.587
130 0.025 7.335 130 0.030 7.209 130 0.035 8.259
130 0.040 6.530 130 0.045 8.400 130 0.050 7.218
130 0.055 9.167 130 0.060 9.082 130 0.065 7.680
130 0.070 7.139 130 0.075 7.275 130 0.080 7.544
140 0.005 4.860 140 0.010 5.932 140 0.015 3.685
140 0.020 5.581 140 0.025 4.935 140 0.030 5.197
140 0.035 5.559 140 0.040 4.836 140 0.045 5.795
140 0.050 5.524 140 0.055 7.736 140 0.060 5.628
140 0.065 6.644 140 0.070 3.785 140 0.075 4.853
140 0.080 6.006
;

title2 ’Raw data’;
proc g3d data=ExperimentA;
   plot Temperature*Catalyst=Yield
        / zmin=2 zmax=11;
run;
```

The plot is displayed in Output 30.3.1.

Output 30.3.1. Surface Plot of Yield by Temperature and Amount of Catalyst



A surface fitted to the plot of Output 30.3.1 by PROC LOESS will be of a very general (and flexible) type, since the procedure requires only weak assumptions about the structure of the dependencies among the data. PROC GAM, on the other hand, makes stronger structural assumptions by restricting the fitted surface to an additive form. These differences will be demonstrated in this example.

The following code requests that both PROC LOESS and PROC GAM fit surfaces to the data.

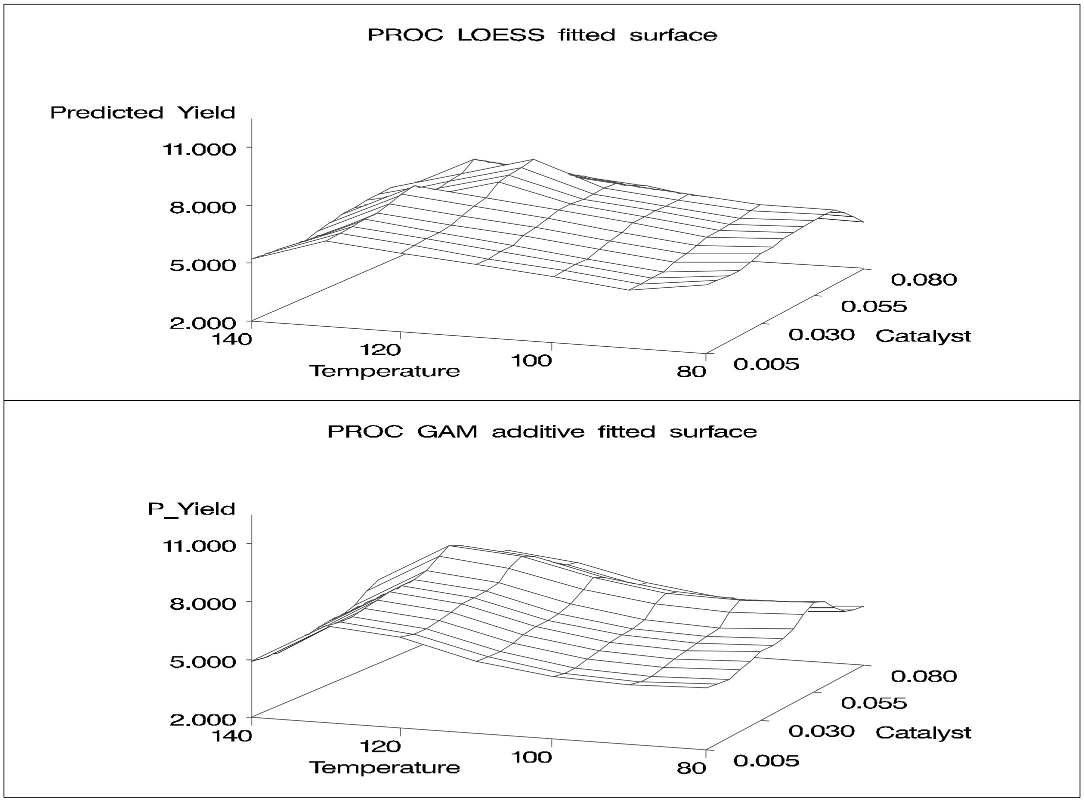
```
ods output OutputStatistics=PredLOESS;
proc loess data=ExperimentA;
   model Yield = Temperature Catalyst
         / scale=sd degree=2 select=gcv;
run;
ods output close;

proc gam data=ExperimentA;
   model Yield = loess(Temperature) loess(Catalyst)
                 / method=gcv;
   output out=PredGAM;
run;
```

In both cases the smoothing parameter was chosen as the value that minimizes GCV. This is performed automatically by PROC LOESS and PROC GAM.

The following code generates plots of the predicted yields, which are displayed in Output 30.3.2.

```
title2 'PROC LOESS fitted surface';
proc g3d data=PredLOESS;
   format pred f6.3;
   plot Temperature*Catalyst=pred
        / name='LOESSA' zmin=2 zmax=11;
run;

title2 'PROC GAM additive fitted surface';
proc g3d data=PredGAM;
   format P_Yield f6.3;
   plot Temperature*Catalyst=P_Yield
        / name='GAMA' zmin=2 zmax=11;
run;

goptions display;
proc greplay nofs tc=sashelp.templt template=v2;
   igout=gseg;
   treplay 1:loessa 2:gama;
run; quit;
```

Output 30.3.2. Fitted Regression Surfaces



Though both PROC LOESS and PROC GAM use the statistical technique loess, it is apparent from Output 30.3.2 that the manner in which it is applied is very different. By smoothing out the data in local neighborhoods, PROC LOESS essentially fits a surface to the data in pieces, one neighborhood at a time. The local regions are treated independently, so separate areas of the fitted surface are only weakly related. PROC GAM imposes additive structure, requiring that cross sections of the fitted surface always have the same shape and thereby relating regions that have a common value of the same individual regressor variable. Under that restriction, the loess technique need not be applied to the entire multidimensional scatter plot, but only to one-dimensional cross sections of the data.

The advantage of using additive model fitting is that its statistical power is directed toward univariate smoothing, and so it is able to discern the finer details of any underlying structure in the data. Regression data may be very sparse when viewed in the context of multidimensional space, even when every individual set of regressor values densely covers its range. This is the familiar curse of dimensionality. Sparse data greatly restricts the effectiveness of nonparametric procedures, but additive model fitting, when appropriate, is one way to overcome this limitation.

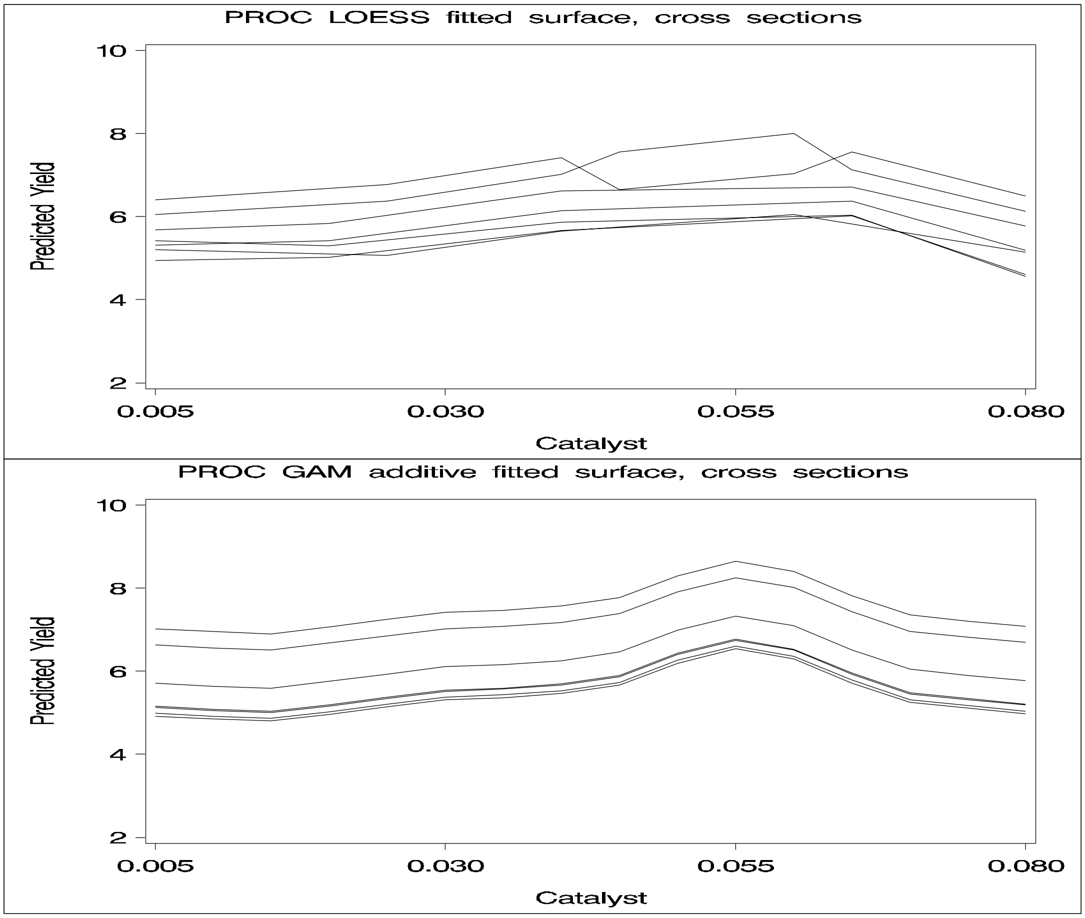
To examine these properties, plots of cross sections of unrestricted (PROC LOESS) and additive (PROC GAM) fitted surfaces for the variable Catalyst are generated by the following code. The code for creating the cross section plots and overlaying them is somewhat complicated, so a macro %XPlot is employed to make it easy to create this plot for the results of each procedure.

```
axis1 minor=none order=(2 to 11 by 2)
      label=(angle=90 rotate=0 "Predicted Yield");
axis2 minor=none order=(0.005 to 0.080 by 0.025)
      label=("Catalyst");
symbol1 color=blue interpol=join value=none
        line=1 width=1;

%macro XPLOT(proc=,name=);

proc sort data=Pred&proc;
   by Catalyst Temperature;
run;

data PredX&proc;
   keep Pred80 Pred90 Pred100 Pred110 Pred120 Pred130
        Pred140  Catalyst;
   array xPred{8:14} Pred80 Pred90 Pred100 Pred110
         Pred120 Pred130 Pred140;
   retain Pred80 Pred90 Pred100 Pred110 Pred120
          Pred130 Pred140;
   set Pred&proc;
   %if &proc=LOESS %then %do;
       xPred{Temperature/10} = pred;
   %end;
   %else %if &proc=GAM %then %do;
         xPred{Temperature/10} = P_Yield;
   %end;
   if abs(Temperature-140)<1 then output;
run;

proc gplot data=PredX&proc;
   plot Pred140*Catalyst=1 Pred130*Catalyst=1
```

```
        Pred120*Catalyst=1 Pred110*Catalyst=1
        Pred100*Catalyst=1 Pred90*Catalyst=1
        Pred80*Catalyst=1
        / overlay cframe=ligr name=&name
          vaxis=axis1 haxis=axis2;
run; quit;

%mend;

title;
title2 'PROC LOESS fitted surface, cross sections';
%XPLOT(proc=LOESS,name='XLOESSA');
title2 'PROC GAM additive fitted surface, cross sections';
%XPLOT(proc=GAM,name='XGAMA');

goptions display;
proc greplay nofs tc=sashelp.templt template=v2;
     igout=gseg;
     treplay 1:xloessa 2:xgama;
run; quit;
```

The plots are displayed in Output 30.3.3.

Output 30.3.3. Cross sections of Fitted Regression Surfaces



Notice that the graphs in the top panel (PROC LOESS) of Output 30.3.3 have varying shapes, while every graph in the bottom panel (PROC GAM) is the same curve shifted vertically. This illustrates precisely the kind of structural differences that distinguish additive models. A second important comparison to make in Output 30.3.2 and Output 30.3.3 is the level of detail in the fitted regression surfaces. Cross sections of the PROC LOESS surface are rather flat, but those of the additive surface have a clear shape. In particular, the ridge near Catalyst=0.055 is only vaguely evident in the PROC LOESS surface, but it is plainly revealed by the additive procedure.

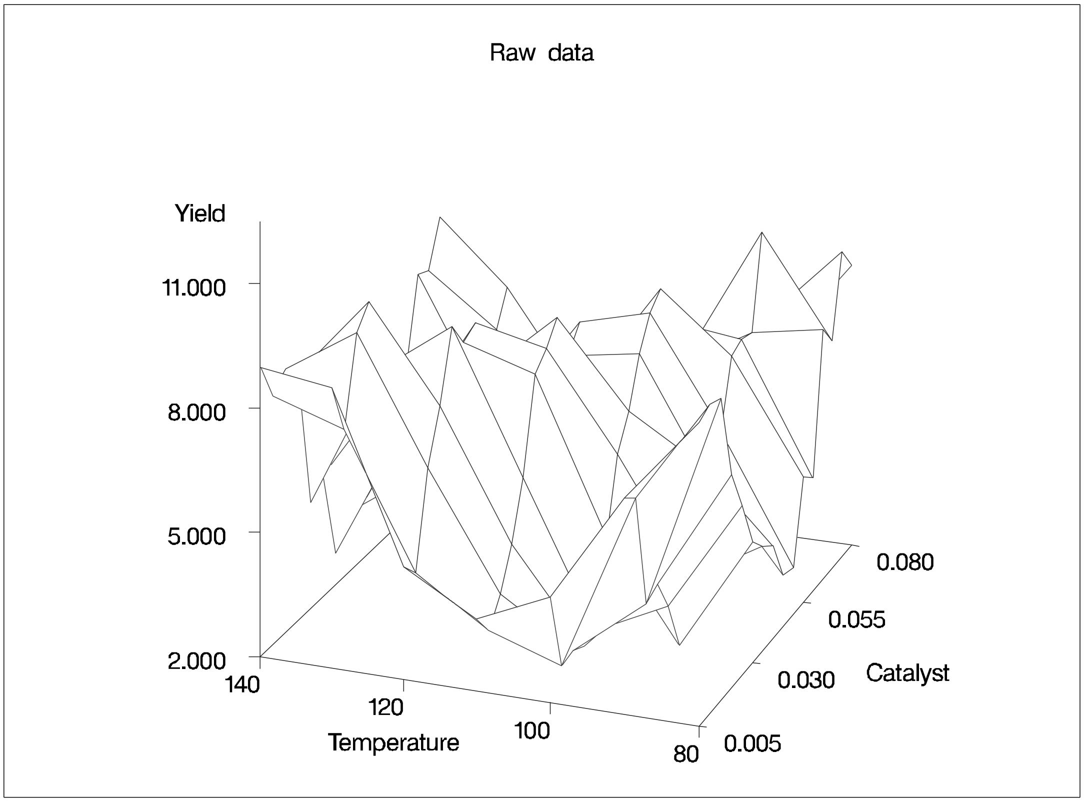
For an example of a situation where unrestricted multidimensional fitting is preferred over additive regression, consider the following simulated data from a similar experiment. The following code creates another SAS data set and plot.

```
data ExperimentB;
   format Temperature f4.0 Catalyst f6.3 Yield f8.3;
   input Temperature Catalyst Yield @@;
   datalines;
 80 0.005  9.115  80 0.010  9.275  80 0.015  9.160
 80 0.020  7.065  80 0.025  6.054  80 0.030  4.899
 80 0.035  4.504  80 0.040  4.238  80 0.045  3.232
 80 0.050  3.135  80 0.055  5.100  80 0.060  4.802
 80 0.065  8.218  80 0.070  7.679  80 0.075  9.669
 80 0.080  9.071  90 0.005  7.085  90 0.010  6.814
 90 0.015  4.009  90 0.020  4.199  90 0.025  3.377
 90 0.030  2.141  90 0.035  3.500  90 0.040  5.967
 90 0.045  5.268  90 0.050  6.238  90 0.055  7.847
 90 0.060  7.992  90 0.065  7.904  90 0.070 10.184
 90 0.075  7.914  90 0.080  6.842 100 0.005  4.497
100 0.010  2.565 100 0.015  2.637 100 0.020  2.436
100 0.025  2.525 100 0.030  4.474 100 0.035  6.238
100 0.040  7.029 100 0.045  8.183 100 0.050  8.939
100 0.055  9.283 100 0.060  8.246 100 0.065  6.927
100 0.070  7.062 100 0.075  5.615 100 0.080  4.687
110 0.005  3.706 110 0.010  3.154 110 0.015  3.726
110 0.020  4.634 110 0.025  5.970 110 0.030  8.219
110 0.035  8.590 110 0.040  9.097 110 0.045  7.887
110 0.050  8.480 110 0.055  6.818 110 0.060  7.666
110 0.065  4.375 110 0.070  3.994 110 0.075  3.630
110 0.080  2.685 120 0.005  4.697 120 0.010  4.268
120 0.015  6.507 120 0.020  7.747 120 0.025  9.412
120 0.030  8.761 120 0.035  8.997 120 0.040  7.538
120 0.045  7.003 120 0.050  6.010 120 0.055  3.886
120 0.060  4.897 120 0.065  2.562 120 0.070  2.714
120 0.075  3.141 120 0.080  5.081 130 0.005  8.729
130 0.010  7.460 130 0.015  9.549 130 0.020 10.049
130 0.025  8.131 130 0.030  7.553 130 0.035  6.191
130 0.040  6.272 130 0.045  4.649 130 0.050  3.884
130 0.055  2.522 130 0.060  4.366 130 0.065  3.272
130 0.070  4.906 130 0.075  6.538 130 0.080  7.380
140 0.005  8.991 140 0.010  8.029 140 0.015  8.417
140 0.020  8.049 140 0.025  4.608 140 0.030  5.025
140 0.035  2.795 140 0.040  3.123 140 0.045  3.407
140 0.050  4.183 140 0.055  3.750 140 0.060  6.316
```

```
140 0.065  5.799 140 0.070  7.992 140 0.075  7.835
140 0.080  8.985
;
run;

title2 'Raw data';
proc g3d data=ExperimentB;
   plot Temperature*Catalyst=Yield
        / zmin=2 zmax=11;
run;
```

A plot of the raw data is displayed in Output 30.3.4.

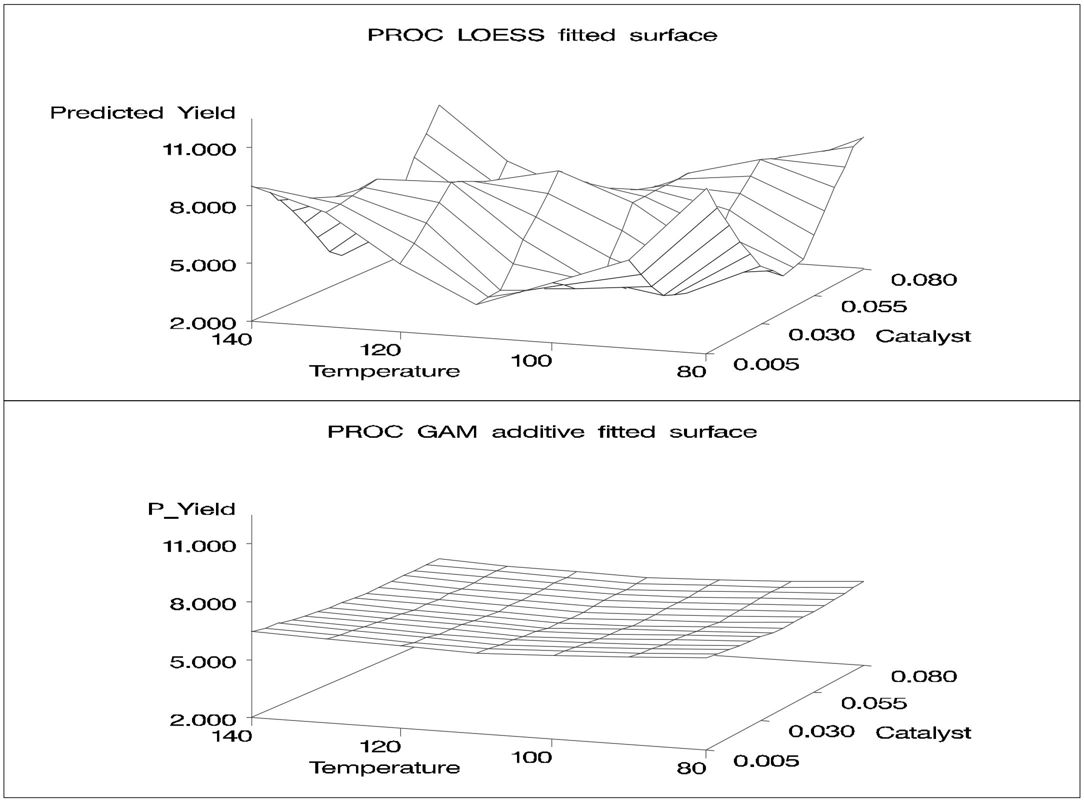
Output 30.3.4. Raw Data from Experiment B



Though the surface displayed in Output 30.3.4 is quite jagged, a distinct feature of the plot is a large ridge that runs diagonally across its surface. One would expect that the ridge would appear in the fitted regression surface of an appropriate nonparametric procedure. Nevertheless, between PROC LOESS and PROC GAM, only PROC LOESS is able to capture this significant feature.

The SAS code for fitting the new data is essentially the same as that for the data set from the first experiment. Similar to Output 30.3.2, multivariate and additive fitted surfaces for these data are displayed in Output 30.3.5.

Output 30.3.5. Fitted Regression Surfaces



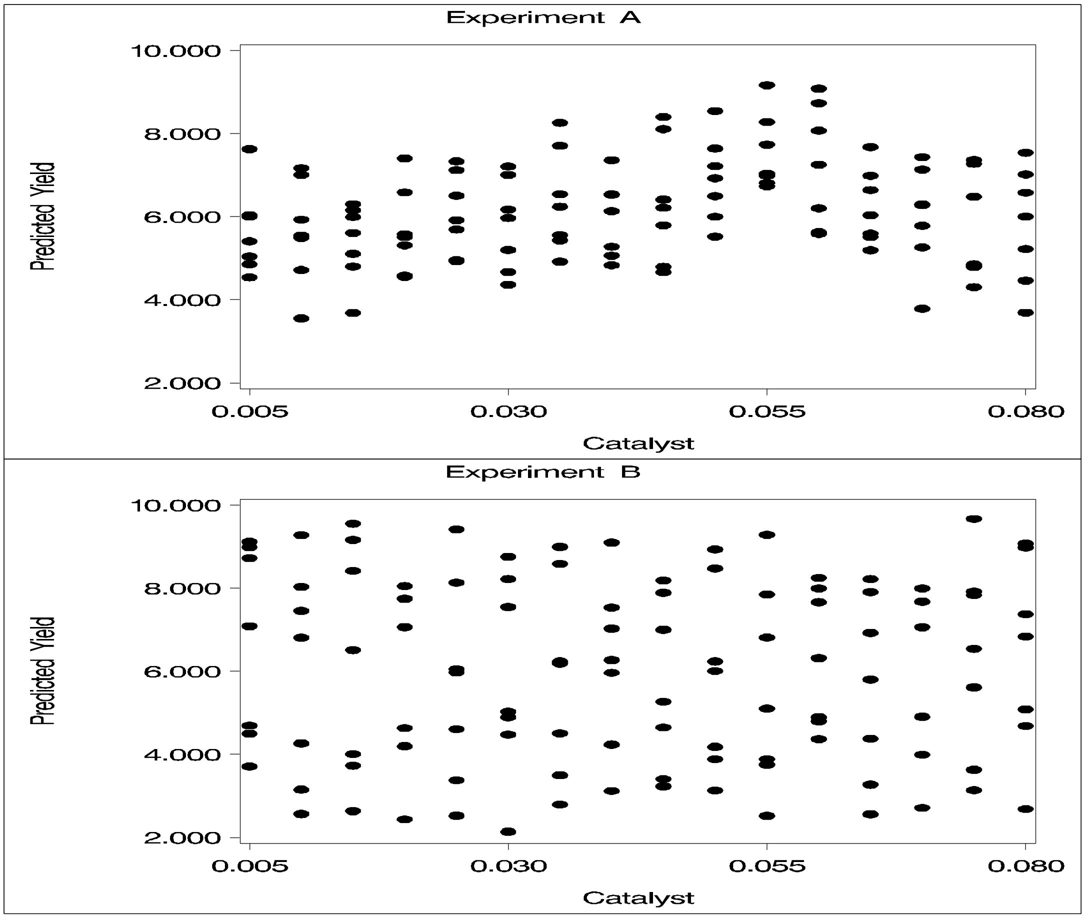
It is clear from Output 30.3.5 that the results of PROC LOESS and PROC GAM are completely different. While the plot in the top panel resembles the raw data plot in Output 30.3.4, the plot in the bottom panel is essentially featureless.

To understand what is happening, compare the scatter plots of Yield by Catalyst for the two data sets in this example. These are generated by the following code and displayed in Output 30.3.6.

```
axis1 minor=none order=(2 to 11 by 2)
      label=(angle=90 rotate=0 "Predicted Yield");
axis2 minor=none order=(0.005 to 0.080 by 0.025) label=("Catalyst");
symbol2 c=yellow v=dot i=none;

title2 'Experiment A';
proc gplot data=ExperimentA;
   plot Yield*Catalyst=2
        / cframe=ligr name='ExptA'
          vaxis=axis1 haxis=axis2;
run; quit;

title2 'Experiment B';
proc gplot data=ExperimentB;
     plot Yield*Catalyst=2
          / cframe=ligr name='ExptB'
            vaxis=axis1 haxis=axis2;
run; quit;
```

```
proc greplay nofs tc=sashelp.templt template=v2;
     igout=gseg;
     treplay 1:expta 2:exptb;
run; quit;
```

Output 30.3.6. Scatterplots of Yield by Catalyst



The top panel of Output 30.3.6 hints at the same kind of structure exhibited in the fitted cross sections of Output 30.3.3. In PROC GAM, the additive model component corresponding to Catalyst is fit to a similar scatter plot, with the partial residuals computed in the backfitting algorithm, so it is able to capture the trend seen here. In contrast, when the second data set is viewed from the perspective of Output 30.3.6, the diagonal ridge apparent in Output 30.3.4 is washed out, and no clear structure shows up in the scatter plot. As a result, the additive model fit produced by PROC GAM is relatively featureless.

References

Allen, D.M. (1974), “The Relationship Between Variable Selection and Data Augmentation and a Method of Prediction,” *Technometrics*, 16, 125–127.

Bell, D.F., Walker J.L., O’Connor G., Tibshirani R. (1994), “Spinal Deformity After Multiple-Level Cervical Laminectomy in Children.” *Spine*, 19, 406–411.

Cleveland, W.S., Devlin, S.J., and Grosse, E. (1988), “Regression by Local Fitting,” *Journal of Econometrics*, 37, 87–114.

Duchon, J. (1976), “Fonctions-Spline et Esperances Conditionnelles de Champs Gaussiens,” *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14, 19–27.

Duchon, J. (1977), “Splines Minimizing Rotation-Invariant Semi-Norms in Sovolev Spaces,” in *Constructive Theory of Functions of Several Variables*, ed. W. Schempp and K. Zeller, New York: Springer-Verlag, 85–100.

Friedman, J.H. and Stuetzle, W. (1981), “Projection Pursuit Regression,” *Journal of the American Statistical Association*, 76, 817–823.

Hastie, T.J. and Tibshirani, R.J. (1986), “Generalized Additive Models (with discussion),” *Statistical Science*, 1, 297–318.

Hastie, T.J. and Tibshirani, R.J. (1990), *Generalized Additive Models*, New York: Chapman and Hall.

Houghton, A.N., Flannery, J., and Viola, M.V. (1980), “Malignant Melanoma in Connecticut and Denmark,” *International Journal of Cancer*, 25, 95–104.

McCullagh, P. and Nelder, J.A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman and Hall.

Meinguet, J. (1979), “Multivariate Interpolation at Arbitrary Points Made Simple,” *Journal of Applied Mathematics and Physics (ZAMP)*, 30, 292–304.

Nelder, J.A. and Wedderburn, R.W.M. (1972), “Generalized Linear Models,” *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

SAS Institute Inc. (1999), *SAS Language Reference: Concepts, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS/STAT User’s Guide, Version 8*, Cary, NC: SAS Institute Inc.

Sockett, E.B., Daneman, D., Clarson, C., and Ehrich, R.M. (1987), “Factors Affecting and Patterns of Residual Insulin Secretion During the First Year of Type I (Insulin Dependent) Diabetes Mellitus in Children,” *Diabet*, 30, 453–459.

Stone, C.J. (1985), “Additive Regression and Other Nonparametric Models,” *Annals of Statistics*, 13, 689–705.

Wahba, G. (1983), “Bayesian ‘Confidence Intervals’ for the Cross Validated Smoothing Spline,” *Journal of the Royal Statistical Society, Series B*, 45, 133–150.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: Society for Industrial and Applied Mathematics.

Wahba, G. and Wendelberger, J. (1980), “Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation,” *Monthly Weather Review*, 108, 1122–1145.

Chapter 31
The GENMOD Procedure

Chapter Contents

Chapter 31
The GENMOD Procedure

Overview

The GENMOD procedure fits generalized linear models, as defined by Nelder and Wedderburn (1972). The class of generalized linear models is an extension of traditional linear models that allows the mean of a population to depend on a *linear predictor* through a nonlinear *link function* and allows the response probability distribution to be any member of an exponential family of distributions. Many widely used statistical models are generalized linear models. These include classical linear models with normal errors, logistic and probit models for binary data, and log-linear models for multinomial data. Many other useful statistical models can be formulated as generalized linear models by the selection of an appropriate link function and response probability distribution. Refer to McCullagh and Nelder (1989) for a discussion of statistical modeling using generalized linear models. The books by Aitkin, Anderson, Francis, and Hinde (1989) and Dobson (1990) are also excellent references with many examples of applications of generalized linear models. Firth (1991) provides an overview of generalized linear models.

The analysis of correlated data arising from repeated measurements when the measurements are assumed to be multivariate normal has been studied extensively. However, the normality assumption may not always be reasonable; for example, different methodology must be used in the data analysis when the responses are discrete and correlated. Generalized Estimating Equations (GEEs) provide a practical method with reasonable statistical efficiency to analyze such data.

Liang and Zeger (1986) introduced GEEs as a method of dealing with correlated data when, except for the correlation among responses, the data can be modeled as a generalized linear model. For example, correlated binary and count data in many cases can be modeled in this way.

The GENMOD procedure can fit models to correlated responses by the GEE method. You can use PROC GENMOD to fit models with most of the correlation structures from Liang and Zeger (1986) using GEEs. Refer to Liang and Zeger (1986), Diggle, Liang, and Zeger (1994), and Lipsitz, Fitzmaurice, Orav, and Laird (1994) for more details on GEEs.

Experimental graphics are now available with the GENMOD procedure for model assessment. For more information, see the "ODS Graphics" section on page 1695.

What Is a Generalized Linear Model?

A traditional linear model is of the form

$$y_i = \mathbf{x}_i{}'\boldsymbol{\beta} + \varepsilon_i$$

where y_i is the response variable for the ith observation. The quantity $\mathbf{x}_i$ is a column vector of covariates, or explanatory variables, for observation i that is known from the experimental setting and is considered to be fixed, or nonrandom. The vector of unknown coefficients $\boldsymbol{\beta}$ is estimated by a least squares fit to the data $\mathbf{y}$. The ε_i are assumed to be independent, normal random variables with zero mean and constant variance. The expected value of y_i, denoted by μ_i, is

$$\mu_i = \mathbf{x}_i{}'\boldsymbol{\beta}$$

While traditional linear models are used extensively in statistical data analysis, there are types of problems for which they are not appropriate.

- It may not be reasonable to assume that data are normally distributed. For example, the normal distribution (which is continuous) may not be adequate for modeling counts or measured proportions that are considered to be discrete.
- If the mean of the data is naturally restricted to a range of values, the traditional linear model may not be appropriate, since the linear predictor $\mathbf{x}_i{}'\boldsymbol{\beta}$ can take on any value. For example, the mean of a measured proportion is between 0 and 1, but the linear predictor of the mean in a traditional linear model is not restricted to this range.
- It may not be realistic to assume that the variance of the data is constant for all observations. For example, it is not unusual to observe data where the variance increases with the mean of the data.

A generalized linear model extends the traditional linear model and is, therefore, applicable to a wider range of data analysis problems. A generalized linear model consists of the following components:

- The linear component is defined just as it is for traditional linear models:

  $$\eta_i = \mathbf{x}_i{}'\boldsymbol{\beta}$$

- A monotonic differentiable link function g describes how the expected value of y_i is related to the linear predictor η_i:

  $$g(\mu_i) = \mathbf{x}_i{}'\boldsymbol{\beta}$$

- The response variables y_i are independent for i = 1, 2,... and have a probability distribution from an exponential family. This implies that the variance of the response depends on the mean μ through a *variance function* V:

  $$\mathit{var}(y_i) = \frac{\phi V(\mu_i)}{w_i}$$

where ϕ is a constant and w_i is a known weight for each observation. The *dispersion parameter* ϕ is either known (for example, for the binomial or Poisson distribution, $\phi = 1$) or it must be estimated.

See the section "Response Probability Distributions" on page 1650 for the form of a probability distribution from the exponential family of distributions.

As in the case of traditional linear models, fitted generalized linear models can be summarized through statistics such as parameter estimates, their standard errors, and goodness-of-fit statistics. You can also make statistical inference about the parameters using confidence intervals and hypothesis tests. However, specific inference procedures are usually based on asymptotic considerations, since exact distribution theory is not available or is not practical for all generalized linear models.

Examples of Generalized Linear Models

You construct a generalized linear model by deciding on response and explanatory variables for your data and choosing an appropriate link function and response probability distribution. Some examples of generalized linear models follow. Explanatory variables can be any combination of continuous variables, classification variables, and interactions.

Traditional Linear Model

- response variable: a continuous variable
- distribution: normal
- link function: identity $g(\mu) = \mu$

Logistic Regression

- response variable: a proportion
- distribution: binomial
- link function: logit $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$

Poisson Regression in Log Linear Model

- response variable: a count
- distribution: Poisson
- link function: log $g(\mu) = \log(\mu)$

Gamma Model with Log Link

- response variable: a positive, continuous variable
- distribution: gamma
- link function: log $g(\mu) = \log(\mu)$

The GENMOD Procedure

The GENMOD procedure fits a generalized linear model to the data by maximum likelihood estimation of the parameter vector $\boldsymbol{\beta}$. There is, in general, no closed form solution for the maximum likelihood estimates of the parameters. The GENMOD procedure estimates the parameters of the model numerically through an iterative fitting process. The dispersion parameter ϕ is also estimated by maximum likelihood or, optionally, by the residual deviance or by Pearson's chi-square divided by the degrees of freedom. Covariances, standard errors, and are computed for the estimated parameters based on the asymptotic normality of maximum likelihood estimators.

A number of popular link functions and probability distributions are available in the GENMOD procedure. The built-in link functions are

- identity: $g(\mu) = \mu$
- logit: $g(\mu) = \log(\mu/(1-\mu))$
- probit: $g(\mu) = \Phi^{-1}(\mu)$, where Φ is the standard normal cumulative distribution function
- power: $g(\mu) = \begin{cases} \mu^{\lambda} & \text{if } \lambda \neq 0 \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$
- log: $g(\mu) = \log(\mu)$
- complementary log-log: $g(\mu) = \log(-\log(1-\mu))$

The available distributions and associated variance functions are

- normal: $V(\mu) = 1$
- binomial (proportion): $V(\mu) = \mu(1-\mu)$
- Poisson: $V(\mu) = \mu$
- gamma: $V(\mu) = \mu^2$
- inverse Gaussian: $V(\mu) = \mu^3$
- negative binomial: $V(\mu) = \mu + k\mu^2$
- multinomial

The negative binomial is a distribution with an additional parameter k in the variance function. PROC GENMOD estimates k by maximum likelihood, or you can optionally set it to a constant value. Refer to McCullagh and Nelder (1989, Chapter 11), Hilbe (1994), or Lawless (1987) for discussions of the negative binomial distribution.

The multinomial distribution is sometimes used to model a response that can take values from a number of categories. The binomial is a special case of the multinomial with two categories. See the section "Multinomial Models" on page 1671 and refer to McCullagh and Nelder (1989, Chapter 5) for a description of the multinomial distribution.

In addition, you can easily define your own link functions or distributions through DATA step programming statements used within the procedure.

An important aspect of generalized linear modeling is the selection of explanatory variables in the model. Changes in goodness-of-fit statistics are often used to evaluate the contribution of subsets of explanatory variables to a particular model. The deviance, defined to be twice the difference between the maximum attainable log likelihood and the log likelihood of the model under consideration, is often used as a measure of goodness of fit. The maximum attainable log likelihood is achieved with a model that has a parameter for every observation. See the section "Goodness of Fit" on page 1656 for formulas for the deviance.

One strategy for variable selection is to fit a sequence of models, beginning with a simple model with only an intercept term, and then include one additional explanatory variable in each successive model. You can measure the importance of the additional explanatory variable by the difference in deviances or fitted log likelihoods between successive models. Asymptotic tests computed by the GENMOD procedure enable you to assess the statistical significance of the additional term.

The GENMOD procedure enables you to fit a sequence of models, up through a maximum number of terms specified in a MODEL statement. A table summarizes twice the difference in log likelihoods between each successive pair of models. This is called a *Type 1* analysis in the GENMOD procedure, because it is analogous to Type I (sequential) sums of squares in the GLM procedure. As with the PROC GLM Type I sums of squares, the results from this process depend on the order in which the model terms are fit.

The GENMOD procedure also generates a *Type 3* analysis analogous to Type III sums of squares in the GLM procedure. A Type 3 analysis does not depend on the order in which the terms for the model are specified. A GENMOD procedure Type 3 analysis consists of specifying a model and computing likelihood ratio statistics for Type III contrasts for each term in the model. The contrasts are defined in the same way as they are in the GLM procedure. The GENMOD procedure optionally computes Wald statistics for Type III contrasts. This is computationally less expensive than likelihood ratio statistics, but it is thought to be less accurate because the specified significance level of hypothesis tests based on the Wald statistic may not be as close to the actual significance level as it is for likelihood ratio tests.

A Type 3 analysis generalizes the use of Type III estimable functions in linear models. Briefly, a Type III estimable function (contrast) for an effect is a linear function of the model parameters that involves the parameters of the effect and any interactions with that effect. A test of the hypothesis that the Type III contrast for a main effect is equal to 0 is intended to test the significance of the main effect in the presence of interactions. See Chapter 32, "The GLM Procedure," and Chapter 11, "The Four Types of Estimable Functions," for more information about Type III estimable functions. Also refer to Littell, Freund, and Spector (1991).

Additional features of the GENMOD procedure are

- likelihood ratio statistics for user-defined contrasts, that is, linear functions of the parameters, and p-values based on their asymptotic chi-square distributions
- estimated values, standard errors, and confidence limits for user-defined contrasts and least-squares means

- ability to create a SAS data set corresponding to most tables displayed by the procedure (see Table 31.3 on page 1694)
- confidence intervals for model parameters based on either the profile likelihood function or asymptotic normality
- syntax similar to that of PROC GLM for the specification of the response and model effects, including interaction terms and automatic coding of classification variables
- ability to fit GEE models for clustered response data

Getting Started

Poisson Regression

You can use the GENMOD procedure to fit a variety of statistical models. A typical use of PROC GENMOD is to perform Poisson regression.

You can use the Poisson distribution to model the distribution of cell counts in a multiway contingency table. Aitkin, Anderson, Francis, and Hinde (1989) have used this method to model insurance claims data. Suppose the following hypothetical insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels).

```
data insure;
   input n c car$ age;
   ln = log(n);
   datalines;
   500   42  small  1
   1200  37  medium 1
   100    1  large  1
   400  101  small  2
   500   73  medium 2
   300   14  large  2
   ;
run;
```

In the preceding data set, the variable n represents the number of insurance policyholders and the variable c represents the number of insurance claims. The variable car is the type of car involved (classified into three groups) and the variable age is the age group of a policyholder (classified into two groups).

You can use PROC GENMOD to perform a Poisson regression analysis of these data with a log link function. This type of model is sometimes called a *log-linear model*.

Assume that the number of claims c has a Poisson probability distribution and that its mean, μ_i, is related to the factors car and age for observation i by

$$
\begin{aligned}
\log(\mu_i) &= \log(\mathrm{n}_i) + \mathbf{x}_i{}'\boldsymbol{\beta} \\
&= \log(\mathrm{n}_i) + \beta_0 + \\
&\quad \mathrm{car}_i(1)\beta_1 + \mathrm{car}_i(2)\beta_2 + \mathrm{car}_i(3)\beta_3 + \\
&\quad \mathrm{age}_i(1)\beta_4 + \mathrm{age}_i(2)\beta_5
\end{aligned}
$$

The indicator variables $\mathrm{car}_i(j)$ and $\mathrm{age}_i(j)$ are associated with the jth level of the variables car and age for observation i

$$
\mathrm{car}_i(j) = \begin{cases} 1 & \text{if car} = j \\ 0 & \text{if car} \neq j \end{cases}
$$

The βs are unknown parameters to be estimated by the procedure. The logarithm of the variable n is used as an *offset*, that is, a regression variable with a constant coefficient of 1 for each observation. A log linear relationship between the mean and the factors car and age is specified by the log link function. The log link function ensures that the mean number of insurance claims for each car and age group predicted from the fitted model is positive.

The following statements invoke the GENMOD procedure to perform this analysis:

```
proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln;
run;
```

The variables car and age are specified as CLASS variables so that PROC GENMOD automatically generates the indicator variables associated with car and age.

The MODEL statement specifies c as the response variable and car and age as explanatory variables. An intercept term is included by default. Thus, the model matrix $\mathbf{X}$ (the matrix which has as its ith row the transpose of the covariate vector for the ith observation) consists of a column of 1s representing the intercept term and columns of 0s and 1s derived from indicator variables representing the levels of the car and age variables.

That is, the model matrix is

$$
\mathbf{X} = \left[\begin{array}{c|ccc|cc}
1 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 & 1 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1
\end{array}\right]
$$

where the first column corresponds to the intercept, the next three columns correspond to the variable car, and the last two columns correspond to the variable age.

The response distribution is specified as Poisson, and the link function is chosen to be log. That is, the Poisson mean parameter μ is related to the linear predictor by

$$\log(\mu) = \mathbf{x}_i{}'\boldsymbol{\beta}$$

The logarithm of n is specified as an offset variable, as is common in this type of analysis. In this case, the offset variable serves to normalize the fitted cell means to a per policyholder basis, since the total number of claims, not individual policyholder claims, are observed.

PROC GENMOD produces the following default output from the preceding statements.

```
                              The GENMOD Procedure

                               Model Information

                        Data Set                    WORK.INSURE
                        Distribution                    Poisson
                        Link Function                       Log
                        Dependent Variable                    c
                        Offset Variable                      ln
```

Figure 31.1. Model Information

The “Model Information” table displayed in Figure 31.1 provides information about the specified model and the input data set.

```
                            Class Level Information

                     Class      Levels    Values

                     car             3    large medium small
                     age             2    1 2
```

Figure 31.2. Class Level Information

Figure 31.2 displays the “Class Level Information” table, which identifies the levels of the classification variables that are used in the model. Note that car is a character variable, and the values are sorted in alphabetical order. This is the default sort order, but you can select different sort orders with the ORDER= option in the PROC GENMOD statement (see the ORDER= option on page 1625 for details).

```
                      Criteria For Assessing Goodness Of Fit

           Criterion                 DF           Value        Value/DF

           Deviance                   2          2.8207          1.4103
           Scaled Deviance            2          2.8207          1.4103
           Pearson Chi-Square         2          2.8416          1.4208
           Scaled Pearson X2          2          2.8416          1.4208
           Log Likelihood                      837.4533
```

Figure 31.3. Goodness Of Fit

The “Criteria For Assessing Goodness Of Fit” table displayed in Figure 31.3 contains statistics that summarize the fit of the specified model. These statistics are helpful in judging the adequacy of a model and in comparing it with other models under consideration. If you compare the deviance of 2.8207 with its asymptotic chi-square with 2 degrees of freedom distribution, you find that the p-value is 0.24. This indicates that the specified model fits the data reasonably well.

```
                         Analysis Of Parameter Estimates

                                       Standard   Wald 95% Confidence      Chi-
 Parameter           DF    Estimate       Error          Limits          Square

 Intercept            1     -1.3168      0.0903    -1.4937    -1.1398    212.73
 car         large    1     -1.7643      0.2724    -2.2981    -1.2304     41.96
 car         medium   1     -0.6928      0.1282    -0.9441    -0.4414     29.18
 car         small    0      0.0000      0.0000     0.0000     0.0000       .
 age         1        1     -1.3199      0.1359    -1.5863    -1.0536     94.34
 age         2        0      0.0000      0.0000     0.0000     0.0000       .
 Scale                0      1.0000      0.0000     1.0000     1.0000

                         Analysis Of Parameter Estimates

                         Parameter            Pr > ChiSq

                         Intercept                <.0001
                         car         large        <.0001
                         car         medium       <.0001
                         car         small          .
                         age         1            <.0001
                         age         2              .
                         Scale

NOTE: The scale parameter was held fixed.
```

Figure 31.4. Analysis Of Parameter Estimates

Figure 31.4 displays the “Analysis Of Parameter Estimates” table, which summarizes the results of the iterative parameter estimation process. For each parameter in the model, PROC GENMOD displays columns with the parameter name, the degrees of freedom associated with the parameter, the estimated parameter value, the standard error of the parameter estimate, the confidence intervals, and the Wald chi-square statistic and associated p-value for testing the significance of the parameter to the model. If a column of the model matrix corresponding to a parameter is found to be linearly dependent, or *aliased*, with columns corresponding to parameters preceding it in the model, PROC GENMOD assigns it zero degrees of freedom and displays a value of zero for both the parameter estimate and its standard error.

This table includes a row for a scale parameter, even though there is no free scale parameter in the Poisson distribution. See the "Response Probability Distributions" section on page 1650 for the form of the Poisson probability distribution. PROC GENMOD allows the specification of a scale parameter to fit overdispersed Poisson and binomial distributions. In such cases, the SCALE row indicates the value of the overdispersion scale parameter used in adjusting output statistics. See the section "Overdispersion" on page 1659 for more on overdispersion and the meaning of the SCALE parameter output by the GENMOD procedure. PROC GENMOD displays a note indicating that the scale parameter is fixed, that is, not estimated by the iterative fitting process.

It is usually of interest to assess the importance of the main effects in the model. Type 1 and Type 3 analyses generate statistical tests for the significance of these effects. You can request these analyses with the TYPE1 and TYPE3 options in the MODEL statement.

```
proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       type1
                       type3;
run;
```

The results of these analyses are summarized in the tables that follow.

```
                              The GENMOD Procedure

                     LR Statistics For Type 1 Analysis

                                                  Chi-
          Source          Deviance        DF    Square    Pr > ChiSq

          Intercept       175.1536
          car             107.4620         2     67.69        <.0001
          age               2.8207         1    104.64        <.0001
```

Figure 31.5. Type 1 Analysis

In the table for Type 1 analysis displayed in Figure 31.5, each entry in the deviance column represents the deviance for the model containing the effect for that row and all effects preceding it in the table. For example, the deviance corresponding to car in the table is the deviance of the model containing an intercept and car. As more terms are included in the model, the deviance decreases.

Entries in the chi-square column are likelihood ratio statistics for testing the significance of the effect added to the model containing all the preceding effects. The chi-square value of 67.69 for car represents twice the difference in log likelihoods between fitting a model with only an intercept term and a model with an intercept and car. Since the scale parameter is set to 1 in this analysis, this is equal to the difference in deviances. Since two additional parameters are involved, this statistic can be compared with a chi-square distribution with two degrees of freedom. The resulting p-value (labeled Pr>Chi) of less than 0.0001 indicates that this variable is highly significant. Similarly, the chi-square value of 104.64 for age represents the difference in log likelihoods between the model with the intercept and car and the model with the intercept, car, and age. This effect is also highly significant, as indicated by the small p-value.

```
                     LR Statistics For Type 3 Analysis

                                          Chi-
                Source           DF      Square    Pr > ChiSq

                car               2       72.82        <.0001
                age               1      104.64        <.0001
```

Figure 31.6. Type 3 Analysis

The Type 3 analysis results in the same conclusions as the Type 1 analysis. The Type 3 chi-square value for the car variable, for example, is twice the difference between the log likelihood for the model with the variables Intercept, car, and age included and the log likelihood for the model with the car variable excluded. The hypothesis tested in this case is the significance of the variable car given that the variable age is in the model. In other words, it tests the additional contribution of car in the model.

The values of the Type 3 likelihood ratio statistics for the car and age variables indicate that both of these factors are highly significant in determining the claims performance of the insurance policyholders.

Generalized Estimating Equations

This section illustrates the use of the REPEATED statement to fit a GEE model, using repeated measures data from the “Six Cities” study of the health effects of air pollution (Ware et al. 1984). The data analyzed are the 16 selected cases in Lipsitz, Fitzmaurice, et al. (1994). The binary response is the wheezing status of 16 children at ages 9, 10, 11, and 12 years. The mean response is modeled as a logistic regression model using the explanatory variables city of residence, age, and maternal smoking status at the particular age. The binary responses for individual children are assumed to be equally correlated, implying an exchangeable correlation structure.

The data set and SAS statements that fit the model by the GEE method are as follows:

```
      data six;
         input case city$ @@;
         do i=1 to 4;
            input age smoke wheeze @@;
            output;
         end;
         datalines;
          1 portage   9 0 1  10 0 1  11 0 1  12 0 0
          2 kingston  9 1 1  10 2 1  11 2 0  12 2 0
          3 kingston  9 0 1  10 0 0  11 1 0  12 1 0
          4 portage   9 0 0  10 0 1  11 0 1  12 1 0
          5 kingston  9 0 0  10 1 0  11 1 0  12 1 0
          6 portage   9 0 0  10 1 0  11 1 0  12 1 0
          7 kingston  9 1 0  10 1 0  11 0 0  12 0 0
          8 portage   9 1 0  10 1 0  11 1 0  12 2 0
          9 portage   9 2 1  10 2 0  11 1 0  12 1 0
         10 kingston  9 0 0  10 0 0  11 0 0  12 1 0
         11 kingston  9 1 1  10 0 0  11 0 1  12 0 1
         12 portage   9 1 0  10 0 0  11 0 0  12 0 0
         13 kingston  9 1 0  10 0 1  11 1 1  12 1 1
         14 portage   9 1 0  10 2 0  11 1 0  12 2 1
         15 kingston  9 1 0  10 1 0  11 1 0  12 2 1
         16 portage   9 1 1  10 1 1  11 2 0  12 1 0
         ;
      run;


   proc genmod data=six ;
      class case city ;
      model  wheeze = city age smoke  /  dist=bin;
      repeated  subject=case / type=exch covb corrw;
   run;
```

The CLASS statement and the MODEL statement specify the model for the mean of the wheeze variable response as a logistic regression with city, age, and smoke as independent variables, just as for an ordinary logistic regression.

The REPEATED statement invokes the GEE method, specifies the correlation structure, and controls the displayed output from the GEE model. The option SUBJECT=CASE specifies that individual subjects are identified in the input data set by the variable case. The SUBJECT= variable case must be listed in the CLASS statement. Measurements on individual subjects at ages 9, 10, 11, and 12 are in the proper order in the data set, so the WITHINSUBJECT= option is not required. The TYPE=EXCH option specifies an exchangeable working correlation structure, the COVB option specifies that the parameter estimate covariance matrix be displayed, and the CORRW option specifies that the final working correlation be displayed.

Initial parameter estimates for iterative fitting of the GEE model are computed as in an ordinary generalized linear model, as described previously. Results of the initial model fit displayed as part of the generated output are not shown here. Statistics for the initial model fit such as parameter estimates, standard errors, deviances, and

Pearson chi-squares do not apply to the GEE model, and are only valid for the initial model fit. The following tables display information that applies to the GEE model fit.

Figure 31.7 displays general information about the GEE model fit.

```
                              The GENMOD Procedure

                             GEE Model Information

              Correlation Structure                    Exchangeable
              Subject Effect                        case (16 levels)
              Number of Clusters                                  16
              Correlation Matrix Dimension                         4
              Maximum Cluster Size                                 4
              Minimum Cluster Size                                 4
```

Figure 31.7. GEE Model Information

Figure 31.8 displays the parameter estimate covariance matrices specified by the COVB option. Both model-based and empirical covariances are produced.

```
                        Covariance Matrix (Model-Based)

                     Prm1           Prm2           Prm4           Prm5

        Prm1      5.74947       -0.22257       -0.53472        0.01655
        Prm2     -0.22257        0.45478      -0.002410        0.01876
        Prm4     -0.53472      -0.002410        0.05300       -0.01658
        Prm5      0.01655        0.01876       -0.01658        0.19104


                         Covariance Matrix (Empirical)

                     Prm1           Prm2           Prm4           Prm5

        Prm1      9.33994       -0.85104       -0.83253       -0.16534
        Prm2     -0.85104        0.47368        0.05736        0.04023
        Prm4     -0.83253        0.05736        0.07778      -0.002364
        Prm5     -0.16534        0.04023      -0.002364        0.13051
```

Figure 31.8. GEE Parameter Estimate Covariance Matrices

The exchangeable working correlation matrix specified by the CORRW option is displayed in Figure 31.9.

```
                           Working Correlation Matrix

                        Col1          Col2          Col3          Col4

            Row1      1.0000        0.1648        0.1648        0.1648
            Row2      0.1648        1.0000        0.1648        0.1648
            Row3      0.1648        0.1648        1.0000        0.1648
            Row4      0.1648        0.1648        0.1648        1.0000
```

Figure 31.9. GEE Working Correlation Matrix

The parameter estimates table, displayed in Figure 31.10, contains parameter estimates, standard errors, confidence intervals, Z scores, and p-values for the parameter estimates. Empirical standard error estimates are used in this table. A table using model-based standard errors can be created by using the REPEATED statement option MODELSE.

```
                       Analysis Of GEE Parameter Estimates
                       Empirical Standard Error Estimates

                                 Standard   95% Confidence
Parameter           Estimate    Error       Limits            Z Pr > |Z|

Intercept            -1.2751    3.0561  -7.2650    4.7148    -0.42    0.6765
city      kingston  -0.1223    0.6882  -1.4713    1.2266    -0.18    0.8589
city      portage     0.0000    0.0000   0.0000    0.0000      .        .
age                   0.2036    0.2789  -0.3431    0.7502     0.73    0.4655
smoke                 0.0935    0.3613  -0.6145    0.8016     0.26    0.7957
```

Figure 31.10. GEE Parameter Estimates Table

Syntax

You can specify the following statements in the GENMOD procedure. Items within the <> are optional.

PROC GENMOD < *options* > ;
  ASSESS | **ASSESSMENT** *keyword* < / *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  CONTRAST *'label' effect values* < . . . *effect values* > < */options* > ;
  DEVIANCE *variable = expression* ;
  ESTIMATE *'label' effect values* < . . . *effect values* > < */options* > ;
  FREQ | **FREQUENCY** *variable* ;
  FWDLINK *variable = expression* ;
  INVLINK *variable = expression* ;
  LSMEANS *effects* < / *options* > ;
  MODEL *response* = < *effects* >< */options* > ;
  OUTPUT < **OUT=***SAS-data-set* >
      < *keyword=name*. . .*keyword=name* > ;
  programming statements
  REPEATED **SUBJECT=** *subject-effect* < / *options* > ;
  WEIGHT | **SCWGT** *variable* ;
  VARIANCE *variable = expression* ;

The PROC GENMOD statement invokes the procedure. All statements other than the MODEL statement are optional. The CLASS statement, if present, must precede the MODEL statement, and the CONTRAST statement must come after the MODEL statement.

PROC GENMOD Statement

PROC GENMOD < *options* > **;**

The PROC GENMOD statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*

specifies the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DESCENDING | DESCEND | DESC

specifies that the levels of the response variable for the ordinal multinomial model and the binomial model with single variable response syntax be sorted in the reverse of the default order. For example, if RORDER=FORMATTED (the default), the DESCENDING option causes the levels to be sorted from highest to lowest instead of from lowest to highest. If RORDER=FREQ, the DESCENDING option causes the levels to be sorted from lowest frequency count to highest instead of from highest to lowest.

NAMELEN=n

specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

ORDER=*keyword*

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST or ESTIMATE statement. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table displays the valid *keywords* and describes how PROC GENMOD interprets them.

ORDER= *keyword*	**Levels Sorted by**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. For more information on sorting order, refer to the chapter titled “The SORT Procedure” in the *SAS Procedures Guide*.

RORDER=*keyword*

specifies the sorting order for the levels of the response variable. This ordering determines which intercept parameter in the model corresponds to each level in the data. If RORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the response variable. The change was implemented because the former default behavior for RORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or RORDER=INTERNAL to get the more natural ordering. The following table displays the valid *keywords* and describes how PROC GENMOD interprets them.

RORDER= *keyword*	**Levels Sorted by**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, RORDER=FORMATTED. For RORDER=FORMATTED and RORDER=INTERNAL, the sort order is machine dependent. The DESCENDING option in the PROC GENMOD statement causes the response variable to be sorted in the reverse of the order displayed in the previous table. For more information on sorting order, refer to the chapter on the SORT procedure in the *SAS Procedures Guide*.

The NOPRINT option, which suppresses displayed output in other SAS procedures, is not available in the PROC GENMOD statement. However, you can use the Output Delivery System (ODS) to suppress all displayed output, store all output on disk for further analysis, or create SAS data sets from selected output. You can suppress all displayed output with the statement ODS SELECT NONE;, and you can turn displayed output back on with the statement ODS SELECT ALL;. See Table 31.3 on page 1694 for the names of output tables available from PROC GENMOD. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

ASSESS Statement (Experimental)

ASSESS|ASSESSMENT *VAR=(effect)* | *LINK* < / *options* > **;**

The ASSESS statement computes and plots, using ODS graphics, model-checking statistics based on aggregates of residuals. See the "Assessment of Models Based on Aggregates of Residuals" section on page 1680 for details about the model assessment methods available in GENMOD.

The types of aggregates available are cumulative residuals, moving sums of residuals, and lowess smoothed residuals. If you do not specify which aggregate to use, the assessments are based on cumulative sums. PROC GENMOD uses experimental ODS graphics for graphical displays. For specific information about the experimental graphics available in GENMOD, see the "ODS Graphics" section on page 1695.

You must specify either LINK or VAR= in order to create an analysis.

LINK

request the assessment of the link function by performing the analysis with respect to the linear predictor.

VAR=(*effect***)**

specifies the functional form of a covariate be checked by performing the analysis with respect to the variable identified by the effect. The effect must be specified in the MODEL statement, and must contain only continuous variables (variables not listed in a CLASS statement).

You can specify the following options after the slash (/).

CRPANEL

requests a plot with four panels, each containing aggregates of the observed residuals and two simulated curves, be created.

LOWESS<**(***number***)**>

requests model assessment based on lowess smoothed residuals with optional *number* the fraction of data used. *number* must be between zero and one. If *number* is not specified, the default value one-third is used.

NPATHS|NPATH|PATHS|PATH=*number*

specifies the number of simulated paths to plot on the default aggregate residuals plot.

RESAMPLE|RESAMPLES<**=***number*>

specifies a *p*-value be computed based on 1,000 simulated paths, or *number* paths, if *number* is specified.

SEED=*number*

specifies a seed for the normal random number generator used in creating simulated realizations of aggregates of residuals for plots and estimating p-values. Specifying a seed allows you to produce identical graphs and p-values from run to run of the procedure. If a seed is not specified, or if *number* is negative or zero, a random number seed is derived from the time of day.

WINDOW<**(***number***)**>

requests assessment based on a moving sum window of width *number*. If *number* is not specified, a value of one-half of the range of the *x*-coordinate is used.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GENMOD to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

Since sorting the data changes the order in which PROC GENMOD reads the data, this can affect the sorting order for the levels of classification variables if you have specified ORDER=DATA in the PROC GENMOD statement. This, in turn, affects specifications in the CONTRAST statement.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GENMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

CLASS Statement

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *v-options* specified on any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *v-options* override the global *v-options*.

DESCENDING
DESC
reverses the sorting order of the classification variable.

MISSING
allows missing value ('.' for a numeric variable and blanks for a character variables) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST or ESTIMATE statement. If ORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table shows how PROC GENMOD interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter

on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=GLM. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the "CLASS Variable Parameterization" section on page 1661 for further details.

EFFECT	specifies effect coding
GLM	specifies less-than-full-rank, reference-cell coding; this option can only be used as a global option
ORDINAL THERMOMETER	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding
REFERENCE REF	specifies reference cell coding
ORTHEFFECT	orthogonalizes PARAM=EFFECT
ORTHORDINAL ORTHOTHERM	orthogonalizes PARAM=ORDINAL
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL
ORTHREF	orthogonalizes PARAM=REFERENCE

The EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

REF=*'level'* **|** *keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference
LAST	designates the last ordered level as reference

TRUNCATE<=*n*>

specifies the length n of CLASS variable values to use in determining CLASS variable levels. If you specify TRUNCATE without the length n, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases previous to Version 9. The default is to use the full formatted length of the CLASS variable. The TRUNCATE option is only available as a global option.

CONTRAST Statement

CONTRAST *'label' effect values* < *,. . . effect values* >< */options* > **;**

The CONTRAST statement provides a means for obtaining a test for a specified hypothesis concerning the model parameters. This is accomplished by specifying a matrix $\mathbf{L}$ for testing the hypothesis $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$. You must be familiar with the details of the model parameterization that PROC GENMOD uses. For more information, see the "Parameterization Used in PROC GENMOD" section on page 1661 and the "CLASS Variable Parameterization" section on page 1661. Computed statistics are based on the asymptotic chi-square distribution of the likelihood ratio statistic, or the generalized score statistic for GEE models, with degrees of freedom determined by the number of linearly independent rows in the $\mathbf{L}$ matrix. You can request Wald chi-square statistics with the Wald option in the CONTRAST statement.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement. Statistics for multiple CONTRAST statements are displayed in a single table.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.

effect identifies an effect that appears in the MODEL statement. The value INTERCEPT or intercept can be used as an effect when an intercept is included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ vector associated with the effect.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas.

If you use the default less-than-full-rank GLM CLASS variable parameterization, each row of the $\mathbf{L}$ matrix is checked for estimability. If PROC GENMOD finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. Refer to Searle (1971) for a discussion of estimable functions. If the elements of $\mathbf{L}$ are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A.

When you use any of the full-rank PARAM= CLASS variable options, all parameters are directly estimable, and rows of **L** are not checked for estimability.

If an effect is not specified in the CONTRAST statement, all of its coefficients in the **L** matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

PROC GENMOD handles missing level combinations of classification variables in the same manner as the GLM and MIXED procedures. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement.

If you specify the WALD option, the test of hypothesis is based on a Wald chi-square statistic. If you omit the WALD option, the test statistic computed depends on whether an ordinary generalized linear model or a GEE-type model is specified.

For an ordinary generalized linear model, the CONTRAST statement computes the likelihood ratio statistic. This is defined to be twice the difference between the log likelihood of the model unconstrained by the contrast and the log likelihood with the model fitted under the constraint that the linear function of the parameters defined by the contrast is equal to 0. A p-value is computed based on the asymptotic chi-square distribution of the chi-square statistic.

If you specify a GEE model with the REPEATED statement, the test is based on a score statistic. The GEE model is fit under the constraint that the linear function of the parameters defined by the contrast is equal to 0. The score chi-square statistic is computed based on the generalized score function. See the "Generalized Score Statistics" section on page 1680 for more information.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement, that is, the rank of **L**.

You can specify the following options after a slash (/).

E

requests that the **L** matrix be displayed.

SINGULAR = *number*

tunes the estimability checking. If **v** is a vector, define ABS($\mathbf{v}$) to be the absolute value of the element of **v** with the largest absolute value. Let $\mathbf{K}'$ be any row in the contrast matrix **L**. Define C to be equal to ABS$(\mathbf{K}')$ if ABS$(\mathbf{K}')$ is greater than 0; otherwise, C equals 1. If ABS$(\mathbf{K}' - \mathbf{K}'\mathbf{T})$ is greater than C$*$*number*, then **K** is declared nonestimable. **T** is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X})$, and $(\mathbf{X}'\mathbf{X})^{-}$ represents a generalized inverse of the matrix $\mathbf{X}'\mathbf{X}$. The value for $number$ must be between 0 and 1; the default value is 1E$-$4.

WALD

requests that a Wald chi-square statistic be computed for the contrast rather than the default likelihood ratio or score statistic. The Wald statistic for testing $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$ is defined by

$$S = (\mathbf{L}'\hat{\boldsymbol{\beta}})'(\mathbf{L}'\boldsymbol{\Sigma}\mathbf{L})^{-}(\mathbf{L}'\hat{\boldsymbol{\beta}})$$

where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate and $\boldsymbol{\Sigma}$ is its estimated covariance matrix. The asymptotic distribution of S is χ^2_r, where r is the rank of $\mathbf{L}$. Computed p-values are based on this distribution.

If you specify a GEE model with the REPEATED statement, $\boldsymbol{\Sigma}$ is the empirical covariance matrix estimate.

DEVIANCE Statement

DEVIANCE *variable = expression* **;**

You can specify a probability distribution other than those available in PROC GENMOD by using the DEVIANCE and VARIANCE statements. You do not need to specify the DEVIANCE or VARIANCE statements if you use the DIST= MODEL statement option to specify a probability distribution. The *variable* identifies the deviance contribution from a single observation to the procedure, and it must be a valid SAS variable name that does not appear in the input data set. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence of the deviance on the mean and the response. You use the automatic variables _MEAN_ and _RESP_ to represent the mean and response in the *expression*.

Alternatively, the deviance function can be defined using programming statements (see the section "Programming Statements" on page 1645) and assigned to a variable, which is then listed as the *expression*. This form is convenient for using complex statements such as if-then-else clauses.

The DEVIANCE statement is ignored unless the VARIANCE statement is also specified.

ESTIMATE Statement

ESTIMATE *'label' effect values . . . < /options >* **;**

The ESTIMATE statement is similar to a CONTRAST statement, except only one-row $\mathbf{L}'$ matrices are permitted.

If you use the default less-than-full-rank GLM CLASS variable parameterization, each row is checked for estimability. If PROC GENMOD finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. Refer to Searle (1971) for a discussion of estimable functions.

The actual estimate, $\mathbf{L}'\boldsymbol{\beta}$, its approximate standard error, and its confidence limits are displayed. A Wald chi-square test that $\mathbf{L}'\boldsymbol{\beta}$ = 0 is also displayed.

The approximate standard error of the estimate is computed as the square root of $\mathbf{L}'\hat{\mathbf{\Sigma}}\mathbf{L}$, where $\hat{\Sigma}$ is the estimated covariance matrix of the parameter estimates. If you specify a GEE model in the REPEATED statement, $\hat{\mathbf{\Sigma}}$ is the empirical covariance matrix estimate.

If you specify the EXP option, then $\exp(\mathbf{L}'\boldsymbol{\beta})$, its standard error, and its confidence limits are also displayed.

The construction of the **L** vector for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.

You can specify the following options in the ESTIMATE statement after a slash (/).

ALPHA=*number*
requests that a confidence interval be constructed with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default value is 0.05.

E
requests that the **L** matrix coefficients be displayed.

EXP
requests that $\exp(\mathbf{L}'\boldsymbol{\beta})$, its standard error, and its confidence limits be computed.

FREQ Statement

FREQ | **FREQUENCY** *variable* **;**

The *variable* in the FREQ statement identifies a variable in the input data set containing the frequency of occurrence of each observation. PROC GENMOD treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If it is less than 1 or if it is missing, the observation is not used.

FWDLINK Statement

FWDLINK *variable = expression* **;**

You can define a link function other than a built-in link function by using the FWDLINK statement. If you use the MODEL statement option LINK= to specify a link function, you do not need to use the FWDLINK statement. The *variable* identifies the link function to the procedure. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence on the mean.

Alternatively, the link function can be defined by using programming statements (see the "Programming Statements" section on page 1645) and assigned to a variable, which is then listed as the *expression*. The second form is convenient for using complex statements such as if-then-else clauses. The GENMOD procedure automatically computes derivatives of the link function required for iterative fitting. You must specify the inverse of the link function in the INVLINK statement when you specify the FWDLINK statement to define the link function. You use the automatic variable _MEAN_ to represent the mean in the preceding *expression*.

INVLINK Statement

INVLINK *variable = expression* **;**

If you define a link function in the FWDLINK statement, then you must define the inverse link function using the INVLINK statement. If you use the MODEL statement option LINK= to specify a link function, you do not need to use the INVLINK statement. The *variable* identifies the inverse link function to the procedure. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence on the linear predictor.

Alternatively, the inverse link function can be defined using programming statements (see the section "Programming Statements" on page 1645) and assigned to a variable, which is then listed as the *expression*. The second form is convenient for using complex statements such as if-then-else clauses. The automatic variable _XBETA_ represents the linear predictor in the preceding *expression*.

LSMEANS Statement

LSMEANS *effects* < / *options* > **;**

The LSMEANS statement computes least-squares means (LS-means) corresponding to the specified effects for the linear predictor part of the model. The $\mathbf{L}$ matrix constructed to compute them is precisely the same as the one formed in PROC GLM.

The LSMEANS statement is not available for multinomial distribution models for ordinal response data.

Each LS-mean is computed as $\mathbf{L}'\widehat{\boldsymbol{\beta}}$, where $\mathbf{L}$ is the coefficient matrix associated with the least-squares mean and $\widehat{\boldsymbol{\beta}}$ is the estimate of the parameter vector. The approximate standard errors for the LS-mean is computed as the square root of $\mathbf{L}'\hat{\boldsymbol{\Sigma}}\mathbf{L}$, where $\hat{\boldsymbol{\Sigma}}$ is the estimated covariance matrix of the parameter estimates. If you specify a GEE model in the REPEATED statement, $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix estimate.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement.

As in the ESTIMATE statement, the $\mathbf{L}$ matrix is tested for estimability, and if this test fails, PROC GENMOD displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC GENMOD constructs a Wald chi-square test to test the null hypothesis that the associated population quantity equals zero.

You can specify the following options in the LSMEANS statement after a slash (/).

ALPHA=*number*

requests that a confidence interval be constructed for each of the LS-means with confidence level $(1 - number) \times 100\%$. The value of *number* must be between 0 and 1; the default value is 0.05, corresponding to a 95% confidence interval.

CL

requests that confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR

displays the estimated correlation matrix of the LS-means as part of the “Least Squares Means” table.

COV

displays the estimated covariance matrix of the LS-means as part of the “Least Squares Means” table.

DIFF

requests that differences of the LS-means be displayed. All possible differences of LS-means, standard errors, and a Wald chi-square test are computed. Confidence limits are computed if the CL option is also specified.

E

requests that the **L** matrix coefficients for all LSMEANS effects be displayed.

MODEL Statement

MODEL *response* = < *effects* >< */options* > **;**

MODEL *events/trials* = < *effects* >< */options* > **;**

The MODEL statement specifies the response, or dependent variable, and the effects, or explanatory variables. If you omit the explanatory variables, the procedure fits an intercept-only model. An intercept term is included in the model by default. The intercept can be removed with the NOINT option.

You can specify the response in the form of a single variable or in the form of a ratio of two variables denoted *events/trials*. The first form is applicable to all responses. The second form is applicable only to summarized binomial response data. When each observation in the input data set contains the number of events (for example, successes) and the number of trials from a set of binomial trials, use the *events/trials* syntax.

In the *events/trials* model syntax, you specify two variables that contain the event and trial counts. These two variables are separated by a slash (/). The values of both *events* and (*trials*−*events*) must be nonnegative, and the value of the *trials* variable must be greater than 0 for an observation to be valid. The variable *events* or *trials* may take noninteger values.

When each observation in the input data set contains a single trial from a binomial or multinomial experiment, use the first form of the preceding MODEL statements. The response variable can be numeric or character. The ordering of response levels is critical in these models. You can use the RORDER= option in the PROC GENMOD statement to specify the response level ordering.

Responses for the Poisson distribution must be positive, but they can be noninteger values.

The effects in the MODEL statement consist of an explanatory variable or combination of variables. Explanatory variables can be continuous or classification variables. Classification variables can be character or numeric. Explanatory variables representing nominal, or classification, data must be declared in a CLASS statement. Interactions between variables can also be included as effects. Columns of the design matrix are automatically generated for classification variables and interactions. The syntax for specification of effects is the same as for the GLM procedure. See the "Specification of Effects" section on page 1659 for more information. Also refer to Chapter 32, "The GLM Procedure."

You can specify the following options in the MODEL statement after a slash (/).

AGGREGATE= *(variable-list)*
AGGREGATE= *variable*

specifies the subpopulations on which the Pearson chi-square and the deviance are calculated. This option applies only to the multinomial distribution or the binomial distribution with binary (single trial syntax) response. It is ignored if specified for other cases. Observations with common values in the given list of variables are regarded as coming from the same subpopulation. This affects the computation of the deviance and Pearson chi-square statistics. Variables in the list can be any variables in the input data set.

ALPHA | ALPH | A=*number*

sets the confidence coefficientfor parameter confidence intervals to $1-$*number*. The value of *number* must be between 0 and 1. The default value of *number* is 0.05.

CICONV=*number*

sets the convergence criterion for profile likelihood confidence intervals. See the section "Confidence Intervals for Parameters" on page 1666 for the definition of convergence. The value of *number* must be between 0 and 1. By default, CICONV=1E$-$4.

CL

requests that confidence limits for predicted values be displayed. See the OBSTATS option.

CODING=EFFECT | FULLRANK

specifies effect coding be used for all class variables in the model. This is the same as specifying PARAM=EFFECT as a CLASS statement option.

CONVERGE=*number*

sets the convergence criterion. The value of *number* must be between 0 and 1. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E$-$4. This convergence criterion is used in parameter estimation for a single model fit, Type 1 statistics, and likelihood ratio statistics for Type 3 analyses and CONTRAST statements.

CONVH=*number*

sets the relative Hessian convergence criterion. The value of *number* must be between 0 and 1. After convergence is determined with the change in parameter criterion specified with the CONVERGE= option, the quantity $tc = \frac{\mathbf{g}'\mathbf{H}^{-1}\mathbf{g}}{|f|}$ is computed and compared to *number*, where **g** is the gradient vector, **H** is the Hessian matrix for the model parameters, and f is the log-likelihood function. If tc is greater than *number*, a warning that the relative Hessian convergence criterion has been exceeded is printed. This criterion detects the occasional case where the change in parameter convergence criterion is satisfied, but a maximum in the log-likelihood function has not been attained. By default, CONVH=1E−4.

CORRB

requests that the parameter estimate correlation matrix be displayed.

COVB

requests that the parameter estimate covariance matrix be displayed.

DIST | D | ERROR | ERR = *keyword*

specifies the built-in probability distribution to use in the model. If you specify the DIST= option and you omit a user-defined link function, a default link function is chosen as displayed in the following table. If you specify no distribution and no link function, then the GENMOD procedure defaults to the normal distribution with the identity link function.

DIST=	**Distribution**	**Default Link Function**
BINOMIAL | BIN | B	binomial	logit
GAMMA | GAM | G	gamma	inverse (power(−1))
IGAUSSIAN | IG	inverse Gaussian	inverse squared (power(−2))
MULTINOMIAL | MULT	multinomial	cumulative logit
NEGBIN | NB	negative binomial	log
NORMAL | NOR | N	normal	identity
POISSON | POI | P	Poisson	log

EXPECTED

requests that the expected Fisher information matrix be used to compute parameter estimate covariances and the associated statistics. The default action is to use the observed Fisher information matrix. See the SCORING= option.

ID=*variable*

causes the values of *variable* in the input data set to be displayed in the OBSTATS table. If an explicit format for *variable* has been defined, the formatted values are displayed. If the OBSTATS option is not specified, this option has no effect.

INITIAL=*numbers*

sets initial values for parameter estimates in the model. The default initial parameter values are weighted least squares estimates based on using the response data as the initial mean estimate. This option can be useful in case of convergence difficulty. The intercept parameter is initialized with the INTERCEPT= option and is not included here. The values are assigned to the variables in the MODEL statement in the same order in which they appear in the MODEL statement. The order of levels for CLASS variables is determined by the ORDER= option. Note that some levels of class variables can be aliased; that is, they correspond to linearly dependent parameters that are not estimated by the procedure. Initial values must be assigned to all levels of class variables, regardless of whether they are aliased or not. The procedure ignores initial values corresponding to parameters not being estimated. If you specify a BY statement, all class variables must take on the same number of levels in each BY group. Otherwise, class variables in some of the BY groups are assigned incorrect initial values. Types of INITIAL= specifications are illustrated in the following table.

Type of List	**Specification**
list separated by blanks	INITIAL = 3 4 5
list separated by commas	INITIAL = 3, 4, 5
x to y	INITIAL = 3 to 5
x to y by z	INITIAL = 3 to 5 by 1
combination of list types	INITIAL = 1, 3 to 5, 9

INTERCEPT=*number*

initializes the intercept term to *number* for parameter estimation. If you specify both the INTERCEPT= and the NOINT options, the intercept term is not estimated, but an intercept term of *number* is included in the model.

ITPRINT

displays the iteration history for all iterative processes: parameter estimation, fitting constrained models for contrasts and Type 3 analyses, and profile likelihood confidence intervals. The last evaluation of the gradient and the negative of the Hessian (second derivative) matrix are also displayed for parameter estimation. This option may result in a large amount of displayed output, especially if some of the optional iterative processes are selected.

LINK = *keyword*

specifies the link function to use in the model. The keywords and their associated built-in link functions are as follows.

LINK=	**Link Function**
CUMCLL | CCLL	cumulative complementary log-log
CUMLOGIT | CLOGIT	cumulative logit
CUMPROBIT | CPROBIT	cumulative probit
CLOGLOG | CLL	complementary log-log
IDENTITY | ID	identity
LOG	log
LOGIT	logit
PROBIT	probit
POWER(*number*) | POW(*number*)	power with λ= *number*

If no LINK= option is supplied and there is a user-defined link function, the user-defined link function is used. If you specify neither the LINK= option nor a user-defined link function, then the default canonical link function is used if you specify the DIST= option. Otherwise, if you omit the DIST= option, the identity link function is used.

The cumulative link functions are appropriate only for the multinomial distribution.

LRCI

requests that two-sided confidence intervals for all model parameters be computed based on the profile likelihood function. This is sometimes called the partially maximized likelihood function. See the "Confidence Intervals for Parameters" section on page 1666 for more information on the profile likelihood function. This computation is iterative and can consume a relatively large amount of CPU time. The confidence coefficient can be selected with the ALPHA=*number* option. The resulting confidence coefficient is $1-$*number*. The default confidence coefficient is 0.95.

MAXITER=*number*
MAXIT=*number*

sets the maximum allowable number of iterations for all iterative computation processes in PROC GENMOD. By default, MAXITER=50.

NOINT

requests that no intercept term be included in the model. An intercept is included unless this option is specified.

NOSCALE

holds the scale parameter fixed. Otherwise, for the normal, inverse Gaussian, and gamma distributions, the scale parameter is estimated by maximum likelihood. If you omit the SCALE= option, the scale parameter is fixed at the value 1.

OFFSET=*variable*

specifies a variable in the input data set to be used as an offset variable. This variable cannot be a CLASS variable, and it cannot be the response variable or one of the explanatory variables.

OBSTATS

specifies that an additional table of statistics be displayed. For each observation, the following items are displayed:

- the value of the response variable (variables if the data are binomial), frequency, and weight variables
- the values of the regression variables
- predicted mean, $\hat{\mu} = g^{-1}(\eta)$, where $\eta = \mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$ is the linear predictor and g is the link function. If there is an offset, it is included in $\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$.
- estimate of the linear predictor $\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$. If there is an offset, it is included in $\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$.
- standard error of the linear predictor $\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$
- the value of the Hessian weight at the final iteration

- lower confidence limit of the predicted value of the mean. The confidence coefficient is specified with the ALPHA= option. See the section "Confidence Intervals on Predicted Values" on page 1669 for the computational method.
- upper confidence limit of the predicted value of the mean
- raw residual, defined as $Y - \mu$
- Pearson, or chi residual, defined as the square root of the contribution for the observation to the Pearson chi-square, that is

  $$\frac{Y - \mu}{\sqrt{V(\mu)/w}}$$

  where Y is the response, μ is the predicted mean, w is the value of the prior weight variable specified in a WEIGHT statement, and V(μ) is the variance function evaluated at μ.
- the standardized Pearson residual
- deviance residual, defined as the square root of the deviance contribution for the observation, with sign equal to the sign of the raw residual
- the standardized deviance residual
- the likelihood residual

The RESIDUALS, PREDICTED, XVARS, and CL options cause only subgroups of the observation statistics to be displayed. You can specify more than one of these options to include different subgroups of statistics.

The ID=*variable* option causes the values of *variable* in the input data set to be displayed in the table. If an explicit format for *variable* has been defined, the formatted values are displayed.

If a REPEATED statement is present, a table is displayed for the GEE model specified in the REPEATED statement. Only the regression variables, response values, predicted values, confidence limits for the predicted values, linear predictor, raw residuals, and Pearson residuals for each observation in the input data set are available.

PREDICTED
PRED
P

requests that predicted values, the linear predictor, its standard error, and the Hessian weight be displayed. See the OBSTATS option.

RESIDUALS
R

requests that residuals and standardized residuals be displayed. See the OBSTATS option.

SCALE=*number*
SCALE=PEARSON
SCALE=P
PSCALE
SCALE=DEVIANCE
SCALE=D
DSCALE

sets the value used for the scale parameter where the NOSCALE option is used. For the binomial and Poisson distributions, which have no free scale parameter, this can be used to specify an *overdispersed* model. In this case, the parameter covariance matrix and the likelihood function are adjusted by the scale parameter. See the "Dispersion Parameter" section (page 1658) and the "Overdispersion" section (page 1659) for more information. If the NOSCALE option is not specified, then *number* is used as an initial estimate of the scale parameter.

Specifying SCALE=PEARSON or SCALE=P is the same as specifying the PSCALE option. This fixes the scale parameter at the value 1 in the estimation procedure. After the parameter estimates are determined, the exponential family dispersion parameter is assumed to be given by Pearson's chi-square statistic divided by the degrees of freedom, and all statistics such as standard errors and likelihood ratio statistics are adjusted appropriately.

Specifying SCALE=DEVIANCE or SCALE=D is the same as specifying the DSCALE option. This fixes the scale parameter at a value of 1 in the estimation procedure.

After the parameter estimates are determined, the exponential family dispersion parameter is assumed to be given by the deviance divided by the degrees of freedom. All statistics such as standard errors and likelihood ratio statistics are adjusted appropriately.

SCORING=*number*

requests that on iterations up to *number*, the Hessian matrix is computed using the Fisher's scoring method. For further iterations, the full Hessian matrix is computed. The default value is 1. A value of 0 causes all iterations to use the full Hessian matrix, and a value greater than or equal to the value of the MAXITER option causes all iterations to use Fisher's scoring. The value of the SCORING= option must be 0 or a positive integer.

SINGULAR=*number*

sets the tolerance for testing singularity of the information matrix and the crossproducts matrix. Roughly, the test requires that a pivot be at least this number times the original diagonal value. By default, *number* is 10^7 times the machine epsilon. The default *number* is approximately 10^{-9} on most machines.

TYPE1

requests that a Type 1, or sequential, analysis be performed. This consists of sequentially fitting models, beginning with the null (intercept term only) model and continuing up to the model specified in the MODEL statement. The likelihood ratio statistic between each successive pair of models is computed and displayed in a table.

A Type 1 analysis is not available for GEE models, since there is no associated likelihood.

TYPE3

requests that statistics for Type 3 contrasts be computed for each effect specified in the MODEL statement. The default analysis is to compute likelihood ratio statistics for the contrasts or score statistics for GEEs. Wald statistics are computed if the WALD option is also specified.

WALD

requests Wald statistics for Type 3 contrasts. You must also specify the TYPE3 option in order to compute Type 3 Wald statistics.

WALDCI

requests that two-sided Wald confidence intervals for all model parameters be computed based on the asymptotic normality of the parameter estimators. This computation is not as time consuming as the LRCI method, since it does not involve an iterative procedure. However, it is not thought to be as accurate, especially for small sample sizes. The confidence coefficient can be selected with the ALPHA= option in the same way as for the LRCI option.

XVARS

requests that the regression variables be included in the OBSTATS table.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* >
< *keyword=name . . . keyword=name* > **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors (XBETA) and their standard error estimates, the weights for the Hessian matrix, predicted values of the mean, confidence limits for predicted values, and residuals.

You can also request these statistics with the OBSTATS, PREDICTED, RESIDUALS, CL, or XVARS options in the MODEL statement. You can then create a SAS data set containing them with ODS OUTPUT commands. You may prefer to specify the OUTPUT statement for requesting these statistics since

- the OUTPUT statement produces no tabular output
- the OUTPUT statement creates a SAS data set more efficiently than ODS. This can be an advantage for large data sets.
- you can specify the individual statistics to be included in the SAS data set

If you use the multinomial distribution with one of the cumulative link functions for ordinal data, the data set also contains variables named _ORDER_ and _LEVEL_ that indicate the levels of the ordinal response variable and the values of the variable in the input data set corresponding to the sorted levels. These variables indicate that the predicted value for a given observation is the probability that the response variable is as large as the value of the Value variable.

The estimated linear predictor, its standard error estimate, and the predicted values and their confidence intervals are computed for all observations in which the explanatory variables are all nonmissing, even if the response is missing. By adding observations with missing response values to the input data set, you can compute these statistics for new observations or for settings of the explanatory variables not present in the data without affecting the model fit.

The following list explains specifications in the OUTPUT statement.

OUT= *SAS-data-set*

specifies the output data set. If you omit the OUT=option, the output data set is created and given a default name using the DATAn convention.

keyword=name

specifies the statistics to be included in the output data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the name of the new variable or variables to contain the statistic. You can list only one variable after the equal sign. Although you can use the OUTPUT statement without any *keyword=name* specifications, the output data set then contains only the original variables and, possibly, the variables Level and Value (if you use the multinomial model with ordinal data). Note that the residuals are not available for the multinomial model with ordinal data. Formulas for the statistics are given in the section "Predicted Values of the Mean" on page 1669 and the "Residuals" section on page 1669. The keywords allowed and the statistics they represent are as follows:

HESSWGT diagonal element of the weight matrix used in computing the Hessian matrix

LOWER | L lower confidence limit for the predicted value of the mean, or the lower confidence limit for the probability that the response is less than or equal to the value of Level or Value. The confidence coefficient is determined by the ALPHA=$number$ option in the MODEL statement as $(1 - number) \times 100\%$. The default confidence coefficient is 95%.

PREDICTED | PRED | PROB | P predicted value of the mean or the predicted probability that the response variable is less than or equal to the value of Level or Value if the multinomial model for ordinal data is used (in other words, $\Pr(\mathrm{Y} \leq \mathrm{Value})$, where Y is the response variable)

RESCHI Pearson (Chi) residual for identifying observations that are poorly accounted for by the model

RESDEV deviance residual for identifying poorly fitted observations

RESLIK likelihood residual for identifying poorly fitted observations

STDXBETA standard error estimate of XBETA (see the XBETA keyword)

STDRESCHI standardized Pearson (Chi) residual for identifying observations that are poorly accounted for by the model

STDRESDEV standardized deviance residual for identifying poorly fitted observations

UPPER | U upper confidence limit for the predicted value of the mean, or the lower confidence limit for the probability that the response is less than or equal to the value of Level or Value. The confidence coefficient is determined by the ALPHA=$number$ option in the MODEL statement as $(1 - number) \times 100\%$. The default confidence coefficient is 95%.

XBETA estimate of the linear predictor $\mathbf{x}_i{}'\boldsymbol{\beta}$ for observation i, or $\alpha_j + \mathbf{x}_i{}'\boldsymbol{\beta}$, where j is the corresponding ordered value of the response variable for the multinomial model with ordinal data. If there is an offset, it is included in $\mathbf{x}_i{}'\boldsymbol{\beta}$.

Programming Statements

Although the most commonly used link and probability distributions are available as built-in functions, the GENMOD procedure enables you to define your own link functions and response probability distributions using the FWDLINK, INVLINK, VARIANCE, and DEVIANCE statements. The variables assigned in these statements can have values computed in programming statements. These programming statements can occur anywhere between the PROC GENMOD statement and the RUN statement. Variable names used in programming statements must be unique. Variables from the input data set may be referenced in programming statements. The mean, linear predictor, and response are represented by the automatic variables _MEAN_, _XBETA_, and _RESP_, which can be referenced in your programming statements. Programming statements are used to define the functional dependencies of the link function, the inverse link function, the variance function, and the deviance function on the mean, linear predictor, and response variable.

The following code illustrates the use of programming statements. Even though you usually request the Poisson distribution by specifying DIST=POISSON as a MODEL statement option, you can define the variance and deviance functions for the Poisson distribution by using the VARIANCE and DEVIANCE statements. For example, the following code performs the same analysis as the Poisson regression example in the "Getting Started" section on page 1616. The code must be in logical order for computation, just as in a DATA step.

```
proc genmod ;
   class car age;
   a = _MEAN_;
   y = _RESP_;
   d = 2 * ( y * log( y / a ) - ( y - a ) );
   variance var = a;
   deviance dev = d;
   model c = car age / link = log offset = ln;
run;
```

The variables var and dev are dummy variables used internally by the procedure to identify the variance and deviance functions. Any valid SAS variable names can be used.

Similarly, the log link function and its inverse could be defined with the FWDLINK and INVLINK statements.

```
fwdlink link = log(_MEAN_);
invlink ilink = exp(_XBETA_);
```

This code is for illustration, and it works well for most Poisson regression problems. If, however, in the iterative fitting process, the mean parameter becomes too close to 0, or a 0 response value occurs, an error condition occurs when the procedure attempts to evaluate the log function. You can circumvent this kind of problem by using if-then-else clauses or other conditional statements to check for possible error conditions and appropriately define the functions for these cases.

Data set variables can be referenced in user definitions of the link function and response distributions using programming statements and the FWDLINK, INVLINK, DEVIANCE, and VARIANCE statements.

See the DEVIANCE, VARIANCE, FWDLINK, and INVLINK statements for more information.

REPEATED Statement

REPEATED SUBJECT= *subject-effect* < / *options* > **;**

The REPEATED statement specifies the covariance structure of multivariate responses for GEE model fitting in the GENMOD procedure. In addition, the REPEATED statement controls the iterative fitting algorithm used in GEEs and specifies optional output. Other GENMOD procedure statements, such as the MODEL and CLASS statements, are used in the same way as they are for ordinary generalized linear models to specify the regression model for the mean of the responses.

SUBJECT=*subject-effect*

identifies subjects in the input data set. The *subject-effect* can be a single variable, an interaction effect, a nested effect, or a combination. Each distinct value, or level, of the effect identifies a different subject, or cluster. Responses from different subjects are assumed to be statistically independent, and responses within subjects are assumed to be correlated. A *subject-effect* must be specified, and variables used in defining the *subject-effect* must be listed in the CLASS statement. The input data set does not need to be sorted by subject. See the SORTED option.

The *options* control how the model is fit and what output is produced. You can specify the following options after a slash (/).

ALPHAINIT=*numbers*

specifies initial values for log odds ratio regression parameters if the LOGOR= option is specified for binary data. If this option is not specified, an initial value of 0.01 is used for all the parameters.

CONVERGE=*number*

specifies the convergence criterion for GEE parameter estimation. If the maximum absolute difference between regression parameter estimates is less than the value of *number* on two successive iterations, convergence is declared. If the absolute value of a regression parameter estimate is greater than 0.08, then the absolute difference normalized by the regression parameter value is used instead of the absolute difference. The default value of *number* is 0.0001.

CORRW

displays the estimated working correlation matrix.

CORRB

displays the estimated regression parameter correlation matrix. Both model-based and empirical correlations are displayed.

COVB

displays the estimated regression parameter covariance matrix. Both model-based and empirical covariances are displayed.

ECORRB

displays the estimated regression parameter empirical correlation matrix.

ECOVB

displays the estimated regression parameter empirical covariance matrix.

INTERCEPT=*number*

specifies either an initial or a fixed value of the intercept regression parameter in the GEE model. If you specify the NOINT option in the MODEL statement, then the intercept is fixed at the value of *number*.

INITIAL=*numbers*

specifies initial values of the regression parameters estimation, other than the intercept parameter, for GEE estimation. If this option is not specified, the estimated regression parameters assuming independence for all responses are used for the initial values.

LOGOR=*log odds ratio structure keyword*

specifies the regression structure of the log odds ratio used to model the association of the responses from subjects for binary data. The response syntax must be of the single variable type, the distribution must be binomial, and the data must be binary. The following table displays the log odds ratio structure keywords and the corresponding log odds ratio regression structures. See the "Alternating Logistic Regressions" section on page 1676 for definitions of the log odds ratio types and examples of specifying log odds ratio models. You should specify either the LOGOR= or the TYPE= option, but not both.

Table 31.1. Log Odds Ratio Regression Structures

Keyword	**Log Odds Ratio** **Regression Structure**
EXCH	exchangeable
FULLCLUST	fully parameterized clusters
LOGORVAR(*variable*)	indicator variable for specifying block effects
NESTK	*k*-nested
NEST1	1-nested
ZFULL	fully specified *z*-matrix specified in ZDATA= data set
ZREP	single cluster specification for replicated *z*-matrix specified in ZDATA= data set
ZREP(matrix)	single cluster specification for replicated *z*-matrix

MAXITER=*number*
MAXIT=*number*

specifies the maximum number of iterations allowed in the iterative GEE estimation process. The default number is 50.

MCORRB

displays the estimated regression parameter model-based correlation matrix.

MCOVB

displays the estimated regression parameter model-based covariance matrix.

MODELSE

displays an analysis of parameter estimates table using model-based standard errors. By default, an “Analysis of Parameter Estimates” table based on empirical standard errors is displayed.

RUPDATE=*number*

specifies the number of iterations between updates of the working correlation matrix. For example, RUPDATE=5 specifies that the working correlation is updated once for every five regression parameter updates. The default value of *number* is 1; that is, the working correlation is updated every time the regression parameters are updated.

SORTED

specifies that the input data are grouped by subject and sorted within subject. If this option is not specified, then the procedure internally sorts by *subject-effect* and *within subject-effect*, if a *within subject-effect* is specified.

SUBCLUSTER=*variable*
SUBCLUST=*variable*

specifies a variable defining subclusters for the 1-nested or *k*-nested log odds ratio association modeling structures.

TYPE | CORR=*correlation-structure keyword*

specifies the structure of the working correlation matrix used to model the correlation of the responses from subjects. The following table displays the correlation structure keywords and the corresponding correlation structures. The default working correlation type is the independent (CORR=IND). See the “Details” section on page 1650 for definitions of the correlation matrix types. You should specify LOGOR= or TYPE= but not both.

Table 31.2. Correlation Structure Types

Keyword	**Correlation Matrix Type**
AR | AR(1)	autoregressive(1)
EXCH | CS	exchangeable
IND	independent
MDEP(number)	m-dependent with m=number
UNSTR | UN	unstructured
USER | FIXED (matrix)	fixed, user-specified correlation matrix

For example, you can specify a fixed 4×4 correlation matrix with the option

```
TYPE=USER( 1.0  0.9  0.8  0.6
           0.9  1.0  0.9  0.8
           0.8  0.9  1.0  0.9
           0.6  0.8  0.9  1.0 )
```

V6CORR

specifies that the 'Version 6' method of computing the normalized Pearson chi-square be used for working correlation estimation and for model-based covariance matrix scale factor.

WITHINSUBJECT | WITHIN=*within subject-effect*

defines an effect specifying the order of measurements within subjects. Each distinct level of the *within subject-effect* defines a different response from the same subject. If the data are in proper order within each subject, you do not need to specify this option.

If some measurements do not appear in the data for some subjects, this option properly orders the existing measurements and treats the omitted measurements as missing values. If the WITHINSUBJECT= option is not used in this situation, measurements may be improperly ordered and missing values assumed for the last measurements in a cluster.

Variables used in defining the *within subject-effect* must be listed in the CLASS statement.

YPAIR=*variable-list*

specifies the variables in the ZDATA= data set corresponding to pairs of responses for log odds ratio association modeling.

ZDATA=*SAS-data-set*

specifies a SAS data set containing either the full *z*-matrix for log odds ratio association modeling or the *z*-matrix for a single complete cluster to be replicated for all clusters.

ZROW=*variable-list*

specifies the variables in the ZDATA= data set corresponding to rows of the *z*-matrix for log odds ratio association modeling.

VARIANCE Statement

VARIANCE *variable = expression* **;**

You can specify a probability distribution other than the built-in distributions by using the VARIANCE and DEVIANCE statements. The variable name *variable* identifies the variance function to the procedure. The *expression* is used to define the functional dependence on the mean, and it can be any arithmetic expression supported by the DATA step language. You use the automatic variable _MEAN_ to represent the mean in the expression.

Alternatively, you can define the variance function with programming statements, as detailed in the section "Programming Statements" on page 1645. This form is convenient for using complex statements such as if-then-else clauses. Derivatives of the variance function for use during optimization are computed automatically. The DEVIANCE statement must also appear when the VARIANCE statement is used to define the variance function.

WEIGHT Statement

WEIGHT | SCWGT *variable* **;**

The WEIGHT statement identifies a variable in the input data set to be used as the exponential family dispersion parameter weight for each observation. The exponential family dispersion parameter is divided by the WEIGHT variable value for each observation. This is true regardless of whether the parameter is estimated by the procedure or specified in the MODEL statement with the SCALE= option. It is also true for distributions such as the Poisson and binomial that are not usually defined to have a dispersion parameter. For these distributions, a WEIGHT variable weights the overdispersion parameter, which has the default value of 1.

The WEIGHT variable does not have to be an integer; if it is less than or equal to 0 or if it is missing, the corresponding observation is not used.

Details

Generalized Linear Models Theory

This is a brief introduction to the theory of generalized linear models. See the "References" section on page 1728 for sources of more detailed information.

Response Probability Distributions

In generalized linear models, the response is assumed to possess a probability distribution of the exponential form. That is, the probability density of the response Y for continuous response variables, or the probability function for discrete responses, can be expressed as

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}$$

for some functions a, b, and c that determine the specific distribution. For fixed ϕ, this is a one parameter exponential family of distributions. The functions a and c are such that $a(\phi) = \phi/w$ and $c = c(y, \phi/w)$, where w is a known weight for each observation. A variable representing w in the input data set may be specified in the WEIGHT statement. If no WEIGHT statement is specified, $w_i = 1$ for all observations.

Standard theory for this type of distribution gives expressions for the mean and variance of Y.

$$\begin{aligned} E(Y) &= b'(\theta) \\ \mathit{Var}(Y) &= \frac{b''(\theta)\phi}{w} \end{aligned}$$

where the primes denote derivatives with respect to θ. If μ represents the mean of Y, then the variance expressed as a function of the mean is

$$\mathit{Var}(Y) = \frac{V(\mu)\phi}{w}$$

where V is the *variance function*.

Probability distributions of the response Y in generalized linear models are usually parameterized in terms of the mean μ and dispersion parameter ϕ instead of the *natural parameter* θ. The probability distributions that are available in the GENMOD procedure are shown in the following list. The PROC GENMOD scale parameter and the variance of Y are also shown.

- Normal:

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] \quad \text{for } -\infty < y < \infty \\ \phi &= \sigma^2 \\ \mathit{scale} &= \sigma \\ Var(Y) &= \sigma^2 \end{aligned}$$

- Inverse Gaussian:

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi y^3}\sigma} \exp\left[-\frac{1}{2y} \left(\frac{y-\mu}{\mu\sigma} \right)^2 \right] \quad \text{for } 0 < y < \infty \\ \phi &= \sigma^2 \\ \textit{scale} &= \sigma \\ \textit{Var}(Y) &= \sigma^2\mu^3 \end{aligned}$$

- Gamma:

$$\begin{aligned} f(y) &= \frac{1}{\Gamma(\nu)y} \left(\frac{y\nu}{\mu} \right)^\nu \exp\left(-\frac{y\nu}{\mu} \right) \quad \text{for } 0 < y < \infty \\ \phi &= \nu^{-1} \\ \textit{scale} &= \nu \\ \textit{Var}(Y) &= \frac{\mu^2}{\nu} \end{aligned}$$

- Negative Binomial:

$$\begin{aligned} f(y) &= \frac{\Gamma(y+1/k)}{\Gamma(y+1)\Gamma(1/k)} \frac{(k\mu)^y}{(1+k\mu)^{y+1/k}} \quad \text{for } y = 0, 1, 2, \ldots \\ \textit{dispersion} &= k \\ \textit{Var}(Y) &= \mu + k\mu^2 \end{aligned}$$

- Poisson:

$$\begin{aligned} f(y) &= \frac{\mu^y e^{-\mu}}{y!} \quad \text{for } y = 0, 1, 2, \ldots \\ \phi &= 1 \\ \textit{Var}(Y) &= \mu \end{aligned}$$

- Binomial:

$$f(y) = \binom{n}{r} \mu^r (1-\mu)^{n-r} \quad \text{for } y = \frac{r}{n},\ r = 0, 1, 2, \ldots, n$$

$$\begin{aligned}\phi &= 1 \\ \mathit{Var}(Y) &= \frac{\mu(1-\mu)}{n}\end{aligned}$$

- Multinomial:

$$f(y_1, y_2, \cdots, y_k) = \frac{m!}{y_1! y_2! \cdots y_k!} p_1^{y_1} p_2^{y_2} \cdots p_k^{y_k}$$

The negative binomial distribution contains a parameter k, called the negative binomial dispersion parameter. This is not the same as the generalized linear model dispersion ϕ, but it is an additional distribution parameter that must be estimated or set to a fixed value.

For the binomial distribution, the response is the binomial proportion $Y = \mathit{events}/\mathit{trials}$. The variance function is $V(\mu) = \mu(1-\mu)$, and the binomial trials parameter n is regarded as a weight w.

If a weight variable is present, ϕ is replaced with ϕ/w, where w is the weight variable.

PROC GENMOD works with a scale parameter that is related to the exponential family dispersion parameter ϕ instead of with ϕ itself. The scale parameters are related to the dispersion parameter as shown previously with the probability distribution definitions. Thus, the scale parameter output in the “Analysis of Parameter Estimates” table is related to the exponential family dispersion parameter. If you specify a constant scale parameter with the SCALE= option in the MODEL statement, it is also related to the exponential family dispersion parameter in the same way.

Link Function

The mean μ_i of the response in the ith observation is related to a linear predictor through a monotonic differentiable link function g.

$$g(\mu_i) = \mathbf{x}_i{}'\boldsymbol{\beta}$$

Here, $\mathbf{x}_i$ is a fixed known vector of explanatory variables, and $\boldsymbol{\beta}$ is a vector of unknown parameters.

Log-Likelihood Functions

Log-likelihood functions for the distributions that are available in the procedure are parameterized in terms of the means μ_i and the dispersion parameter ϕ. The term y_i represents the response for the ith observation, and w_i represents the known dispersion weight. The log-likelihood functions are of the form

$$L(\mathbf{y}, \boldsymbol{\mu}, \phi) = \sum_i \log\left(f(y_i, \mu_i, \phi)\right)$$

where the sum is over the observations. The forms of the individual contributions

$$l_i = \log\left(f(y_i, \mu_i, \phi)\right)$$

are shown in the following list; the parameterizations are expressed in terms of the mean and dispersion parameters.

- Normal:

$$l_i = -\frac{1}{2}\left[\frac{w_i(y_i-\mu_i)^2}{\phi} + \log\left(\frac{\phi}{w_i}\right) + \log(2\pi)\right]$$

- Inverse Gaussian:

$$l_i = -\frac{1}{2}\left[\frac{w_i(y_i-\mu_i)^2}{y_i\mu^2\phi} + \log\left(\frac{\phi y_i^3}{w_i}\right) + \log(2\pi)\right]$$

- Gamma:

$$l_i = \frac{w_i}{\phi}\log\left(\frac{w_i y_i}{\phi\mu_i}\right) - \frac{w_i y_i}{\phi\mu_i} - \log(y_i) - \log\left(\Gamma\left(\frac{w_i}{\phi}\right)\right)$$

- Negative Binomial:

$$l_i = y\log\left(\frac{k\mu}{w_i}\right) - (y + w_i/k)\log\left(1 + \frac{k\mu}{w_i}\right) + \log\left(\frac{\Gamma(y+w_i/k)}{\Gamma(y+1)\Gamma(w_i/k)}\right)$$

- Poisson:

$$l_i = \frac{w_i}{\phi}[y_i\log(\mu_i) - \mu_i]$$

- Binomial:

$$l_i = \frac{w_i}{\phi}[r_i\log(p_i) + (n_i - r_i)\log(1-p_i)]$$

- Multinomial:

$$l_i = \frac{w_i}{\phi}\sum_j y_{ij}\log(\mu_{ij})$$

For the binomial, multinomial, and Poisson distribution, terms involving binomial coefficients or factorials of the observed counts are dropped from the computation of the log-likelihood function since they do not affect parameter estimates or their estimated covariances. The value of ϕ used in computing the reported log-likelihood is either the final estimated value, or the fixed value, if the dispersion parameter is fixed.

Maximum Likelihood Fitting

The GENMOD procedure uses a ridge-stabilized Newton-Raphson algorithm to maximize the log-likelihood function $L(\mathbf{y}, \boldsymbol{\mu}, \phi)$ with respect to the regression parameters. By default, the procedure also produces maximum likelihood estimates of the scale parameter as defined in the "Response Probability Distributions" section (page 1650) for the normal, inverse Gaussian, negative binomial, and gamma distributions.

On the rth iteration, the algorithm updates the parameter vector $\boldsymbol{\beta}_r$ with

$$\boldsymbol{\beta}_{r+1} = \boldsymbol{\beta}_r - \mathbf{H}^{-1}\mathbf{s}$$

where $\mathbf{H}$ is the Hessian (second derivative) matrix, and $\mathbf{s}$ is the gradient (first derivative) vector of the log-likelihood function, both evaluated at the current value of the parameter vector. That is,

$$\mathbf{s} = [s_j] = \left[\frac{\partial L}{\partial \beta_j}\right]$$

and

$$\mathbf{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \beta_i \partial \beta_j}\right]$$

In some cases, the scale parameter is estimated by maximum likelihood. In these cases, elements corresponding to the scale parameter are computed and included in $\mathbf{s}$ and $\mathbf{H}$.

If $\eta_i = \mathbf{x}_i{}'\boldsymbol{\beta}$ is the linear predictor for observation i and g is the link function, then $\eta_i = g(\mu_i)$, so that $\mu_i = g^{-1}(\mathbf{x}_i{}'\boldsymbol{\beta})$ is an estimate of the mean of the ith observation, obtained from an estimate of the parameter vector $\boldsymbol{\beta}$.

The gradient vector and Hessian matrix for the regression parameters are given by

$$\begin{aligned} \mathbf{s} &= \sum_i \frac{w_i(y_i - \mu_i)\mathbf{x}_i}{V(\mu_i)g'(\mu_i)\phi} \\ \mathbf{H} &= -\mathbf{X}'\mathbf{W}_o\mathbf{X} \end{aligned}$$

where $\mathbf{X}$ is the design matrix, $\mathbf{x}_i$ is the transpose of the ith row of $\mathbf{X}$, and V is the variance function. The matrix $\mathbf{W}_o$ is diagonal with its ith diagonal element

$$w_{oi} = w_{ei} + w_i(y_i - \mu_i)\frac{V(\mu_i)g''(\mu_i) + V'(\mu_i)g'(\mu_i)}{(V(\mu_i))^2(g'(\mu_i))^3\phi}$$

where

$$w_{ei} = \frac{w_i}{\phi V(\mu_i)(g'(\mu_i))^2}$$

The primes denote derivatives of g and V with respect to μ. The negative of $\mathbf{H}$ is called the observed information matrix. The expected value of $\mathbf{W}_o$ is a diagonal matrix $\mathbf{W}_e$ with diagonal values w_{ei}. If you replace $\mathbf{W}_o$ with $\mathbf{W}_e$, then the negative of $\mathbf{H}$ is called the expected information matrix. $\mathbf{W}_e$ is the weight matrix for the Fisher's scoring method of fitting. Either $\mathbf{W}_o$ or $\mathbf{W}_e$ can be used in the update equation. The GENMOD procedure uses Fisher's scoring for iterations up to the number specified by the SCORING option in the MODEL statement, and it uses the observed information matrix on additional iterations.

Covariance and Correlation Matrix

The estimated covariance matrix of the parameter estimator is given by

$$\mathbf{\Sigma} = -\mathbf{H}^{-1}$$

where $\mathbf{H}$ is the Hessian matrix evaluated using the parameter estimates on the last iteration. Note that the dispersion parameter, whether estimated or specified, is incorporated into $\mathbf{H}$. Rows and columns corresponding to aliased parameters are not included in $\mathbf{\Sigma}$.

The correlation matrix is the normalized covariance matrix. That is, if σ_{ij} is an element of $\mathbf{\Sigma}$, then the corresponding element of the correlation matrix is $\sigma_{ij}/\sigma_i\sigma_j$, where $\sigma_i = \sqrt{\sigma_{ii}}$.

Goodness of Fit

Two statistics that are helpful in assessing the goodness of fit of a given generalized linear model are the scaled deviance and Pearson's chi-square statistic. For a fixed value of the dispersion parameter ϕ, the scaled deviance is defined to be twice the difference between the maximum achievable log likelihood and the log likelihood at the maximum likelihood estimates of the regression parameters.

Note that these statistics are not valid for GEE models.

If $l(\mathbf{y}, \boldsymbol{\mu})$ is the log-likelihood function expressed as a function of the predicted mean values $\boldsymbol{\mu}$ and the vector $\mathbf{y}$ of response values, then the scaled deviance is defined by

$$D^*(\mathbf{y}, \boldsymbol{\mu}) = 2(l(\mathbf{y}, \mathbf{y}) - l(\mathbf{y}, \boldsymbol{\mu}))$$

For specific distributions, this can be expressed as

$$D^*(\mathbf{y}, \boldsymbol{\mu}) = \frac{D(\mathbf{y}, \boldsymbol{\mu})}{\phi}$$

where D is the deviance. The following table displays the deviance for each of the probability distributions available in PROC GENMOD.

Distribution	**Deviance**
normal	$\sum_i w_i(y_i - \mu_i)^2$
Poisson	$2\sum_i w_i \left[y_i \log\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i)\right]$
binomial	$2\sum_i w_i m_i \left[y_i \log\left(\frac{y_i}{\mu_i}\right) + (1 - y_i)\log\left(\frac{1-y_i}{1-\mu_i}\right)\right]$
gamma	$2\sum_i w_i \left[-\log\left(\frac{y_i}{\mu_i}\right) + \frac{y_i-\mu_i}{\mu_i}\right]$
inverse Gaussian	$\sum_i \frac{w_i(y_i-\mu_i)^2}{\mu_i^2 y_i}$
multinomial	$\sum_i \sum_j w_i y_{ij} \log\left(\frac{y_{ij}}{p_{ij} m_i}\right)$
negative binomial	$2\sum_i \left[y \log(y/\mu) - (y + w_i/k)\log\left(\frac{y+w_i/k}{\mu+w_i/k}\right)\right]$

In the binomial case, $y_i = r_i/m_i$, where r_i is a binomial count and m_i is the binomial number of trials parameter.

In the multinomial case, y_{ij} refers to the observed number of occurrences of the jth category for the ith subpopulation defined by the AGGREGATE= variable, m_i is the total number in the ith subpopulation, and p_{ij} is the category probability.

Pearson's chi-square statistic is defined as

$$X^2 = \sum_i \frac{w_i(y_i - \mu_i)^2}{V(\mu_i)}$$

and the scaled Pearson's chi-square is X^2/ϕ.

The scaled version of both of these statistics, under certain regularity conditions, has a limiting chi-square distribution, with degrees of freedom equal to the number of observations minus the number of parameters estimated. The scaled version can be used as an approximate guide to the goodness of fit of a given model. Use caution before applying these statistics to ensure that all the conditions for the asymptotic distributions hold. McCullagh and Nelder (1989) advise that differences in deviances for nested models can be better approximated by chi-square distributions than the deviances themselves.

In cases where the dispersion parameter is not known, an estimate can be used to obtain an approximation to the scaled deviance and Pearson's chi-square statistic. One strategy is to fit a model that contains a sufficient number of parameters so that all systematic variation is removed, estimate ϕ from this model, and then use this estimate in computing the scaled deviance of sub-models. The deviance or Pearson's chi-square divided by its degrees of freedom is sometimes used as an estimate of the dispersion parameter ϕ. For example, since the limiting chi-square distribution of the scaled deviance $D^* = D/\phi$ has $n - p$ degrees of freedom, where n is the number of observations and p the number of parameters, equating D^* to its mean and solving for ϕ yields $\hat{\phi} = D/(n - p)$. Similarly, an estimate of ϕ based on Pearson's chi-square X^2 is $\hat{\phi} = X^2/(n - p)$. Alternatively, a maximum likelihood estimate of ϕ can be computed by the procedure, if desired. See the discussion in the "Type 1 Analysis" section on page 1665 for more on the estimation of the dispersion parameter.

Dispersion Parameter

There are several options available in PROC GENMOD for handling the exponential distribution dispersion parameter. The NOSCALE and SCALE options in the MODEL statement affect the way in which the dispersion parameter is treated. If you specify the SCALE=DEVIANCE option, the dispersion parameter is estimated by the deviance divided by its degrees of freedom. If you specify the SCALE=PEARSON option, the dispersion parameter is estimated by Pearson's chi-square statistic divided by its degrees of freedom.

Otherwise, values of the SCALE and NOSCALE options and the resultant actions are displayed in the following table.

NOSCALE	**SCALE=*value***	**Action**
present	present	scale fixed at *value*
present	not present	scale fixed at 1
not present	not present	scale estimated by ML
not present	present	scale estimated by ML, starting point at *value*

The meaning of the scale parameter displayed in the "Analysis Of Parameter Estimates" table is different for the Gamma distribution than for the other distributions. The relation of the scale parameter as used by PROC GENMOD to the exponential family dispersion parameter ϕ is displayed in the following table. For the binomial and Poisson distributions, ϕ is the overdispersion parameter, as defined in the "Overdispersion" section, which follows.

Distribution	**Scale**
normal	$\sqrt{\phi}$
inverse Gaussian	$\sqrt{\phi}$
gamma	$1/\phi$
binomial	$\sqrt{\phi}$
Poisson	$\sqrt{\phi}$

In the case of the negative binomial distribution, PROC GENMOD reports the "dispersion" parameter estimated by maximum likelihood. This is the negative binomial parameter k defined in the "Response Probability Distributions" section (page 1650).

Overdispersion

Overdispersion is a phenomenon that sometimes occurs in data that are modeled with the binomial or Poisson distributions. If the estimate of dispersion after fitting, as measured by the deviance or Pearson's chi-square, divided by the degrees of freedom, is not near 1, then the data may be *overdispersed* if the dispersion estimate is greater than 1 or *underdispersed* if the dispersion estimate is less than 1. A simple way to model this situation is to allow the variance functions of these distributions to have a multiplicative overdispersion factor ϕ.

- binomial: $V(\mu) = \phi\mu(1 - \mu)$
- Poisson: $V(\mu) = \phi\mu$

The models are fit in the usual way, and the parameter estimates are not affected by the value of ϕ. The covariance matrix, however, is multiplied by ϕ, and the scaled deviance and log likelihoods used in likelihood ratio tests are divided by ϕ. The profile likelihood function used in computing confidence intervals is also divided by ϕ. If you specify an WEIGHT statement, ϕ is divided by the value of the WEIGHT variable for each observation. This has the effect of multiplying the contributions of the log-likelihood function, the gradient, and the Hessian by the value of the WEIGHT variable for each observation.

The SCALE= option in the MODEL statement enables you to specify a value of $\sigma = \sqrt{\phi}$ for the binomial and Poisson distributions. If you specify the SCALE=DEVIANCE option in the MODEL statement, the procedure uses the deviance divided by degrees of freedom as an estimate of ϕ, and all statistics are adjusted appropriately. You can use Pearson's chi-square instead of the deviance by specifying the SCALE=PEARSON option.

The function obtained by dividing a log-likelihood function for the binomial or Poisson distribution by a dispersion parameter is not a legitimate log-likelihood function. It is an example of a *quasi-likelihood* function. Most of the asymptotic theory for log likelihoods also applies to quasi-likelihoods, which justifies computing standard errors and likelihood ratio statistics using quasi-likelihoods instead of proper log likelihoods. Refer to McCullagh and Nelder (1989, Chapter 9) and McCullagh (1983) for details on quasi-likelihood functions.

Although the estimate of the dispersion parameter is often used to indicate overdispersion or underdispersion, this estimate may also indicate other problems such as an incorrectly specified model or outliers in the data. You should carefully assess whether this type of model is appropriate for your data.

Specification of Effects

Each term in a model is called an effect. Effects are specified in the MODEL statement in the same way as in the GLM procedure. You specify effects with a special notation that uses variable names and operators. There are two types of variables, *classification* (or *class*) variables and *continuous* variables. There are two primary types of operators, *crossing* and *nesting*. A third type, the *bar* operator, is used to simplify effect specification. Crossing is the type of operator most commonly used in generalized linear models.

Variables that identify classification levels are called *class* variables in the SAS System and are identified in a CLASS statement. These may also be called *categorical, qualitative, discrete,* or *nominal* variables. Class variables can be either character or numeric. The values of class variables are called *levels*. For example, the class variable Sex could have levels 'male' and 'female'.

In a model, an explanatory variable that is not declared in a CLASS statement is assumed to be continuous. Continuous variables must be numeric. For example, the heights and weights of subjects in an experiment are continuous variables.

The types of effects most useful in generalized linear models are shown in the following list. Assume that A, B, and C are class variables and that X1 and X2 are continuous variables.

- Regressor effects are specified by writing continuous variables by themselves: X1, X2.
- Polynomial effects are specified by joining two or more continuous variables with asterisks: X1*X2.
- Main effects are specified by writing class variables by themselves: A, B, C.
- Crossed effects (interactions) are specified by joining two or more class variables with asterisks: A*B, B*C, A*B*C.
- Nested effects are specified by following a main effect or crossed effect with a class variable or list of class variables enclosed in parentheses: B(A), C(B A), A*B(C). In the preceding example, B(A) is "B nested within A."
- Combinations of continuous and class variables can be specified in the same way using the crossing and nesting operators.

The bar operator consists of two effects joined with a vertical bar (|). It is shorthand notation for including the left-hand side, the right-hand side, and the cross between them as effects in the model. For example, A | B is equivalent to A B A*B. The effects in the bar operator can be class variables, continuous variables, or combinations of effects defined using operators. Multiple bars are permitted. For example, A | B | C means A B C A*B A*C B*C A*B*C.

You can specify the maximum number of variables in any effect that results from bar evaluation by specifying the maximum number, preceded by an @ sign. For example, A | B | C@2 results in effects that involve two or fewer variables: A B C A*B A*C B*C.

For further information on types of effects and their specification, see Chapter 32, "The GLM Procedure."

Parameterization Used in PROC GENMOD

Design Matrix

The linear predictor part of a generalized linear model is

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$$

where $\boldsymbol{\beta}$ is an unknown parameter vector and $\mathbf{X}$ is a known design matrix. By default, all models automatically contain an intercept term; that is, the first column of $\mathbf{X}$ contains all 1s. Additional columns of $\mathbf{X}$ are generated for classification variables, regression variables, and any interaction terms included in the model. PROC GENMOD parameterizes main effects and interaction terms using the same ordering rules that PROC GLM uses. This is important to understand when you want to construct likelihood ratios for custom contrasts using the CONTRAST statement. See Chapter 32, "The GLM Procedure," for more details on model parameterization.

Some columns of $\mathbf{X}$ can be linearly dependent on other columns due to specifying an overparameterized model. For example, when you specify a model consisting of an intercept term and a class variable, the column corresponding to any one of the levels of the class variable is linearly dependent on the other columns of $\mathbf{X}$. PROC GENMOD handles this in the same manner as PROC GLM. The columns of $\mathbf{X}'\mathbf{X}$ are checked in the order in which the model is specified for dependence on preceding columns. If a dependency is found, the parameter corresponding to the dependent column is set to 0 along with its standard error to indicate that it is not estimated. The order in which the levels of a class variable are checked for dependencies can be set by the ORDER= option in the PROC GENMOD statement.

You can exclude the intercept term from the model by specifying the NOINT option in the MODEL statement.

Missing Level Combinations

All levels of interaction terms involving classification variables may not be represented in the data. In that case, PROC GENMOD does not include parameters in the model for the missing levels.

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of -1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	A1	A2	A5	A7
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects using the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL

THERMOMETER Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	A2	A5	A7
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects using the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, . . . according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Type 1 Analysis

A Type 1 analysis consists of fitting a sequence of models, beginning with a simple model with only an intercept term, and continuing through a model of specified complexity, fitting one additional effect on each step. Likelihood ratio statistics, that is, twice the difference of the log likelihoods, are computed between successive models. This type of analysis is sometimes called an analysis of deviance since, if the dispersion parameter is held fixed for all models, it is equivalent to computing differences of scaled deviances. The asymptotic distribution of the likelihood ratio statistics, under the hypothesis that the additional parameters included in the model are equal to 0, is a chi-square with degrees of freedom equal to the difference in the number of parameters estimated in the successive models. Thus, these statistics can be used in a test of hypothesis of the significance of each additional term fit.

This type of analysis is not available for GEE models, since the deviance is not computed for this type of model.

If the dispersion parameter ϕ is known, it can be included in the models; if it is unknown, there are two strategies allowed by PROC GENMOD. The dispersion parameter can be estimated from a maximal model by the deviance or Pearson's chi-square divided by degrees of freedom, as discussed in the "Goodness of Fit" section on page 1656, and this value can be used in all models. An alternative is to consider the dispersion to be an additional unknown parameter for each model and estimate it by maximum likelihood on each step. By default, PROC GENMOD estimates scale by maximum likelihood at each step.

A table of likelihood ratio statistics is produced, along with associated p-values based on the asymptotic chi-square distributions.

If you specify either the SCALE=DEVIANCE or the SCALE=PEARSON option in the MODEL statement, the dispersion parameter is estimated using the deviance or Pearson's chi-square statistic, and F statistics are computed in addition to the chi-square statistics for assessing the significance of each additional term in the Type 1 analysis. See the section "F Statistics" on page 1668 for a definition of F statistics.

This Type 1 analysis has the general property that the results depend on the order in which the terms of the model are fitted. The terms are fitted in the order in which they are specified in the MODEL statement.

Type 3 Analysis

A Type 3 analysis is similar to the Type III sums of squares used in PROC GLM, except that likelihood ratios are used instead of sums of squares. First, a Type III estimable function is defined for an effect of interest in exactly the same way as in PROC GLM. Then, maximum likelihood estimation is performed under the constraint that the Type III function of the parameters is equal to 0, using constrained optimization. Let the resulting constrained parameter estimates be $\tilde{\boldsymbol{\beta}}$ and the log likelihood be $l(\tilde{\boldsymbol{\beta}})$. Then the likelihood ratio statistic

$$S = 2(l(\hat{\boldsymbol{\beta}}) - l(\tilde{\boldsymbol{\beta}}))$$

where $\hat{\boldsymbol{\beta}}$ is the unconstrained estimate, has an asymptotic chi-square distribution under the hypothesis that the Type III contrast is equal to 0, with degrees of freedom equal to the number of parameters associated with the effect.

When a Type 3 analysis is requested, PROC GENMOD produces a table that contains the likelihood ratio statistics, degrees of freedom, and p-values based on the limiting chi-square distributions for each effect in the model. If you specify either the DSCALE or PSCALE option in the MODEL statement, F statistics are also computed for each effect.

Options for handling the dispersion parameter are the same as for a Type 1 analysis. The dispersion parameter can be specified to be a known value, estimated from the deviance or Pearson's chi-square divided by degrees of freedom, or estimated by maximum likelihood individually for the unconstrained and constrained models. By default, PROC GENMOD estimates scale by maximum likelihood for each model fit.

The results of this type of analysis do not depend on the order in which the terms are specified in the MODEL statement.

A Type 3 analysis can consume considerable computation time since a constrained model is fitted for each effect. Wald statistics for Type 3 contrasts are computed if you specify the WALD option. Wald statistics for contrasts use less computation time than likelihood ratio statistics but may be less accurate indicators of the significance of the effect of interest. The Wald statistic for testing $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$, where $\mathbf{L}$ is the contrast matrix, is defined by

$$S = (\mathbf{L}'\hat{\boldsymbol{\beta}})'(\mathbf{L}'\hat{\boldsymbol{\Sigma}}\mathbf{L})^{-}(\mathbf{L}'\hat{\boldsymbol{\beta}})$$

where $\boldsymbol{\beta}$ is the maximum likelihood estimate and $\boldsymbol{\Sigma}$ is its estimated covariance matrix. The asymptotic distribution of S is chi-square with r degrees of freedom, where r is the rank of $\mathbf{L}$.

See Chapter 32, "The GLM Procedure," and Chapter 11, "The Four Types of Estimable Functions," for more information about Type III estimable functions. Also refer to Littell, Freund, and Spector (1991).

Generalized score tests for Type III contrasts are computed for GEE models if you specify the TYPE3 option in the MODEL statement when a REPEATED statement is also used. See the section "Generalized Score Statistics" on page 1680 for more information on generalized score statistics. Wald tests are also available with the Wald option in the CONTRAST statement.

Confidence Intervals for Parameters

Likelihood Ratio-Based Confidence Intervals

PROC GENMOD produces likelihood ratio-based confidence intervals, also known as profile likelihood confidence intervals, for parameter estimates for generalized linear models. These are not computed for GEE models, since there is no likelihood for this type of model. Suppose that the parameter vector is $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]'$ and that you want a confidence interval for β_j. The profile likelihood function for β_j is defined as

$$l^*(\beta_j) = \max_{\tilde{\boldsymbol{\beta}}} l(\boldsymbol{\beta})$$

where $\tilde{\boldsymbol{\beta}}$ is the vector $\boldsymbol{\beta}$ with the jth element fixed at β_j and l is the log likelihood function. If $l = l(\hat{\boldsymbol{\beta}})$ is the log likelihood evaluated at the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$, then $2(l - l^*(\beta_j))$ has a limiting chi-square distribution with one degree of freedom if β_j is the true parameter value. A $(1 - \alpha)100\%$ confidence interval for β_j is

$$\left\{\beta_j : l^*(\beta_j) \geq l_0 = l - 0.5\chi^2_{1-\alpha,1}\right\}$$

where $\chi^2_{1-\alpha,1}$ is the $100(1 - \alpha)$ percentile of the chi-square distribution with one degree of freedom. The endpoints of the confidence interval can be found by solving numerically for values of β_j that satisfy equality in the preceding relation. PROC GENMOD solves this by starting at the maximum likelihood estimate of $\boldsymbol{\beta}$. The log likelihood function is approximated with a quadratic surface, for which an exact solution is possible. The process is iterated until convergence to an endpoint is attained. The process is repeated for the other endpoint.

Convergence is controlled by the CICONV= option in the MODEL statement. Suppose ϵ is the number specified in the CICONV= option. The default value of ϵ is 10^{-4}. Let the parameter of interest be β_j and define $\mathbf{r} = \mathbf{u}_j$, the unit vector with a 1 in position j and 0s elsewhere. Convergence is declared on the current iteration if the following two conditions are satisfied:

$$\begin{aligned} |l^*(\beta_j) - l_0| &\leq \epsilon \\ (\mathbf{s} + \lambda\mathbf{r})'\mathbf{H}^{-1}(\mathbf{s} + \lambda\mathbf{r}) &\leq \epsilon \end{aligned}$$

where $l^*(\beta_j)$, $\mathbf{s}$, and $\mathbf{H}$ are the log likelihood, the gradient, and the Hessian evaluated at the current parameter vector and λ is a constant computed by the procedure. The first condition for convergence means that the log-likelihood function must be within ϵ of the correct value, and the second condition means that the gradient vector must be proportional to the restriction vector $\mathbf{r}$.

When you request the LRCI option in the MODEL statement, PROC GENMOD computes profile likelihood confidence intervals for all parameters in the model, including the scale parameter, if there is one. The interval endpoints are displayed in a table as well as the values of the remaining parameters at the solution.

Wald Confidence Intervals

You can request that PROC GENMOD produce Wald confidence intervals for the parameters. The (1-α)100% Wald confidence interval for a parameter β is defined as

$$\hat{\beta} \pm z_{1-\alpha/2}\hat{\sigma}$$

where z_p is the $100p$th percentile of the standard normal distribution, $\hat{\beta}$ is the parameter estimate, and $\hat{\sigma}$ is the estimate of its standard error.

F Statistics

Suppose that D_0 is the deviance resulting from fitting a generalized linear model and that D_1 is the deviance from fitting a submodel. Then, under appropriate regularity conditions, the asymptotic distribution of $(D_1 - D_0)/\phi$ is chi-square with r degrees of freedom, where r is the difference in the number of parameters between the two models and ϕ is the dispersion parameter. If ϕ is unknown, and $\hat{\phi}$ is an estimate of ϕ based on the deviance or Pearson's chi-square divided by degrees of freedom, then, under regularity conditions, $(n - p)\hat{\phi}/\phi$ has an asymptotic chi-square distribution with $n - p$ degrees of freedom. Here, n is the number of observations and p is the number of parameters in the model that is used to estimate ϕ. Thus, the asymptotic distribution of

$$F = \frac{D_1 - D_0}{r\hat{\phi}}$$

is the F distribution with r and $n-p$ degrees of freedom, assuming that $(D_1-D_0)/\phi$ and $(n - p)\hat{\phi}/\phi$ are approximately independent.

This F statistic is computed for the Type 1 analysis, Type 3 analysis, and hypothesis tests specified in CONTRAST statements when the dispersion parameter is estimated by either the deviance or Pearson's chi-square divided by degrees of freedom, as specified by the DSCALE or PSCALE option in the MODEL statement. In the case of a Type 1 analysis, model 0 is the higher-order model obtained by including one additional effect in model 1. For a Type 3 analysis and hypothesis tests, model 0 is the full specified model and model 1 is the sub-model obtained from constraining the Type III contrast or the user-specified contrast to be 0.

Lagrange Multiplier Statistics

When you select the NOINT or NOSCALE option, restrictions are placed on the intercept or scale parameters. Lagrange multiplier, or score, statistics are computed in these cases. These statistics assess the validity of the restrictions, and they are computed as

$$\chi^2 = \frac{s^2}{V}$$

where s is the component of the score vector evaluated at the restricted maximum corresponding to the restricted parameter and $V = \mathbf{I}_{11} - \mathbf{I}_{12}\mathbf{I}_{22}^{-1}\mathbf{I}_{21}$. The matrix $\mathbf{I}$ is the information matrix, 1 refers to the restricted parameter, and 2 refers to the rest of the parameters.

Under regularity conditions, this statistic has an asymptotic chi-square distribution with one degree of freedom, and p-values are computed based on this limiting distribution.

If you set $k = 0$ in a negative binomial model, s is the score statistic of Cameron and Trivedi (1998) for testing for overdispersion in a Poisson model against alternatives of the form $V(\mu) = \mu + k\mu^2$.

Refer to Rao (1973, p. 417) for more details.

Predicted Values of the Mean

Predicted Values

A predicted value, or fitted value, of the mean μ_i corresponding to the vector of covariates $\mathbf{x}_i$ is given by

$$\hat{\mu_i} = g^{-1}(\mathbf{x}_i'\hat{\boldsymbol{\beta}})$$

where g is the link function, regardless of whether $\mathbf{x}_i$ corresponds to an observation or not. That is, the response variable can be missing and the predicted value is still computed for valid $\mathbf{x}_i$. In the case where $\mathbf{x}_i$ does not correspond to a valid observation, $\mathbf{x}_i$ is not checked for estimability. You should check the estimability of $\mathbf{x}_i$ in this case in order to ensure the uniqueness of the predicted value of the mean. If there is an offset, it is included in the predicted value computation.

Confidence Intervals on Predicted Values

Approximate confidence intervals for predicted values of the mean can be computed as follows. The variance of the linear predictor $\eta_i = \mathbf{x}_i'\hat{\boldsymbol{\beta}}$ is estimated by

$$\sigma_x^2 = \mathbf{x}_i'\boldsymbol{\Sigma}\mathbf{x}_i$$

where $\boldsymbol{\Sigma}$ is the estimated covariance of $\hat{\boldsymbol{\beta}}$.

Approximate $100(1-\alpha)\%$ confidence intervals are computed as

$$g^{-1}\left(\mathbf{x}_i'\hat{\beta} \pm z_{1-\alpha/2}\sigma_x\right)$$

where z_p is the $100p$ percentile of the standard normal distribution and g is the link function. If either endpoint in the argument is outside the valid range of arguments for the inverse link function, the corresponding confidence interval endpoint is set to missing.

Residuals

The GENMOD procedure computes three kinds of residuals. The raw residual is defined as

$$r_i = y_i - \mu_i$$

where y_i is the ith response and μ_i is the corresponding predicted mean.

The Pearson residual is the square root of the ith contribution to the Pearson's chi-square.

$$r_{Pi} = (y_i - \mu_i)\sqrt{\frac{w_i}{V(\mu_i)}}$$

Finally, the deviance residual is defined as the square root of the contribution of the ith observation to the deviance, with the sign of the raw residual.

$$r_{Di} = \sqrt{d_i}(\mathit{sign}(y_i - \mu_i))$$

The adjusted Pearson, deviance, and likelihood residuals are defined by Agresti (1990), Williams (1987), and Davison and Snell (1991). These residuals are useful for outlier detection and for assessing the influence of single observations on the fitted model.

For the generalized linear model, the variance of the ith individual observation is given by

$$v_i = \frac{\phi V(\mu_i)}{w_i}$$

where ϕ is the dispersion parameter, w_i is a user-specified prior weight (if not specified, $w_i = 1$), μ_i is the mean, and $V(\mu_i)$ is the variance function. Let

$$w_{ei} = v_i^{-1}(g'(\mu_i))^{-2}$$

for the ith observation, where $g'(\mu_i)$ is the derivative of the link function, evaluated at μ_i. Let $\mathbf{W}_e$ be the diagonal matrix with w_{ei} denoting the ith diagonal element. The weight matrix $\mathbf{W}_e$ is used in computing the expected information matrix.

Define h_i as the ith diagonal element of the matrix

$$\mathbf{W}_e^{\frac{1}{2}}\mathbf{X}(\mathbf{X}'\mathbf{W}_e\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_e^{\frac{1}{2}}$$

The Pearson residuals, standardized to have unit asymptotic variance, are given by

$$r_{Pi} = \frac{y_i - \mu_i}{\sqrt{v_i(1 - h_i)}}$$

The deviance residuals, standardized to have unit asymptotic variance, are given by

$$r_{Di} = \frac{\mathit{sign}(y_i - \mu_i)\sqrt{d_i}}{\sqrt{\phi(1 - h_i)}}$$

where d_i is the square root of the contribution to the total deviance from observation i, and sign$(y_i - \mu_i)$ is 1 if $y_i - \mu_i$ is positive and -1 if $y_i - \mu_i$ is negative. The likelihood residuals are defined by

$$r_{Gi} = \mathit{sign}(y_i - \mu_i)\sqrt{(1 - h_i)r_{Di}^2 + h_i r_{Pi}^2}$$

Multinomial Models

This type of model applies to cases where an observation can fall into one of k categories. Binary data occurs in the special case where $k = 2$. If there are m_i observations in a subpopulation i, then the probability distribution of the number falling into the k categories $\mathbf{y}_i = (y_{i1}, y_{i2}, \cdots y_{ik})$ can be modeled by the multinomial distribution, defined in the "Response Probability Distributions" section (page 1650), with $\sum_j y_{ij} = m_i$. The multinomial model is an *ordinal* model if the categories have a natural order.

The GENMOD procedure orders the response categories for ordinal multinomial models from lowest to highest by default. This is different from the binomial distribution, where the response probability for the lowest of the two categories is modeled. You can change the way GENMOD orders the response levels with the RORDER= option in the PROC GENMOD statement. The order that GENMOD uses is shown in the "Response Profiles" output table described in the section "Response Profile" on page 1685.

The GENMOD procedure supports only the ordinal multinomial model. If $(p_{i1}, p_{i2}, \cdots p_{ik})$ are the category probabilities, the cumulative category probabilities are modeled with the same link functions used for binomial data. Let $P_{ir} = \sum_{j=1}^{r} p_{ij}$, $r = 1, 2, \cdots, k-1$ be the cumulative category probabilities (note that $P_{ik} = 1$). The ordinal model is

$$g(P_{ir}) = \mu_r + \mathbf{x}_i{}'\boldsymbol{\beta} \quad \text{for} \quad r = 1, 2, \cdots k-1$$

where $\mu_1, \mu_2, \cdots \mu_{k-1}$ are intercept terms that depend only on the categories and $\mathbf{x}_i$ is a vector of covariates that does not include an intercept term. The logit, probit, and complementary log-log link functions g are available. These are obtained by specifying the MODEL statement options DIST=MULTINOMIAL and LINK=CUMLOGIT (cumulative logit), LINK=CUMPROBIT (cumulative probit), or LINK=CUMCLL (cumulative complementary log-log). Alternatively,

$$P_{ir} = \mathrm{F}(\mu_r + \mathbf{x}_i{}'\boldsymbol{\beta}) \quad \text{for} \quad r = 1, 2, \cdots k-1$$

where $\mathrm{F} = g^{-1}$ is a cumulative distribution function for the logistic, normal, or extreme value distribution.

PROC GENMOD estimates the intercept parameters $\mu_1, \mu_2, \cdots \mu_{k-1}$ and regression parameters $\boldsymbol{\beta}$ by maximum likelihood.

The subpopulations i are defined by constant values of the AGGREGATE= variable. This has no effect on the parameter estimates, but it does affect the deviance and Pearson chi-square statistics; it also affects parameter estimate standard errors if you specify the SCALE=DEVIANCE or SCALE=PEARSON options.

Generalized Estimating Equations

Let $y_{ij}, j = 1, \ldots, n_i, i = 1, \ldots, K$ represent the jth measurement on the ith subject. There are n_i measurements on subject i and $\sum_{i=1}^{K} n_i$ total measurements.

Correlated data are modeled using the same link function and linear predictor setup (systematic component) as the independence case. The random component is described by the same variance functions as in the independence case, but the covariance structure of the correlated measurements must also be modeled. Let the vector of measurements on the ith subject be $\mathbf{Y}_i = [y_{i1}, \ldots, y_{in_i}]'$ with corresponding vector of means $\boldsymbol{\mu}_i = [\mu_{i1}, \ldots, \mu_{in_i}]'$ and let $\mathbf{V}_i$ be the covariance matrix of $\mathbf{Y}_i$. Let the vector of independent, or explanatory, variables for the jth measurement on the ith subject be

$$\mathbf{x}_{ij} = [x_{ij1}, \ldots, x_{ijp}]'$$

The Generalized Estimating Equation of Liang and Zeger (1986) for estimating the $p\times 1$ vector of regression parameters $\boldsymbol{\beta}$ is an extension of the independence estimating equation to correlated data and is given by

$$\mathbf{S}(\boldsymbol{\beta}) = \sum_{i=1}^{K} \mathbf{D}_i' \mathbf{V}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}_i(\boldsymbol{\beta})) = \mathbf{0}$$

where

$$\mathbf{D}_i = \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

Since

$$g(\mu_{ij}) = \mathbf{x}_{ij}{}'\boldsymbol{\beta}$$

where g is the link function, the $p \times n_i$ matrix of partial derivatives of the mean with respect to the regression parameters for the ith subject is given by

$$\mathbf{D}_i' = \frac{\partial \boldsymbol{\mu}_i'}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{x_{i11}}{g'(\mu_{i1})} & \cdots & \frac{x_{in_i1}}{g'(\mu_{in_i})} \\ \vdots & & \vdots \\ \frac{x_{i1p}}{g'(\mu_{i1})} & \cdots & \frac{x_{in_ip}}{g'(\mu_{in_i})} \end{bmatrix}$$

Working Correlation Matrix

Let $\mathbf{R}_i(\boldsymbol{\alpha})$ be an $n_i \times n_i$ "working" correlation matrix that is fully specified by the vector of parameters $\boldsymbol{\alpha}$. The covariance matrix of $\mathbf{Y}_i$ is modeled as

$$\mathbf{V}_i = \phi \mathbf{A}_i^{\frac{1}{2}} \mathbf{W}_i^{-\frac{1}{2}} \mathbf{R}(\boldsymbol{\alpha}) \mathbf{W}_i^{-\frac{1}{2}} \mathbf{A}_i^{\frac{1}{2}}$$

where $\mathbf{A}_i$ is an $n_i \times n_i$ diagonal matrix with $v(\mu_{ij})$ as the jth diagonal element and $\mathbf{W}_i$ is an $n_i \times n_i$ diagonal matrix with w_{ij} as the jth diagonal where w_{ij} is a weight specified with the WEIGHT statement. If there is no WEIGHT statement, $w_{ij} = 1$ for all i and j. If $\mathbf{R}_i(\boldsymbol{\alpha})$ is the true correlation matrix of $\mathbf{Y}_i$, then $\mathbf{V}_i$ is the true covariance matrix of $\mathbf{Y}_i$.

The working correlation matrix is usually unknown and must be estimated. It is estimated in the iterative fitting process using the current value of the parameter vector $\boldsymbol{\beta}$ to compute appropriate functions of the Pearson residual

$$e_{ij} = \frac{y_{ij} - \mu_{ij}}{\sqrt{v(\mu_{ij})/w_{ij}}}$$

If you specify the working correlation as $\mathbf{R}_0 = \mathbf{I}$, which is the identity matrix, the GEE reduces to the independence estimating equation.

Following are the structures of the working correlation supported by the GENMOD procedure and the estimators used to estimate the working correlations.

Working Correlation Structure	**Estimator**
Fixed $\mathrm{Corr}(Y_{ij}, Y_{ik}) = r_{jk}$ where r_{jk} is the jkth element of a constant, user-specified correlation matrix $\mathbf{R}_0$.	The working correlation is not estimated in this case.
Independent $\mathrm{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$	The working correlation is not estimated in this case.
***m*-dependent** $\mathrm{Corr}(Y_{ij}, Y_{i,j+t}) = \begin{cases} 1 & t = 0 \\ \alpha_t & t = 1, 2, \ldots, m \\ 0 & t > m \end{cases}$	$\hat{\alpha}_t = \frac{1}{(K_t - p)\phi} \sum_{i=1}^{K} \sum_{j \le n_i - t} e_{ij} e_{i,j+t}$ $K_t = \sum_{i=1}^{K} (n_i - t)$
Exchangeable $\mathrm{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ \alpha & j \neq k \end{cases}$	$\hat{\alpha} = \frac{1}{(N^* - p)\phi} \sum_{i=1}^{K} \sum_{j<k} e_{ij} e_{ik}$ $N^* = .5 \sum_{i=1}^{K} n_i (n_i - 1)$
Unstructured $\mathrm{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ \alpha_{jk} & j \neq k \end{cases}$	$\hat{\alpha}_{jk} = \frac{1}{(K - p)\phi} \sum_{i=1}^{K} e_{ij} e_{ik}$
Autoregressive AR(1) $\mathrm{Corr}(Y_{ij}, Y_{i,j+t}) = \alpha^t$ for $t = 0, 1, 2, \ldots, n_i - j$	$\hat{\alpha} = \frac{1}{(K_1 - p)\phi} \sum_{i=1}^{K} \sum_{j \le n_i - 1} e_{ij} e_{i,j+1}$ $K_1 = \sum_{i=1}^{K} (n_i - 1)$

Dispersion Parameter

The dispersion parameter ϕ is estimated by

$$\hat{\phi} = \frac{1}{N - p} \sum_{i=1}^{K} \sum_{j=1}^{n_i} e_{ij}^2$$

where $N = \sum_{i=1}^{K} n_i$ is the total number of measurements and p is the number of regression parameters.

The square root of $\hat{\phi}$ is reported by PROC GENMOD as the scale parameter in the "Analysis of GEE Parameter Estimates Model-Based Standard Error Estimates" output table.

Fitting Algorithm

The following is an algorithm for fitting the specified model using GEEs. Note that this is not in general a likelihood-based method of estimation, so that inferences based on likelihoods are not possible for GEE methods.

1. Compute an initial estimate of $\boldsymbol{\beta}$ with an ordinary generalized linear model assuming independence.
2. Compute the working correlations $\mathbf{R}$ based on the standardized residuals, the current $\boldsymbol{\beta}$, and the assumed structure of $\mathbf{R}$.
3. Compute an estimate of the covariance:

$$\mathbf{V}_i = \phi \mathbf{A}_i^{\frac{1}{2}} \mathbf{W}_i^{-\frac{1}{2}} \hat{\mathbf{R}}(\boldsymbol{\alpha}) \mathbf{W}_i^{-\frac{1}{2}} \mathbf{A}_i^{\frac{1}{2}}$$

4. Update $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}_{r+1} = \boldsymbol{\beta}_r + \left[\sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}} \right]^{-1} \left[\sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}_i) \right]$$

5. Iterate steps 2-4 until convergence

Missing Data

Refer to Diggle, Liang, and Zeger (1994, Chapter 11) for a discussion of missing values in longitudinal data. Suppose that you intend to take measurements $Y_{i1}, \ldots, Y_{in}$ for the ith unit. Missing values for which Y_{ij} are missing whenever Y_{ik} is missing for all $j \geq k$ are called *dropouts*. Otherwise, missing values that occur intermixed with nonmissing values are *intermittent* missing values. The GENMOD procedure can estimate the working correlation from data containing both types of missing values using the *all available pairs* method, in which all nonmissing pairs of data are used in the moment estimators of the working correlation parameters defined previously.

For example, for the unstructured working correlation model,

$$\hat{\alpha}_{jk} = \frac{1}{(K' - p)\phi} \sum e_{ij} e_{ik}$$

where the sum is over the units that have nonmissing measurements at times j and k, and K' is the number of units with nonmissing measurements at j and k. Estimates of the parameters for other working correlation types are computed in a similar manner, using available nonmissing pairs in the appropriate moment estimators.

The contribution of the ith unit to the parameter update equation is computed by omitting the elements of $(\mathbf{Y}_i - \boldsymbol{\mu}_i)$, the columns of $\mathbf{D}_i' = \frac{\partial \boldsymbol{\mu}'}{\partial \boldsymbol{\beta}}$, and the rows and columns of $\mathbf{V}_i$ corresponding to missing measurements.

Parameter Estimate Covariances

The *model-based* estimator of $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ is given by

$$\boldsymbol{\Sigma}_m(\hat{\boldsymbol{\beta}}) = \mathbf{I}_0^{-1}$$

where

$$\mathbf{I}_0 = \sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

This is the GEE equivalent of the inverse of the Fisher information matrix that is often used in generalized linear models as an estimator of the covariance estimate of the maximum likelihood estimator of $\boldsymbol{\beta}$. It is a consistent estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$ if the mean model and the working correlation matrix are correctly specified.

The estimator

$$\boldsymbol{\Sigma}_e = \mathbf{I}_0^{-1}\mathbf{I}_1\mathbf{I}_0^{-1}$$

is called the *empirical*, or *robust*, estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$ where

$$\mathbf{I}_1 = \sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} \mathrm{Cov}(\mathbf{Y}_i) \mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

It has the property of being a consistent estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$, even if the working correlation matrix is misspecified, that is, if $\mathrm{Cov}(\mathbf{Y}_i) \neq \mathbf{V}_i$. In computing $\mathbf{M}$, $\boldsymbol{\beta}$ and ϕ are replaced by estimates, and $\mathrm{Cov}(\mathbf{Y}_i)$ is replaced by the estimate

$$(\mathbf{Y}_i - \boldsymbol{\mu}_i(\hat{\boldsymbol{\beta}}))(\mathbf{Y}_i - \boldsymbol{\mu}_i(\hat{\boldsymbol{\beta}}))'$$

Multinomial GEEs

Lipsitz, Kim, and Zhao (1994) and Miller, Davis, and Landis (1993) describe how to extend GEEs to multinomial data. Currently, only the independent working correlation is available for multinomial models in PROC GENMOD.

Alternating Logistic Regressions

If the responses are binary (that is, they take only two values), then there is an alternative method to account for the association among the measurements. The Alternating Logistic Regressions (ALR) algorithm of Carey, Zeger, and Diggle (1993) models the association between pairs of responses with log odds ratios, instead of with correlations, as ordinary GEEs do.

For binary data, the correlation between the *j*th and *k*th response is, by definition,

$$Corr(Y_{ij}, Y_{ik}) = \frac{\Pr(Y_{ij} = 1, Y_{ik} = 1) - \mu_{ij}\mu_{ik}}{\sqrt{\mu_{ij}(1 - \mu_{ij})\mu_{ik}(1 - \mu_{ik})}}$$

The joint probability in the numerator satisfies the following bounds, by elementary properties of probability, since $\mu_{ij} = \Pr(Y_{ij} = 1)$:

$$\max(0, \mu_{ij} + \mu_{ik} - 1) \leq \Pr(Y_{ij} = 1, Y_{ik} = 1) \leq \min(\mu_{ij}, \mu_{ik})$$

The correlation, therefore, is constrained to be within limits that depend in a complicated way on the means of the data.

The odds ratio, defined as

$$OR(Y_{ij}, Y_{ik}) = \frac{\Pr(Y_{ij} = 1, Y_{ik} = 1)\Pr(Y_{ij} = 0, Y_{ik} = 0)}{\Pr(Y_{ij} = 1, Y_{ik} = 0)\Pr(Y_{ij} = 0, Y_{ik} = 1)}$$

is not constrained by the means and is preferred, in some cases, to correlations for binary data.

The ALR algorithm seeks to model the logarithm of the odds ratio, $\gamma_{ijk} = \log(OR(Y_{ij}, Y_{ik}))$, as

$$\gamma_{ijk} = \mathbf{z}'_{ijk}\alpha$$

where α is a $q \times 1$ vector of regression parameters and $\mathbf{z}_{ijk}$ is a fixed, specified vector of coefficients.

The parameter γ_{ijk} can take any value in $(-\infty, \infty)$ with $\gamma_{ijk} = 0$ corresponding to no association.

The log odds ratio, when modeled in this way with a regression model, can take different values in subgroups defined by $\mathbf{z}_{ijk}$. For example, $\mathbf{z}_{ijk}$ can define subgroups within clusters, or it can define 'block effects' between clusters.

You specify a GEE model for binary data using log odds ratios by specifying a model for the mean, as in ordinary GEEs, and a model for the log odds ratios. You can use any of the link functions appropriate for binary data in the model for the mean, such as logistic, probit, or complementary log-log. The ALR algorithm alternates between a GEE step to update the model for the mean and a logistic regression step to update the log odds ratio model. Upon convergence, the ALR algorithm provides estimates of the regression parameters for the mean, $\boldsymbol{\beta}$, the regression parameters for the log odds ratios, α, their standard errors, and their covariances.

Specifying Log Odds Ratio Models

Specifying a regression model for the log odds ratio requires you to specify rows of the *z*-matrix $\mathbf{z}_{ijk}$ for each cluster i and each unique within-cluster pair (j,k). The GENMOD procedure provides several methods of specifying $\mathbf{z}_{ijk}$. These are controlled by the LOGOR=*keyword* and associated options in the REPEATED statement. The supported keywords and the resulting log odds ratio models are described as follows.

EXCH exchangeable log odds ratios. In this model, the log odds ratio is a constant for all clusters i and pairs (j,k). The parameter α is the common log odds ratio.

$$\mathbf{z}_{ijk} = 1 \quad \text{for all} \quad i,j,k$$

FULLCLUST fully parameterized clusters. Each cluster is parameterized in the same way, and there is a parameter for each unique pair within clusters. If a complete cluster is of size n, then there are $\frac{n(n-1)}{2}$ parameters in the vector α. For example, if a full cluster is of size 4, then there are $\frac{4\times3}{2} = 6$ parameters, and the *z*-matrix is of the form

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements of α correspond to log odds ratios for cluster pairs in the following order:

Pair	Parameter
(1,2)	Alpha1
(1,3)	Alpha2
(1,4)	Alpha3
(2.3)	Alpha4
(2,4)	Alpha5
(3,4)	Alpha6

LOGORVAR(*variable*) log odds ratios by cluster. The argument *variable* is a variable name that defines the ‘block effects’ between clusters. The log odds ratios are constant within clusters, but they take a different value for each different value of the *variable*. For example, if Center is a variable in the input data set taking a different value for k treatment centers, then specifying LOGOR=LOGORVAR(Center) requests a model with different log odds ratios for each of the k centers, constant within center.

NESTK

k-nested log odds ratios. You must also specify the SUBCLUST=*variable* option to define subclusters within clusters. Within each cluster, PROC GENMOD computes a log odds ratio parameter for pairs having the same value of *variable* for both members of the pair and one log odds ratio parameter for each unique combination of different values of *variable*.

NEST1

1-nested log odds ratios. You must also specify the SUBCLUST=*variable* option to define subclusters within clusters. There are two log odds ratio parameters for this model. Pairs having the same value of *variable* correspond to one parameter; pairs having different values of *variable* correspond to the other parameter. For example, if clusters are hospitals and subclusters are wards within hospitals, then patients within the same ward have one log odds ratio parameter, and patients from different wards have the other parameter.

ZFULL

specifies the full *z*-matrix. You must also specify a SAS data set containing the *z*-matrix with the ZDATA=*data-set-name* option. Each observation in the data set corresponds to one row of the *z*-matrix. You must specify the ZDATA data set as if all clusters are complete, that is, as if all clusters are the same size and there are no missing observations. The ZDATA data set has $K[n_{max}(n_{max}-1)/2]$ observations, where K is the number of clusters and n_{max} is the maximum cluster size. If the members of cluster i are ordered as $1, 2, \cdots, n$, then the rows of the *z*-matrix must be specified for pairs in the order $(1,2), (1,3), \cdots, (1,n), (2,3), \cdots, (2,n), \cdots, (n-1,n)$. The variables specified in the REPEATED statement for the SUBJECT effect must also be present in the ZDATA= data set to identify clusters. You must specify variables in the data set that define the columns of the *z*-matrix by the ZROW=*variable-list* option. If there are q columns, (q variables in *variable-list*), then there are q log odds ratio parameters. You can optionally specify variables indicating the cluster pairs corresponding to each row of the *z*-matrix with the YPAIR=(*variable1*, *variable2*) option. If you specify this option, the data from the ZDATA data set is sorted within each cluster by *variable1* and *variable2*. See Example 31.6 for an example of specifying a full *z*-matrix.

ZREP

replicated *z*-matrix. You specify *z*-matrix data exactly as you do for the ZFULL case, except that you specify only one complete cluster. The *z*-matrix for the one cluster is replicated for each cluster. The number of observations in the ZDATA data set is $\frac{n_{max}(n_{max}-1)}{2}$, where n_{max} is the size of a complete cluster (a cluster with no missing observations). See Example 31.6 for an example of specifying a replicated *z*-matrix.

ZREP(matrix) direct input of the replicated *z*-matrix. You specify the *z*-matrix for one cluster with the syntax LOGOR=ZREP ($(y_1 \ \ y_2) z_1 \ \ z_2 \cdots z_q, \cdots$), where y_1 and y_2 are numbers representing a pair of observations and the values $z_1, z_2, \cdots, z_q$ make up the corresponding row of the *z*-matrix. The number of rows specified is $\frac{n_{max}(n_{max}-1)}{2}$, where n_{max} is the size of a complete cluster (a cluster with no missing observations). For example,

```
LOGOR =  ZREP((1 2) 1 0,
              (1 3) 1 0,
              (1 4) 1 0,
              (2 3) 1 1,
              (2 4) 1 1,
              (3 4) 1 1)
```

specifies the $\frac{4 \times 3}{2} = 6$ rows of the *z*-matrix for a cluster of size 4 with $q = 2$ log odds ratio parameters. The log odds ratio for pairs (1 2), (1 3), (1 4) is α_1, and the log odds ratio for pairs (2 3), (2 4), (3 4) is $\alpha_1 + \alpha_2$.

Generalized Score Statistics

Boos (1992) and Rotnitzky and Jewell (1990) describe score tests applicable to testing $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ in GEEs, where $\mathbf{L}$ is a user-specified $r \times p$ contrast matrix or a contrast for a Type 3 test of hypothesis.

Let $\tilde{\boldsymbol{\beta}}$ be the regression parameters resulting from solving the GEE under the restricted model $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, and let $\mathbf{S}(\tilde{\boldsymbol{\beta}})$ be the generalized estimating equation values at $\tilde{\boldsymbol{\beta}}$.

The generalized score statistic is

$$T = \mathbf{S}(\tilde{\boldsymbol{\beta}})' \boldsymbol{\Sigma}_m \mathbf{L}' (\mathbf{L} \boldsymbol{\Sigma}_e \mathbf{L}')^{-1} \mathbf{L} \boldsymbol{\Sigma}_m \mathbf{S}(\tilde{\boldsymbol{\beta}})$$

where $\boldsymbol{\Sigma}_m$ is the model-based covariance estimate and $\boldsymbol{\Sigma}_e$ is the empirical covariance estimate. The *p*-values for T are computed based on the chi-square distribution with *r* degrees of freedom.

Assessment of Models Based on Aggregates of Residuals (Experimental)

Lin, Wei, and Ying (2002) present graphical and numerical methods for model assessment based on the cumulative sums of residuals over certain coordinates (e.g., covariates or linear predictors) or some related aggregates of residuals. The distributions of these stochastic processes under the assumed model can be approximated by the distributions of certain zero-mean Gaussian processes whose realizations can be generated by simulation. Each observed residual pattern can then be compared, both graphically and numerically, with a number of realizations from the null distribution. Such comparisons enable you to assess objectively whether the observed residual pattern reflects anything beyond random fluctuation. These procedures are useful in determining appropriate functional forms of covariates and link function. You use the ASSESS|ASSESSMENT statement to perform this kind of model-checking with cumulative sums of residuals, moving sums of residuals, or LOWESS smoothed residuals. See Example 31.8 and Example 31.9 for examples of model assessment.

Let the model for the mean be

$$g(\mu_i) = \mathbf{x}_i{}'\boldsymbol{\beta}$$

where μ_i is the mean of the response y_i and $\mathbf{x}_i$ is the vector of covariates for the ith observation. Denote the raw residual resulting from fitting the model as

$$e_i = y_i - \hat{\mu}_i$$

and let x_{ij} be the value of the jth covariate in the model for observation i. Then to check the functional form of the jth covariate, consider the cumulative sum of residuals with respect to x_{ij}

$$W_j(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} I(x_{ij} \le x) e_i$$

where $I()$ is the indicator function. For any x, $W_j(x)$ is the sum of the residuals with values of x_j less than or equal to x.

Denote the score, or gradient vector by

$$U(\boldsymbol{\beta}) = \sum_{i=1}^{n} h(\mathbf{x}_i{}'\boldsymbol{\beta})\mathbf{x}_i(y_i - \nu(\mathbf{x}_i{}'\boldsymbol{\beta}))$$

where $\nu(r) = g^{-1}(r)$, and

$$h(r) = \frac{1}{g\prime(\nu(r))V(\nu(r))}$$

Let J be the Fisher information matrix

$$J(\boldsymbol{\beta}) = -\frac{\partial U(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'}$$

Define

$$\hat{W}_j(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} [I(x_{ij} \le x) + \eta\prime(x; \hat{\boldsymbol{\beta}}) J^{-1}(\hat{\boldsymbol{\beta}})\mathbf{x}_i h(\mathbf{x}_i{}'\hat{\boldsymbol{\beta}})] e_i Z_i$$

where

$$\eta(x;\boldsymbol{\beta}) = -\sum_{i=1}^{n} I(x_{ij} \le x) \frac{\partial \nu(\mathbf{x}_i{}'\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

and Z_i are independent $N(0,1)$ random variables. Then the conditional distribution of $\hat{W}_j(x)$, given $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, under the null hypothesis H_0 that the model for the mean is correct, is the same asymptotically as $n \rightarrow \infty$ as the unconditional distribution of $W_j(x)$ (Lin, Wei, and Ying, 2002).

You can approximate realizations from the null hypothesis distribution of $W_j(x)$ by repeatedly generating normal samples $Z_i, i = 1, \ldots, n$ while holding $(y_i, \mathbf{x}_i), i = 1, \ldots, n$ at their observed values and computing $\hat{W}_j(x)$ for each sample.

You can assess the functional form of covariate j by plotting a few realizations of $\hat{W}_j(x)$ on the same plot as the observed $W_j(x)$ and visually comparing to see how typical the observed $W_j(x)$ is of the null distribution samples.

You can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j be the observed value of $S_j = \sup_x |W_j(x)|$. The p-value $\Pr[S_j \ge s_j]$ is approximated by $\Pr[\hat{S}_j \ge s_j]$, where $\hat{S}_j = \sup_x |\hat{W}_j(x)|$. $\Pr[\hat{S}_j \ge s_j]$ is estimated by generating realizations of $\hat{W}_j(.)$ (1,000 is the default number of realizations).

You can check the link function instead of the jth covariate by using values of the linear predictor $\mathbf{x}_i{}'\hat{\boldsymbol{\beta}}$ in place of values of the jth covariate x_{ij}. The graphical and numerical methods described above are then sensitive to inadequacies in the link function.

An alternative aggregate of residuals is the moving sum statistic

$$W_j(x,b) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} I(x - b \le x_{ij} \le x) e_i$$

If you specify the keyword WINDOW(b), then the moving sum statistic with window size b is used instead of the cumulative sum of residuals, with $I(x - b \le x_{ij} \le x)$ replacing $I(x_{ij} \le x)$ above.

If you specify the keyword LOWESS(f), LOWESS smoothed residuals are used in the formulas above, where f is the fraction of the data to be used at a given point. If f is not specified, $f = \frac{1}{3}$ is used. Define, for data $(Y_i, X_i), i = 1, \ldots, n$, r as the nearest integer to nf and h as the rth smallest among $|X_i - x|, i = 1, \ldots, n$. Let

$$K_i(x) = K(\frac{X_i - x}{h})$$

where

$$K(t) = \frac{70}{81}(1 - |t|^3)^3 I(-1 \le t \le 1)$$

Define

$$w_i(x) = K_i(x)[S_2(x) - (X_i - x)S_1(x)]$$

where

$$S_1(x) = \sum_{i=1}^{n} K_i(x)(X_i - x)$$

$$S_2(x) = \sum_{i=1}^{n} K_i(x)(X_i - x)^2$$

Then the LOWESS estimate of Y at x is defined by

$$\hat{Y}(x) = \sum_{i=1}^{n} \frac{w_i(x)}{\sum_{i=1}^{n} w_i(x)} Y_i$$

LOWESS smoothed residuals for checking the functional form of the jth covariate are defined by replacing Y_i with e_i and X_i with x_{ij}. To implement the graphical and numerical assessment methods, $I(x_{ij} \leq x)$ is replaced with $\frac{w_i(x)}{\sum_{i=1}^{n} w_i(x)}$ in the formulas for $W_j(x)$ and $\hat{W}_j(x)$.

You can perform the model checking described above for marginal models for dependent responses fit by generalized estimating equations (GEEs). Let y_{ik} denote the kth measurement on the ith cluster, $i = 1, \ldots, K$, $k = 1, \ldots, n_i$, and $\mathbf{x}_{ik}$ the corresponding vector of covariates. The marginal mean of the response $\mu_{ik} = \mathrm{E}(y_{ik})$ is assumed to depend on the covariate vector by

$$g(\mu_{ik}) = \mathbf{x}_{ik}'\boldsymbol{\beta}$$

where g is the link function.

Define the vector of residuals for the ith cluster as

$$\mathbf{e}_i = (e_{i1}, \ldots, e_{in_i})' = (y_{i1} - \hat{\mu}_{i1}, \ldots, y_{in_i} - \hat{\mu}_{in_i})'$$

You use the following extension of $W_j(x)$ defined above to check the functional form of the jth covariate:

$$W_j(x) = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} \sum_{k=1}^{n_i} I(x_{ikj} \leq x) e_{ik}$$

where x_{ikj} is the jth component of $\mathbf{x}_{ik}$.

The null distribution of $W_j(x)$ can be approximated by the conditional distribution of

$$\hat{W}_j(x) = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} \left\{ \sum_{k=1}^{n_i} I(x_{ikj} \le x) e_{ik} + \eta'(x, \hat{\boldsymbol{\beta}}) \mathbf{I}_0^{-1} \hat{\mathbf{D}}_i' \hat{\mathbf{V}}_i^{-1} \mathbf{e}_i \right\} Z_i$$

where $\hat{\mathbf{D}}_i$ and $\hat{\mathbf{V}}_i$ are defined as in the section "Generalized Estimating Equations" on page 1672 with the unknown parameters replaced by their estimated values,

$$\eta(x, \boldsymbol{\beta}) = -\sum_{i=1}^{K} \sum_{k=1}^{n_i} I(x_{ikj} \le x) \frac{\partial \mu_{ik}}{\partial \boldsymbol{\beta}}$$

$$\mathbf{I}_0 = \sum_{i=1}^{K} \hat{\mathbf{D}}_i' \hat{\mathbf{V}}_i^{-1} \hat{\mathbf{D}}_i$$

and $Z_i, i = 1, \ldots, K$, are independent $N(0, 1)$ random variables. You replace x_{ikj} with the linear predictor $\mathbf{x}_{ik}'\hat{\boldsymbol{\beta}}$ in the preceding formulas to check the link function.

Displayed Output

The following output is produced by the GENMOD procedure. Note that some of the tables are optional and appear only in conjunction with the REPEATED statement and its options or with options in the MODEL statement. For details, see the section "ODS Table Names" on page 1693.

Model Information

PROC GENMOD displays the following model information:

- data set name
- response distribution
- link function
- response variable name
- offset variable name
- frequency variable name
- scale weight variable name
- number of observations used
- number of events if events/trials format is used for response
- number of trials if events/trials format is used for response
- sum of frequency weights
- number of missing values in data set
- number of invalid observations (for example, negative or 0 response values with gamma distribution or number of observations with events greater than trials with binomial distribution)

Class Level Information

If you use classification variables in the model, PROC GENMOD displays the levels of classification variables specified in the CLASS statement and in the MODEL statement. The levels are displayed in the same sorted order used to generate columns in the design matrix.

Response Profile

If you specify an ordinal model for the multinomial distribution, a table titled "Response Profile" is displayed containing the ordered values of the response variable and the number of occurrences of the values used in the model.

Iteration History for Parameter Estimates

If you specify the ITPRINT model option, PROC GENMOD displays a table containing the following for each iteration in the Newton-Raphson procedure for model fitting:

- iteration number
- ridge value
- log likelihood
- values of all parameters in the model

Criteria for Assessing Goodness of Fit

PROC GENMOD displays the following criteria for assessing goodness of fit:

- degrees of freedom for deviance and Pearson's chi-square, equal to the number of observations minus the number of regression parameters estimated
- deviance
- deviance divided by degrees of freedom
- scaled deviance
- scaled deviance divided by degrees of freedom
- Pearson's chi-square
- Pearson's chi-square divided by degrees of freedom
- scaled Pearson's chi-square
- scaled Pearson's chi-square divided by degrees of freedom
- log likelihood

Last Evaluation of the Gradient

If you specify the model option ITPRINT, the GENMOD procedure displays the last evaluation of the gradient vector.

Last Evaluation of the Hessian

If you specify the model option ITPRINT, the GENMOD procedure displays the last evaluation of the Hessian matrix.

Analysis of (Initial) Parameter Estimates

The "Analysis of (Initial) Parameter Estimates" table contains the results from fitting a generalized linear model to the data. If you specify the REPEATED statement, these GLM parameter estimates are used as initial values for the GEE solution. For each parameter in the model, PROC GENMOD displays the following:

- the parameter name
  - the variable name for continuous regression variables
  - the variable name and level for classification variables and interactions involving classification variables
  - SCALE for the scale variable related to the dispersion parameter
- degrees of freedom for the parameter
- estimate value
- standard error
- Wald chi-square value
- p-value based on the chi-square distribution
- confidence limits (Wald or profile likelihood) for parameters

Estimated Covariance Matrix

If you specify the model option COVB, the GENMOD procedure displays the estimated covariance matrix, defined as the inverse of the information matrix at the final iteration. This is based on the expected information matrix if the EXPECTED option is specified in the MODEL statement. Otherwise, it is based on the Hessian matrix used at the final iteration. This is, by default, the observed Hessian unless altered by the SCORING option in the MODEL statement.

Estimated Correlation Matrix

If you specify the CORRB model option, PROC GENMOD displays the estimated correlation matrix. This is based on the expected information matrix if the EXPECTED option is specified in the MODEL statement. Otherwise, it is based on the Hessian matrix used at the final iteration. This is, by default, the observed Hessian unless altered by the SCORING option in the MODEL statement.

Iteration History for LR Confidence Intervals

If you specify the ITPRINT and LRCI model options, PROC GENMOD displays an iteration history table for profile likelihood-based confidence intervals. For each parameter in the model, PROC GENMOD displays the following:

- parameter identification number
- iteration number
- log likelihood value
- parameter values

Likelihood Ratio-Based Confidence Intervals for Parameters

If you specify the LRCI and the ITPRINT options in the MODEL statement, a table is displayed summarizing profile likelihood-based confidence intervals for all parameters. The table contains the following for each parameter in the model:

- confidence coefficient
- parameter identification number
- lower and upper endpoints of confidence intervals for the parameter
- values of all other parameters at the solution

LR Statistics for Type 1 Analysis

If you specify the TYPE1 model option, a table containing the following is displayed for each effect in the model:

- name of effect
- deviance for the model including the effect and all previous effects
- degrees of freedom for the effect
- likelihood ratio statistic for testing the significance of the effect
- p-value computed from the chi-square distribution with

  effect degrees of freedom

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option in the MODEL statement, columns containing the following are displayed:

- name of effect
- deviance for the model including the effect and all previous effects
- numerator degrees of freedom
- denominator degrees of freedom
- chi-square statistic for testing the significance of the effect
- p-value computed from the chi-square distribution with numerator degrees of freedom
- F statistic for testing the significance of the effect
- p-value based on the F distribution

Iteration History for Type 3 Contrasts

If you specify the model options ITPRINT and TYPE3, an iteration history table is displayed for fitting the model with Type 3 contrast constraints for each effect. The table contains the following:

- effect name
- iteration number
- ridge value
- log likelihood
- values of all parameters

LR Statistics for Type 3 Analysis

If you specify the TYPE3 model option, a table containing the following is displayed for each effect in the model:

- name of the effect
- likelihood ratio statistic for testing the significance of the effect
- degrees of freedom for effect
- p-value computed from the chi-square distribution

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option in the MODEL statement, columns containing the following are displayed:

- name of the effect
- likelihood ratio statistic for testing the significance of the effect
- F statistic for testing the significance of the effect
- numerator degrees of freedom
- denominator degrees of freedom
- p-value based on the F distribution
- p-value computed from the chi-square distribution with numerator degrees of freedom

Wald Statistics for Type 3 Analysis

If you specify the TYPE3 and WALD model options, a table containing the following is displayed for each effect in the model:

- name of effect
- degrees of freedom for effect
- Wald statistic for testing the significance of the effect
- p-value computed from the chi-square distribution

Parameter Information

If you specify the ITPRINT, COVB, CORRB, WALDCI, or LRCI option in the MODEL statement, or if you specify a CONTRAST statement, a table is displayed that identifies parameters with numbers, rather than names, for use in tables and matrices where a compact identifier for parameters is helpful. For each parameter, the table contains the following:

- a number that identifies the parameter
- the parameter name, including level information for effects containing classification variables

Observation Statistics

If you specify the OBSTATS option in the MODEL statement, PROC GENMOD displays a table containing miscellaneous statistics. For each observation in the input data set, the following are displayed:

- the value of the response variable, denoted by the variable name
- the predicted value of the mean, denoted by PRED
- the value of the linear predictor, denoted by XBETA. The value of an OFFSET variable is added to the linear predictor.
- the estimated standard error of the linear predictor, denoted by STD
- the value of the negative of the weight in the Hessian matrix at the final iteration, denoted by HESSWGT. This is the expected weight if the EXPECTED option is specified in the MODEL statement. Otherwise, it is the weight used in the final iteration. That is, it is the observed weight unless the SCORING= option has been specified.
- approximate lower and upper endpoints for a confidence interval for the predicted value of the mean, denoted by LOWER and UPPER
- raw residual, denoted by RESRAW
- Pearson residual, denoted by RESCHI
- deviance residual, denoted by RESDEV
- standardized Pearson residual, denoted by STDRESCHI
- standardized deviance residual, denoted by STDRESDEV
- likelihood residual, denoted by RESLIK

ESTIMATE Statement Results

If you specify a REPEATED statement, the ESTIMATE statement results apply to the specified GEE model. Otherwise, they apply to the specified generalized linear model.

The following are displayed for each ESTIMATE statement:

- contrast label
- estimated value of the contrast
- standard error of the estimate
- significance level α
- $(1 - \alpha) \times 100\%$ confidence intervals for contrast
- Wald chi-square statistic for the contrast
- p-value computed from the chi-square distribution

If you specify the EXP option, an additional row is displayed with statistics for the exponentiated value of the contrast.

CONTRAST Coefficients

If you specify the CONTRAST or ESTIMATE statement and you specify the E option, a table titled "Coefficients For Contrast *label*" is displayed, where *label* is the label specified in the CONTRAST statement. The table contains the following:

- the contrast label
- the rows of the contrast matrix

Iteration History for Contrasts

If you specify the ITPRINT option, an iteration history table is displayed for fitting the model with contrast constraints for each effect. The table contains the following for each contrast defined in a CONTRAST statement:

- contrast label
- iteration number
- ridge value
- log likelihood
- values of all parameters

CONTRAST Statement Results

If you specify a REPEATED statement, the CONTRAST statement results apply to the specified GEE model. Otherwise, they apply to the specified generalized linear model.

The following are displayed for each CONTRAST statement:

- contrast label
- degrees of freedom for the contrast
- likelihood ratio, score, or Wald statistic for testing the significance of the contrast. Score statistics are used in GEE models, likelihood ratio statistics are used in generalized linear models, and Wald statistics are used in both.

- p-value computed from the chi-square distribution
- the type of statistic computed for this contrast: Wald, LR, or score

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option for generalized linear models, columns containing the following are displayed:

- contrast label
- likelihood ratio statistic for testing the significance of the contrast
- F statistic for testing the significance of the contrast
- numerator degrees of freedom
- denominator degrees of freedom
- p-value based on the F distribution
- p-value computed from the chi-square distribution with numerator degrees of freedom

LSMEANS Coefficients

If you specify the LSMEANS statement and you specify the E option, a table titled "Coefficients for *effect* Least Squares Means" is displayed, where *effect* is the effect specified in the LSMEANS statement. The table contains the following:

- the effect names
- the rows of least squares means coefficients

Least Squares Means

If you specify the LSMEANS statement a table titled "Least Squares Means" is displayed. The table contains the following:

- the effect names
- for each level of each effect,
  - the least squares mean estimate
  - standard error
  - chi-square value
  - p-value computed from the chi-square distribution

If you specify the DIFF option, a table titled "Differences of Least Squares Means" is displayed containing corresponding statistics for the differences between the least squares means for the levels of each effect.

GEE Model Information

If you specify the REPEATED statement, the following are displayed:

- correlation structure of the working correlation matrix or the log odds ratio structure
- within-subject effect
- subject effect
- number of clusters
- correlation matrix dimension
- minimum and maximum cluster size

Log Odds Ratio Parameter Information

If you specify the REPEATED statement and specify a log odds ratio model for binary data with the LOGOR= option, then a table is displayed showing the correspondence between data pairs and log odds ratio model parameters.

Iteration History for GEE Parameter Estimates

If you specify the REPEATED statement and the MODEL statement option ITPRINT, an iteration history table for GEE parameter estimates is displayed. The table contains the following:

- parameter identification number
- iteration number
- values of all parameters

Last Evaluation of the Generalized Gradient and Hessian

If you specify the REPEATED statement and select ITPRINT as a model option, PROC GENMOD displays the last generalized gradient and Hessian matrix in the GEE iterative parameter estimation process.

GEE Parameter Estimate Covariance Matrices

If you specify the REPEATED statement and the COVB option, PROC GENMOD displays both model-based and empirical parameter estimate covariance matrices.

GEE Parameter Estimate Correlation Matrices

If you specify the REPEATED statement and the CORRB option, PROC GENMOD displays both model-based and empirical parameter estimate covariance matrices.

GEE Working Correlation Matrix

If you specify the REPEATED statement and the CORRW option, PROC GENMOD displays the exchangeable working correlation matrix.

Analysis of GEE Parameter Estimates

If you specify the REPEATED statement, PROC GENMOD uses empirical standard error estimates to compute and display the following for each parameter in the model:

- the parameter name
  - the variable name for continuous regression variables
  - the variable name and level for classification variables and interactions involving classification variables
  - "Scale" for the scale variable related to the dispersion parameter
- parameter estimate
- standard error
- 95% confidence interval
- Z score and p-value

If you specify the MODELSE option in the REPEATED statement, a table based on model-based standard errors is also produced.

GEE Observation Statistics

If you specify the OBSTATS option in the REPEATED statement, PROC GENMOD displays a table containing miscellaneous statistics. For each observation in the input data set, the following are displayed:

- the value of the response variable and all other variables in the model, denoted by the variable names
- the predicted value of the mean, denoted by PRED
- the value of the linear predictor, denoted by XBETA
- the standard error of the linear predictor, denoted by STD
- confidence limits for the predicted values, denoted by LOWER and UPPER
- raw residual, denoted by RESRAW
- Pearson residual, denoted by RESCHI

ODS Table Names

PROC GENMOD assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 31.3. ODS Tables Produced in PROC GENMOD

ODS Table Name	Description	Statement	Option
AssessmentSummary (experimental)	Model assessment summary	ASSESS	default
ClassLevels	Class variable levels	CLASS	default
Contrasts	Tests of contrasts	CONTRAST	default
ContrastCoef	Contrast coefficients	CONTRAST	E
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
Estimates	Estimates of contrasts	ESTIMATE	default
EstimateCoef	Contrast coefficients	ESTIMATE	E
GEEEmpPEst	GEE parameter estimates with empirical standard errors	REPEATED	default
GEELogORInfo	GEE log odds ratio model information	REPEATED	LOGOR=
GEEModInfo	GEE model information	REPEATED	default
GEEModPEst	GEE parameter estimates with model-based standard errors	REPEATED	MODELSE
GEENCorr	GEE model-based correlation matrix	REPEATED	MCORRB
GEENCov	GEE model-based covariance matrix	REPEATED	MCOVB
GEERCorr	GEE empirical correlation matrix	REPEATED	ECORRB
GEERCov	GEE empirical covariance matrix	REPEATED	ECOVB
GEEWCorr	GEE working correlation matrix	REPEATED	CORRW
IterContrasts	Iteration history for contrasts	MODEL CONTRAST	ITPRINT
IterLRCI	Iteration history for likelihood ratio confidence intervals	MODEL	LRCI ITPRINT
IterParms	Iteration history for parameter estimates	MODEL	ITPRINT
IterParmsGEE	Iteration history for GEE parameter estimates	MODEL REPEATED	ITPRINT
IterType3	Iteration history for Type 3 statistics	MODEL	TYPE3 ITPRINT
LRCI	Likelihood ratio confidence intervals	MODEL	LRCI ITPRINT

Table 31.3. (continued)

ODS Table Name	Description	Statement	Option
LSMeanCoef	Coefficients for least squares means	LSMEANS	E
LSMeanDiffs	Least squares means differences	LSMEANS	DIFF
LSMeans	Least squares means	LSMEANS	default
LagrangeStatistics	Lagrange statistics	MODEL	NOINT | NOSCALE
LastGEEGrad	Last evaluation of the generalized gradient and Hessian	MODEL REPEATED	ITPRINT
LastGradHess	Last evaluation of the gradient and Hessian	MODEL	ITPRINT
LinDep	Linearly dependent rows of contrasts	CONTRAST*	default
ModelInfo	Model information	MODEL	default
Modelfit	Goodness-of-fit statistics	MODEL	default
NObs	Number of observations summary	default	
NonEst	Nonestimable rows of contrasts	CONTRAST*	default
ObStats	Observation-wise statistics	MODEL	OBSTATS | CL | PREDICTED | RESIDUALS | XVARS
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL*	default
ResponseProfiles	Frequency counts for multinomial models	MODEL	DIST=MULTINOMIAL
Type1	Type 1 tests	MODEL	TYPE1
Type3	Type 3 tests	MODEL	TYPE3

*Depends on data.

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the GENMOD procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the ASSESS statement and options. For more information on the ASSESS statement, see the "ASSESS Statement" section on page 1627. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

ODS Graph Names

PROC GENMOD assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 31.4.

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 31.4.

Table 31.4. ODS Graphics Produced by PROC GENMOD

ODS Graph Name	Description	Statement	Option
CumResidPanel	Panel plot of aggregates of residuals	ASSESS	CRPANEL
CumulativeResiduals	Model assessment based on aggregates of residuals	ASSESS	default

Examples

The following examples illustrate some of the capabilities of the GENMOD procedure. These are not intended to represent definitive analyses of the data sets presented here. You should refer to the texts cited in the "References" section on page 1728 for guidance on complete analysis of data using generalized linear models.

Example 31.1. Logistic Regression

In an experiment comparing the effects of five different drugs, each drug is tested on a number of different subjects. The outcome of each experiment is the presence or absence of a positive response in a subject. The following artificial data represent the number of responses r in the n subjects for the five different drugs, labeled A through E. The response is measured for different levels of a continuous covariate x for each drug. The drug type and the continuous covariate x are explanatory variables in this experiment. The number of responses r is modeled as a binomial random variable for each combination of the explanatory variable values, with the binomial number of trials parameter equal to the number of subjects n and the binomial probability equal to the probability of a response.

The following DATA step creates the data set.

```
data drug;
   input drug$ x r n @@;
   datalines;
   A  .1   1  10   A  .23  2  12   A  .67  1   9
   B  .2   3  13   B  .3   4  15   B  .45  5  16   B  .78  5  13
   C  .04  0  10   C  .15  0  11   C  .56  1  12   C  .7   2  12
   D  .34  5  10   D  .6   5   9   D  .7   8  10
   E  .2  12  20   E  .34 15  20   E  .56 13  15   E  .8  17  20
   ;
run;
```

A logistic regression for these data is a generalized linear model with response equal to the binomial proportion r/n. The probability distribution is binomial, and the link function is logit. For these data, drug and x are explanatory variables. The probit and the complementary log-log link functions are also appropriate for binomial data.

PROC GENMOD performs a logistic regression on the data in the following SAS statements:

```
proc genmod data=drug;
   class drug;
   model r/n = x drug / dist = bin
                        link = logit
                        lrci;
run;
```

Since these data are binomial, you use the events/trials syntax to specify the response in the MODEL statement. Profile likelihood confidence intervals for the regression parameters are computed using the LRCI option.

General model and data information is produced in Output 31.1.1.

Output 31.1.1. Model Information

```
                              The GENMOD Procedure

                               Model Information

                    Data Set                              WORK.DRUG
                    Distribution                           Binomial
                    Link Function                             Logit
                    Response Variable (Events)                    r
                    Response Variable (Trials)                    n
```

The five levels of the CLASS variable DRUG are displayed in Output 31.1.2.

Output 31.1.2. Class Variable Levels

```
                            Class Level Information

                        Class      Levels    Values

                        drug            5    A B C D E
```

In the “Criteria For Assessing Goodness Of Fit” table displayed in Output 31.1.3, the value of the deviance divided by its degrees of freedom is less than 1. A p-value is not computed for the deviance; however, a deviance that is approximately equal to its degrees of freedom is a possible indication of a good model fit. Asymptotic distribution theory applies to binomial data as the number of binomial trials parameter n becomes large for each combination of explanatory variables. McCullagh and

Nelder (1989) caution against the use of the deviance alone to assess model fit. The model fit for each observation should be assessed by examination of residuals. The OBSTATS option in the MODEL statement produces a table of residuals and other useful statistics for each observation.

Output 31.1.3. Goodness of Fit Criteria

```
                     Criteria For Assessing Goodness Of Fit

          Criterion                 DF           Value        Value/DF

          Deviance                  12          5.2751          0.4396
          Scaled Deviance           12          5.2751          0.4396
          Pearson Chi-Square        12          4.5133          0.3761
          Scaled Pearson X2         12          4.5133          0.3761
          Log Likelihood                     -114.7732
```

In the “Analysis Of Parameter Estimates” table displayed in Output 31.1.4, chi-square values for the explanatory variables indicate that the parameter values other than the intercept term are all significant. The scale parameter is set to 1 for the binomial distribution. When you perform an overdispersion analysis, the value of the overdispersion parameter is indicated here. See the the section “Overdispersion” on page 1659 for a discussion of overdispersion.

Output 31.1.4. Parameter Estimates

```
                          Analysis Of Parameter Estimates

                                            Likelihood Ratio
                                 Standard    95% Confidence       Chi-
 Parameter      DF  Estimate      Error          Limits          Square   Pr > ChiSq

 Intercept       1    0.2792     0.4196   -0.5336    1.1190       0.44       0.5057
 x               1    1.9794     0.7660    0.5038    3.5206       6.68       0.0098
 drug       A    1   -2.8955     0.6092   -4.2280   -1.7909      22.59       <.0001
 drug       B    1   -2.0162     0.4052   -2.8375   -1.2435      24.76       <.0001
 drug       C    1   -3.7952     0.6655   -5.3111   -2.6261      32.53       <.0001
 drug       D    1   -0.8548     0.4838   -1.8072    0.1028       3.12       0.0773
 drug       E    0    0.0000     0.0000    0.0000    0.0000        .            .
 Scale           0    1.0000     0.0000    1.0000    1.0000

NOTE: The scale parameter was held fixed.
```

The preceding table contains the profile likelihood confidence intervals for the explanatory variable parameters requested with the LRCI option. Wald confidence intervals are displayed by default. Profile likelihood confidence intervals are considered to be more accurate than Wald intervals (refer to Aitkin et al. 1989), especially with small sample sizes. You can specify the confidence coefficient with the ALPHA= option in the MODEL statement. The default value of 0.05, corresponding to 95% confidence limits, is used here. See the section “Confidence Intervals for Parameters” on page 1666 for a discussion of profile likelihood confidence intervals.

Example 31.2. Normal Regression, Log Link

Consider the following data, where x is an explanatory variable, and y is the response variable. It appears that y varies nonlinearly with x and that the variance is approximately constant. A normal distribution with a log link function is chosen to model these data; that is, $\log(\mu_i) = \mathbf{x}_i{}'\boldsymbol{\beta}$ so that $\mu_i = \exp(\mathbf{x}_i{}'\boldsymbol{\beta})$.

```
data nor;
   input x y;
   datalines;
   0 5
   0 7
   0 9
   1 7
   1 10
   1 8
   2 11
   2 9
   3 16
   3 13
   3 14
   4 25
   4 24
   5 34
   5 32
   5 30
   ;
run;
```

The following SAS statements produce the analysis with the normal distribution and log link:

```
proc genmod data=nor;
   model y = x / dist = normal
                 link = log;

   output out       = Residuals
          pred      = Pred
          resraw    = Resraw
          reschi    = Reschi
          resdev    = Resdev
          stdreschi = Stdreschi
          stdresdev = Stdresdev
          reslik    = Reslik;
run;
```

The OUTPUT statement is specified to produce a data set that contains predicted values and residuals for each observation. This data set can be useful for further analysis, such as residual plotting.

The output from these statements is displayed in Output 31.2.1.

Output 31.2.1. Log Linked Normal Regression

```
                              The GENMOD Procedure

                               Model Information

                         Data Set              WORK.NOR
                         Distribution            Normal
                         Link Function              Log
                         Dependent Variable           y


                     Criteria For Assessing Goodness Of Fit

          Criterion                 DF           Value        Value/DF

          Deviance                  14         52.3000          3.7357
          Scaled Deviance           14         16.0000          1.1429
          Pearson Chi-Square        14         52.3000          3.7357
          Scaled Pearson X2         14         16.0000          1.1429
          Log Likelihood                      -32.1783


                        Analysis Of Parameter Estimates

                             Standard       Wald 95%          Chi-
   Parameter  DF  Estimate      Error   Confidence Limits   Square  Pr > ChiSq

   Intercept   1    1.7214     0.0894    1.5461    1.8966   370.76      <.0001
   x           1    0.3496     0.0206    0.3091    0.3901   286.64      <.0001
   Scale       1    1.8080     0.3196    1.2786    2.5566

NOTE: The scale parameter was estimated by maximum likelihood.
```

The PROC GENMOD scale parameter, in the case of the normal distribution, is the standard deviation. By default, the scale parameter is estimated by maximum likelihood. You can specify a fixed standard deviation by using the NOSCALE and SCALE= options in the MODEL statement.

Output 31.2.2. Data Set of Predicted Values and Residuals

```
Obs x  y    Pred    Reschi    Resdev    Resraw  Stdreschi Stdresdev   Reslik

  1 0  5  5.5921 -0.59212 -0.59212 -0.59212  -0.34036  -0.34036 -0.34036
  2 0  7  5.5921  1.40788  1.40788  1.40788   0.80928   0.80928  0.80928
  3 0  9  5.5921  3.40788  3.40788  3.40788   1.95892   1.95892  1.95892
  4 1  7  7.9324 -0.93243 -0.93243 -0.93243  -0.54093  -0.54093 -0.54093
  5 1 10  7.9324  2.06757  2.06757  2.06757   1.19947   1.19947  1.19947
  6 1  8  7.9324  0.06757  0.06757  0.06757   0.03920   0.03920  0.03920
  7 2 11 11.2522 -0.25217 -0.25217 -0.25217  -0.14686  -0.14686 -0.14686
  8 2  9 11.2522 -2.25217 -2.25217 -2.25217  -1.31166  -1.31166 -1.31166
  9 3 16 15.9612  0.03878  0.03878  0.03878   0.02249   0.02249  0.02249
 10 3 13 15.9612 -2.96122 -2.96122 -2.96122  -1.71738  -1.71738 -1.71738
 11 3 14 15.9612 -1.96122 -1.96122 -1.96122  -1.13743  -1.13743 -1.13743
 12 4 25 22.6410  2.35897  2.35897  2.35897   1.37252   1.37252  1.37252
 13 4 24 22.6410  1.35897  1.35897  1.35897   0.79069   0.79069  0.79069
 14 5 34 32.1163  1.88366  1.88366  1.88366   1.22914   1.22914  1.22914
 15 5 32 32.1163 -0.11634 -0.11634 -0.11634  -0.07592  -0.07592 -0.07592
 16 5 30 32.1163 -2.11634 -2.11634 -2.11634  -1.38098  -1.38098 -1.38098
```

The data set of predicted values and residuals (Output 31.2.2) is created by the OUTPUT statement. With this data set, you can construct residual plots using the GPLOT procedure to aid in assessing model fit. Note that raw, Pearson, and deviance residuals are equal in this example. This is a characteristic of the normal distribution and is not true in general for other distributions.

Example 31.3. Gamma Distribution Applied to Life Data

Life data are sometimes modeled with the gamma distribution. Although PROC GENMOD does not analyze censored data or provide other useful lifetime distributions such as the Weibull or lognormal, it can be used for modeling complete (uncensored) data with the gamma distribution, and it can provide a statistical test for the exponential distribution against other gamma distribution alternatives. Refer to Lawless (1982) or Nelson (1982) for applications of the gamma distribution to life data.

The following data represent failure times of machine parts, some of which are manufactured by manufacturer A and some by manufacturer B.

```
data A;
   input lifetime@@ ;
   mfg = ’A’;
   datalines;
   620  470  260  89   388  242
   103  100  39   460  284  1285
   218  393  106  158  152  477
   403  103  69   158  818  947
   399  1274 32   12   134  660
   548  381  203  871  193  531
   317  85   1410 250  41   1101
   32   421  32   343  376  1512
   1792 47   95   76   515  72
```

```
      1585 253  6    860  89   1055
      537  101  385  176  11   565
      164  16   1267 352  160  195
      1279 356  751  500  803  560
      151  24   689  1119 1733 2194
      763  555  14   45   776  1
      ;

   data B;
      input lifetime@@ ;
      mfg = 'B';
      datalines;
      1747 945  12   1453 14   150
      20   41   35   69   195  89
      1090 1868 294  96   618  44
      142  892  1307 310  230  30
      403  860  23   406  1054 1935
      561  348  130  13   230  250
      317  304  79   1793 536  12
      9    256  201  733  510  660
      122  27   273  1231 182  289
      667  761  1096 43   44   87
      405  998  1409 61   278  407
      113  25   940  28   848  41
      646  575  219  303  304  38
      195  1061 174  377  388  10
      246  323  198  234  39   308
      55   729  813  1216 1618 539
      6    1566 459  946  764  794
      35   181  147  116  141  19
      380  609  546
      ;

   data lifdat;
      set A B;
   run;
```

The following SAS statements use PROC GENMOD to compute Type 3 statistics to test for differences between the two manufacturers in machine part life. Type 3 statistics are identical to Type 1 statistics in this case, since there is only one effect in the model. The log link function is selected to ensure that the mean is positive.

```
proc genmod data = lifdat;
   class mfg;
   model lifetime = mfg / dist=gamma
                          link=log
                          type3;
run;
```

The output from these statements is displayed in Output 31.3.1.

Output 31.3.1. Gamma Model of Life Data

```
                             The GENMOD Procedure

                              Model Information

                       Data Set                WORK.LIFDAT
                       Distribution                  Gamma
                       Link Function                   Log
                       Dependent Variable         lifetime


                           Class Level Information

                          Class      Levels    Values

                          mfg             2    A B


                    Criteria For Assessing Goodness Of Fit

          Criterion                 DF           Value        Value/DF

          Deviance                 199        287.0591          1.4425
          Scaled Deviance          199        237.5335          1.1936
          Pearson Chi-Square       199        211.6870          1.0638
          Scaled Pearson X2        199        175.1652          0.8802
          Log Likelihood                    -1432.4177


                      Analysis Of Parameter Estimates

                                Standard       Wald 95%          Chi-
 Parameter     DF  Estimate       Error   Confidence Limits    Square  Pr > ChiSq

 Intercept      1    6.1302      0.1043    5.9257    6.3347   3451.61      <.0001
 mfg        A   1    0.0199      0.1559   -0.2857    0.3255      0.02      0.8985
 mfg        B   0    0.0000      0.0000    0.0000    0.0000       .           .
 Scale          1    0.8275      0.0714    0.6987    0.9800

NOTE: The scale parameter was estimated by maximum likelihood.


                       LR Statistics For Type 3 Analysis

                                          Chi-
                   Source           DF     Square    Pr > ChiSq

                   mfg               1       0.02        0.8985
```

The p-value of 0.8985 for the chi-square statistic in the Type 3 table indicates that there is no significant difference in the part life for the two manufacturers.

Using the following statements, you can refit the model without using the manufacturer as an effect. The LRCI option in the MODEL statement is specified to compute profile likelihood confidence intervals for the mean life and scale parameters.

```
proc genmod data = lifdat;
   model lifetime = / dist=gamma
                      link=log
                      lrci;
run;
```

Output 31.3.2. Refitting of the Gamma Model: Omitting the mfg Effect

```
                              The GENMOD Procedure

                         Analysis Of Parameter Estimates

                                           Likelihood Ratio
                                Standard    95% Confidence       Chi-
  Parameter  DF  Estimate       Error          Limits          Square  Pr > ChiSq

  Intercept   1    6.1391      0.0775     5.9904     6.2956  6268.10      <.0001
  Scale       1    0.8274      0.0714     0.6959     0.9762

NOTE: The scale parameter was estimated by maximum likelihood.
```

The intercept is the estimated log mean of the fitted gamma distribution, so that the mean life of the parts is

$$\mu = \exp(\text{INTERCEPT}) = \exp(6.1391) = 463.64$$

The SCALE parameter used in PROC GENMOD is the inverse of the gamma dispersion parameter, and it is sometimes called the gamma *index parameter*. See the "Response Probability Distributions" section on page 1650 for the definition of the gamma probability density function. A value of 1 for the index parameter corresponds to the exponential distribution . The estimated value of the scale parameter is 0.8274. The 95% profile likelihood confidence interval for the scale parameter is (0.6959, 0.9762), which does not contain 1. The hypothesis of an exponential distribution for the data is, therefore, rejected at the 0.05 level. A confidence interval for the mean life is

$$(\exp(5.99), \exp(6.30)) = (399.57, 542.18)$$

Example 31.4. Ordinal Model for Multinomial Data

This example illustrates how you can use the GENMOD procedure to fit a model to data measured on an ordinal scale. The following statements create a SAS data set called icecream. The data set contains the results of a hypothetical taste test of three brands of ice cream. The three brands are rated for taste on a five point scale from very good (vg) to very bad (vb). An analysis is performed to assess the differences in the ratings for the three brands. The variable taste contains the ratings and brand contains the brands tested. The variable count contains the number of testers rating each brand in each category.

The following statements create the icecream data set.

```
data icecream;
   input count brand$ taste$;
   datalines;
   70  ice1 vg
   71  ice1 g
   151 ice1 m
   30  ice1 b
   46  ice1 vb
   20  ice2 vg
   36  ice2 g
   130 ice2 m
   74  ice2 b
   70  ice2 vb
   50  ice3 vg
   55  ice3 g
   140 ice3 m
   52  ice3 b
   50  ice3 vb
   ;
run;
```

The following statements fit a cumulative logit model to the ordinal data with the variable taste as the response and the variable brand as a covariate. The variable count is used as a FREQ variable.

```
proc genmod rorder=data;
   freq count;
   class brand;
   model taste = brand / dist=multinomial
                         link=cumlogit
                         aggregate=brand
                         type1;

   estimate ’LogOR12’ brand 1 -1 / exp;
   estimate ’LogOR13’ brand 1  0  -1 / exp;
   estimate ’LogOR23’ brand 0  1  -1 / exp;
run;
```

The AGGREGATE=BRAND option in the MODEL statement specifies the variable brand as defining multinomial populations for computing deviances and Pearson chi-squares. The RORDER=DATA option specifies that the taste variable levels be ordered by their order of appearance in the input data set, that is, from very good (vg) to very bad (vb). By default, the response is sorted in increasing ASCII order. Always check the “Response Profiles” table to verify that response levels are appropriately ordered. The TYPE1 option requests a Type 1 test for the significance of the covariate brand.

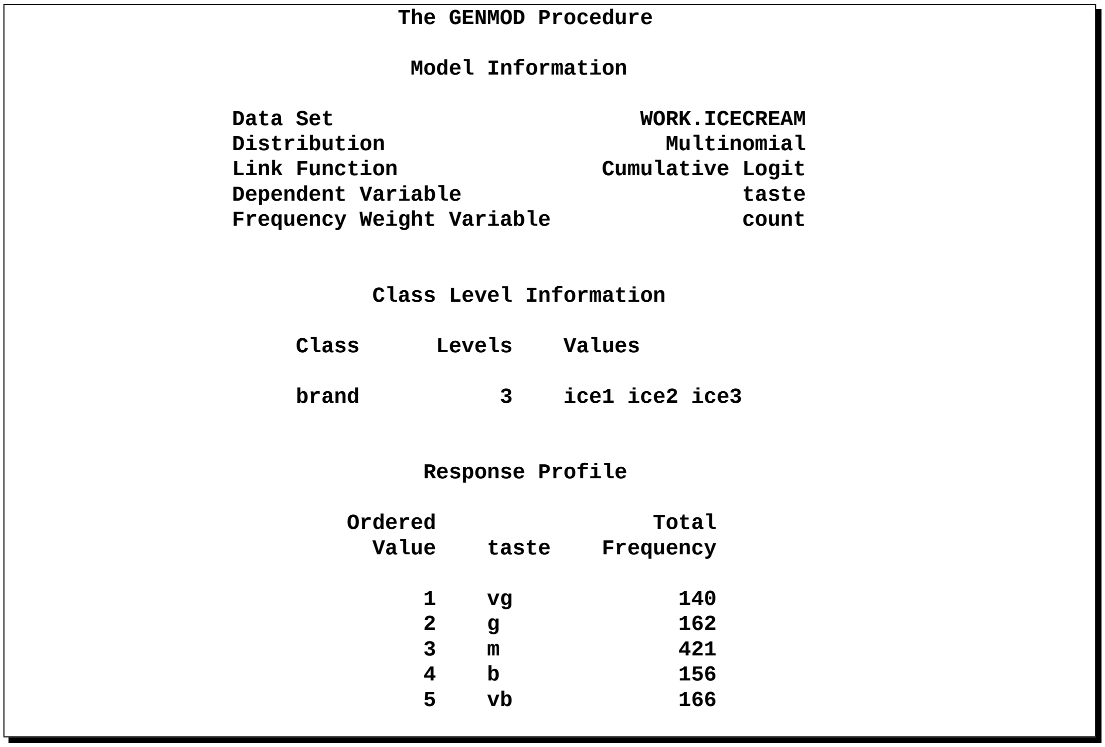
If $\gamma_j(\mathbf{x}) = \mathrm{Pr}(\mathrm{taste} \leq j)$ is the cumulative probability of the jth or lower taste category, then the odds ratio comparing $\mathbf{x}_1$ to $\mathbf{x}_2$ is as follows:

$$\frac{\gamma_j(\mathbf{x}_1)/(1-\gamma_j(\mathbf{x}_1))}{\gamma_j(\mathbf{x}_2)/(1-\gamma_j(\mathbf{x}_2))} = \exp[(\mathbf{x}_1 - \mathbf{x}_2)'\boldsymbol{\beta}]$$

Refer to McCullagh and Nelder (1989, Chapter 5) for details on the cumulative logit model. The ESTIMATE statements compute log odds ratios comparing each of brands. The EXP option in the ESTIMATE statements exponentiates the log odds ratios to form odds ratio estimates. Standard errors and confidence intervals are also computed.

Output 31.4.1 displays general information about the model and data, the levels of the CLASS variable brand, and the total number of occurrences of the ordered levels of the response variable taste.

Output 31.4.1. Ordinal Model Information



```
                              The GENMOD Procedure

                               Model Information

                 Data Set                              WORK.ICECREAM
                 Distribution                            Multinomial
                 Link Function                      Cumulative Logit
                 Dependent Variable                            taste
                 Frequency Weight Variable                     count


                            Class Level Information

                      Class      Levels    Values

                      brand           3    ice1 ice2 ice3


                               Response Profile

                          Ordered                  Total
                            Value    taste    Frequency

                                1    vg             140
                                2    g              162
                                3    m              421
                                4    b              156
                                5    vb             166
```

Output 31.4.2 displays estimates of the intercept terms and covariates and associated statistics. The intercept terms correspond to the four cumulative logits defined on the taste categories in the order shown in Output 31.4.1. That is, Intercept1 is the intercept for the first cumulative logit, $\log(\frac{p_1}{1-p_1})$, Intercept2 is the intercept for the second cumulative logit $\log(\frac{p_1+p_2}{1-(p_1+p_2)})$, and so forth.

Output 31.4.2. Parameter Estimates

```
                         Analysis Of Parameter Estimates

                                       Standard   Wald 95% Confidence      Chi-
  Parameter          DF    Estimate       Error         Limits           Square

  Intercept1          1     -1.8578      0.1219    -2.0967    -1.6189    232.35
  Intercept2          1     -0.8646      0.1056    -1.0716    -0.6576     67.02
  Intercept3          1      0.9231      0.1060     0.7154     1.1308     75.87
  Intercept4          1      1.8078      0.1191     1.5743     2.0413    230.32
  brand        ice1   1      0.3847      0.1370     0.1162     0.6532      7.89
  brand        ice2   1     -0.6457      0.1397    -0.9196    -0.3719     21.36
  brand        ice3   0      0.0000      0.0000     0.0000     0.0000       .
  Scale               0      1.0000      0.0000     1.0000     1.0000

                              Analysis Of Parameter
                                    Estimates

                           Parameter           Pr > ChiSq

                           Intercept1              <.0001
                           Intercept2              <.0001
                           Intercept3              <.0001
                           Intercept4              <.0001
                           brand        ice1       0.0050
                           brand        ice2       <.0001
                           brand        ice3         .
                           Scale

NOTE: The scale parameter was held fixed.
```

The Type 1 test displayed in Output 31.4.3 indicates that Brand is highly significant; that is, there are significant differences in the brands. The log odds ratios and odds ratios in the “ESTIMATE Statement Results” table indicate the relative differences between the brands. For example, the odds ratio of 2.8 in the “Exp(LogOR12)” row indicates that the odds of brand 1 being in lower taste categories is 2.8 times the odds of brand 2 being in lower taste categories. Since, in this ordering, the lower categories represent the more favorable taste results, this indicates that brand 1 scored significantly better than brand 2. This is also apparent from the data in this example.

Output 31.4.3. Type 1 Tests and Odds Ratios

```
                       LR Statistics For Type 1 Analysis

                                                    Chi-
          Source                Deviance        DF     Square    Pr > ChiSq

          Intercepts             65.9576
          brand                   9.8654         2      56.09        <.0001


                           Contrast Estimate Results

                        Standard                                Chi-
Label         Estimate     Error   Alpha   Confidence Limits  Square  Pr > ChiSq

LogOR12         1.0305    0.1401    0.05    0.7559    1.3050   54.11      <.0001
Exp(LogOR12)    2.8024    0.3926    0.05    2.1295    3.6878
LogOR13         0.3847    0.1370    0.05    0.1162    0.6532    7.89      0.0050
Exp(LogOR13)    1.4692    0.2013    0.05    1.1233    1.9217
LogOR23        -0.6457    0.1397    0.05   -0.9196   -0.3719   21.36      <.0001
Exp(LogOR23)    0.5243    0.0733    0.05    0.3987    0.6894
```

Example 31.5. GEE for Binary Data with Logit Link Function

Table 31.5 displays a partial listing of a SAS data set of clinical trial data comparing two treatments for a respiratory disorder. See “Gee Model for Binary Data” in the SAS/STAT Sample Program Library for the complete data set. These data are from Stokes, Davis, and Koch (1995), where a SAS macro is used to fit a GEE model. A GEE model is fit, using the REPEATED statement in the GENMOD procedure.

Table 31.5. Respiratory Disorder Data

Patients in each of two centers are randomly assigned to groups receiving the active treatment or a placebo. During treatment, respiratory status (coded here as 0=poor, 1=good) is determined for each of four visits. The variables center, treatment, sex, and baseline (baseline respiratory status) are classification variables with two levels. The variable age (age at time of entry into the study) is a continuous variable.

Explanatory variables in the model are Intercept (x_{ij1}), treatment (x_{ij2}), center (x_{ij3}), sex (x_{ij4}), age (x_{ij6}), and baseline (x_{ij6}), so that $\mathbf{x}'_{ij} = [x_{ij1}, x_{ij2}, \ldots, x_{ij6}]$ is the vector of explanatory variables. Indicator variables for the classification explanatory variables can be automatically generated by listing them in the CLASS statement in PROC GENMOD. However, in order to be consistent with the analysis in Stokes, Davis, and Koch (1995), the four classification explanatory variables are coded as follows:

$$x_{ij2} = \begin{cases} 0 \text{ placebo} \\ 1 \text{ active} \end{cases} \qquad x_{ij3} = \begin{cases} 0 \text{ center 1} \\ 1 \text{ center 2} \end{cases}$$

$$x_{ij4} = \begin{cases} 0 \text{ male} \\ 1 \text{ female} \end{cases} \qquad x_{ij6} = \begin{cases} 0 \text{ poor} \\ 1 \text{ good} \end{cases}$$

Suppose y_{ij} represents the respiratory status of patient i at the jth visit, $j = 1, \ldots, 4$, and $\mu_{ij} = \mathrm{E}(y_{ij})$ represents the mean of the respiratory status. Since the response data are binary, you can use the variance function for the binomial distribution $v(\mu_{ij}) = \mu_{ij}(1 - \mu_{ij})$ and the logit link function $g(\mu_{ij}) = \log(\mu_{ij}/(1 - \mu_{ij}))$. The model for the mean is $g(\mu_{ij}) = \mathbf{x}_{ij}{}'\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a vector of regression parameters to be estimated.

Further manipulation of the data set creates an observation for each visit with the respiratory status at each visit represented by the binary variable outcome and indicator variables for treatment (active), center (center2), and sex (female). A partial listing of the resulting data set is shown in Output 31.5.1.

Output 31.5.1. Respiratory Disorder Data

```
Obs   center   id   age   baseline   active   center2   female   visit   outcome

  1      1      1    46       0         0        0         0        1        0
  2      1      1    46       0         0        0         0        2        0
  3      1      1    46       0         0        0         0        3        0
  4      1      1    46       0         0        0         0        4        0
  5      1      2    28       0         0        0         0        1        0
  6      1      2    28       0         0        0         0        2        0
  7      1      2    28       0         0        0         0        3        0
  8      1      2    28       0         0        0         0        4        0
  9      1      3    23       1         1        0         0        1        1
 10      1      3    23       1         1        0         0        2        1
 11      1      3    23       1         1        0         0        3        1
 12      1      3    23       1         1        0         0        4        1
 13      1      4    44       1         0        0         0        1        1
 14      1      4    44       1         0        0         0        2        1
 15      1      4    44       1         0        0         0        3        1
 16      1      4    44       1         0        0         0        4        0
 17      1      5    13       1         0        0         1        1        1
 18      1      5    13       1         0        0         1        2        1
 19      1      5    13       1         0        0         1        3        1
 20      1      5    13       1         0        0         1        4        1
```

The GEE solution is requested with the REPEATED statement in the GENMOD procedure. The option SUBJECT=ID(CENTER) specifies that the observations in a single cluster are uniquely identified by center and id within center. The option TYPE=UNSTR specifies the unstructured working correlation structure. The MODEL statement specifies the regression model for the mean with the binomial distribution variance function.

```
proc genmod data=resp descend;
   class id center;
   model outcome=center2 active female age baseline / dist=bin;
   repeated  subject=id(center) / corr=unstr corrw;
run;
```

These statements first produce the usual output (not shown) for fitting the generalized linear (GLM) model specified in the MODEL statement. The parameter estimates from the GLM model are used as initial values for the GEE solution. The

DESCEND option in the PROC GENMOD statement specifies that the probability that outcome $= 1$ be modeled. If the DESCEND option had not been specified, the probability that outcome $= 0$ would be modeled by default.

Information about the GEE model is displayed in Output 31.5.2. The results of GEE model fitting are displayed in Output 31.5.3. If you specify no other options, the standard errors, confidence intervals, Z scores, and p-values are based on empirical standard error estimates. You can specify the MODELSE option in the REPEATED statement to create a table based on model-based standard error estimates.

Output 31.5.2. Model Fitting Information

```
                              The GENMOD Procedure

                             GEE Model Information

             Correlation Structure                          Unstructured
             Subject Effect                     id(center) (111 levels)
             Number of Clusters                                      111
             Correlation Matrix Dimension                              4
             Maximum Cluster Size                                      4
             Minimum Cluster Size                                      4
```

Output 31.5.3. Results of Model Fitting

```
                           Working Correlation Matrix

                         Col1          Col2          Col3          Col4

             Row1       1.0000        0.3351        0.2140        0.2953
             Row2       0.3351        1.0000        0.4429        0.3581
             Row3       0.2140        0.4429        1.0000        0.3964
             Row4       0.2953        0.3581        0.3964        1.0000


                        Analysis Of GEE Parameter Estimates
                         Empirical Standard Error Estimates

                          Standard   95% Confidence
          Parameter Estimate    Error       Limits           Z Pr > |Z|

          Intercept  -0.8882   0.4568  -1.7835   0.0071   -1.94   0.0519
          center2     0.6558   0.3512  -0.0326   1.3442    1.87   0.0619
          active      1.2442   0.3455   0.5669   1.9214    3.60   0.0003
          female      0.1128   0.4408  -0.7512   0.9768    0.26   0.7981
          age        -0.0175   0.0129  -0.0427   0.0077   -1.36   0.1728
          baseline    1.8981   0.3441   1.2237   2.5725    5.52   <.0001
```

The non-significance of age and female make them candidates for omission from the model.

Example 31.6. Log Odds Ratios and the ALR Algorithm

Since the respiratory data in Example 31.5 are binary, you can use the ALR algorithm to model the log odds ratios instead of using working correlations to model associations. Here, a “fully parameterized cluster” model for the log odds ratio is fit. That is, there is a log odds ratio parameter for each unique pair of responses within clusters, and all clusters are parameterized identically. The following statements fit the same regression model for the mean as in Example 31.5 but use a regression model for the log odds ratios instead of a working correlation. The LOGOR=FULLCLUST option specifies a fully parameterized log odds ratio model.

```
proc genmod data=resp descend;
   class id center;
   model outcome=center2 active female age baseline / dist=bin;
   repeated  subject=id(center) / logor=fullclust;
run;
```

The results of fitting the model are displayed in Output 31.6.1 along with a table that shows the correspondence between the log odds ratio parameters and the within cluster pairs.

Output 31.6.1. Results of Model Fitting

```
                           The GENMOD Procedure

                              Log Odds Ratio
                          Parameter Information

                          Parameter       Group

                          Alpha1          (1, 2)
                          Alpha2          (1, 3)
                          Alpha3          (1, 4)
                          Alpha4          (2, 3)
                          Alpha5          (2, 4)
                          Alpha6          (3, 4)


                   Analysis Of GEE Parameter Estimates
                    Empirical Standard Error Estimates

                         Standard   95% Confidence
Parameter Estimate    Error       Limits            Z Pr > |Z|

Intercept  -0.9266   0.4513 -1.8111  -0.0421   -2.05   0.0400
center2     0.6287   0.3486 -0.0545   1.3119    1.80   0.0713
active      1.2611   0.3406  0.5934   1.9287    3.70   0.0002
female      0.1024   0.4362 -0.7526   0.9575    0.23   0.8144
age        -0.0162   0.0125 -0.0407   0.0084   -1.29   0.1977
baseline    1.8980   0.3404  1.2308   2.5652    5.58   <.0001
Alpha1      1.6109   0.4892  0.6522   2.5696    3.29   0.0010
Alpha2      1.0771   0.4834  0.1297   2.0246    2.23   0.0259
Alpha3      1.5875   0.4735  0.6594   2.5155    3.35   0.0008
Alpha4      2.1224   0.5022  1.1381   3.1068    4.23   <.0001
Alpha5      1.8818   0.4686  0.9634   2.8001    4.02   <.0001
Alpha6      2.1046   0.4949  1.1347   3.0745    4.25   <.0001
```

You can fit the same model by fully specifying the *z*-matrix. The following statements create a data set containing the full *z*-matrix.

```
   data zin;
      keep id center z1-z6 y1 y2;
      array zin(6) z1-z6;
      set resp ;
      by center id;
      if first.id
         then do;
            t = 0;
            do m = 1 to 4;
               do n = m+1 to 4;
                  do j = 1 to 6;
                     zin(j) = 0;
                  end;
               y1 = m;
               y2 = n;
               t + 1;
               zin(t) = 1;
               output;
               end;
            end;
         end;
   run;


proc print data=zin (obs=12);
```

Output 31.6.2 displays the full *z*-matrix for the first two clusters. The *z*-matrix is identical for all clusters in this example.

Output 31.6.2. Full *z*-Matrix Data Set

```
Obs    z1    z2    z3    z4    z5    z6    center    id    y1    y2

  1     1     0     0     0     0     0       1       1     1     2
  2     0     1     0     0     0     0       1       1     1     3
  3     0     0     1     0     0     0       1       1     1     4
  4     0     0     0     1     0     0       1       1     2     3
  5     0     0     0     0     1     0       1       1     2     4
  6     0     0     0     0     0     1       1       1     3     4
  7     1     0     0     0     0     0       1       2     1     2
  8     0     1     0     0     0     0       1       2     1     3
  9     0     0     1     0     0     0       1       2     1     4
 10     0     0     0     1     0     0       1       2     2     3
 11     0     0     0     0     1     0       1       2     2     4
 12     0     0     0     0     0     1       1       2     3     4
```

The following statements fit the model for fully parameterized clusters by fully specifying the *z*-matrix. The results are identical to those shown previously.

```
proc genmod data=resp descend;
   class id center;
   model outcome=center2 active female age baseline / dist=bin;
   repeated  subject=id(center) / logor=zfull
                                  zdata=zin
                                  zrow =(z1-z6)
                                  ypair=(y1 y2) ;
run;
```

Example 31.7. Log-Linear Model for Count Data

These data, from Thall and Vail (1990), are concerned with the treatment of people suffering from epileptic seizure episodes. These data are also analyzed in Diggle, Liang, and Zeger (1994). The data consist of the number of epileptic seizures in an eight-week baseline period, before any treatment, and in each of four two-week treatment periods, in which patients received either a placebo or the drug Progabide in addition to other therapy. A portion of the data is displayed in Table 31.6. See "Gee Model for Count Data, Exchangeable Correlation" in the SAS/STAT Sample Program Library for the complete data set.

Table 31.6. Epileptic Seizure Data

Patient ID	Treatment	Baseline	Visit1	Visit2	Visit3	Visit4
104	Placebo	11	5	3	3	3
106	Placebo	11	3	5	3	3
107	Placebo	6	2	4	0	5
.						
.						
.						
101	Progabide	76	11	14	9	8
102	Progabide	38	8	7	9	4
103	Progabide	19	0	4	3	0
.						
.						
.						

Model the data as a log-linear model with $V(\mu) = \mu$ (the Poisson variance function) and

$$\begin{aligned}\log(E(Y_{ij})) &= \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + \\ &\quad x_{i1}x_{i2}\beta_3 + \log(t_{ij})\end{aligned}$$

where

- Y_{ij} = number of epileptic seizures in interval j
- t_{ij} = length of interval j

- $x_{i1} = \begin{cases} 1: & \text{weeks } 8-16 \text{ (treatment)} \\ 0: & \text{weeks } 0-8 \text{ (baseline)} \end{cases}$
- $x_{i2} = \begin{cases} 1: & \text{progabide group} \\ 0: & \text{placebo group} \end{cases}$

The correlations between the counts are modeled as $r_{ij} = \alpha$, $i \neq j$ (exchangeable correlations). For comparison, the correlations are also modeled as independent (identity correlation matrix). In this model, the regression parameters have the interpretation in terms of the log seizure rate displayed in Table 31.7.

Table 31.7. Interpretation of Regression Parameters

Treatment	**Visit**	$\log(E(Y_{ij})/t_{ij})$
Placebo	Baseline	β_0
	1-4	$\beta_0 + \beta_1$
Progabide	Baseline	$\beta_0 + \beta_2$
	1-4	$\beta_0 + \beta_1 + \beta_2 + \beta_3$

The difference between the log seizure rates in the pretreatment (baseline) period and the treatment periods is β_1 for the placebo group and $\beta_1 + \beta_3$ for the Progabide group. A value of $\beta_3 < 0$ indicates a reduction in the seizure rate.

Output 31.7.1 is a listing of the first 14 observations of the data, which are arranged as one visit per observation:

Output 31.7.1. Partial Listing of the Seizure Data

```
Obs     id    y    visit    trt    bline    age

  1    104    5      1       0       11      31
  2    104    3      2       0       11      31
  3    104    3      3       0       11      31
  4    104    3      4       0       11      31
  5    106    3      1       0       11      30
  6    106    5      2       0       11      30
  7    106    3      3       0       11      30
  8    106    3      4       0       11      30
  9    107    2      1       0        6      25
 10    107    4      2       0        6      25
 11    107    0      3       0        6      25
 12    107    5      4       0        6      25
 13    114    4      1       0        8      36
 14    114    4      2       0        8      36
```

Some further data manipulations create an observation for the baseline measures, a log time interval variable for use as an offset, and an indicator variable for whether the observation is for a baseline measurement or a visit measurement. Patient 207 is deleted as an outlier, as in the Diggle, Liang, and Zeger (1994) analysis.

```
data new;
   set thall;
   output;
   if visit=1 then do;
      y=bline;
      visit=0;
      output;
   end;
run;

data new;
   set new;
   if id ne 207;
   if visit=0 then do;
      x1=0;
      ltime=log(8);
   end;
   else do;
      x1=1;
      ltime=log(2);
   end;
run;
```

The GEE solution is requested by using the REPEATED statement in the GENMOD procedure. The SUBJECT=ID option indicates that the id variable describes the observations for a single cluster, and the CORRW option displays the working correlation matrix. The TYPE= option specifies the correlation structure; the value EXCH indicates the exchangeable structure.

```
proc genmod data=new;
   class id;
   model y=x1 | trt / d=poisson offset=ltime;
   repeated subject=id / corrw covb type=exch;
run;
```

These statements first produce the usual output from fitting a generalized linear model (GLM) to these data. The estimates are used as initial values for the GEE solution.

Information about the GEE model is displayed in Output 31.7.3. The results of fitting the model are displayed in Output 31.7.4. Compare these with the model of independence displayed in Output 31.7.2. The parameter estimates are nearly identical, but the standard errors for the independence case are underestimated. The coefficient of the interaction term, β_3, is highly significant under the independence model and marginally significant with the exchangeable correlations model.

Output 31.7.2. Independence Model

```
                              The GENMOD Procedure

                    Analysis Of Initial Parameter Estimates

                            Standard       Wald 95%         Chi-
   Parameter  DF  Estimate     Error   Confidence Limits   Square  Pr > ChiSq

   Intercept   1    1.3476    0.0341    1.2809    1.4144  1565.44      <.0001
   x1          1    0.1108    0.0469    0.0189    0.2027     5.58      0.0181
   trt         1   -0.1080    0.0486   -0.2034   -0.0127     4.93      0.0264
   x1*trt      1   -0.3016    0.0697   -0.4383   -0.1649    18.70      <.0001
   Scale       0    1.0000    0.0000    1.0000    1.0000

NOTE: The scale parameter was held fixed.
```

Output 31.7.3. GEE Model Information

```
                             GEE Model Information

                 Correlation Structure           Exchangeable
                 Subject Effect                id (58 levels)
                 Number of Clusters                        58
                 Correlation Matrix Dimension               5
                 Maximum Cluster Size                       5
                 Minimum Cluster Size                       5
```

Output 31.7.4. GEE Parameter Estimates

```
                       Analysis Of GEE Parameter Estimates
                        Empirical Standard Error Estimates

                            Standard   95% Confidence
         Parameter Estimate    Error       Limits            Z Pr > |Z|

         Intercept   1.3476   0.1574   1.0392   1.6560    8.56   <.0001
         x1          0.1108   0.1161  -0.1168   0.3383    0.95   0.3399
         trt        -0.1080   0.1937  -0.4876   0.2716   -0.56   0.5770
         x1*trt     -0.3016   0.1712  -0.6371   0.0339   -1.76   0.0781
```

Table 31.8 displays the regression coefficients, standard errors, and normalized coefficients that result from fitting the model using independent and exchangeable working correlation matrices.

Table 31.8. Results of Model Fitting

Variable	**Correlation** Structure	**Coef.**	**Std. Error**	**Coef./S.E.**
Intercept	Exchangeable	1.35	0.16	8.56
	Independent	1.35	0.03	39.52
Visit (x_1)	Exchangeable	0.11	0.12	0.95
	Independent	0.11	0.05	2.36
Treat (x_2)	Exchangeable	−0.11	0.19	−0.56
	Independent	−0.11	0.05	−2.22
$x_1 * x_2$	Exchangeable	−0.30	0.17	−1.76
	Independent	−0.30	0.07	−4.32

The fitted exchangeable correlation matrix is specified with the CORRW option and is displayed in Output 31.7.5.

Output 31.7.5. Working Correlation Matrix

```
                          Working Correlation Matrix

                   Col1          Col2          Col3          Col4          Col5

     Row1        1.0000        0.5941        0.5941        0.5941        0.5941
     Row2        0.5941        1.0000        0.5941        0.5941        0.5941
     Row3        0.5941        0.5941        1.0000        0.5941        0.5941
     Row4        0.5941        0.5941        0.5941        1.0000        0.5941
     Row5        0.5941        0.5941        0.5941        0.5941        1.0000
```

If you specify the COVB option, you produce both the model-based (naive) and the empirical (robust) covariance matrices. Output 31.7.6 contains these estimates.

Output 31.7.6. Covariance Matrices

```
                        Covariance Matrix (Model-Based)

                         Prm1          Prm2          Prm3          Prm4

        Prm1          0.01223      0.001520      -0.01223     -0.001520
        Prm2         0.001520       0.01519     -0.001520      -0.01519
        Prm3         -0.01223     -0.001520       0.02495      0.005427
        Prm4        -0.001520      -0.01519      0.005427       0.03748


                         Covariance Matrix (Empirical)

                         Prm1          Prm2          Prm3          Prm4

        Prm1          0.02476     -0.001152      -0.02476      0.001152
        Prm2        -0.001152       0.01348      0.001152      -0.01348
        Prm3         -0.02476      0.001152       0.03751     -0.002999
        Prm4         0.001152      -0.01348     -0.002999       0.02931
```

The two covariance estimates are similar, indicating an adequate correlation model.

Example 31.8. Model Assessment of Multiple Regression Using Aggregates of Residuals (Experimental)

Neter et al. (1996, Section 8.2) describe a study of 54 patients undergoing a certain kind of liver operation in a surgical unit. The data consist of the survival time and certain covariates. After a model selection procedure, they arrived at the following model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

where Y is the logarithm (base 10) of the survival time, X_1, X_2, X_3 are *blood-clotting score*, *prognostic index*, and *enzyme function*, and ϵ is a normal error term. A listing of the SAS data set containing the data is shown in Output 31.8.1. The variables Y, X1, X2, X3 correspond to Y, X_1, X_2, X_3, and LogX1 is $\log(X_1)$. The GENMOD fit of the model is shown in Output 31.8.2. The analysis first focuses on the adequacy of the functional form of X_1, *blood-clotting score*.

Output 31.8.1. Surgical Unit Example Data

```
                    Obs        Y        X1      X2      X3      LogX1

                      1     2.3010     6.7     62      81    0.82607
                      2     2.0043     5.1     59      66    0.70757
                      3     2.3096     7.4     57      83    0.86923
                      4     2.0043     6.5     73      41    0.81291
                      5     2.7067     7.8     65     115    0.89209
                      6     1.9031     5.8     38      72    0.76343
                      7     1.9031     5.7     46      63    0.75587
                      8     2.1038     3.7     68      81    0.56820
                      9     2.3054     6.0     67      93    0.77815
                     10     2.3075     3.7     76      94    0.56820
                     11     2.5172     6.3     84      83    0.79934
                     12     1.8129     6.7     51      43    0.82607
                     13     2.9191     5.8     96     114    0.76343
                     14     2.5185     5.8     83      88    0.76343
                     15     2.2253     7.7     62      67    0.88649
                     16     2.3365     7.4     74      68    0.86923
                     17     1.9395     6.0     85      28    0.77815
                     18     1.5315     3.7     51      41    0.56820
                     19     2.3324     7.3     68      74    0.86332
                     20     2.2355     5.6     57      87    0.74819
                     21     2.0374     5.2     52      76    0.71600
                     22     2.1335     3.4     83      53    0.53148
                     23     1.8451     6.7     26      68    0.82607
                     24     2.3424     5.8     67      86    0.76343
                     25     2.4409     6.3     59     100    0.79934
                     26     2.1584     5.8     61      73    0.76343
                     27     2.2577     5.2     52      86    0.71600
                     28     2.7589    11.2     76      90    1.04922
                     29     1.8573     5.2     54      56    0.71600
                     30     2.2504     5.8     76      59    0.76343
                     31     1.8513     3.2     64      65    0.50515
                     32     1.7634     8.7     45      23    0.93952
                     33     2.0645     5.0     59      73    0.69897
                     34     2.4698     5.8     72      93    0.76343
                     35     2.0607     5.4     58      70    0.73239
                     36     2.2648     5.3     51      99    0.72428
                     37     2.0719     2.6     74      86    0.41497
                     38     2.0792     4.3      8     119    0.63347
                     39     2.1790     4.8     61      76    0.68124
                     40     2.1703     5.4     52      88    0.73239
                     41     1.9777     5.2     49      72    0.71600
                     42     1.8751     3.6     28      99    0.55630
                     43     2.6840     8.8     86      88    0.94448
                     44     2.1847     6.5     56      77    0.81291
                     45     2.2810     3.4     77      93    0.53148
                     46     2.0899     6.5     40      84    0.81291
                     47     2.4928     4.5     73     106    0.65321
                     48     2.5999     4.8     86     101    0.68124
                     49     2.1987     5.1     67      77    0.70757
                     50     2.4914     3.9     82     103    0.59106
                     51     2.0934     6.6     77      46    0.81954
                     52     2.0969     6.4     85      40    0.80618
                     53     2.2967     6.4     59      85    0.80618
                     54     2.4955     8.8     78      72    0.94448
```

Output 31.8.2. Regression Model for Linear X1

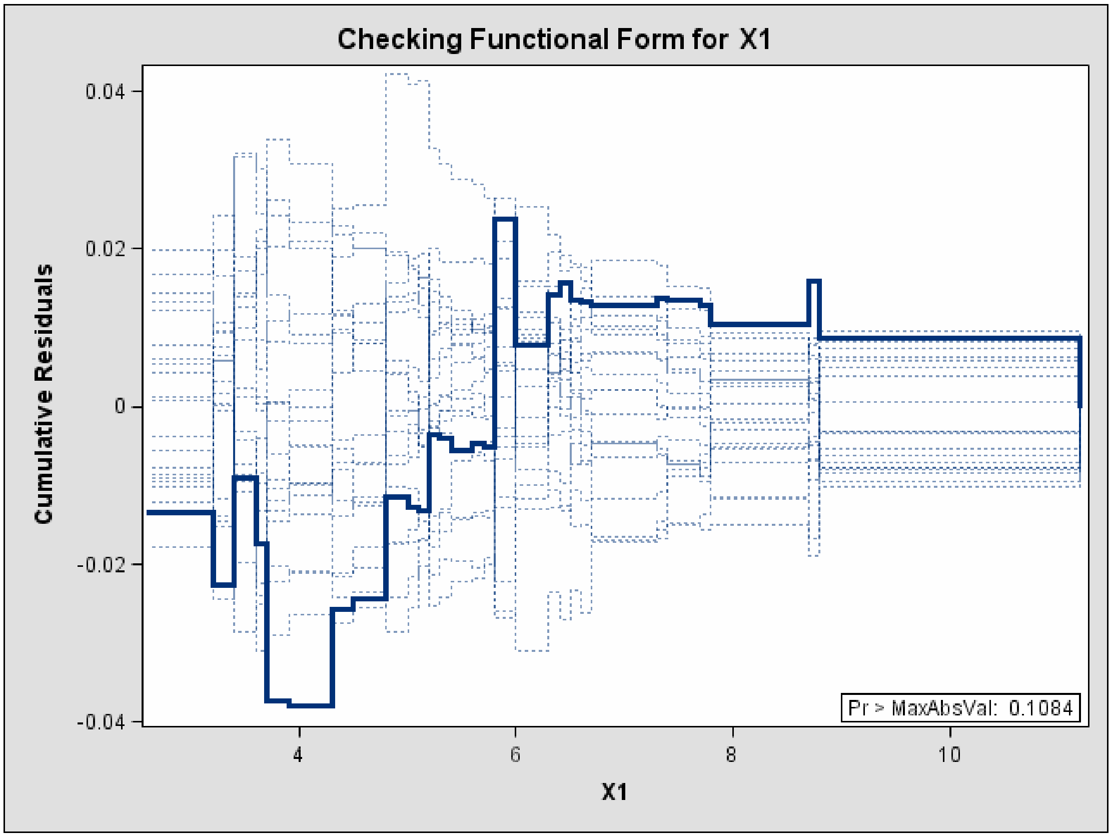
```
                              The GENMOD Procedure

                        Analysis Of Parameter Estimates

                            Standard       Wald 95%         Chi-
  Parameter  DF  Estimate      Error   Confidence Limits   Square  Pr > ChiSq

  Intercept   1    0.4836     0.0426    0.4001    0.5672   128.71      <.0001
  X1          1    0.0692     0.0041    0.0612    0.0772   288.17      <.0001
  X2          1    0.0093     0.0004    0.0085    0.0100   590.45      <.0001
  X3          1    0.0095     0.0003    0.0089    0.0101   966.07      <.0001
  Scale       0    0.0469     0.0000    0.0469    0.0469

NOTE: The scale parameter was estimated by the square root of Pearson's
      Chi-Square/DOF.
```

In order to assess the adequacy of the fitted multiple regression model, the ASSESS statement in the following SAS statements was used to create the plots of cumulative residuals against X1 shown in Output 31.8.3 and Output 31.8.4 and the summary table in Output 31.8.5. The RESAMPLE= keyword specifies that a p-value be computed based on a sample of 10,000 simulated residual paths. A random number seed is specified by the SEED= keyword for reproducibility. If you do not specify the seed, one is derived from the time of day. The keyword CRPANEL specifies that the panel of four cumulative residual plots shown in Output 31.8.4 be created, each with two simulated paths. The single residual plot with 20 simulated paths in Output 31.8.3 is created by default.

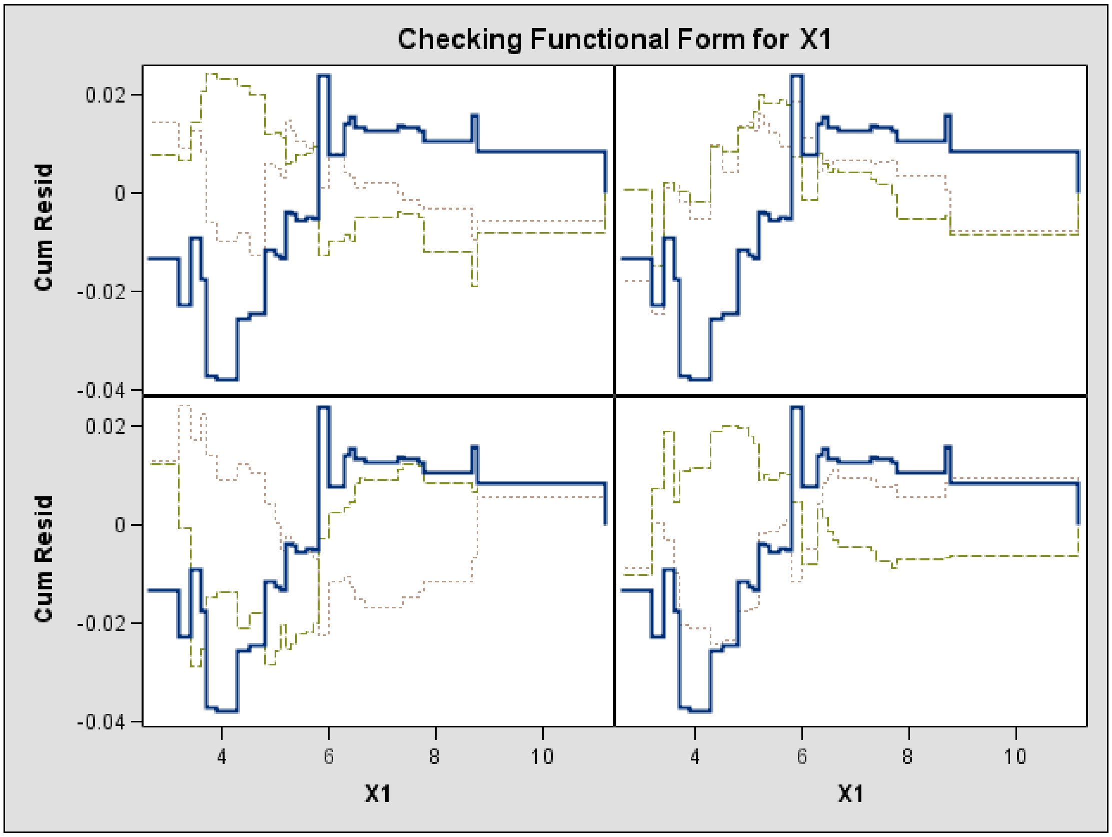
```
ods html;
ods graphics on;

proc genmod data=Surg;
   model Y = X1 X2 X3 / scale=Pearson;
   assess var=(X1) / resample=10000
                     seed=603708000
                     crpanel ;
run;

ods graphics off;
ods html close;
```

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental ASSESS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the GENMOD procedure, see the “ODS Graphics” section on page 1695.

Output 31.8.3. Cumulative Residual Plot for Linear X1 Fit (Experimental)



Output 31.8.4. Cumulative Residual Panel Plot for Linear X1 Fit (Experimental)



Output 31.8.5. Summary of Model Assessment

```
                             Assessment Summary

               Maximum
Assessment    Absolute                                          Pr >
Variable         Value    Replications          Seed     MaxAbsVal

X1              0.0380           10000     603708000        0.1084
```

The p-value of 0.1084 reported on Output 31.8.3 and Output 31.8.5 suggests that a more adequate model may be possible. The observed cumulative residuals on Output 31.8.3 and Output 31.8.4, represented by the heavy lines, seem atypical of the simulated curves, represented by the light lines, reinforcing the conclusion that a more appropriate functional form for X1 is possible.

The cumulative residual plots in Output 31.8.6 provide guidance in determining a more appropriate functional form. The four curves were created from simple forms of model misspecification using simulated data. The mean models of the data and the fitted model are shown in Table 31.9.

Output 31.8.6. Typical Cumulative Residual Patterns

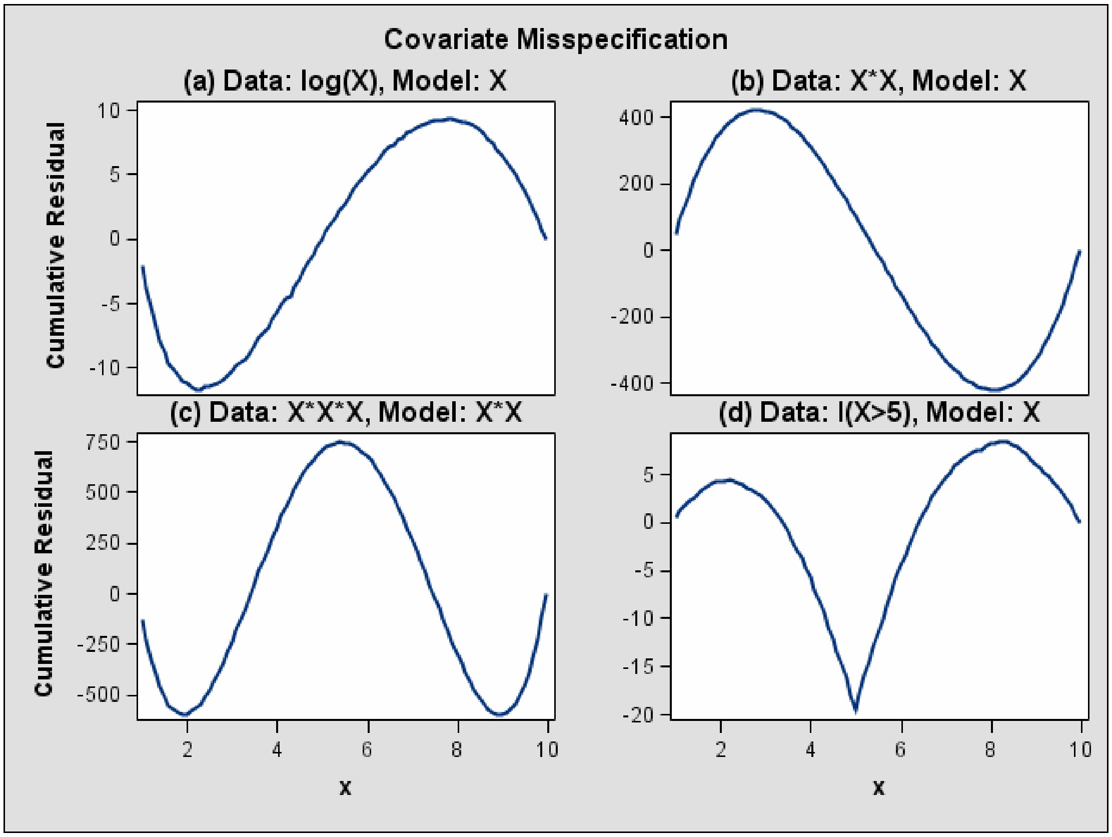


Table 31.9. Model Misspecifications

Plot	Data E(Y)	Fitted Model E(Y)
(a)	log(X)	X
(b)	$X + X^2$	X
(c)	$X + X^2 + X^3$	$X + X^2$
(d)	$I(X > 5)$	X

The observed cumulative residual pattern in Output 31.8.3 and Output 31.8.4 most resembles the behavior of the curve in plot (a) of Output 31.8.6, indicating that log(X_1) might be a more appropriate term in the model than X_1.

The following SAS statements fit a model with LogX1 in place of X1 and request a model assessment.

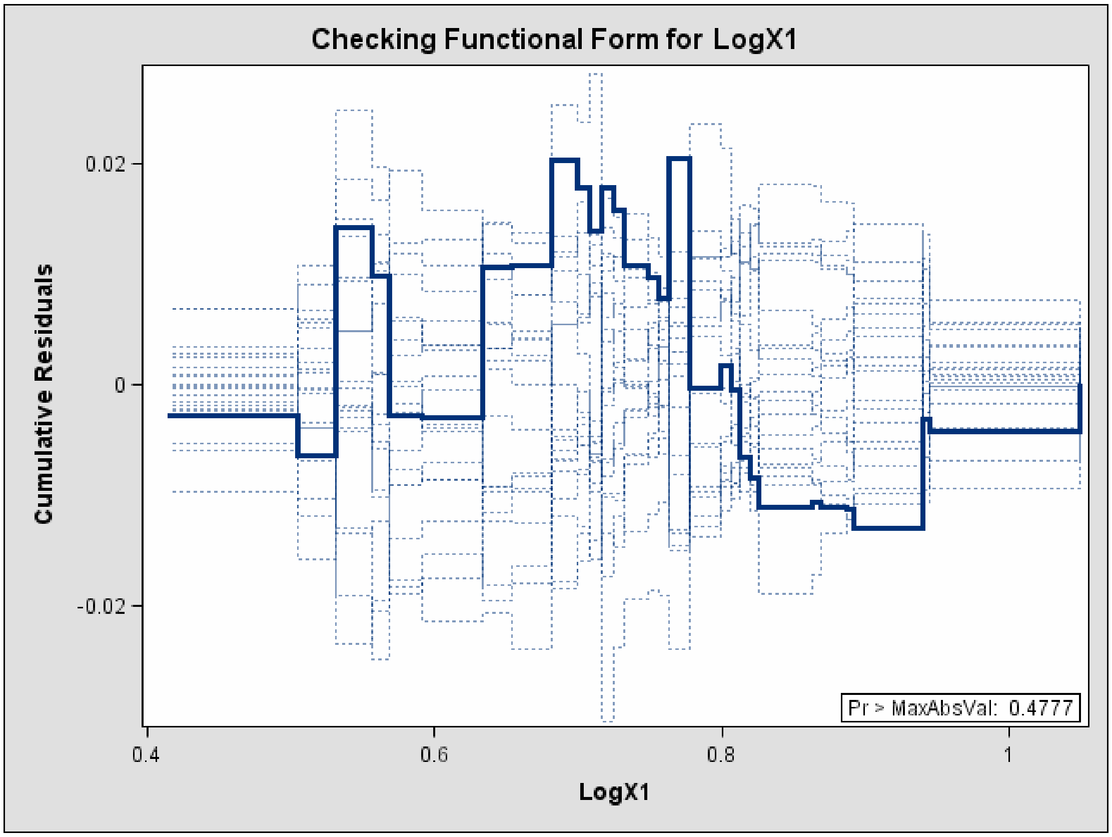
```
ods html;
ods graphics on;

proc genmod data=Surg;
   model Y = LogX1 X2 X3 / scale=Pearson;
   assess var=(LogX1) / resample=10000
                        seed=603708000;
run;

ods graphics off;
ods html close;
```

The revised model fit is shown in Output 31.8.7, the p-value from the simulation is 0.4777, and the cumulative residuals plotted on Output 31.8.8 show no systematic trend. The log-transformation for X1 is more appropriate. Under the revised model, the p-values for testing the functional forms of X2 and X3 are 0.20 and 0.63, and the p-value for testing the linearity of the model is is 0.65. Thus, the revised model seems reasonable.

Output 31.8.7. Multiple Regression Model With Log(X1)

```
                              The GENMOD Procedure

                         Analysis Of Parameter Estimates

                                 Standard      Wald 95%          Chi-
   Parameter  DF  Estimate      Error    Confidence Limits   Square  Pr > ChiSq

   Intercept   1    0.1844     0.0504    0.0857    0.2832     13.41      0.0003
   LogX1       1    0.9121     0.0491    0.8158    1.0083    345.05      <.0001
   X2          1    0.0095     0.0004    0.0088    0.0102    728.62      <.0001
   X3          1    0.0096     0.0003    0.0090    0.0101   1139.73      <.0001
   Scale       0    0.0434     0.0000    0.0434    0.0434

NOTE: The scale parameter was estimated by the square root of Pearson’s
      Chi-Square/DOF.
```

Output 31.8.8. Cumulative Residual Plot With Log(X1)(Experimental)



Example 31.9. Assessment of a Marginal Model for Dependent Data Using Aggregates of Residuals (Experimental)

This example illustrates the use of cumulative residuals to assess the adequacy of a marginal model for dependent data fit by generalized estimating equations (GEEs). The assessment methods are applied to CD4 count data from an AIDS clinical trial reported by Fischl et al. (1990), and reanalyzed by Lin, Wei, and Ying (2002). The study randomly assigned 360 HIV patients to AZT and 351 to placebo. CD4 counts were measured repeatedly over the course of the study. The data used here are the 4328 measurements taken in the first 40 weeks of the study.

The analysis focuses on the time trend of the response. The first model considered is

$$E(y_{ik}) = \beta_0 + \beta_1 T_{ik} + \beta_2 T_{ik}^2 + \beta_3 R_i T_{ik} + \beta_4 R_i T_{ik}^2$$

where T_{ik} is the time (in weeks) of the kth measurement on the ith patient, y_{ik} is the CD4 count at T_{ik} for the ith patient, and R_i is the indicator of AZT for the ith patient. Normal errors and an independent working correlation are assumed.

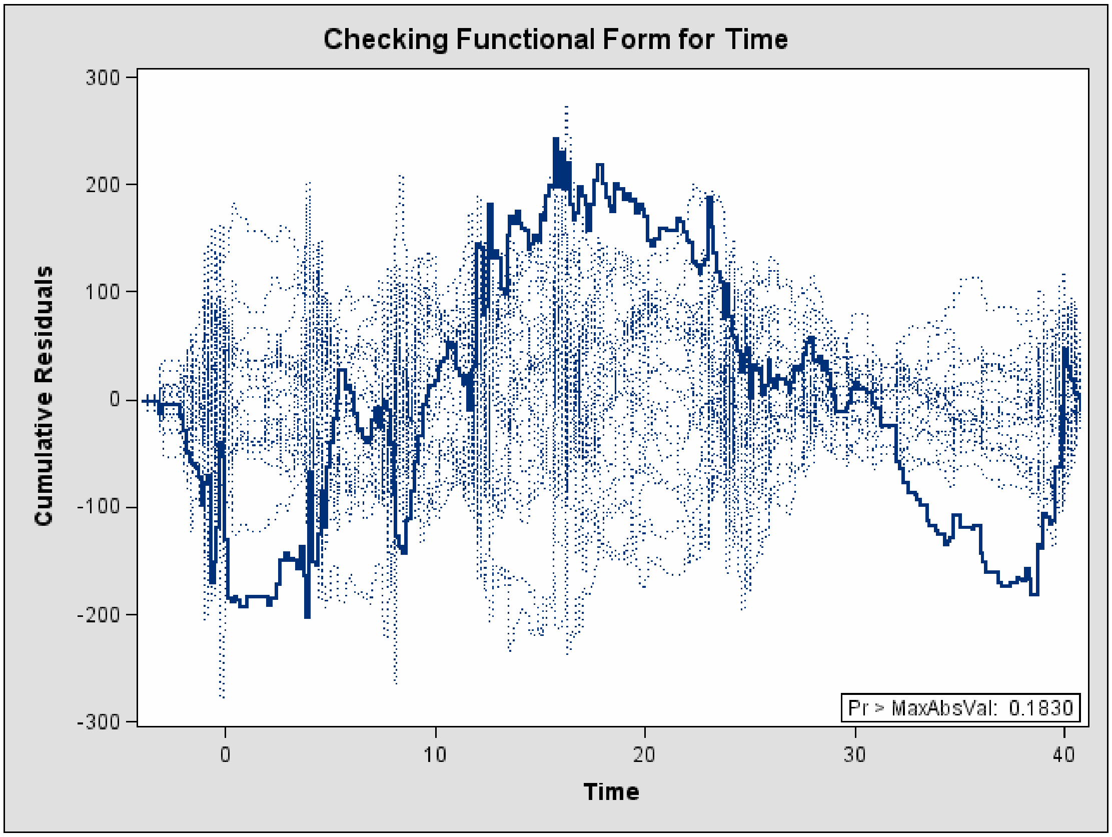
The following SAS statements fit the preceding model, create the cumulative residual plot in Output 31.9.1, and compute a p-value for the model.

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental ASSESS statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the GENMOD procedure, see the "ODS Graphics" section on page 1695.

Here, the SAS data set variables Time, Time2, TrtTime, and TrtTime2 correspond to T_{ik}, T_{ik}^2, R_iT_{ik}, and $R_iT_{ik}^2$, respectively. The variable Id identifies individual patients.

```
ods html;
ods graphics on;

proc genmod data=cd4;
   class Id;
   model Y = Time Time2 TrtTime TrtTime2;
   repeated sub=Id;
   assess var=(Time) / resample
                       seed=603708000;
run;

ods graphics off;
ods html close;
```

Output 31.9.1. Cumulative Residual Plot for Quadratic Time Fit (Experimental)

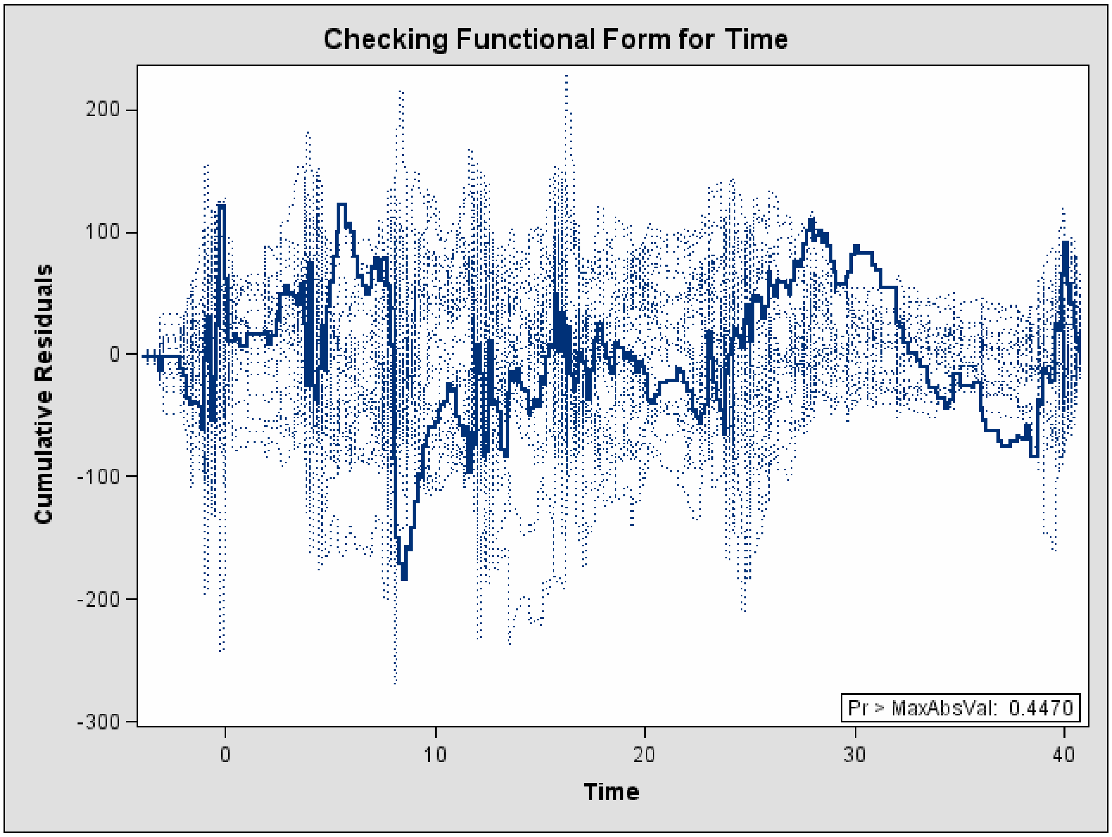


The cumulative residual plot in Output 31.9.1 displays cumulative residuals versus time for the model and 20 simulated realizations. The associated p-value, also shown on Output 31.9.1, is 0.18. These results indicate that a more satisfactory model might be possible. The observed cumulative residual pattern most resembles plot (c) in Output 31.8.6, suggesting cubic time trends.

The following SAS statements fit the model, create the plot in Output 31.9.2, and compute a p-value for a model with the additional terms T_{ik}^3 and $R_iT_{ik}^3$.

```
ods html;
ods graphics on;

proc genmod data=cd4;
   class Id;
   model Y = Time Time2 Time3 TrtTime TrtTime2 TrtTime3;
   repeated sub=Id;
   assess var=(Time) / resample
                       seed=603708000;
run;

ods graphics off;
ods html close;
```

Output 31.9.2. Cumulative Residual Plot for Cubic Time Fit (Experimental)



The observed cumulative residual pattern appears more typical of the simulated realizations, and the p-value is 0.45, indicating that the model with cubic time trends is more appropriate.

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Aitkin, M. (1987), "Modelling Variance Heterogeneity in Normal Regression Using GLIM," *Applied Statistics*, 36, 332–339.

Aitkin, M., Anderson, D., Francis, B., and Hinde, J. (1989), *Statistical Modelling in GLIM*, Oxford: Oxford Science Publications.

Boos, D. (1992), "On Generalized Score Tests," *The American Statistician*, 46, 327–333.

Cameron, A.C. and Trivedi, P.K. (1998), *Regression Analysis of Count Data*, Cambridge: Cambridge University Press.

Carey, V., Zeger, S.L., and Diggle, P. (1993), "Modelling Multivariate Binary Data with Alternating Logistic Regressions," *Biometrika*, 80, 517–526.

Davison, A.C. and Snell, E.J. (1991), "Residuals and Diagnostics," in *Statistical Theory and Modelling*, ed. D.V. Hinkley, N. Reid, and E.J. Snell, London: Chapman and Hall.

Diggle, P.J., Liang, K.Y., and Zeger, S.L. (1994), *Analysis of Longitudinal Data*, Oxford: Clarendon Press.

Dobson, A. (1990), *An Introduction to Generalized Linear Models*, London: Chapman and Hall.

Firth, D. (1991), "Generalized Linear Models," in *Statistical Theory and Modelling*, ed. Hinkley, D.V., Reid, N., and Snell, E.J., London: Chapman and Hall.

Fischl, M.A., Richman, D.D., Hansen, N., et al. (1990), "The Safety and Efficacy of Zidovudine (AZT) in the Treatment of Subjects with Mildly Symptomatic Human Immunodeficiency Virus Type I (HIV) Infection," *Annals of Internal Medicine*, 112, 727–737.

Hilbe, J. (1994), "Log Negative Binomial Regression Using the GENMOD Procedure," *Proceedings of the Nineteenth Annual SAS User's Group International Conference*, 14, 1199–1204.

Jennrich, R.I. and Schluchter, M.D. (1986), "Unbalanced Repeated-Measures Models with Structured Covariance Matrices," *Biometrics*, 42, 805–820.

Lawless, J.E. (1982), *Statistical Models and Methods for Lifetime Data*, New York: John Wiley & Sons, Inc.

Lawless, J.E. (1987), "Negative Binomial and Mixed Poisson Regression," *The Canadian Journal of Statistics*, 15, 209–225.

Liang, K.Y. and Zeger, S.L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Lin, D.Y., Wei, L.J., and Ying, Z. (2002), "Model-Checking Techniques Based on Cumulative Residuals," *Biometrics*, 58, 1–12.

Lipsitz, S.H., Fitzmaurice, G.M., Orav, E.J., and Laird, N.M. (1994), "Performance of Generalized Estimating Equations in Practical Situations," *Biometrics*, 50, 270–278.

Lipsitz, S.H., Kim, K., and Zhao, L. (1994), "Analysis of Repeated Categorical Data Using Generalized Estimating Equations," *Statistics in Medicine*, 13, 1149–1163.

Lipsitz, S.H., Laird, N.M., and Harrington, D.P., (1991), "Generalized Estimating Equations for Correlated Binary Data: Using the Odds Ratio as a Measure of Association," *Biometrika*, 78, 153–160.

Littell, Ramon C., Freund, Rudolf J., and Spector, Philip C. (1991), *SAS System for Linear Models*, Third Edition, Cary, NC: SAS Institute Inc.

McCullagh, P. (1983), "Quasi-Likelihood Functions," *Annals of Statistics*, 11, 59–67.

McCullagh, P. and Nelder, J.A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman and Hall.

Miller, M.E., Davis, C.S., and Landis, J.R. (1993), "The Analysis of Longitudinal Polytomous Data: Generalized Estimating Equations and Connections with Weighted Least Squares," *Biometrics*, 49, 1033–1044.

Nelder, J.A. and Wedderburn, R.W.M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370–384.

Nelson, W. (1982), *Applied Life Data Analysis*, New York: John Wiley & Sons, Inc.

Neter, J., Kutner, M.H., Nachtsheim, C.J., and Wasserman, W. (1996), *Applied Linear Statistical Models,* Fourth Edition, Chicago: IRWIN.

Rao, C.R. (1973), *Linear Statistical Inference and Its Applications*, New York: John Wiley & Sons, Inc.

Rotnitzky, A. and Jewell, N.P., (1990), "Hypothesis Testing of Regression Parameters in Semiparametric Generalized Linear Models for Cluster Correlated Data," *Biometrika*, 77, 485–497.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Selvin, S. (1995), *Practical Biostatistical Methods*, Belmont, California: Duxbury Press

Stokes, M.E., Davis, C.S., and Koch, G.G (1995), *Categorical Data Analysis Using the SAS System*, Cary NC: SAS Institute Inc.

Thall, P.F. and Vail, S.C. (1990), "Some Covariance Models for Longitudinal Count Data with Overdispersion," *Biometrics,* 46, 657–671.

Ware, J.H., Dockery, Spiro A. III, Speizer, F.E., and Ferris, B.G., Jr. (1984), "Passive Smoking, Gas Cooking, and Respiratory Health of Children Living in Six Cities," *American Review of Respiratory Diseases*, 129, 366–374.

Williams, D.A. (1987), "Generalized Linear Model Diagnostics Using the Deviance and Single Case Deletions," *Applied Statistics*, 36, 181–191.

Chapter 32
The GLM Procedure

Chapter Contents

Chapter 32
The GLM Procedure

Overview

The GLM procedure uses the method of least squares to fit general linear models. Among the statistical methods available in PROC GLM are regression, analysis of variance, analysis of covariance, multivariate analysis of variance, and partial correlation.

PROC GLM analyzes data within the framework of General linear models. PROC GLM handles models relating one or several continuous dependent variables to one or several independent variables. The independent variables may be either *classification* variables, which divide the observations into discrete groups, or *continuous* variables. Thus, the GLM procedure can be used for many different analyses, including

- simple regression
- multiple regression
- analysis of variance (ANOVA), especially for unbalanced data
- analysis of covariance
- response-surface models
- weighted regression
- polynomial regression
- partial correlation
- multivariate analysis of variance (MANOVA)
- repeated measures analysis of variance

PROC GLM Features

The following list summarizes the features in PROC GLM:

- PROC GLM enables you to specify any degree of interaction (crossed effects) and nested effects. It also provides for polynomial, continuous-by-class, and continuous-nesting-class effects.
- Through the concept of estimability, the GLM procedure can provide tests of hypotheses for the effects of a linear model regardless of the number of missing cells or the extent of confounding. PROC GLM displays the Sum of Squares (SS) associated with each hypothesis tested and, upon request, the form of the estimable functions employed in the test. PROC GLM can produce the general form of all estimable functions.

- The REPEATED statement enables you to specify effects in the model that represent repeated measurements on the same experimental unit for the same response, providing both univariate and multivariate tests of hypotheses.
- The RANDOM statement enables you to specify random effects in the model; expected mean squares are produced for each Type I, Type II, Type III, Type IV, and contrast mean square used in the analysis. Upon request, F tests using appropriate mean squares or linear combinations of mean squares as error terms are performed.
- The ESTIMATE statement enables you to specify an $\mathbf{L}$ vector for estimating a linear function of the parameters $\mathbf{L}\beta$.
- The CONTRAST statement enables you to specify a contrast vector or matrix for testing the hypothesis that $\mathbf{L}\beta = 0$. When specified, the contrasts are also incorporated into analyses using the MANOVA and REPEATED statements.
- The MANOVA statement enables you to specify both the hypothesis effects and the error effect to use for a multivariate analysis of variance.
- PROC GLM can create an output data set containing the input dataset in addition to predicted values, residuals, and other diagnostic measures.
- PROC GLM can be used interactively. After specifying and running a model, a variety of statements can be executed without recomputing the model parameters or sums of squares.
- For analysis involving multiple dependent variables but not the MANOVA or REPEATED statements, a missing value in one dependent variable does not eliminate the observation from the analysis for other dependent variables. PROC GLM automatically groups together those variables that have the same pattern of missing values within the data set or within a BY group. This ensures that the analysis for each dependent variable brings into use all possible observations.
- Experimental graphics are now available with the GLM procedure. For more information, see the "ODS Graphics" section on page 1846.

PROC GLM Contrasted with Other SAS Procedures

As described previously, PROC GLM can be used for many different analyses and has many special features not available in other SAS procedures. However, for some types of analyses, other procedures are available. As discussed in the "PROC GLM for Unbalanced ANOVA" and "PROC GLM for Quadratic Least Squares Regression" sections (beginning on page 1735), sometimes these other procedures are more efficient than PROC GLM. The following procedures perform some of the same analyses as PROC GLM:

ANOVA performs analysis of variance for balanced designs. The ANOVA procedure is generally more efficient than PROC GLM for these designs.

MIXED fits mixed linear models by incorporating covariance structures in the model fitting process. Its RANDOM and REPEATED statements are similar to those in PROC GLM but offer different functionalities.

NESTED performs analysis of variance and estimates variance components for nested random models. The NESTED procedure is generally more efficient than PROC GLM for these models.

NPAR1WAY performs nonparametric one-way analysis of rank scores. This can also be done using the RANK procedure and PROC GLM.

REG performs simple linear regression. The REG procedure allows several MODEL statements and gives additional regression diagnostics, especially for detection of collinearity. PROC REG also creates plots of model summary statistics and regression diagnostics.

RSREG performs quadratic response-surface regression, and canonical and ridge analysis. The RSREG procedure is generally recommended for data from a response surface experiment.

TTEST compares the means of two groups of observations. Also, tests for equality of variances for the two groups are available. The TTEST procedure is usually more efficient than PROC GLM for this type of data.

VARCOMP estimates variance components for a general linear model.

Getting Started

PROC GLM for Unbalanced ANOVA

Analysis of variance, or ANOVA, typically refers to partitioning the variation in a variable's values into variation between and within several groups or classes of observations. The GLM procedure can perform simple or complicated ANOVA for balanced or unbalanced data.

This example discusses a 2 × 2 ANOVA model. The experimental design is a full factorial, in which each level of one treatment factor occurs at each level of the other treatment factor. The data are shown in a table and then read into a SAS data set.

		A	
		1	2
B	1	12 14	20 18
	2	11 9	17

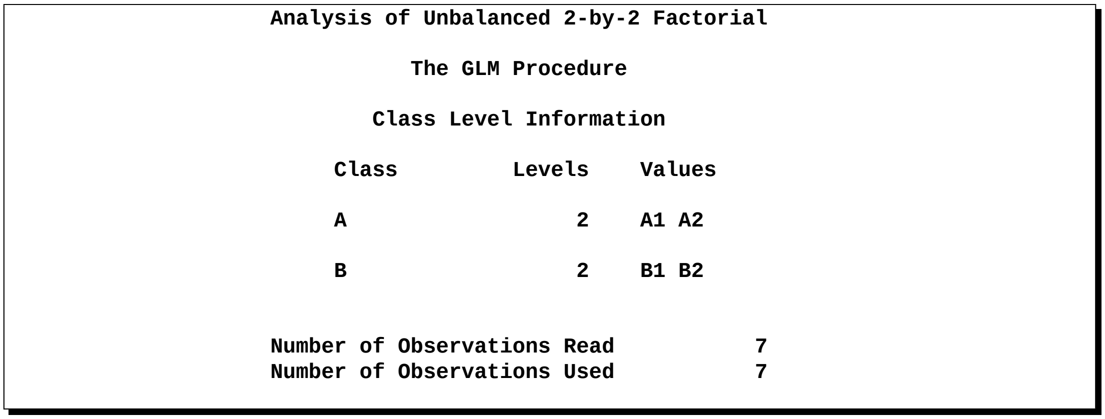
```
title 'Analysis of Unbalanced 2-by-2 Factorial';
data exp;
   input A $ B $ Y @@;
   datalines;
A1 B1 12 A1 B1 14     A1 B2 11 A1 B2 9
A2 B1 20 A2 B1 18     A2 B2 17
;
```

Note that there is only one value for the cell with A='A2' and B='B2'. Since one cell contains a different number of values from the other cells in the table, this is an unbalanced design.

The following PROC GLM invocation produces the analysis.

```
proc glm;
   class A B;
   model Y=A B A*B;
run;
```

Both treatments are listed in the CLASS statement because they are classification variables. A*B denotes the interaction of the A effect and the B effect. The results are shown in Figure 32.1 and Figure 32.2.



```
                    Analysis of Unbalanced 2-by-2 Factorial

                               The GLM Procedure

                            Class Level Information

                         Class          Levels    Values

                         A                   2    A1 A2

                         B                   2    B1 B2


                    Number of Observations Read           7
                    Number of Observations Used           7
```

Figure 32.1. Class Level Information

Figure 32.1 displays information about the classes as well as the number of observations in the data set. Figure 32.2 shows the ANOVA table, simple statistics, and tests of effects.

```
                    Analysis of Unbalanced 2-by-2 Factorial

                              The GLM Procedure

Dependent Variable: Y

                                          Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                        3      91.71428571    30.57142857     15.29   0.0253

 Error                        3       6.00000000     2.00000000

 Corrected Total              6      97.71428571


               R-Square     Coeff Var      Root MSE        Y Mean

               0.938596      9.801480      1.414214      14.42857


 Source                      DF        Type I SS    Mean Square   F Value   Pr > F

 A                            1      80.04761905    80.04761905     40.02   0.0080
 B                            1      11.26666667    11.26666667      5.63   0.0982
 A*B                          1       0.40000000     0.40000000      0.20   0.6850


 Source                      DF      Type III SS    Mean Square   F Value   Pr > F

 A                            1      67.60000000    67.60000000     33.80   0.0101
 B                            1      10.00000000    10.00000000      5.00   0.1114
 A*B                          1       0.40000000     0.40000000      0.20   0.6850
```

Figure 32.2. ANOVA Table and Tests of Effects

The degrees of freedom may be used to check your data. The Model degrees of freedom for a 2×2 factorial design with interaction are $(ab - 1)$, where a is the number of levels of A and b is the number of levels of B; in this case, $(2 \times 2 - 1) = 3$. The Corrected Total degrees of freedom are always one less than the number of observations used in the analysis; in this case, $7 - 1 = 6$.

The overall F test is significant $(F = 15.29, p = 0.0253)$, indicating strong evidence that the means for the four different A×B cells are different. You can further analyze this difference by examining the individual tests for each effect.

Four types of estimable functions of parameters are available for testing hypotheses in PROC GLM. For data with no missing cells, the Type III and Type IV estimable functions are the same and test the same hypotheses that would be tested if the data were balanced. Type I and Type III sums of squares are typically not equal when the data are unbalanced; Type III sums of squares are preferred in testing effects in unbalanced cases because they test a function of the underlying parameters that is independent of the number of observations per treatment combination.

According to a significance level of 5% $(\alpha = 0.05)$, the A*B interaction is not significant $(F = 0.20, p = 0.6850)$. This indicates that the effect of A does not depend on the level of B and vice versa. Therefore, the tests for the individual effects are valid, showing a significant A effect $(F = 33.80, p = 0.0101)$ but no significant B effect $(F = 5.00, p = 0.1114)$.

PROC GLM for Quadratic Least Squares Regression

In polynomial regression, the values of a dependent variable (also called a response variable) are described or predicted in terms of polynomial terms involving one or more independent or explanatory variables. An example of quadratic regression in PROC GLM follows. These data are taken from Draper and Smith (1966, p. 57). Thirteen specimens of 90/10 Cu-Ni alloys are tested in a corrosion-wheel setup in order to examine corrosion. Each specimen has a certain iron content. The wheel is rotated in salt sea water at 30 ft/sec for 60 days. Weight loss is used to quantify the corrosion. The fe variable represents the iron content, and the loss variable denotes the weight loss in milligrams/square decimeter/day in the following DATA step.

```
title 'Regression in PROC GLM';
data iron;
   input fe loss @@;
   datalines;
0.01 127.6   0.48 124.0   0.71 110.8   0.95 103.9
1.19 101.5   0.01 130.1   0.48 122.0   1.44  92.3
0.71 113.1   1.96  83.7   0.01 128.0   1.44  91.4
1.96  86.2
;
```

The GPLOT procedure is used to request a scatter plot of the response variable versus the independent variable.

```
symbol1 c=blue;
proc gplot;
   plot loss*fe / vm=1;
run;
```

The plot in Figure 32.3 displays a strong negative relationship between iron content and corrosion resistance, but it is not clear whether there is curvature in this relationship.

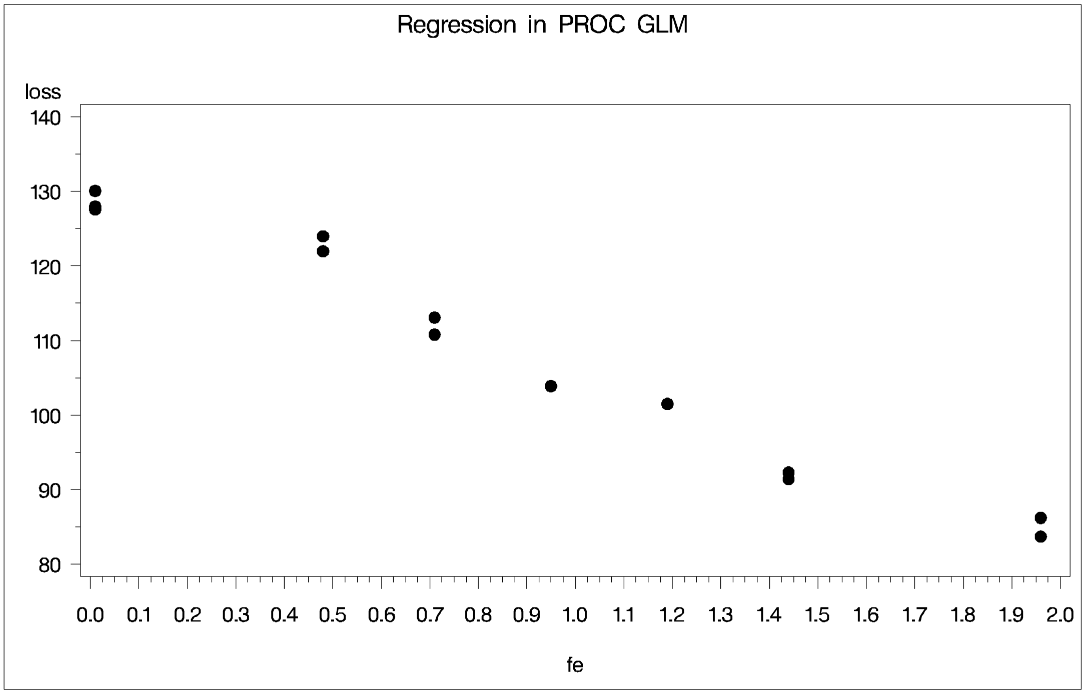


Figure 32.3. Plot of LOSS vs. FE

The following statements fit a quadratic regression model to the data. This enables you to estimate the linear relationship between iron content and corrosion resistance and test for the presence of a quadratic component. The intercept is automatically fit unless the NOINT option is specified.

```
proc glm;
   model loss=fe fe*fe;
run;
```

The CLASS statement is omitted because a regression line is being fitted. Unlike PROC REG, PROC GLM allows polynomial terms in the MODEL statement.

```
                           Regression in PROC GLM

                             The GLM Procedure

                  Number of Observations Read          13
                  Number of Observations Used          13
```

Figure 32.4. Class Level Information

The preliminary information in Figure 32.4 informs you that the GLM procedure has been invoked and states the number of observations in the data set. If the model involves classification variables, they are also listed here, along with their levels.

Figure 32.5 shows the overall ANOVA table and some simple statistics. The degrees of freedom can be used to check that the model is correct and that the data have been read correctly. The Model degrees of freedom for a regression is the number of parameters in the model minus 1. You are fitting a model with three parameters in this case,

$$\text{loss} \quad = \quad \beta_0 + \beta_1 \times (\text{fe}) + \beta_2 \times (\text{fe})^2 + \textit{error}$$

so the degrees of freedom are $3 - 1 = 2$. The Corrected Total degrees of freedom are always one less than the number of observations used in the analysis.

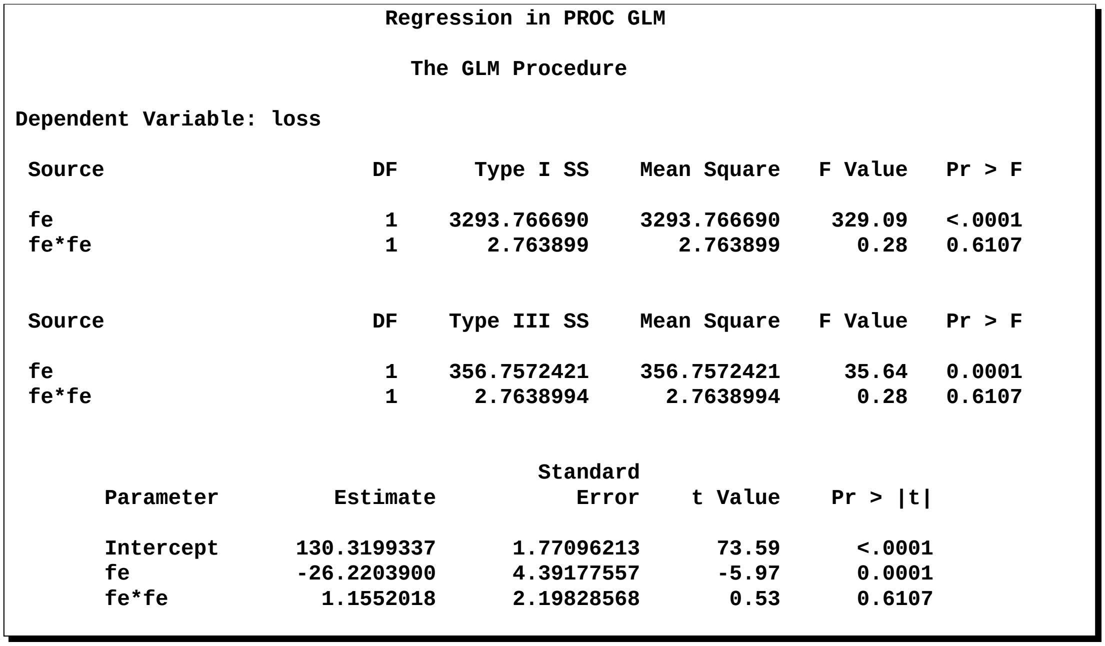
```
                             Regression in PROC GLM

                               The GLM Procedure

Dependent Variable: loss

                                            Sum of
 Source                      DF           Squares     Mean Square    F Value    Pr > F

 Model                        2       3296.530589     1648.265295     164.68    <.0001

 Error                       10        100.086334       10.008633

 Corrected Total             12       3396.616923


              R-Square     Coeff Var      Root MSE     loss Mean

              0.970534      2.907348      3.163642      108.8154
```

Figure 32.5. ANOVA Table

The R^2 indicates that the model accounts for 97% of the variation in LOSS. The coefficient of variation (C.V.), Root MSE (Mean Square for Error), and mean of the dependent variable are also listed.

The overall F test is significant $(F = 164.68, p < 0.0001)$, indicating that the model as a whole accounts for a significant amount of the variation in LOSS. Thus, it is appropriate to proceed to testing the effects.

Figure 32.6 contains tests of effects and parameter estimates. The latter are displayed by default when the model contains only continuous variables.



```
                              Regression in PROC GLM

                                The GLM Procedure

Dependent Variable: loss

 Source                      DF     Type I SS    Mean Square   F Value   Pr > F

 fe                           1   3293.766690    3293.766690    329.09   <.0001
 fe*fe                        1      2.763899       2.763899      0.28   0.6107


 Source                      DF   Type III SS    Mean Square   F Value   Pr > F

 fe                           1   356.7572421    356.7572421     35.64   0.0001
 fe*fe                        1     2.7638994      2.7638994      0.28   0.6107


                                            Standard
       Parameter          Estimate             Error    t Value    Pr > |t|

       Intercept       130.3199337        1.77096213      73.59      <.0001
       fe              -26.2203900        4.39177557      -5.97      0.0001
       fe*fe             1.1552018        2.19828568       0.53      0.6107
```

Figure 32.6. Tests of Effects and Parameter Estimates

The t tests provided are equivalent to the Type III F tests. The quadratic term is not significant $(F = 0.28, p = 0.6107; t = 0.53, p = 0.6107)$ and thus can be removed from the model; the linear term is significant $(F = 35.64, p = 0.0001; t = -5.97, p = 0.0001)$. This suggests that there is indeed a straight line relationship between loss and fe.

Fitting the model without the quadratic term provides more accurate estimates for β_0 and β_1. PROC GLM allows only one MODEL statement per invocation of the procedure, so the PROC GLM statement must be issued again. The statements used to fit the linear model are

```
proc glm;
   model loss=fe;
run;
```

Figure 32.7 displays the output produced by these statements. The linear term is still significant $(F = 352.27, p < 0.0001)$. The estimated model is now

$$\text{loss} = 129.79 - 24.02 \times \text{fe}$$

```
                               Regression in PROC GLM

                                 The GLM Procedure

Dependent Variable: loss

                                            Sum of
 Source                      DF            Squares     Mean Square    F Value    Pr > F

 Model                        1        3293.766690     3293.766690     352.27    <.0001

 Error                       11         102.850233        9.350021

 Corrected Total             12        3396.616923


              R-Square     Coeff Var      Root MSE     loss Mean

              0.969720      2.810063      3.057780      108.8154


 Source                      DF          Type I SS     Mean Square    F Value    Pr > F

 fe                           1        3293.766690     3293.766690     352.27    <.0001


 Source                      DF        Type III SS     Mean Square    F Value    Pr > F

 fe                           1        3293.766690     3293.766690     352.27    <.0001


                                                Standard
       Parameter         Estimate                  Error    t Value    Pr > |t|

       Intercept      129.7865993             1.40273671      92.52      <.0001
       fe             -24.0198934             1.27976715     -18.77      <.0001
```

Figure 32.7. Linear Model Output

Syntax

The following statements are available in PROC GLM.

PROC GLM *< options >* ;
  CLASS *variables < / option >* ;
  MODEL *dependents=independents < / options >* ;
  ABSORB *variables* ;
  BY *variables* ;
  FREQ *variable* ;
  ID *variables* ;
  WEIGHT *variable* ;
  CONTRAST *’label’ effect values < . . . effect values > < / options >* ;
  ESTIMATE *’label’ effect values < . . . effect values > < / options >* ;
  LSMEANS *effects < / options >* ;
  MANOVA *< test-options >< / detail-options >* ;

MEANS *effects* < */ options* > **;**
OUTPUT < **OUT=***SAS-data-set* >
keyword=names < . . . *keyword=names* > < */ option* > **;**
RANDOM *effects* < */ options* > **;**
REPEATED *factor-specification* < */ options* > **;**
TEST < **H=***effects* > **E=***effect* < */ options* > **;**

Although there are numerous statements and options available in PROC GLM, many applications use only a few of them. Often you can find the features you need by looking at an example or by quickly scanning through this section.

To use PROC GLM, the PROC GLM and MODEL statements are required. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run). If your model contains classification effects, the classification variables must be listed in a CLASS statement, and the CLASS statement must appear before the MODEL statement. In addition, if you use a CONTRAST statement in combination with a MANOVA, RANDOM, REPEATED, or TEST statement, the CONTRAST statement must be entered first in order for the contrast to be included in the MANOVA, RANDOM, REPEATED, or TEST analysis.

The following table summarizes the positional requirements for the statements in the GLM procedure.

Table 32.1. Positional Requirements for PROC GLM Statements

Statement	Must Appear Before the	Must Appear After the
ABSORB	first RUN statement	
BY	first RUN statement	
CLASS	MODEL statement	
CONTRAST	MANOVA, REPEATED, or RANDOM statement	MODEL statement
ESTIMATE		MODEL statement
FREQ	first RUN statement	
ID	first RUN statement	
LSMEANS		MODEL statement
MANOVA		CONTRAST or MODEL statement
MEANS		MODEL statement
MODEL	CONTRAST, ESTIMATE, LSMEANS, or MEANS statement	CLASS statement
OUTPUT		MODEL statement
RANDOM		CONTRAST or MODEL statement
REPEATED		CONTRAST, MODEL, or TEST statement
TEST	MANOVA or REPEATED statement	MODEL statement
WEIGHT	first RUN statement	

The following table summarizes the function of each statement (other than the PROC statement) in the GLM procedure:

Table 32.2. Statements in the GLM Procedure

Statement	Description
ABSORB	absorbs classification effects in a model
BY	specifies variables to define subgroups for the analysis
CLASS	declares classification variables
CONTRAST	constructs and tests linear functions of the parameters
ESTIMATE	estimates linear functions of the parameters
FREQ	specifies a frequency variable
ID	identifies observations on output
LSMEANS	computes least-squares (marginal) means
MANOVA	performs a multivariate analysis of variance
MEANS	computes and optionally compares arithmetic means
MODEL	defines the model to be fit
OUTPUT	requests an output data set containing diagnostics for each observation

Table 32.2. (continued)

Statement	Description
RANDOM	declares certain effects to be random and computes expected mean squares
REPEATED	performs multivariate and univariate repeated measures analysis of variance
TEST	constructs tests using the sums of squares for effects and the error term you specify
WEIGHT	specifies a variable for weighting observations

The rest of this section gives detailed syntax information for each of these statements, beginning with the PROC GLM statement. The remaining statements are covered in alphabetical order.

PROC GLM Statement

PROC GLM < *options* > **;**

The PROC GLM statement starts the GLM procedure. You can specify the following options in the PROC GLM statement:

ALPHA=*p*

specifies the level of significance p for $100(1-p)\%$ confidence intervals. The value must be between 0 and 1; the default value of $p = 0.05$ results in 95% intervals. This value is used as the default confidence level for limits computed by the following options.

Statement	Options
LSMEANS	CL
MEANS	CLM CLDIFF
MODEL	CLI CLM CLPARM
OUTPUT	UCL= LCL= UCLM= LCLM=

You can override the default in each of these cases by specifying the ALPHA= option for each statement individually.

DATA=*SAS-data-set*

names the SAS data set used by the GLM procedure. By default, PROC GLM uses the most recently created SAS data set.

MANOVA

requests the multivariate mode of eliminating observations with missing values. If any of the dependent variables have missing values, the procedure eliminates that observation from the analysis. The MANOVA option is useful if you use PROC GLM in interactive mode and plan to perform a multivariate analysis.

MULTIPASS

requests that PROC GLM reread the input data set when necessary, instead of writing the necessary values of dependent variables to a utility file. This option decreases disk space usage at the expense of increased execution times, and is useful only in rare situations where disk space is at an absolute premium.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters long, where n is a value between 20 and 200 characters. The default length is 20 characters.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of all classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use CONTRAST or ESTIMATE statements. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table shows how PROC GLM interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter

on the SORT procedure in the *SAS Procedures Guide*, and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUTSTAT=*SAS-data-set*

names an output data set that contains sums of squares, degrees of freedom, F statistics, and probability levels for each effect in the model, as well as for each CONTRAST that uses the overall residual or error mean square (MSE) as the denominator in constructing the F statistic. If you use the CANONICAL option in the MANOVA statement and do not use an M= specification in the MANOVA statement, the data set also contains results of the canonical analysis. See the section "Output Data Sets" on page 1840 for more information.

ABSORB Statement

ABSORB *variables* **;**

Absorption is a computational technique that provides a large reduction in time and memory requirements for certain types of models. The *variables* are one or more variables in the input data set.

For a main effect variable that does not participate in interactions, you can absorb the effect by naming it in an ABSORB statement. This means that the effect can be adjusted out before the construction and solution of the rest of the model. This is particularly useful when the effect has a large number of levels.

Several variables can be specified, in which case each one is assumed to be nested in the preceding variable in the ABSORB statement.

Note: When you use the ABSORB statement, the data set (or each BY group, if a BY statement appears) must be sorted by the variables in the ABSORB statement. The GLM procedure cannot produce predicted values or least-squares means (LS-means) or create an output data set of diagnostic values if an ABSORB statement is used. If the ABSORB statement is used, it must appear before the first RUN statement or it is ignored.

When you use an ABSORB statement and also use the INT option in the MODEL statement, the procedure ignores the option but computes the uncorrected total sum of squares (SS) instead of the corrected total sums of squares.

See the "Absorption" section on page 1799 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

Since sorting the data changes the order in which PROC GLM reads observations, the sorting order for the levels of the classification variables may be affected if you have also specified ORDER=DATA in the PROC GLM statement. This, in turn, affects specifications in CONTRAST and ESTIMATE statements.

If you specify the BY statement, it must appear before the first RUN statement or it is ignored. When you use a BY statement, the interactive features of PROC GLM are disabled.

When both BY and ABSORB statements are used, observations must be sorted first by the variables in the BY statement, and then by the variables in the ABSORB statement.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Contents*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < */ option* > **;**

The CLASS statement names the classification variables to be used in the model. Typical class variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. If you specify the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide*, and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

The GLM procedure displays a table summarizing the class variables and their levels, and you can use this to check the ordering of levels and, hence, of the corresponding parameters for main effects. If you need to check the ordering of parameters for interaction effects, use the E option in the MODEL, CONTRAST, ESTIMATE, and LSMEANS statements. See the "Parameterization of PROC GLM Models" section on page 1787 for more information.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

CONTRAST Statement

CONTRAST *'label' effect values* $<$ . . . *effect values* $>$ $<$ **/** *options* $>$ **;**

The CONTRAST statement enables you to perform custom hypothesis tests by specifying an $\mathbf{L}$ vector or matrix for testing the univariate hypothesis $\mathbf{L}\beta = 0$ or the multivariate hypothesis $\mathbf{LBM} = 0$. Thus, to use this feature you must be familiar with the details of the model parameterization that PROC GLM uses. For more information, see the "Parameterization of PROC GLM Models" section on page 1787. All of the elements of the $\mathbf{L}$ vector may be given, or if only certain portions of the $\mathbf{L}$ vector are given, the remaining elements are constructed by PROC GLM from the context (in a manner similar to rule 4 discussed in the "Construction of Least-Squares Means" section on page 1820).

There is no limit to the number of CONTRAST statements you can specify, but they must appear after the MODEL statement. In addition, if you use a CONTRAST statement and a MANOVA, REPEATED, or TEST statement, appropriate tests for contrasts are carried out as part of the MANOVA, REPEATED, or TEST analysis. If you use a CONTRAST statement and a RANDOM statement, the expected mean square of the contrast is displayed. As a result of these additional analyses, the CONTRAST statement must appear before the MANOVA, REPEATED, RANDOM, or TEST statement.

In the CONTRAST statement,

label identifies the contrast on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. The INTERCEPT effect can be used when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ vector associated with the effect.

You can specify the following options in the CONTRAST statement after a slash(/):

E

displays the entire $\mathbf{L}$ vector. This option is useful in confirming the ordering of parameters for specifying $\mathbf{L}$.

E=*effect*

specifies an error term, which must be one of the effects in the model. The procedure uses this effect as the denominator in F tests in univariate analysis. In addition, if you use a MANOVA or REPEATED statement, the procedure uses the effect specified by the E= option as the basis of the $\mathbf{E}$ matrix. By default, the procedure uses the overall residual or error mean square (MSE) as an error term.

ETYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to Type I, II, III, and IV tests, respectively) of the E= effect. If the E= option is specified and the ETYPE= option is not, the procedure uses the highest type computed in the analysis.

SINGULAR=*number*

tunes the estimability checking. If ABS$(\mathbf{L} - \mathbf{LH}) > C\times$*number* for any row in the contrast, then $\mathbf{L}$ is declared nonestimable. $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS$(\mathbf{L})$ except for rows where $\mathbf{L}$ is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

As stated previously, the CONTRAST statement enables you to perform custom hypothesis tests. If the hypothesis is testable in the univariate case, SS(H_0: $\mathbf{L}\beta = 0$) is computed as

$$(\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$. This is the sum of squares displayed on the analysis-of-variance table.

For multivariate testable hypotheses, the usual multivariate tests are performed using

$$\mathbf{H} = \mathrm{M}'(\mathbf{LB})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{LB})\mathrm{M}$$

where $\mathbf{B} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}$ is the matrix of multivariate responses or dependent variables. The degrees of freedom associated with the hypothesis is equal to the row rank of $\mathbf{L}$. The sum of squares computed in this situation is equivalent to the sum of squares computed using an $\mathbf{L}$ matrix with any row deleted that is a linear combination of previous rows.

Multiple-degree-of-freedom hypotheses can be specified by separating the rows of the $\mathbf{L}$ matrix with commas.

For example, for the model

```
proc glm;
   class A B;
   model Y=A B;
run;
```

with A at 5 levels and B at 2 levels, the parameter vector is

$$(\mu \;\; \alpha_1 \;\; \alpha_2 \;\; \alpha_3 \;\; \alpha_4 \;\; \alpha_5 \;\; \beta_1 \;\; \beta_2)$$

To test the hypothesis that the pooled A linear and A quadratic effect is zero, you can use the following $\mathbf{L}$ matrix:

$$\mathbf{L} = \begin{bmatrix} 0 & -2 & -1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & -1 & -2 & -1 & 2 & 0 & 0 \end{bmatrix}$$

The corresponding CONTRAST statement is

```
contrast 'A LINEAR & QUADRATIC'
         a -2 -1  0  1  2,
         a  2 -1 -2 -1  2;
```

If the first level of A is a control level and you want a test of control versus others, you can use this statement:

```
contrast 'CONTROL VS OTHERS'  a -1 0.25 0.25 0.25 0.25;
```

See the following discussion of the ESTIMATE statement and the "Specification of ESTIMATE Expressions" section on page 1801 for rules on specification, construction, distribution, and estimability in the CONTRAST statement.

ESTIMATE Statement

ESTIMATE *'label' effect values* < . . . *effect values* > < **/** *options* > **;**

The ESTIMATE statement enables you to estimate linear functions of the parameters by multiplying the vector $\mathbf{L}$ by the parameter estimate vector $\mathbf{b}$ resulting in $\mathbf{Lb}$. All of the elements of the $\mathbf{L}$ vector may be given, or, if only certain portions of the $\mathbf{L}$ vector are given, the remaining elements are constructed by PROC GLM from the context (in a manner similar to rule 4 discussed in the "Construction of Least-Squares Means" section on page 1820).

The linear function is checked for estimability. The estimate $\mathbf{Lb}$, where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$, is displayed along with its associated standard error, $\sqrt{\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'s^2}$, and t test. If you specify the CLPARM option in the MODEL statement (see page 1771), confidence limits for the true value are also displayed.

There is no limit to the number of ESTIMATE statements that you can specify, but they must appear after the MODEL statement. In the ESTIMATE statement,

label identifies the estimate on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. The INTERCEPT effect can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are the elements of the **L** vector associated with the preceding effect. For example,

```
estimate 'A1 VS A2' A  1 -1;
```

forms an estimate that is the difference between the parameters estimated for the first and second levels of the CLASS variable A.

You can specify the following options in the ESTIMATE statement after a slash:

DIVISOR=*number*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integer numerators. For example, you can use

```
estimate '1/3(A1+A2) - 2/3A3' a 1 1 -2 / divisor=3;
```

instead of

```
estimate '1/3(A1+A2) - 2/3A3' a 0.33333 0.33333 -0.66667;
```

E

displays the entire **L** vector. This option is useful in confirming the ordering of parameters for specifying **L**.

SINGULAR=*number*

tunes the estimability checking. If ABS$(\mathbf{L} - \mathbf{LH}) > C\times$*number*, then the **L** vector is declared nonestimable. **H** is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS$(\mathbf{L})$ except for rows where **L** is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

See also the "Specification of ESTIMATE Expressions" section on page 1801.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times.

The analysis produced using a FREQ statement reflects the expanded number of observations. For example, means and total degrees of freedom reflect the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set are identical. Each observation in the old data set is replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

If you specify the FREQ statement, it must appear before the first RUN statement or it is ignored.

ID Statement

ID *variables* **;**

When predicted values are requested as a MODEL statement option, values of the variables given in the ID statement are displayed beside each observed, predicted, and residual value for identification. Although there are no restrictions on the length of ID variables, PROC GLM may truncate the number of values listed in order to display them on one line. The GLM procedure displays a maximum of five ID variables.

If you specify the ID statement, it must appear before the first RUN statement or it is ignored.

LSMEANS Statement

LSMEANS *effects* < **/** *options* > **;**

Least-squares means (LS-means) are computed for each *effect* listed in the LSMEANS statement. You may specify only classification effects in the LSMEANS statement—that is, effects that contain only classification variables. You may also specify options to perform multiple comparisons. In contrast to the MEANS statement, the LSMEANS statement performs multiple comparisons on interactions as well as main effects.

LS-means are *predicted population margins*; that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs. Each LS-mean is computed as $\mathbf{L}'\mathbf{b}$ for a certain column vector $\mathbf{L}$, where $\mathbf{b}$ is the vector of parameter estimates—that is, the solution of the normal equations. For further information, see the section "Construction of Least-Squares Means" on page 1820.

Multiple effects can be specified in one LSMEANS statement, or multiple LSMEANS statements can be used, but they must all appear after the MODEL statement. For example,

```
proc glm;
   class A B;
   model Y=A B A*B;
   lsmeans A B A*B;
run;
```

LS-means are displayed for each level of the A, B, and A*B effects.

You can specify the following options in the LSMEANS statement after a slash:

ADJUST=BON
ADJUST=DUNNETT
ADJUST=SCHEFFE
ADJUST=SIDAK
ADJUST=SIMULATE <(*simoptions*)>
ADJUST=SMM | GT2
ADJUST=TUKEY
ADJUST=T

requests a multiple comparison adjustment for the *p*-values and confidence limits for the differences of LS-means. The ADJUST= option modifies the results of the TDIFF and PDIFF options; thus, if you omit the TDIFF or PDIFF option then the ADJUST= option has no effect. By default, PROC GLM analyzes all pairwise differences unless you specify ADJUST=DUNNETT, in which case PROC GLM analyzes all differences with a control level. The default is ADJUST=T, which really signifies no adjustment for multiple comparisons.

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in the "Multiple Comparisons" section on page 1806 and in Chapter 48, "The MULTTEST Procedure." When you specify ADJUST=TUKEY and your data are unbalanced, PROC GLM uses the approximation described in Kramer (1956) and identifies the adjustment as "Tukey-Kramer" in the results. Similarly, when you specify ADJUST=DUNNETT and the LS-means are correlated, PROC GLM uses the factor-analytic covariance approximation described in Hsu (1992) and identifies the adjustment as "Dunnett-Hsu" in the results. The preceding references also describe the SCHEFFE and SMM adjustments.

The SIMULATE adjustment computes the adjusted *p*-values from the simulated distribution of the maximum or maximum absolute value of a multivariate *t* random vector. The simulation estimates q, the true $(1-\alpha)$th quantile, where $1-\alpha$ is the confidence coefficient. The default α is the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You can change this value with the ALPHA= option in the LSMEANS statement.

The number of samples for the SIMULATE adjustment is set so that the tail area for the simulated q is within a certain *accuracy radius* γ of $1-\alpha$ with an *accuracy confidence* of $100(1-\epsilon)$%. In equation form,

$$P(|F(\hat{q}) - (1-\alpha)| \le \gamma) \quad = \quad 1-\epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; refer to Edwards and Berry (1987) for details. By default, γ = 0.005 and ϵ = 0.01 so that the tail area of $\hat{q}$ is within 0.005 of 0.95 with 99% confidence.

You can specify the following *simoptions* in parentheses after the ADJUST=SIMULATE option.

ACC=*value* specifies the target accuracy radius γ of a $100(1-\epsilon)$% confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The default value is ACC=0.005. Note that, if you also specify the CVADJUST *simoption*, then the actual accuracy radius will probably be substantially less than this target.

CVADJUST specifies that the quantile should be estimated by the control variate adjustment method of Hsu and Nelson (1998) instead of simply as the quantile of the simulated sample. Specifying the CVADJUST option typically has the effect of significantly reducing the accuracy radius γ of a $100 \times (1-\epsilon)$% confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The control-variate-adjusted quantile estimate takes roughly twice as long to compute, but it is typically much more accurate than the sample quantile.

EPS=*value* specifies the value ϵ for a $100 \times (1-\epsilon)$% confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The default value for the accuracy confidence is 99%, corresponding to EPS=0.01.

NSAMP=*n* specifies the sample size for the simulation. By default, n is set based on the values of the target accuracy radius γ and accuracy confidence $100 \times (1-\epsilon)$true probability content of the estimated $(1-\alpha)$th quantile. With the default values for γ, ϵ, and α (0.005, 0.01, and 0.05, respectively), NSAMP=12604 by default.

REPORT specifies that a report on the simulation should be displayed, including a listing of the parameters, such as γ, ϵ, and α as well as an analysis of various methods for estimating or approximating the quantile.

SEED=*number* specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

ALPHA=*p*

specifies the level of significance p for $100(1-p)$% confidence intervals. This option is useful only if you also specify the CL option, and, optionally, the PDIFF option. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified, This value is used to set the endpoints for confidence intervals for the individual means as well as for differences between means.

AT *variable* = *value*
AT (*variable-list*) = (*value-list*)
AT MEANS

enables you to modify the values of the covariates used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to set the covariates to whatever values you consider interesting. For more information, see the section "Setting Covariate Values" on page 1821.

BYLEVEL

requests that PROC GLM process the OM data set by each level of the LS-mean effect in question. For more details, see the entry for the OM option in this section.

CL

requests confidence limits for the individual LS-means. If you specify the PDIFF option, confidence limits for differences between means are produced as well. You can control the confidence level with the ALPHA= option. Note that, if you specify an ADJUST= option, the confidence limits for the differences are adjusted for multiple inference but the confidence intervals for individual means are **not** adjusted.

COV

includes variances and covariances of the LS-means in the output data set specified in the OUT= option in the LSMEANS statement. Note that this is the covariance matrix for the LS-means themselves, not the covariance matrix for the differences between the LS-means, which is used in the PDIFF computations. If you omit the OUT= option, the COV option has no effect. When you specify the COV option, you can specify only one effect in the LSMEANS statement.

E

displays the coefficients of the linear functions used to compute the LS-means.

E=*effect*

specifies an effect in the model to use as an error term. The procedure uses the mean square for the *effect* as the error mean square when calculating estimated standard errors (requested with the STDERR option) and probabilities (requested with the STDERR, PDIFF, or TDIFF option). Unless you specify STDERR, PDIFF or TDIFF, the E= option is ignored. By default, if you specify the STDERR, PDIFF, or TDIFF option and do not specify the E= option, the procedure uses the error mean square for calculating standard errors and probabilities.

ETYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to Type I, II, III, and IV tests, respectively) of the E= effect. If you specify the E= option but not the ETYPE= option, the highest type computed in the analysis is used. If you omit the E= option, the ETYPE= option has no effect.

NOPRINT

suppresses the normal display of results from the LSMEANS statement. This option is useful when an output data set is created with the OUT= option in the LSMEANS statement.

OBSMARGINS
OM

specifies a potentially different weighting scheme for computing LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the input data set. For more information, see the section “Changing the Weighting Scheme” on page 1822.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across the entire data set, the procedure computes separate margins for each level of the LS-mean effect in question. The resulting LS-means are actually equal to raw means in this case. If you specify the BYLEVEL option, it disables the AT option.

OUT=*SAS-data-set*

creates an output data set that contains the values, standard errors, and, optionally, the covariances (see the COV option) of the LS-means. For more information, see the “Output Data Sets” section on page 1840.

PDIFF*<=difftype>*

requests that p-values for differences of the LS-means be produced. The optional *difftype* specifies which differences to display. Possible values for *difftype* are ALL, CONTROL, CONTROLL, and CONTROLU. The ALL value requests all pairwise differences, and it is the default. The CONTROL value requests the differences with a control that, by default, is the first level of each of the specified LS-mean effects.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are class variables, each having two levels, ’1’ and ’2’, the following LSMEANS statement specifies the ’1’ ’2’ level of A*B and the ’2’ ’1’ level of B*C as controls:

```
lsmeans A*B B*C / pdiff=control('1' '2', '2' '1');
```

For multiple effect situations such as this one, the ordering of the list is significant, and you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL difftype. For one-tailed results, use either the CONTROLL or CONTROLU difftype.

- PDIFF=CONTROLL tests whether the noncontrol levels are less than the control; you declare a noncontrol level to be significantly less than the control if the associated upper confidence limit for the noncontrol level minus the control is less than zero, and you ignore the associated lower confidence limits (which are set to minus infinity).
- PDIFF=CONTROLU tests whether the noncontrol levels are greater than the control; you declare a noncontrol level to be significantly greater than the control if the associated lower confidence limit for the noncontrol level minus the control is greater than zero, and you ignore the associated upper confidence limits (which are set to infinity).

The default multiple comparisons adjustment for each *difftype* is shown in the following table.

difftype	**Default ADJUST=**
Not specified	T
ALL	TUKEY
CONTROL CONTROLL CONTROLU	DUNNETT

If no *difftype* is specified, the default for the ADJUST= option is T (that is, no adjustment); for PDIFF=ALL, ADJUST=TUKEY is the default; in all other instances, the default value for the ADJUST= option is DUNNETT. If there is a conflict between the PDIFF= and ADJUST= options, the ADJUST= option takes precedence.

For example, in order to compute one-sided confidence limits for differences with a control, adjusted according to Dunnett's procedure, the following statements are equivalent:

```
   lsmeans Treatment / pdiff=controll cl;
   lsmeans Treatment / pdiff=controll cl adjust=dunnett;
```

SLICE = *fixed-effect*
SLICE = (*fixed-effects***)**

specifies effects within which to test for differences between interaction LS-mean effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant and you want to test for the effect of A within each level of B. The appropriate LSMEANS statement is

```
   lsmeans A*B / slice=B;
```

This code tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and using them to form an *F*-test as performed by the CONTRAST statement.

SINGULAR=*number*

tunes the estimability checking. If ABS$(\mathbf{L} - \mathbf{LH}) > C \times$*number* for any row, then $\mathbf{L}$ is declared nonestimable. $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS$(\mathbf{L})$ except for rows where $\mathbf{L}$ is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

STDERR

produces the standard error of the LS-means and the probability level for the hypothesis H_0: LS-mean $= 0$.

TDIFF

produces the t values for all hypotheses H_0: LS-mean(i) $=$ LS-mean(j) and the corresponding probabilities.

MANOVA Statement

MANOVA < *test-options* >< / *detail-options* > **;**

If the MODEL statement includes more than one dependent variable, you can perform multivariate analysis of variance with the MANOVA statement. The *test-options* define which effects to test, while the *detail-options* specify how to execute the tests and what results to display.

When a MANOVA statement appears before the first RUN statement, PROC GLM enters a multivariate mode with respect to the handling of missing values; in addition to observations with missing independent variables, observations with *any* missing dependent variables are excluded from the analysis. If you want to use this mode of handling missing values and do not need any multivariate analyses, specify the MANOVA option in the PROC GLM statement.

If you use both the CONTRAST and MANOVA statements, the MANOVA statement must appear after the CONTRAST statement.

Test Options

The following options can be specified in the MANOVA statement as *test-options* in order to define which multivariate tests to perform.

H=*effects* | **INTERCEPT** | **_ALL_**

specifies effects in the preceding model to use as hypothesis matrices. For each $\mathbf{H}$ matrix (the SSCP matrix associated with an effect), the H= specification displays the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$ (where $\mathbf{E}$ is the matrix associated with the error effect), Hotelling-Lawley trace, Pillai's trace, Wilks' criterion, and Roy's maximum root criterion. By default, these statistics are tested with approximations based on the F distribution. To test them with exact (but computationally intensive) calculations, use the MSTAT=EXACT option.

Use the keyword INTERCEPT to produce tests for the intercept. To produce tests for all effects listed in the MODEL statement, use the keyword _ALL_ in place of a list of effects. For background and further details, see the "Multivariate Analysis of Variance" section on page 1823.

E=*effect*

specifies the error effect. If you omit the E= specification, the GLM procedure uses the error SSCP (residual) matrix from the analysis.

M=*equation,. . .,equation* | **(***row-of-matrix,. . .,row-of-matrix***)**

specifies a transformation matrix for the dependent variables listed in the MODEL statement. The equations in the M= specification are of the form

$$\begin{aligned} c_1 \times \textit{dependent-variable} &\pm c_2 \times \textit{dependent-variable} \\ \cdots &\pm c_n \times \textit{dependent-variable} \end{aligned}$$

where the c_i values are coefficients for the various *dependent-variables*. If the value of a given c_i is 1, it can be omitted; in other words $1 \times Y$ is the same as Y.

Equations should involve two or more dependent variables. For sample syntax, see the "Examples" section on page 1762.

Alternatively, you can input the transformation matrix directly by entering the elements of the matrix with commas separating the rows and parentheses surrounding the matrix. When this alternate form of input is used, the number of elements in each row must equal the number of dependent variables. Although these combinations actually represent the columns of the $\mathbf{M}$ matrix, they are displayed by rows.

When you include an M= specification, the analysis requested in the MANOVA statement is carried out for the variables defined by the equations in the specification, not the original dependent variables. If you omit the M= option, the analysis is performed for the original dependent variables in the MODEL statement.

If an M= specification is included without either the MNAMES= or PREFIX= option, the variables are labeled MVAR1, MVAR2, and so forth, by default. For further information, see the "Multivariate Analysis of Variance" section on page 1823.

MNAMES=*names*

provides names for the variables defined by the equations in the M= specification. Names in the list correspond to the M= equations or to the rows of the $\mathbf{M}$ matrix (as it is entered).

PREFIX=*name*

is an alternative means of identifying the transformed variables defined by the M= specification. For example, if you specify PREFIX=DIFF, the transformed variables are labeled DIFF1, DIFF2, and so forth.

Detail Options

You can specify the following options in the MANOVA statement after a slash as *detail-options*.

CANONICAL

displays a canonical analysis of the $\mathbf{H}$ and $\mathbf{E}$ matrices (transformed by the $\mathbf{M}$ matrix, if specified) instead of the default display of characteristic roots and vectors.

ETYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to Type I, II, III, and IV tests, respectively) of the **E** matrix, the SSCP matrix associated with the E= effect. You need this option if you use the E= specification to specify an error effect other than residual error and you want to specify the type of sums of squares used for the effect. If you specify ETYPE=n, the corresponding test must have been performed in the MODEL statement, either by options SSn, En, or the default Type I and Type III tests. By default, the procedure uses an ETYPE= value corresponding to the highest type (largest n) used in the analysis.

HTYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to Type I, II, III, and IV tests, respectively) of the **H** matrix. See the ETYPE= option for more details.

MSTAT=FAPPROX
MSTAT=EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact *p*-values for three of the four tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root) and an improved F-approximation for the fourth (Pillai's Trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's Greatest Root, computations for the exact *p*-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information on the results of MSTAT=EXACT, see the "Multivariate Analysis of Variance" section on page 1823.

ORTH

requests that the transformation matrix in the M= specification of the MANOVA statement be orthonormalized by rows before the analysis.

PRINTE

displays the error SSCP matrix $\mathbf{E}$. If the $\mathbf{E}$ matrix is the error SSCP (residual) matrix from the analysis, the partial correlations of the dependent variables given the independent variables are also produced.

For example, the statement

```
manova / printe;
```

displays the error SSCP matrix and the partial correlation matrix computed from the error SSCP matrix.

PRINTH

displays the hypothesis SSCP matrix $\mathbf{H}$ associated with each effect specified by the H= specification.

SUMMARY

produces analysis-of-variance tables for each dependent variable. When no $\mathbf{M}$ matrix is specified, a table is displayed for each original dependent variable from the MODEL statement; with an $\mathbf{M}$ matrix other than the identity, a table is displayed for each transformed variable defined by the $\mathbf{M}$ matrix.

Examples

The following statements provide several examples of using a MANOVA statement.

```
proc glm;
   class A B;
   model Y1-Y5=A B(A) / nouni;
   manova h=A e=B(A) / printh printe htype=1 etype=1;
   manova h=B(A) / printe;
   manova h=A e=B(A) m=Y1-Y2,Y2-Y3,Y3-Y4,Y4-Y5
          prefix=diff;
   manova h=A e=B(A) m=(1 -1  0  0  0,
                        0  1 -1  0  0,
                        0  0  1 -1  0,
                        0  0  0  1 -1) prefix=diff;
run;
```

Since this MODEL statement requests no options for type of sums of squares, the procedure uses Type I and Type III sums of squares. The first MANOVA statement specifies A as the hypothesis effect and B(A) as the error effect. As a result of the PRINTH option, the procedure displays the hypothesis SSCP matrix associated with the A effect; and, as a result of the PRINTE option, the procedure displays the error SSCP matrix associated with the B(A) effect. The option HTYPE=1 specifies a Type I **H** matrix, and the option ETYPE=1 specifies a Type I **E** matrix.

The second MANOVA statement specifies B(A) as the hypothesis effect. Since no error effect is specified, PROC GLM uses the error SSCP matrix from the analysis as the **E** matrix. The PRINTE option displays this **E** matrix. Since the **E** matrix is the error SSCP matrix from the analysis, the partial correlation matrix computed from this matrix is also produced.

The third MANOVA statement requests the same analysis as the first MANOVA statement, but the analysis is carried out for variables transformed to be successive differences between the original dependent variables. The option PREFIX=DIFF labels the transformed variables as DIFF1, DIFF2, DIFF3, and DIFF4.

Finally, the fourth MANOVA statement has the identical effect as the third, but it uses an alternative form of the M= specification. Instead of specifying a set of equations, the fourth MANOVA statement specifies rows of a matrix of coefficients for the five dependent variables.

As a second example of the use of the M= specification, consider the following:

```
proc glm;
   class group;
   model dose1-dose4=group / nouni;
   manova h = group
          m = -3*dose1 -   dose2 +   dose3 + 3*dose4,
                 dose1 -   dose2 -   dose3 +   dose4,
                -dose1 + 3*dose2 - 3*dose3 +   dose4
          mnames = Linear Quadratic Cubic
          / printe;
run;
```

The M= specification gives a transformation of the dependent variables dose1 through dose4 into orthogonal polynomial components, and the MNAMES= option labels the transformed variables LINEAR, QUADRATIC, and CUBIC, respectively. Since the PRINTE option is specified and the default residual matrix is used as an error term, the partial correlation matrix of the orthogonal polynomial components is also produced.

MEANS Statement

MEANS *effects* < **/** *options* > **;**

Within each group corresponding to each effect specified in the MEANS statement, PROC GLM computes the arithmetic means and standard deviations of all continuous variables in the model (both dependent and independent). You may specify only classification effects in the MEANS statement—that is, effects that contain only classification variables.

Note that the arithmetic means are not adjusted for other effects in the model; for adjusted means, see the "LSMEANS Statement" section on page 1753. If you use a WEIGHT statement, PROC GLM computes weighted means; see the "Weighted Means" section on page 1820.

You may also specify options to perform multiple comparisons. However, the MEANS statement performs multiple comparisons only for main effect means; for multiple comparisons of interaction means, see the "LSMEANS Statement" section on page 1753.

You can use any number of MEANS statements, provided that they appear after the MODEL statement. For example, suppose A and B each have two levels. Then, if you use the following statements

```
proc glm;
   class A B;
   model Y=A B A*B;
   means A B / tukey;
   means A*B;
run;
```

the means, standard deviations, and Tukey's multiple comparisons tests are displayed for each level of the main effects A and B, and just the means and standard deviations are displayed for each of the four combinations of levels for A*B. Since multiple comparisons tests apply only to main effects, the single MEANS statement

```
means A B A*B / tukey;
```

produces the same results.

PROC GLM does not compute means for interaction effects containing continuous variables. Thus, if you have the model

```
class A;
model Y=A X A*X;
```

then the effects X and A*X cannot be used in the MEANS statement. However, if you specify the effect A in the means statement

```
means A;
```

then PROC GLM, by default, displays within-A arithmetic means of both Y and X. Use the DEPONLY option to display means of only the dependent variables.

```
means A / deponly;
```

If you use a WEIGHT statement, PROC GLM computes weighted means and estimates their variance as inversely proportional to the corresponding sum of weights (see the "Weighted Means" section on page 1820). However, note that the statistical interpretation of multiple comparison tests for weighted means is not well understood. See the "Multiple Comparisons" section on page 1806 for formulas. The following table summarizes categories of options available in the MEANS statement.

Task	Available options
Modify output	DEPONLY
Perform multiple comparison tests	BON DUNCAN DUNNETT DUNNETTL DUNNETTU GABRIEL GT2 LSD REGWQ SCHEFFE SIDAK SMM SNK T TUKEY WALLER
Specify additional details for multiple comparison tests	ALPHA= CLDIFF CLM E= ETYPE= HTYPE= KRATIO= LINES NOSORT
Test for homogeneity of variances	HOVTEST
Compensate for heterogeneous variances	WELCH

These options are described in the following list.

ALPHA=*p*

specifies the level of significance for comparisons among the means. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You can specify any value greater than 0 and less than 1.

BON

performs Bonferroni t tests of differences between means for all main effect means in the MEANS statement. See the CLDIFF and LINES options for a discussion of how the procedure displays results.

CLDIFF

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, GT2, T, LSD, and TUKEY options as confidence intervals for all pairwise differences between means, and the results of the DUNNETT, DUNNETTU, and DUNNETTL options as confidence intervals for differences with the control. The CLDIFF option is the default for unequal cell sizes unless the DUNCAN, REGWQ, SNK, or WALLER option is specified.

CLM

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, T, and LSD options as intervals for the mean of each level of the variables specified in the MEANS statement. For all options except GABRIEL, the intervals are confidence intervals for the true means. For the GABRIEL option, they are *comparison intervals* for comparing means pairwise: in this case, if the intervals corresponding to two means overlap, then the difference between them is insignificant according to Gabriel's method.

DEPONLY

displays only means for the dependent variables. By default, PROC GLM produces means for all continuous variables, including continuous independent variables.

DUNCAN

performs Duncan's multiple range test on all main effect means given in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

DUNNETT < **(***formatted-control-values***)** >

performs Dunnett's two-tailed t test, testing if any treatments are significantly different from a single control for all main effects means in the MEANS statement.

To specify which level of the effect is the control, enclose the formatted value in quotes in parentheses after the keyword. If more than one effect is specified in the MEANS statement, you can use a list of control values within the parentheses. By default, the first level of the effect is used as the control. For example,

```
means A  / dunnett('CONTROL');
```

where CONTROL is the formatted control value of A. As another example,

```
means A B C / dunnett('CNTLA' 'CNTLB' 'CNTLC');
```

where CNTLA, CNTLB, and CNTLC are the formatted control values for A, B, and C, respectively.

DUNNETTL < **(***formatted-control-value***)** >

performs Dunnett's one-tailed t test, testing if any treatment is significantly less than the control. Control level information is specified as described for the DUNNETT option.

DUNNETTU < **(***formatted-control-value***)** >

performs Dunnett's one-tailed t test, testing if any treatment is significantly greater than the control. Control level information is specified as described for the DUNNETT option.

E=*effect*

specifies the error mean square used in the multiple comparisons. By default, PROC GLM uses the overall residual or error mean square (MS). The effect specified with the E= option must be a term in the model; otherwise, the procedure uses the residual MS.

ETYPE=*n*

specifies the type of mean square for the error effect. When you specify E=*effect*, you may need to indicate which type (1, 2, 3, or 4) of MS is to be used. The n value must be one of the types specified in or implied by the MODEL statement. The default MS type is the highest type used in the analysis.

GABRIEL

performs Gabriel's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

GT2

see the SMM option.

HOVTEST
HOVTEST=BARTLETT
HOVTEST=BF
HOVTEST=LEVENE < **(TYPE= ABS | SQUARE)**>
HOVTEST=OBRIEN < **(W=*number*)**>

requests a homogeneity of variance test for the groups defined by the MEANS effect. You can optionally specify a particular test; if you do not specify a test, Levene's test (Levene 1960) with TYPE=SQUARE is computed. Note that this option is ignored unless your MODEL statement specifies a simple one-way model.

The HOVTEST=BARTLETT option specifies Bartlett's test (Bartlett 1937), a modification of the normal-theory likelihood ratio test.

The HOVTEST=BF option specifies Brown and Forsythe's variation of Levene's test (Brown and Forsythe 1974).

The HOVTEST=LEVENE option specifies Levene's test (Levene 1960), which is widely considered to be the standard homogeneity of variance test. You can use the TYPE= option in parentheses to specify whether to use the absolute residuals (TYPE=ABS) or the squared residuals (TYPE=SQUARE) in Levene's test. TYPE=SQUARE is the default.

The HOVTEST=OBRIEN option specifies O'Brien's test (O'Brien 1979), which is basically a modification of HOVTEST=LEVENE(TYPE=SQUARE). You can use the W= option in parentheses to tune the variable to match the suspected kurtosis of the underlying distribution. By default, W=0.5, as suggested by O'Brien (1979; 1981).

See the "Homogeneity of Variance in One-Way Models" section on page 1818 for more details on these methods. Example 32.10 on page 1892 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

HTYPE=*n*

specifies the MS type for the hypothesis MS. The HTYPE= option is needed only when the WALLER option is specified. The default HTYPE= value is the highest type used in the model.

KRATIO=*value*

specifies the Type 1/Type 2 error seriousness ratio for the Waller-Duncan test. Reasonable values for the KRATIO= option are 50, 100, 500, which roughly correspond for the two-level case to ALPHA levels of 0.1, 0.05, and 0.01, respectively. By default, the procedure uses the value of 100.

LINES

presents results of the BON, DUNCAN, GABRIEL, REGWQ, SCHEFFE, SIDAK, SMM, GT2, SNK, T, LSD, TUKEY, and WALLER options by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding means. The LINES option is appropriate for equal cell sizes, for which it is the default. The LINES option is also the default if the DUNCAN, REGWQ, SNK, or WALLER option is specified, or if there are only two cells of unequal size. The LINES option cannot be used in combination with the DUNNETT, DUNNETTL, or DUNNETTU option. In addition, the procedure has a restriction that no more than 24 overlapping groups of means can exist. If a mean belongs to more than 24 groups, the procedure issues an error message. You can either reduce the number of levels of the variable or use a multiple comparison test that allows the CLDIFF option rather than the LINES option.

Note: If the cell sizes are unequal, the harmonic mean of the cell sizes is used to compute the critical ranges. This approach is reasonable if the cell sizes are not too different, but it can lead to liberal tests if the cell sizes are highly disparate. In this case, you should not use the LINES option for displaying multiple comparisons results; use the TUKEY and CLDIFF options instead.

LSD

see the T option.

NOSORT

prevents the means from being sorted into descending order when the CLDIFF or CLM option is specified.

REGWQ

performs the Ryan-Einot-Gabriel-Welsch multiple range test on all main effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

SCHEFFE

performs Scheffé's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SIDAK

performs pairwise t tests on differences between means with levels adjusted according to Sidak's inequality for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SMM
GT2

performs pairwise comparisons based on the studentized maximum modulus and Sidak's uncorrelated-t inequality, yielding Hochberg's GT2 method when sample sizes are unequal, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SNK

performs the Student-Newman-Keuls multiple range test on all main effect means in the MEANS statement. See the LINES option for discussions of how the procedure displays results.

T
LSD

performs pairwise t tests, equivalent to Fisher's least-significant-difference test in the case of equal cell sizes, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

TUKEY

performs Tukey's studentized range test (HSD) on all main effect means in the MEANS statement. (When the group sizes are different, this is the Tukey-Kramer test.) See the CLDIFF and LINES options for discussions of how the procedure displays results.

WALLER

performs the Waller-Duncan k-ratio t test on all main effect means in the MEANS statement. See the KRATIO= and HTYPE= options for information on controlling details of the test, and the LINES option for a discussion of how the procedure displays results.

WELCH

requests the variance-weighted one-way ANOVA of Welch (1951). This alternative to the usual analysis of variance for a one-way model is robust to the assumption of equal within-group variances. This option is ignored unless your MODEL statement specifies a simple one-way model.

Note that using the WELCH option merely produces one additional table consisting of Welch's ANOVA. It does not affect all of the other tests displayed by the GLM procedure, which still require the assumption of equal variance for exact validity.

See the "Homogeneity of Variance in One-Way Models" section on page 1818 for more details on Welch's ANOVA. Example 32.10 on page 1892 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

MODEL Statement

MODEL *dependents=independents* < / *options* > **;**

The MODEL statement names the dependent variables and independent effects. The syntax of effects is described in the "Specification of Effects" section on page 1784. For any model effect involving classification variables (interactions as well as main effects), the number of levels can not exceed 32,767. If no independent effects are specified, only an intercept term is fit. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run).

The following table summarizes options available in the MODEL statement.

Task	Options
Produce tests for the intercept	INTERCEPT
Omit the intercept parameter from model	NOINT
Produce parameter estimates	SOLUTION
Produce tolerance analysis	TOLERANCE
Suppress univariate tests and output	NOUNI
Display estimable functions	E E1 E2 E3 E4 ALIASING
Control hypothesis tests performed	SS1 SS2 SS3 SS4
Produce confidence intervals	ALPHA= CLI CLM CLPARM
Display predicted and residual values	P
Display intermediate calculations	INVERSE XPX
Tune sensitivity	SINGULAR= ZETA=

These options are described in the following list.

ALIASING

specifies that the estimable functions should be displayed as an *aliasing structure*, for which each row says which linear combination of the parameters is estimated by each estimable function; also, adds a column of the same information to the table of parameter estimates, giving for each parameter the expected value of the estimate associated with that parameter. This option is most useful in fractional factorial experiments that can be analyzed without a CLASS statement.

ALPHA=*p*

specifies the level of significance p for $100(1-p)$% confidence intervals. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement, or 0.05 if that option is not specified. You may use values between 0 and 1.

CLI

produces confidence limits for individual predicted values for each observation. The CLI option is ignored if the CLM option is also specified.

CLM

produces confidence limits for a mean predicted value for each observation.

CLPARM

produces confidence limits for the parameter estimates (if the SOLUTION option is also specified) and for the results of all ESTIMATE statements.

E

displays the general form of all estimable functions. This is useful for determining the order of parameters when writing CONTRAST and ESTIMATE statements.

E1

displays the Type I estimable functions for each effect in the model and computes the corresponding sums of squares.

E2

displays the Type II estimable functions for each effect in the model and computes the corresponding sums of squares.

E3

displays the Type III estimable functions for each effect in the model and computes the corresponding sums of squares.

E4

displays the Type IV estimable functions for each effect in the model and computes the corresponding sums of squares.

INTERCEPT
INT

produces the hypothesis tests associated with the intercept as an effect in the model. By default, the procedure includes the intercept in the model but does not display associated tests of hypotheses. Except for producing the uncorrected total sum of squares instead of the corrected total sum of squares, the INT option is ignored when you use an ABSORB statement.

INVERSE
I

displays the augmented inverse (or generalized inverse) $\mathbf{X'X}$ matrix:

$$\begin{bmatrix} (X'X)^{-} & (X'X)^{-}X'Y \\ Y'X(X'X)^{-} & Y'Y - Y'X(X'X)^{-}X'Y \end{bmatrix}$$

The upper left-hand corner is the generalized inverse of $\mathbf{X'X}$, the upper right-hand corner is the parameter estimates, and the lower right-hand corner is the error sum of squares.

NOINT

omits the intercept parameter from the model.

NOUNI

suppresses the display of univariate statistics. You typically use the NOUNI option with a multivariate or repeated measures analysis of variance when you do not need the standard univariate results. The NOUNI option in a MODEL statement does not affect the univariate output produced by the REPEATED statement.

P

displays observed, predicted, and residual values for each observation that does not contain missing values for independent variables. The Durbin-Watson statistic is also displayed when the P option is specified. The PRESS statistic is also produced if either the CLM or CLI option is specified.

SINGULAR=*number*

tunes the sensitivity of the regression routine to linear dependencies in the design. If a diagonal pivot element is less than $C \times number$ as PROC GLM sweeps the $\mathbf{X'X}$ matrix, the associated design column is declared to be linearly dependent with previous columns, and the associated parameter is zeroed.

The C value adjusts the check to the relative scale of the variable. The C value is equal to the corrected sum of squares for the variable, unless the corrected sum of squares is 0, in which case C is 1. If you specify the NOINT option but not the ABSORB statement, PROC GLM uses the uncorrected sum of squares instead.

The default value of the SINGULAR= option, 10^{-7}, may be too small, but this value is necessary in order to handle the high-degree polynomials used in the literature to compare regression routines.

SOLUTION

produces a solution to the normal equations (parameter estimates). PROC GLM displays a solution by default when your model involves no classification variables, so you need this option only if you want to see the solution for models with classification effects.

SS1

displays the sum of squares associated with Type I estimable functions for each effect. These are also displayed by default.

SS2

displays the sum of squares associated with Type II estimable functions for each effect.

SS3

displays the sum of squares associated with Type III estimable functions for each effect. These are also displayed by default.

SS4

displays the sum of squares associated with Type IV estimable functions for each effect.

TOLERANCE

displays the tolerances used in the SWEEP routine. The tolerances are of the form C/USS or C/CSS, as described in the discussion of the SINGULAR= option. The tolerance value for the intercept is not divided by its uncorrected sum of squares.

XPX

displays the augmented $\mathbf{X'X}$ crossproducts matrix:

$$\begin{bmatrix} X'X & X'Y \\ Y'X & Y'Y \end{bmatrix}$$

ZETA=*value*

tunes the sensitivity of the check for estimability for Type III and Type IV functions. Any element in the estimable function basis with an absolute value less than the ZETA= option is set to zero. The default value for the ZETA= option is 10^{-8}.

Although it is possible to generate data for which this absolute check can be defeated, the check suffices in most practical examples. Additional research needs to be performed to make this check relative rather than absolute.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > *keyword=names*
< . . . *keyword=names* > < / *option* > **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated after fitting the model. At least one specification of the form *keyword=names* is required.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of diagnostic measures that are calculated for each observation in the data set. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets).

Details on the specifications in the OUTPUT statement follow.

keyword=names

specifies the statistics to include in the output data set and provides names to the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable or variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains that statistic for the first dependent variable listed in the MODEL statement; the second variable contains the statistic for the second dependent variable in the MODEL statement, and so on. The list of variables following the equal sign can be shorter than the list of dependent variables in the MODEL statement. In this case, the procedure creates the new names in order of the dependent variables in the MODEL statement. See the “Examples” section on page 1775.

The keywords allowed and the statistics they represent are as follows:

COOKD Cook’s D influence statistic

COVRATIO standard influence of observation on covariance of parameter estimates

DFFITS standard influence of observation on predicted value

H leverage, $h_i = x_i(\mathbf{X}'\mathbf{X})^{-1}x_i'$

LCL lower bound of a $100(1-p)$% confidence interval for an individual prediction. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set then $p = 0.05$ by default, resulting in the lower bound for a 95% confidence interval. The interval also depends on the variance of the error, as well as the variance of the parameter estimates. For the corresponding upper bound, see the UCL keyword.

LCLM lower bound of a $100(1-p)$% confidence interval for the expected value (mean) of the predicted value. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set then $p = 0.05$ by default, resulting in the lower bound for a 95% confidence interval. For the corresponding upper bound, see the UCLM keyword.

PREDICTED | P predicted values

PRESS residual for the ith observation that results from dropping it and predicting it on the basis of all other observations. This is the residual divided by $(1-h_i)$ where h_i is the leverage, defined previously.

RESIDUAL | R residuals, calculated as ACTUAL − PREDICTED

RSTUDENT a studentized residual with the current observation deleted

STDI standard error of the individual predicted value

STDP standard error of the mean predicted value

STDR standard error of the residual

STUDENT studentized residuals, the residual divided by its standard error

UCL — upper bound of a $100(1-p)$% confidence interval for an individual prediction. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set then $p = 0.05$ by default, resulting in the upper bound for a 95% confidence interval. The interval also depends on the variance of the error, as well as the variance of the parameter estimates. For the corresponding lower bound, see the LCL keyword.

UCLM — upper bound of a $100(1-p)$% confidence interval for the expected value (mean) of the predicted value. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set then $p = 0.05$ by default, resulting in the upper bound for a 95% confidence interval. For the corresponding lower bound, see the LCLM keyword.

OUT=*SAS-data-set*

gives the name of the new data set. By default, the procedure uses the DATAn convention to name the new data set.

The following option is available in the OUTPUT statement and is specified after a slash(/):

ALPHA=*p*

specifies the level of significance p for $100(1-p)$% confidence intervals. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You may use values between 0 and 1.

See Chapter 2, “Introduction to Regression Procedures,” and the “Influence Diagnostics” section on page 3898 in Chapter 61, “The REG Procedure,” for details on the calculation of these statistics.

Examples

The following statements show the syntax for creating an output data set with a single dependent variable.

```
proc glm;
   class a b;
   model y=a b a*b;
   output out=new p=yhat r=resid stdr=eresid;
run;
```

These statements create an output data set named new. In addition to all the variables from the original data set, new contains the variable yhat, with values that are predicted values of the dependent variable y; the variable resid, with values that are the residual values of y; and the variable eresid, with values that are the standard errors of the residuals.

The following statements show a situation with five dependent variables.

```
proc glm;
   by group;
   class a;
   model y1-y5=a x(a);
   output out=pout predicted=py1-py5;
run;
```

Data set pout contains five new variables, py1 through py5. The values of py1 are the predicted values of y1; the values of py2 are the predicted values of y2; and so on.

For more information on the data set produced by the OUTPUT statement, see the section "Output Data Sets" on page 1840.

RANDOM Statement

RANDOM *effects* < **/** *options* > **;**

When some model effects are random (that is, assumed to be sampled from a normal population of effects), you can specify these effects in the RANDOM statement in order to compute the expected values of mean squares for various model effects and contrasts and, optionally, to perform random effects analysis of variance tests. You can use as many RANDOM statements as you want, provided that they appear after the MODEL statement. If you use a CONTRAST statement with a RANDOM statement and you want to obtain the expected mean squares for the contrast hypothesis, you must enter the CONTRAST statement before the RANDOM statement.

Note: PROC GLM uses only the information pertaining to expected mean squares when you specify the TEST option in the RANDOM statement and, even then, only in the extra F tests produced by the RANDOM statement. Other features in the GLM procedure—including the results of the LSMEANS and ESTIMATE statements—assume that all effects are fixed, so that all tests and estimability checks for these statements are based on a fixed effects model, even when you use a RANDOM statement. Therefore, you should use the MIXED procedure to compute tests involving these features that take the random effects into account; see the section "PROC GLM versus PROC MIXED for Random Effects Analysis" on page 1833 and Chapter 46, "The MIXED Procedure," for more information.

When you use the RANDOM statement, by default the GLM procedure produces the Type III expected mean squares for model effects and for contrasts specified before the RANDOM statement in the program code. In order to obtain expected values for other types of mean squares, you need to specify which types of mean squares are of interest in the MODEL statement. See the section "Computing Type I, II, and IV Expected Mean Squares" on page 1835 for more information.

The list of effects in the RANDOM statement should contain one or more of the pure classification effects specified in the MODEL statement (that is, main effects, crossed effects, or nested effects involving only class variables). The coefficients corresponding to each effect specified are assumed to be normally and independently distributed with common variance. Levels in different effects are assumed to be independent.

You can specify the following options in the RANDOM statement after a slash:

Q

displays all quadratic forms in the fixed effects that appear in the expected mean squares. For some designs, large mixed-level factorials, for example, the Q option may generate a substantial amount of output.

TEST

performs hypothesis tests for each effect specified in the model, using appropriate error terms as determined by the expected mean squares.

Caution: PROC GLM does not automatically declare interactions to be random when the effects in the interaction are declared random. For example,

```
random a b / test;
```

does not produce the same expected mean squares or tests as

```
random a b a*b / test;
```

To ensure correct tests, you need to list all random interactions and random main effects in the RANDOM statement.

See the section "Random Effects Analysis" on page 1833 for more information on the calculation of expected mean squares and tests and on the similarities and differences between the GLM and MIXED procedures. See Chapter 3, "Introduction to Analysis-of-Variance Procedures," and Chapter 46, "The MIXED Procedure," for more information on random effects.

REPEATED Statement

REPEATED *factor-specification* < */ options* > **;**

When values of the dependent variables in the MODEL statement represent repeated measurements on the same experimental unit, the REPEATED statement enables you to test hypotheses about the measurement factors (often called *within-subject factors*) as well as the interactions of within-subject factors with independent variables in the MODEL statement (often called *between-subject factors*). The REPEATED statement provides multivariate and univariate tests as well as hypothesis tests for a variety of single-degree-of-freedom contrasts. There is no limit to the number of within-subject factors that can be specified.

The REPEATED statement is typically used for handling repeated measures designs with one repeated response variable. Usually, the variables on the left-hand side of the equation in the MODEL statement represent one repeated response variable. This does not mean that only one factor can be listed in the REPEATED statement. For example, one repeated response variable (hemoglobin count) might be measured 12 times (implying variables Y1 to Y12 on the left-hand side of the equal sign in the MODEL statement), with the associated within-subject factors treatment and time (implying two factors listed in the REPEATED statement). See the "Examples" section on page 1781 for an example of how PROC GLM handles this case. Designs with two or more repeated response variables can, however, be handled with the IDENTITY transformation; see page 1779 for more information, and Example 32.9 on page 1886 for an example of analyzing a doubly-multivariate repeated measures design.

When a REPEATED statement appears, the GLM procedure enters a multivariate mode of handling missing values. If any values for variables corresponding to each combination of the within-subject factors are missing, the observation is excluded from the analysis.

If you use a CONTRAST or TEST statement with a REPEATED statement, you must enter the CONTRAST or TEST statement before the REPEATED statement.

The simplest form of the REPEATED statement requires only a *factor-name*. With two repeated factors, you must specify the *factor-name* and number of levels (*levels*) for each factor. Optionally, you can specify the actual values for the levels (*level-values*), a *transformation* that defines single-degree-of freedom contrasts, and *options* for additional analyses and output. When you specify more than one within-subject factor, the *factor-names* (and associated level and transformation information) must be separated by a comma in the REPEATED statement. These terms are described in the following section, "Syntax Details."

Syntax Details

You can specify the following terms in the REPEATED statement.

factor-specification

The *factor-specification* for the REPEATED statement can include any number of individual factor specifications, separated by commas, of the following form:

factor-name levels < **(***level-values***)** > < *transformation* >

where

factor-name names a factor to be associated with the dependent variables. The name should not be the same as any variable name that already exists in the data set being analyzed and should conform to the usual conventions of SAS variable names.

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently.

levels gives the number of levels associated with the factor being defined. When there is only one within-subject factor, the number of levels is equal to the number of dependent variables. In this case, *levels* is optional. When more than one within-subject factor is defined, however, *levels* is required, and the product of the number of levels of all the factors must equal the number of dependent variables in the MODEL statement.

(*level-values*) gives values that correspond to levels of a repeated-measures factor. These values are used to label output and as spacings for constructing orthogonal polynomial contrasts if you specify a POLYNOMIAL transformation. The number of values specified must correspond to the number of levels for that factor in the REPEATED statement. Enclose the *level-values* in parentheses.

The following *transformation* keywords define single-degree-of-freedom contrasts for factors specified in the REPEATED statement. Since the number of contrasts generated is always one less than the number of levels of the factor, you have some control over which contrast is omitted from the analysis by which transformation you select. The only exception is the IDENTITY transformation; this transformation is not composed of contrasts and has the same degrees of freedom as the factor has levels. By default, the procedure uses the CONTRAST transformation.

CONTRAST < **(***ordinal-reference-level* **)** > generates contrasts between levels of the factor and a reference level. By default, the procedure uses the last level as the reference level; you can optionally specify a reference level in parentheses after the keyword CONTRAST. The reference level corresponds to the ordinal value of the level rather than the level value specified. For example, to generate contrasts between the first level of a factor and the other levels, use

```
contrast(1)
```

HELMERT generates contrasts between each level of the factor and the mean of subsequent levels.

IDENTITY generates an identity transformation corresponding to the associated factor. This transformation is *not* composed of contrasts; it has n degrees of freedom for an n-level factor, instead of $n - 1$. This can be used for doubly-multivariate repeated measures.

MEAN < **(***ordinal-reference-level* **)** > generates contrasts between levels of the factor and the mean of all other levels of the factor. Specifying a reference level eliminates the contrast between that level and the mean. Without a reference level, the contrast involving the last level is omitted. See the CONTRAST transformation for an example.

POLYNOMIAL generates orthogonal polynomial contrasts. Level values, if provided, are used as spacings in the construction of the polynomials; otherwise, equal spacing is assumed.

PROFILE generates contrasts between adjacent levels of the factor.

You can specify the following options in the REPEATED statement after a slash.

CANONICAL

performs a canonical analysis of the $\mathbf{H}$ and $\mathbf{E}$ matrices corresponding to the transformed variables specified in the REPEATED statement.

HTYPE=*n*

specifies the type of the $\mathbf{H}$ matrix used in the multivariate tests and the type of sums of squares used in the univariate tests. See the HTYPE= option in the specifications for the MANOVA statement for further details.

MEAN

generates the overall arithmetic means of the within-subject variables.

MSTAT=FAPPROX
MSTAT=EXACT

specifies the method of evaluating the test statistics for the multivariate analysis. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact *p*-values for three of the four tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root) and an improved F-approximation for the fourth (Pillai's Trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's Greatest Root, computations for the exact *p*-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information on the results of MSTAT=EXACT, see the "Multivariate Analysis of Variance" section on page 1823.

NOM

displays only the results of the univariate analyses.

NOU

displays only the results of the multivariate analyses.

PRINTE

displays the $\mathbf{E}$ matrix for each combination of within-subject factors, as well as partial correlation matrices for both the original dependent variables and the variables defined by the transformations specified in the REPEATED statement. In addition, the PRINTE option provides sphericity tests for each set of transformed variables. If the requested transformations are not orthogonal, the PRINTE option also provides a sphericity test for a set of orthogonal contrasts.

PRINTH

displays the $\mathbf{H}$ (SSCP) matrix associated with each multivariate test.

PRINTM

displays the transformation matrices that define the contrasts in the analysis. PROC GLM always displays the $\mathbf{M}$ matrix so that the transformed variables are defined by the rows, not the columns, of the displayed $\mathbf{M}$ matrix. In other words, PROC GLM actually displays $\mathbf{M}'$.

PRINTRV
displays the characteristic roots and vectors for each multivariate test.

SUMMARY
produces analysis-of-variance tables for each contrast defined by the within-subject factors. Along with tests for the effects of the independent variables specified in the MODEL statement, a term labeled MEAN tests the hypothesis that the overall mean of the contrast is zero.

Examples

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently. For example, assume that three treatments are administered at each of four times, for a total of twelve dependent variables on each experimental unit. If the variables are listed in the MODEL statement as Y1 through Y12, then the following REPEATED statement

```
proc glm;
    classes group;
    model Y1-Y12=group / nouni;
    repeated trt 3, time 4;
run;
```

implies the following structure:

	Dependent Variables											
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
Value of trt	1	1	1	1	2	2	2	2	3	3	3	3
Value of time	1	2	3	4	1	2	3	4	1	2	3	4

The REPEATED statement always produces a table like the preceding one. For more information, see the section "Repeated Measures Analysis of Variance" on page 1825.

TEST Statement

TEST < **H=***effects* > **E=***effect* < **/** *options* > **;**

Although an F value is computed for all sums of squares in the analysis using the residual MS as an error term, you may request additional F tests using other effects as error terms. You need a TEST statement when a nonstandard error structure (as in a split-plot design) exists. Note, however, that this may not be appropriate if the design is unbalanced, since in most unbalanced designs with nonstandard error structures, mean squares are not necessarily independent with equal expectations under the null hypothesis.

Caution: The GLM procedure does not check any of the assumptions underlying the F statistic. When you specify a TEST statement, you assume sole responsibility for the validity of the F statistic produced. To help validate a test, you can use the RANDOM statement and inspect the expected mean squares, or you can use the TEST option of the RANDOM statement.

You may use as many TEST statements as you want, provided that they appear after the MODEL statement.

You can specify the following terms in the TEST statement.

H=*effects* specifies which effects in the preceding model are to be used as hypothesis (numerator) effects.

E=*effect* specifies one, and only one, effect to use as the error (denominator) term. The E= specification is required.

By default, the sum of squares type for all hypothesis sum of squares and error sum of squares is the highest type computed in the model. If the hypothesis type or error type is to be another type that was computed in the model, you should specify one or both of the following options after a slash.

ETYPE=*n*

specifies the type of sum of squares to use for the error term. The type must be a type computed in the model (n=1, 2, 3, or 4).

HTYPE=*n*

specifies the type of sum of squares to use for the hypothesis. The type must be a type computed in the model (n=1, 2, 3, or 4).

This example illustrates the TEST statement with a split-plot model:

```
proc glm;
   class a b c;
   model y=a  b(a) c a*c b*c(a);
   test h=a e=b(a)/ htype=1 etype=1;
   test h=c a*c e=b*c(a) / htype=1 etype=1;
run;
```

WEIGHT Statement

WEIGHT *variable* **;**

When a WEIGHT statement is used, a weighted residual sum of squares

$$\sum_i w_i(y_i - \hat{y}_i)^2$$

is minimized, where w_i is the value of the variable specified in the WEIGHT statement, y_i is the observed value of the response variable, and $\hat{y}_i$ is the predicted value of the response variable.

If you specify the WEIGHT statement, it must appear before the first RUN statement or it is ignored.

An observation is used in the analysis only if the value of the WEIGHT statement variable is nonmissing and greater than zero.

The WEIGHT statement has no effect on degrees of freedom or number of observations, but it is used by the MEANS statement when calculating means and performing multiple comparison tests (as described in the "MEANS Statement" section beginning on page 1763). The normal equations used when a WEIGHT statement is present are

$$\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{Y}$$

where $\mathbf{W}$ is a diagonal matrix consisting of the values of the variable specified in the WEIGHT statement.

If the weights for the observations are proportional to the reciprocals of the error variances, then the weighted least-squares estimates are best linear unbiased estimators (BLUE).

Details

Statistical Assumptions for Using PROC GLM

The basic statistical assumption underlying the least-squares approach to general linear modeling is that the observed values of each dependent variable can be written as the sum of two parts: a fixed component $x'\beta$, which is a linear function of the independent coefficients, and a random noise, or error, component ϵ:

$$y \quad = \quad x'\beta + \epsilon$$

The independent coefficients x are constructed from the model effects as described in the "Parameterization of PROC GLM Models" section on page 1787. Further, the errors for different observations are assumed to be uncorrelated with identical variances. Thus, this model can be written

$$E(Y) \quad = \quad X\beta, \qquad \text{Var}(Y) \quad = \quad \sigma^2 I$$

where Y is the vector of dependent variable values, X is the matrix of independent coefficients, I is the identity matrix, and σ^2 is the common variance for the errors. For multiple dependent variables, the model is similar except that the errors for different dependent variables within the same observation are not assumed to be uncorrelated. This yields a multivariate linear model of the form

$$E(Y) \quad = \quad XB, \qquad \text{Var}(\text{vec}(Y)) \quad = \quad \Sigma \otimes I$$

where Y and B are now matrices, with one column for each dependent variable, vec(Y) strings Y out by rows, and $\otimes$ indicates the Kronecker matrix product.

Under the assumptions thus far discussed, the least-squares approach provides estimates of the linear parameters that are unbiased and have minimum variance among linear estimators. Under the further assumption that the errors have a normal (or Gaussian) distribution, the least-squares estimates are the maximum likelihood estimates and their distribution is known. All of the significance levels ("p values") and confidence limits calculated by the GLM procedure require this assumption of normality in order to be exactly valid, although they are good approximations in many other cases.

Specification of Effects

Each term in a model, called an *effect*, is a variable or combination of variables. Effects are specified with a special notation using variable names and operators. There are two kinds of variables: *classification* (or *class*) *variables* and *continuous variables*. There are two primary operators: *crossing* and *nesting*. A third operator, the *bar operator*, is used to simplify effect specification.

In an analysis-of-variance model, independent variables must be variables that identify classification levels. In the SAS System, these are called *class variables* and are declared in the CLASS statement. (They can also be called *categorical*, *qualitative*, *discrete*, or *nominal variables*.) Class variables can be either *numeric* or *character*. The values of a class variable are called *levels*. For example, the class variable Sex has the levels "male" and "female."

In a model, an independent variable that is not declared in the CLASS statement is assumed to be continuous. Continuous variables, which must be numeric, are used for response variables and covariates. For example, the heights and weights of subjects are continuous variables.

Types of Effects

There are seven different types of effects used in the GLM procedure. In the following list, assume that A, B, C, D, and E are class variables and that X1, X2, and Y are continuous variables:

- Regressor effects are specified by writing continuous variables by themselves: X1 X2.
- Polynomial effects are specified by joining two or more continuous variables with asterisks: X1*X1 X1*X2.
- Main effects are specified by writing class variables by themselves: A B C.
- Crossed effects (interactions) are specified by joining class variables with asterisks: A*B B*C A*B*C.

- Nested effects are specified by following a main effect or crossed effect with a class variable or list of class variables enclosed in parentheses. The main effect or crossed effect is nested within the effects listed in parentheses:

  B(A) C(B*A) D*E(C*B*A) .

  In this example, B(A) is read "B nested within A."
- Continuous-by-class effects are written by joining continuous variables and class variables with asterisks: X1*A.
- Continuous-nesting-class effects consist of continuous variables followed by a class variable interaction enclosed in parentheses: X1(A) X1*X2(A*B).

One example of the general form of an effect involving several variables is

X1*X2*A*B*C(D*E)

This example contains crossed continuous terms by crossed classification terms nested within more than one class variable. The continuous list comes first, followed by the crossed list, followed by the nesting list in parentheses. Note that asterisks can appear within the nested list but not immediately before the left parenthesis. For details on how the design matrix and parameters are defined with respect to the effects specified in this section, see the section "Parameterization of PROC GLM Models" on page 1787.

The MODEL statement and several other statements use these effects. Some examples of MODEL statements using various kinds of effects are shown in the following table; a, b, and c represent class variables, and y, y1, y2, x, and z represent continuous variables.

Specification	Kind of Model
model y=x;	simple regression
model y=x z;	multiple regression
model y=x x*x;	polynomial regression
model y1 y2=x z;	multivariate regression
model y=a;	one-way ANOVA
model y=a b c;	main effects model
model y=a b a*b;	factorial model (with interaction)
model y=a b(a) c(b a);	nested model
model y1 y2=a b;	multivariate analysis of variance (MANOVA)
model y=a x;	analysis-of-covariance model
model y=a x(a);	separate-slopes model
model y=a x x*a;	homogeneity-of-slopes model

The Bar Operator

You can shorten the specification of a large factorial model using the bar operator. For example, two ways of writing the model for a full three-way factorial model are

```
proc glm;                         and           proc glm;
   class A B C;                                    class A B C;
   model Y=A B C A*B                               model Y=A|B|C;
         A*C B*C A*B*C;                         run;
run;
```

When the bar (|) is used, the right- and left-hand sides become effects, and the cross of them becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules 2–4 given in Searle (1971, p. 390).

- Multiple bars are evaluated left to right. For instance, A|B|C is evaluated as follows.

$$\begin{aligned} \mathsf{A \mid B \mid C} &\rightarrow \{\, \mathsf{A \mid B} \,\} \mid \mathsf{C} \\ &\rightarrow \{\, \mathsf{A\ \ B\ \ A{*}B} \,\} \mid \mathsf{C} \\ &\rightarrow \mathsf{A\ \ B\ \ A{*}B\ \ C\ \ A{*}C\ \ B{*}C\ \ A{*}B{*}C} \end{aligned}$$

- Crossed and nested groups of variables are combined. For example, A(B) | C(D) generates A*C(B D), among other terms.
- Duplicate variables are removed. For example, A(C) | B(C) generates A*B(C C), among other terms, and the extra C is removed.
- Effects are discarded if a variable occurs on both the crossed and nested parts of an effect. For instance, A(B) | B(D E) generates A*B(B D E), but this effect is eliminated immediately.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect. For example, the specification A | B | C@2 would result in only those effects that contain 2 or fewer variables: in this case, A B A*B C A*C and B*C.

The following table gives more examples of using the bar and at operators.

A | C(B)	is equivalent to	A C(B) A*C(B)
A(B) | C(B)	is equivalent to	A(B) C(B) A*C(B)
A(B) | B(D E)	is equivalent to	A(B) B(D E)
A | B(A) | C	is equivalent to	A B(A) C A*C B*C(A)
A | B(A) | C@2	is equivalent to	A B(A) C A*C
A | B | C | D@2	is equivalent to	A B A*B C A*C B*C D A*D B*D C*D
A*B(C*D)	is equivalent to	A*B(C D)

Using PROC GLM Interactively

You can use the GLM procedure interactively. After you specify a model with a MODEL statement and run PROC GLM with a RUN statement, you can execute a variety of statements without reinvoking PROC GLM.

The "Syntax" section (page 1742) describes which statements can be used interactively. These interactive statements can be executed singly or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement cannot be repeated; PROC GLM allows only one MODEL statement.

If you use PROC GLM interactively, you can end the GLM procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement.

When you are using PROC GLM interactively, additional RUN statements do not end the procedure but tell PROC GLM to execute additional statements.

When you specify a WHERE statement with PROC GLM, it should appear before the first RUN statement. The WHERE statement enables you to select only certain observations for analysis without using a subsetting DATA step. For example, the statement `where group ne 5` omits observations with GROUP=5 from the analysis. Refer to *SAS Language Reference: Dictionary* for details on this statement.

When you specify a BY statement with PROC GLM, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

Interactivity is also disabled when there are different patterns of missing values among the dependent variables. For details, see the "Missing Values" section on page 1836.

Parameterization of PROC GLM Models

The GLM procedure constructs a linear model according to the specifications in the MODEL statement. Each effect generates one or more columns in a design matrix $\mathbf{X}$. This section shows precisely how $\mathbf{X}$ is built.

Intercept

All models include a column of 1s by default to estimate an intercept parameter μ. You can use the NOINT option to suppress the intercept.

Regression Effects

Regression effects (covariates) have the values of the variables copied into the design matrix directly. Polynomial terms are multiplied out and then installed in $\mathbf{X}$.

Main Effects

If a class variable has m levels, PROC GLM generates m columns in the design matrix for its main effect. Each column is an indicator variable for one of the levels of the class variable. The default order of the columns is the sort order of the values of their levels; this order can be controlled with the ORDER= option in the PROC GLM statement, as shown in the following table.

Data		Design Matrix					
			A		B		
A	B	μ	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

There are more columns for these effects than there are degrees of freedom for them; in other words, PROC GLM is using an over-parameterized model.

Crossed Effects

First, PROC GLM reorders the terms to correspond to the order of the variables in the CLASS statement; thus, B*A becomes A*B if A precedes B in the CLASS statement. Then, PROC GLM generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the cross index faster than the leftmost variables. No columns are generated corresponding to combinations of levels that do not occur in the data.

Data		Design Matrix											
			A		B			A*B					
A	B	μ	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In this matrix, main-effects columns are not linearly independent of crossed-effect columns; in fact, the column space for the crossed effects contains the space of the main effect.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following statements are the same (but the ordering of the columns is different):

```
model y=a b(a);      (B nested within A)
model y=a a*b;       (omitted main effect for B)
```

The nesting operator in PROC GLM is more a notational convenience than an operation distinct from crossing. Nested effects are characterized by the property that the nested variables never appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables.

Data		Design Matrix								
			A		B(A)					
A	B	μ	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

Continuous-Nesting-Class Effects

When a continuous variable nests with a class variable, the design columns are constructed by multiplying the continuous values into the design columns for the class effect.

Data		Design Matrix				
			A		X(A)	
X	A	μ	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate slope for X within each level of A.

Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. The two models differ by the presence of the continuous variable as a regressor by itself, in addition to being a contributor to X*A.

Data		Design Matrix					
				A		X*A	
X	A	μ	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

Continuous-by-class effects are used to test the homogeneity of slopes. If the continuous-by-class effect is nonsignificant, the effect can be removed so that the response with respect to X is the same for all levels of the class variables.

General Effects

An example that combines all the effects is

X1*X2*A*B*C(D E)

The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses.

The sequencing of parameters is important to learn if you use the CONTRAST or ESTIMATE statement to compute or test some linear function of the parameter estimates.

Effects may be retitled by PROC GLM to correspond to ordering rules. For example, B*A(E D) may be retitled A*B(D E) to satisfy the following:

- Class variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in a CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed part index faster than variables in the nested list.

- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. If the CLASS statement is

```
class A B C D;
```

then the order of the parameters for the effect B*A(C D), which is retitled A*B(C D), is as follows.

$A_1B_1C_1D_1$
$A_1B_2C_1D_1$
$A_2B_1C_1D_1$
$A_2B_2C_1D_1$
$A_1B_1C_1D_2$
$A_1B_2C_1D_2$
$A_2B_1C_1D_2$
$A_2B_2C_1D_2$
$A_1B_1C_2D_1$
$A_1B_2C_2D_1$
$A_2B_1C_2D_1$
$A_2B_2C_2D_1$
$A_1B_1C_2D_2$
$A_1B_2C_2D_2$
$A_2B_1C_2D_2$
$A_2B_2C_2D_2$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect changes fastest because it is rightmost in the (renamed) cross list. Then A changes next fastest. The D effect changes next fastest, and C is the slowest since it is leftmost in the nested list.

When numeric class variables are used, their levels are sorted by their character format, which may not correspond to their numeric sort sequence. Therefore, it is advisable to include a format for numeric class variables or to use the ORDER=INTERNAL option in the PROC GLM statement to ensure that levels are sorted by their internal values.

Degrees of Freedom

For models with classification (categorical) effects, there are more design columns constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns. In this event, the parameters are not jointly estimable; there is an infinite number of least-squares solutions. The GLM procedure uses a generalized (g2) inverse to obtain values for the estimates; see the

"Computational Method" section on page 1840 for more details. The solution values are not produced unless the SOLUTION option is specified in the MODEL statement. The solution has the characteristic that estimates are zero whenever the design column for that parameter is a linear combination of previous columns. (Strictly termed, the solution values should not be called estimates, since the parameters may not be formally estimable.) With this full parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Other procedures (such as the CATMOD procedure) reparameterize models to full rank using certain restrictions on the parameters. PROC GLM does not reparameterize, making the hypotheses that are commonly tested more understandable. See Goodnight (1978a) for additional reasons for not reparameterizing.

PROC GLM does not actually construct the entire design matrix $\mathbf{X}$; rather, a row x_i of $\mathbf{X}$ is constructed for each observation in the data set and used to accumulate the crossproduct matrix $\mathbf{X}'\mathbf{X} = \sum_i x_i' x_i$.

Hypothesis Testing in PROC GLM

See Chapter 11, "The Four Types of Estimable Functions," for a complete discussion of the four standard types of hypothesis tests.

Example

To illustrate the four types of tests and the principles upon which they are based, consider a two-way design with interaction based on the following data:

		B	
		1	2
	1	23.5 23.7	28.7
A	2	8.9	5.6 8.9
	3	10.3 12.5	13.6 14.6

Invoke PROC GLM and specify all the estimable functions options to examine what the GLM procedure can test. The following statements are followed by the summary ANOVA table. See Figure 32.8.

```
data example;
   input a b y @@;
   datalines;
1 1 23.5   1 1 23.7   1 2 28.7   2 1  8.9   2 2  5.6
2 2  8.9   3 1 10.3   3 1 12.5   3 2 13.6   3 2 14.6
;

proc glm;
```

```
      class a b;
      model y=a b a*b / e e1 e2 e3 e4;
   run;
```

```
                               The GLM Procedure

Dependent Variable: y

                                           Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                        5      520.4760000    104.0952000     49.66   0.0011

 Error                        4        8.3850000      2.0962500

 Corrected Total              9      528.8610000


              R-Square     Coeff Var      Root MSE        y Mean

              0.984145      9.633022      1.447843      15.03000
```

Figure 32.8. Summary ANOVA Table from PROC GLM

The following sections show the general form of estimable functions and discuss the four standard tests, their properties, and abbreviated output for the two-way crossed example.

Estimability

Figure 32.9 is the general form of estimable functions for the example. In order to be testable, a hypothesis must be able to fit within the framework displayed here.

```
                               The GLM Procedure

                       General Form of Estimable Functions

                       Effect             Coefficients

                       Intercept          L1

                       a         1        L2
                       a         2        L3
                       a         3        L1-L2-L3

                       b         1        L5
                       b         2        L1-L5

                       a*b       1 1      L7
                       a*b       1 2      L2-L7
                       a*b       2 1      L9
                       a*b       2 2      L3-L9
                       a*b       3 1      L5-L7-L9
                       a*b       3 2      L1-L2-L3-L5+L7+L9
```

Figure 32.9. General Form of Estimable Functions

If a hypothesis is estimable, the Ls in the preceding scheme can be set to values that match the hypothesis. All the standard tests in PROC GLM can be shown in the preceding format, with some of the Ls zeroed and some set to functions of other Ls.

The following sections show how many of the hypotheses can be tested by comparing the model sum-of-squares regression from one model to a submodel. The notation used is

$$\mathrm{SS}(B\ \mathit{effects}|A\ \mathit{effects}) = \mathrm{SS}(B\ \mathit{effects}, A\ \mathit{effects}) - \mathrm{SS}(A\ \mathit{effects})$$

where SS(*A effects*) denotes the regression model sum of squares for the model consisting of *A effects*. This notation is equivalent to the reduction notation defined by Searle (1971) and summarized in Chapter 11, "The Four Types of Estimable Functions."

Type I Tests

Type I sums of squares (SS), also called *sequential sums of squares*, are the incremental improvement in error sums of squares as each effect is added to the model. They can be computed by fitting the model in steps and recording the difference in error sum of squares at each step.

Source	Type I SS
A	$\mathrm{SS}(A \mid \mu)$
B	$\mathrm{SS}(B \mid \mu, A)$
$A * B$	$\mathrm{SS}(A * B\ \mid \mu, A, B)$

Type I sums of squares are displayed by default because they are easy to obtain and can be used in various hand calculations to produce sum of squares values for a series of different models. Nelder (1994) and others have argued that Type I and II sums are essentially the only appropriate ones for testing ANOVA effects; however, refer also to the discussion of Nelder's article, especially Rodriguez et al. (1995) and Searle (1995).

The Type I hypotheses have these properties:

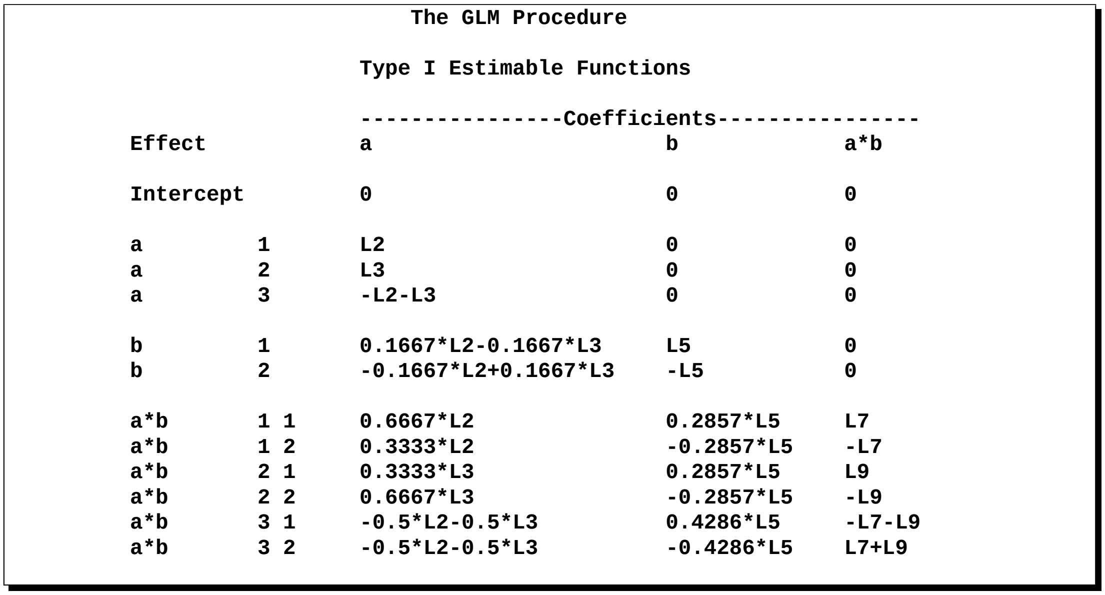
- Type I sum of squares for all effects add up to the model sum of squares. None of the other sum of squares types have this property, except in special cases.
- Type I hypotheses can be derived from rows of the Forward-Dolittle transformation of $\mathbf{X}'\mathbf{X}$ (a transformation that reduces $\mathbf{X}'\mathbf{X}$ to an upper triangular matrix by row operations).
- Type I sum of squares are statistically independent of each other under the usual assumption that the true residual errors are independent and identically normally distributed (see page 1783).
- Type I hypotheses depend on the order in which effects are specified in the MODEL statement.

- Type I hypotheses are uncontaminated by parameters corresponding to effects that precede the effect being tested; however, the hypotheses usually involve parameters for effects following the tested effect in the model. For example, in the model

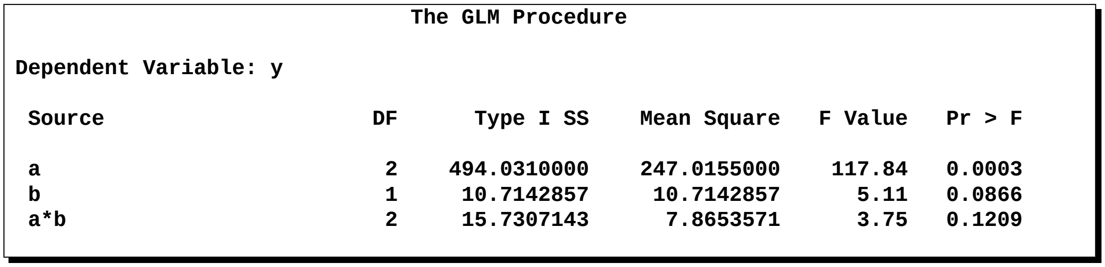
  ```
  Y=A B;
  ```

  the Type I hypothesis for B does not involve A parameters, but the Type I hypothesis for A does involve B parameters.
- Type I hypotheses are functions of the cell counts for unbalanced data; the hypotheses are not usually the same hypotheses that are tested if the data are balanced.
- Type I sums of squares are useful for polynomial models where you want to know the contribution of a term as though it had been made orthogonal to preceding effects. Thus, in polynomial models, Type I sums of squares correspond to tests of the orthogonal polynomial effects.

The Type I estimable functions and associated tests for the example are shown in Figure 32.10. (This combines tables from several pages of output.)



```
                               The GLM Procedure

                           Type I Estimable Functions

                           ----------------Coefficients----------------
         Effect            a                        b              a*b

         Intercept         0                        0              0

         a         1       L2                       0              0
         a         2       L3                       0              0
         a         3       -L2-L3                   0              0

         b         1       0.1667*L2-0.1667*L3      L5             0
         b         2       -0.1667*L2+0.1667*L3     -L5            0

         a*b       1 1     0.6667*L2                0.2857*L5      L7
         a*b       1 2     0.3333*L2                -0.2857*L5     -L7
         a*b       2 1     0.3333*L3                0.2857*L5      L9
         a*b       2 2     0.6667*L3                -0.2857*L5     -L9
         a*b       3 1     -0.5*L2-0.5*L3           0.4286*L5      -L7-L9
         a*b       3 2     -0.5*L2-0.5*L3           -0.4286*L5     L7+L9
```



```
                              The GLM Procedure

Dependent Variable: y

 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 a                            2    494.0310000    247.0155000    117.84   0.0003
 b                            1     10.7142857     10.7142857      5.11   0.0866
 a*b                          2     15.7307143      7.8653571      3.75   0.1209
```

Figure 32.10. Type I Estimable Functions and Associated Tests

Type II Tests

The Type II tests can also be calculated by comparing the error sums of squares (SS) for subset models. The Type II SS are the reduction in error SS due to adding the term after all other terms have been added to the model except terms that contain the effect being tested. An effect is contained in another effect if it can be derived by deleting variables from the latter effect. For example, A and B are both contained in A*B. For this model

Source	Type II SS
A	$\mathrm{SS}(A \mid \mu, B)$
B	$\mathrm{SS}(B \mid \mu, A)$
$A * B$	$\mathrm{SS}(A * B \mid \mu, A, B)$

Type II SS have these properties:

- Type II SS do not necessarily sum to the model SS.
- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- Type II SS are invariant to the ordering of effects in the model.
- For unbalanced designs, Type II hypotheses for effects that are contained in other effects are not usually the same hypotheses that are tested if the data are balanced. The hypotheses are generally functions of the cell counts.

The Type II estimable functions and associated tests for the example are shown in Figure 32.11. (Again, this combines tables from several pages of output.)

```
                               The GLM Procedure

                          Type II Estimable Functions

                           ----------------Coefficients----------------
         Effect            a                        b             a*b

         Intercept         0                        0             0

         a         1       L2                       0             0
         a         2       L3                       0             0
         a         3       -L2-L3                   0             0

         b         1       0                        L5            0
         b         2       0                        -L5           0

         a*b       1 1     0.619*L2+0.0476*L3       0.2857*L5     L7
         a*b       1 2     0.381*L2-0.0476*L3       -0.2857*L5    -L7
         a*b       2 1     -0.0476*L2+0.381*L3      0.2857*L5     L9
         a*b       2 2     0.0476*L2+0.619*L3       -0.2857*L5    -L9
         a*b       3 1     -0.5714*L2-0.4286*L3     0.4286*L5     -L7-L9
         a*b       3 2     -0.4286*L2-0.5714*L3     -0.4286*L5    L7+L9
```

```
                               The GLM Procedure

Dependent Variable: y

 Source                     DF     Type II SS    Mean Square   F Value   Pr > F

 a                           2    499.1202857    249.5601429    119.05   0.0003
 b                           1     10.7142857     10.7142857      5.11   0.0866
 a*b                         2     15.7307143      7.8653571      3.75   0.1209
```

Figure 32.11. Type II Estimable Functions and Associated Tests

Type III and Type IV Tests

Type III and Type IV sums of squares (SS), sometimes referred to as *partial sums of squares*, are considered by many to be the most desirable; see Searle (1987, Section 4.6). These SS cannot, in general, be computed by comparing model SS from several models using PROC GLM's parameterization. However, they can sometimes be computed by reduction for methods that reparameterize to full rank, when such a reparameterization effectively imposes Type III linear constraints on the parameters. In PROC GLM, they are computed by constructing a hypothesis matrix $\mathbf{L}$ and then computing the SS associated with the hypothesis $\mathbf{L}\beta = 0$. As long as there are no missing cells in the design, Type III and Type IV SS are the same.

These are properties of Type III and Type IV SS:

- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- The hypotheses to be tested are invariant to the ordering of effects in the model.

- The hypotheses are the same hypotheses that are tested if there are no missing cells. They are not functions of cell counts.
- The SS do not generally add up to the model SS and, in some cases, can exceed the model SS.

The SS are constructed from the general form of estimable functions. Type III and Type IV tests are different only if the design has missing cells. In this case, the Type III tests have an orthogonality property, while the Type IV tests have a balancing property. These properties are discussed in Chapter 11, “The Four Types of Estimable Functions.” For this example, since the data contains observations for all pairs of levels of A and B, Type IV tests are identical to the Type III tests that are shown in Figure 32.12. (This combines tables from several pages of output.)

```
                               The GLM Procedure

                          Type III Estimable Functions

                            -------------Coefficients-------------
            Effect          a                  b              a*b

            Intercept       0                  0              0

            a         1     L2                 0              0
            a         2     L3                 0              0
            a         3     -L2-L3             0              0

            b         1     0                  L5             0
            b         2     0                  -L5            0

            a*b       1 1   0.5*L2             0.3333*L5      L7
            a*b       1 2   0.5*L2             -0.3333*L5     -L7
            a*b       2 1   0.5*L3             0.3333*L5      L9
            a*b       2 2   0.5*L3             -0.3333*L5     -L9
            a*b       3 1   -0.5*L2-0.5*L3     0.3333*L5      -L7-L9
            a*b       3 2   -0.5*L2-0.5*L3     -0.3333*L5     L7+L9
```

```
                               The GLM Procedure

Dependent Variable: y

 Source                      DF     Type III SS     Mean Square    F Value    Pr > F

 a                            2     479.1078571     239.5539286     114.28    0.0003
 b                            1       9.4556250       9.4556250       4.51    0.1009
 a*b                          2      15.7307143       7.8653571       3.75    0.1209
```

Figure 32.12. Type III Estimable Functions and Associated Tests

Absorption

Absorption is a computational technique used to reduce computing resource needs in certain cases. The classic use of absorption occurs when a blocking factor with a large number of levels is a term in the model.

For example, the statements

```
proc glm;
   absorb herd;
   class a b;
   model y=a b a*b;
run;
```

are equivalent to

```
proc glm;
   class herd a b;
   model y=herd a b a*b;
run;
```

The exception to the previous statements is that the Type II, Type III, or Type IV SS for HERD are not computed when HERD is absorbed.

The algorithm for absorbing variables is similar to the one used by the NESTED procedure for computing a nested analysis of variance. As each new row of $[X|Y]$ (corresponding to the nonabsorbed independent effects and the dependent variables) is constructed, it is adjusted for the absorbed effects in a Type I fashion. The efficiency of the absorption technique is due to the fact that this adjustment can be done in one pass of the data and without solving any linear equations, assuming that the data have been sorted by the absorbed variables.

Several effects can be absorbed at one time. For example, these statements

```
proc glm;
   absorb herd cow;
   class a b;
   model y=a b a*b;
run;
```

are equivalent to

```
proc glm;
   class herd cow a b;
   model y=herd cow(herd) a b a*b;
run;
```

When you use absorption, the size of the $\mathbf{X}'\mathbf{X}$ matrix is a function only of the effects in the MODEL statement. The effects being absorbed do not contribute to the size of the $\mathbf{X}'\mathbf{X}$ matrix.

For the preceding example, a and b can be absorbed:

```
proc glm;
   absorb a b;
   class herd cow;
   model y=herd cow(herd);
run;
```

Although the sources of variation in the results are listed as

```
a b(a) herd cow(herd)
```

all types of estimable functions for herd and cow(herd) are free of a, b, and a*b parameters.

To illustrate the savings in computing using the ABSORB statement, PROC GLM is run on generated data with 1147 degrees of freedom in the model with the following statements:

```
data a;
   do herd=1 to 40;
      do cow=1 to 30;
         do treatment=1 to 3;
            do rep=1 to 2;
               y = herd/5 + cow/10 + treatment + rannor(1);
               output;
            end;
         end;
      end;
   end;

proc glm;
   class herd cow treatment;
   model y=herd cow(herd) treatment;
run;
```

This analysis would have required over 6 megabytes of memory for the $\mathbf{X}'\mathbf{X}$ matrix had PROC GLM solved it directly. However, in the following statements, the GLM procedure needs only a 4×4 matrix for the intercept and treatment because the other effects are absorbed.

```
proc glm;
   absorb herd cow;
   class treatment;
   model y = treatment;
run;
```

These statements produce the results shown in Figure 32.13.

```
                              The GLM Procedure

                           Class Level Information

                       Class          Levels    Values

                       treatment           3    1 2 3


                   Number of Observations Read        7200
                   Number of Observations Used        7200



                              The GLM Procedure

Dependent Variable: y

                                            Sum of
 Source                      DF          Squares     Mean Square    F Value    Pr > F

 Model                     1201      49465.40242        41.18685      41.57    <.0001

 Error                     5998       5942.23647         0.99070

 Corrected Total           7199      55407.63889


               R-Square     Coeff Var      Root MSE        y Mean

               0.892754      13.04236      0.995341      7.631598


 Source                      DF      Type I SS     Mean Square    F Value    Pr > F

 herd                        39    38549.18655       988.44068     997.72    <.0001
 cow(herd)                 1160     6320.18141         5.44843       5.50    <.0001
 treatment                    2     4596.03446      2298.01723    2319.58    <.0001


 Source                      DF    Type III SS     Mean Square    F Value    Pr > F

 treatment                    2    4596.034455     2298.017228    2319.58    <.0001
```

Figure 32.13. Absorption of Effects

Specification of ESTIMATE Expressions

Consider the model

$$E(Y) \quad = \quad \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3.$$

The corresponding MODEL statement for PROC GLM is

```
model y=x1 x2 x3;
```

To estimate the difference between the parameters for x_1 and x_2,

$$\beta_1 - \beta_2 \quad = \quad (\begin{array}{cccc} 0 & 1 & -1 & 0 \end{array})\boldsymbol{\beta}, \text{ where } \boldsymbol{\beta} = (\begin{array}{cccc} \beta_0 & \beta_1 & \beta_2 & \beta_3 \end{array})'$$

you can use the following ESTIMATE statement:

```
estimate 'B1-B2'  x1 1  x2 -1;
```

To predict y at $x_1 = 1$, $x_2 = 0$, and $x_3 = -2$, you can estimate

$$\beta_0 + \beta_1 - 2\beta_3 \quad = \quad (\begin{array}{cccc} 1 & 1 & 0 & -2 \end{array})\boldsymbol{\beta}$$

with the following ESTIMATE statement:

```
estimate 'B0+B1-2B3' intercept 1 x1 1 x3 -2;
```

Now consider models involving class variables such as

```
model y=A B A*B;
```

with the associated parameters:

$$\left(\begin{array}{cccccccccccc} \mu & \alpha_1 & \alpha_2 & \alpha_3 & \beta_1 & \beta_2 & \gamma_{11} & \gamma_{12} & \gamma_{21} & \gamma_{22} & \gamma_{31} & \gamma_{32} \end{array}\right)$$

The LS-mean for the first level of A is $\mathbf{L}\beta$, where

$$\mathbf{L} = (\begin{array}{c|ccc|cc|cccccc} 1 & 1 & 0 & 0 & 0.5 & 0.5 & 0.5 & 0.5 & 0 & 0 & 0 & 0 \end{array})$$

You can estimate this with the following ESTIMATE statement:

```
estimate 'LS-mean(A1)' intercept 1 A 1 B 0.5 0.5 A*B 0.5 0.5;
```

Note in this statement that only one element of $\mathbf{L}$ is specified following the A effect, even though A has three levels. Whenever the list of constants following an effect name is shorter than the effect's number of levels, zeros are used as the remaining constants. (If the list of constants is longer than the number of levels for the effect, the extra constants are ignored, and a warning message is displayed.)

To estimate the A linear effect in the preceding model, assuming equally spaced levels for A, you can use the following $\mathbf{L}$:

$$\mathbf{L} = (\begin{array}{c|ccc|cc|cccccc} 0 & -1 & 0 & 1 & 0 & 0 & -0.5 & -0.5 & 0 & 0 & 0.5 & 0.5 \end{array})$$

The ESTIMATE statement for this $\mathbf{L}$ is written as

```
        estimate 'A Linear' A -1 0 1;
```

If you do not specify the elements of $\mathbf{L}$ for an effect that contains a specified effect, then the elements of the specified effect are equally distributed over the corresponding levels of the higher-order effect. In addition, if you specify the intercept in an ESTIMATE or CONTRAST statement, it is distributed over all classification effects that are not contained by any other specified effect. The distribution of lower-order coefficients to higher-order effect coefficients follows the same general rules as in the LSMEANS statement, and it is similar to that used to construct Type IV tests. In the previous example, the -1 associated with α_1 is divided by the number n_{1j} of γ_{1j} parameters; then each γ_{1j} coefficient is set to $-1/n_{1j}$. The 1 associated with α_3 is distributed among the γ_{3j} parameters in a similar fashion. In the event that an unspecified effect contains several specified effects, only that specified effect with the most factors in common with the unspecified effect is used for distribution of coefficients to the higher-order effect.

Numerous syntactical expressions for the ESTIMATE statement were considered, including many that involved specifying the effect and level information associated with each coefficient. For models involving higher-level effects, the requirement of specifying level information can lead to very bulky specifications. Consequently, the simpler form of the ESTIMATE statement described earlier was implemented. The syntax of this ESTIMATE statement puts a burden on you to know a priori the order of the parameter list associated with each effect. You can use the ORDER= option in the PROC GLM statement to ensure that the levels of the classification effects are sorted appropriately.

Note: If you use the ESTIMATE statement with unspecified effects, use the E option to make sure that the actual $\mathbf{L}$ constructed by the preceding rules is the one you intended.

A Check for Estimability

Each $\mathbf{L}$ is checked for estimability using the relationship: $\mathbf{L} = \mathbf{LH}$ where $\mathbf{H} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$. The $\mathbf{L}$ vector is declared nonestimable, if for any i

$$\mathrm{ABS}(\mathbf{L}_i - (\mathbf{LH})_i) > \begin{cases} \epsilon & \text{if } \mathbf{L}_i = 0 \text{ or} \\ \epsilon \times \mathrm{ABS}(\mathbf{L}_i) & \text{otherwise} \end{cases}$$

where $\epsilon = 10^{-4}$ by default; you can change this with the SINGULAR= option. Continued fractions (like 1/3) should be specified to at least six decimal places, or the DIVISOR parameter should be used.

Comparing Groups

An important task in analyzing data with classification effects is to estimate the typical response for each level of a given effect; often, you also want to compare these estimates to determine which levels are equivalent in terms of the response. You can perform this task in two ways with the GLM procedure: with direct, arithmetic group means; and with so-called *least-squares means* (LS-means).

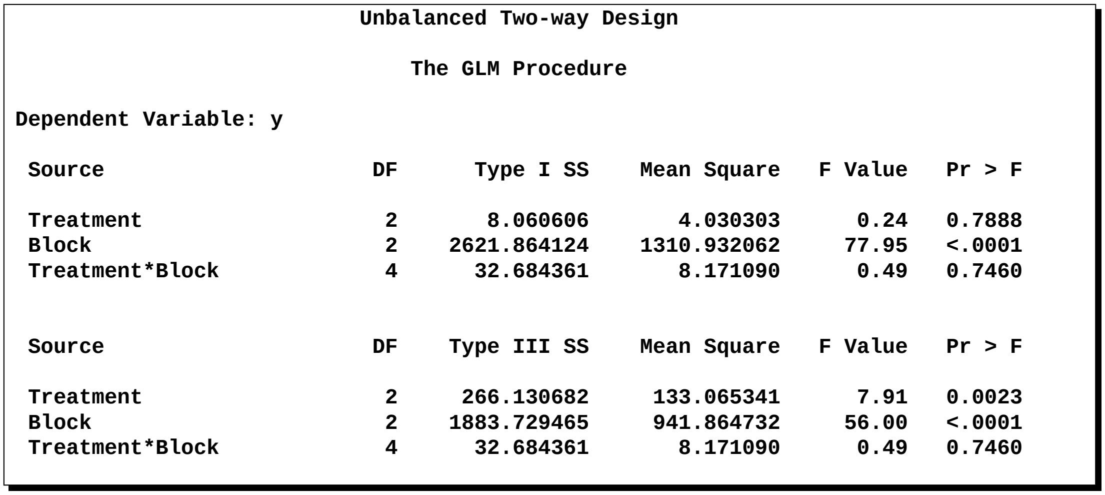
Means Versus LS-Means

Computing and comparing arithmetic means—either simple or weighted within-group averages of the input data—is a familiar and well-studied statistical process. This is the right approach to summarizing and comparing groups for one-way and balanced designs. However, in unbalanced designs with more than one effect, the arithmetic mean for a group may not accurately reflect the "typical" response for that group, since it does not take other effects into account.

For example, consider the following analysis of an unbalanced two-way design:

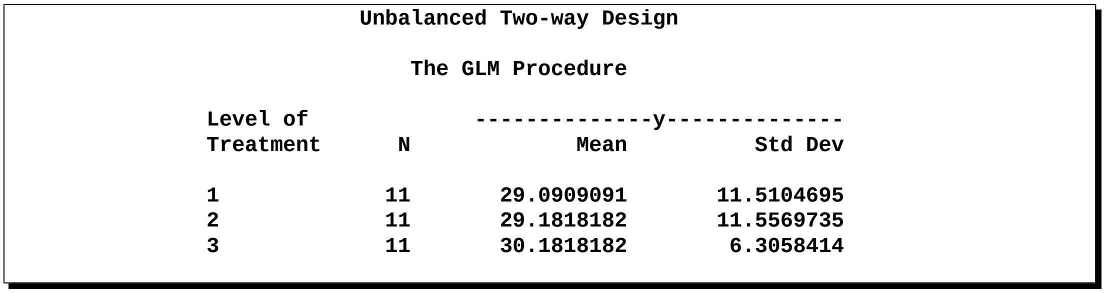
```
data twoway;
   input Treatment Block y @@;
   datalines;
1 1 17   1 1 28   1 1 19   1 1 21   1 1 19
1 2 43   1 2 30   1 2 39   1 2 44   1 2 44
1 3 16
2 1 21   2 1 21   2 1 24   2 1 25
2 2 39   2 2 45   2 2 42   2 2 47
2 3 19   2 3 22   2 3 16
3 1 22   3 1 30   3 1 33   3 1 31
3 2 46
3 3 26   3 3 31   3 3 26   3 3 33   3 3 29   3 3 25
;

title1 "Unbalanced Two-way Design";
ods select ModelANOVA Means LSMeans;
proc glm data=twoway;
   class Treatment Block;
   model y = Treatment|Block;


   means Treatment;
   lsmeans Treatment;
run;
ods select all;
```

The ANOVA results are shown in Figure 32.14.



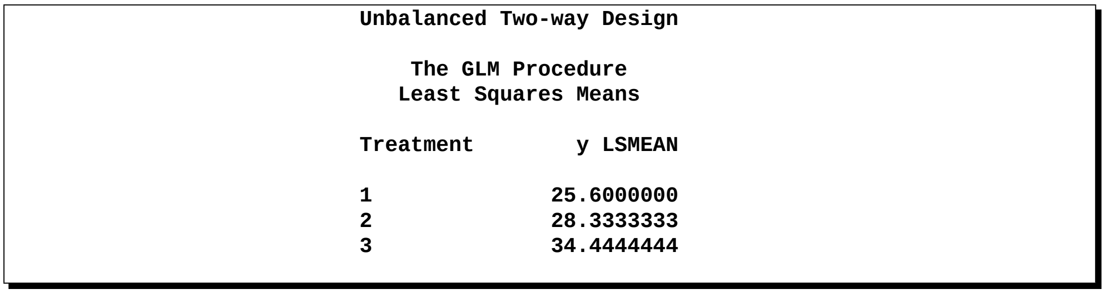
```
                        Unbalanced Two-way Design

                            The GLM Procedure

Dependent Variable: y

 Source                     DF      Type I SS    Mean Square   F Value   Pr > F

 Treatment                   2       8.060606       4.030303      0.24   0.7888
 Block                       2    2621.864124    1310.932062     77.95   <.0001
 Treatment*Block             4      32.684361       8.171090      0.49   0.7460


 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 Treatment                   2     266.130682     133.065341      7.91   0.0023
 Block                       2    1883.729465     941.864732     56.00   <.0001
 Treatment*Block             4      32.684361       8.171090      0.49   0.7460
```

Figure 32.14. ANOVA Results for Unbalanced Two-Way Design



```
                        Unbalanced Two-way Design

                            The GLM Procedure

              Level of             --------------y--------------
              Treatment      N             Mean          Std Dev

              1             11       29.0909091       11.5104695
              2             11       29.1818182       11.5569735
              3             11       30.1818182        6.3058414
```

Figure 32.15. Treatment Means for Unbalanced Two-Way Design



```
                        Unbalanced Two-way Design

                            The GLM Procedure
                           Least Squares Means

                        Treatment        y LSMEAN

                        1              25.6000000
                        2              28.3333333
                        3              34.4444444
```

Figure 32.16. Treatment LS-means for Unbalanced Two-Way Design

No matter how you look at it, this data exhibits a strong effect due to the blocks (F-test $p < 0.0001$) and no significant interaction between treatments and blocks (F-test $p > 0.7$). But the lack of balance affects how the treatment effect is interpreted: in a main-effects-only model, there are no significant differences between the treatment means themselves (Type I F-test $p > 0.7$), but there are highly significant differences between the treatment means corrected for the block effects (Type III F-test $p < 0.01$).

LS-means are, in effect, within-group means appropriately adjusted for the other effects in the model. More precisely, they estimate the marginal means for a balanced population (as opposed to the unbalanced design). For this reason, they are also called *estimated population marginal means* by Searle et al. (1980). In the same way that the Type I F-test assesses differences between the arithmetic treatment means (when the treatment effect comes first in the model), the Type III F-test assesses differences between the LS-means. Accordingly, for the unbalanced two-way design, the discrepancy between the Type I and Type III tests is reflected in the arithmetic treatment means and treatment LS-means, as shown in Figure 32.15 and Figure 32.16. See the section "Construction of Least-Squares Means" on page 1820 for more on LS-means.

Note that, while the arithmetic means are always uncorrelated (under the usual assumptions for analysis of variance; see page 1783), the LS-means may not be. This fact complicates the problem of multiple comparisons for LS-means; see the following section.

Multiple Comparisons

When comparing more than two means, an ANOVA F-test tells you whether the means are significantly different from each other, but it does not tell you which means differ from which other means. Multiple comparison procedures (MCPs), also called *mean separation tests*, give you more detailed information about the differences among the means. The goal in multiple comparisons is to compare the average effects of three or more "treatments" (for example, drugs, groups of subjects) to decide which treatments are better, which ones are worse, and by how much, while controlling the probability of making an incorrect decision. A variety of multiple comparison methods are available with the MEANS and LSMEANS statement in the GLM procedure.

The following classification is due to Hsu (1996). Multiple comparison procedures can be categorized in two ways: by the comparisons they make and by the strength of inference they provide. With respect to which comparisons are made, the GLM procedure offers two types:

- comparisons between all pairs of means
- comparisons between a control and all other means

The strength of inference says what can be inferred about the structure of the means when a test is significant; it is related to what type of error rate the MCP controls. MCPs available in the GLM procedure provide one of the following types of inference, in order from weakest to strongest.

- Individual: differences between means, unadjusted for multiplicity
- Inhomogeneity: means are different
- Inequalities: which means are different
- Intervals: simultaneous confidence intervals for mean differences

Methods that control only individual error rates are not true MCPs at all. Methods that yield the strongest level of inference, simultaneous confidence intervals, are usually preferred, since they enable you not only to say which means are different but also to put confidence bounds on *how much* they differ, making it easier to assess the practical significance of a difference. They are also less likely to lead nonstatisticians to the invalid conclusion that nonsignificantly different sample means imply equal population means. Interval MCPs are available for both arithmetic means and LS-means via the MEANS and LSMEANS statements, respectively.*

Table 32.3 and Table 32.4 display MCPs available in PROC GLM for all pairwise comparisons and comparisons with a control, respectively, along with associated strength of inference and the syntax (when applicable) for both the MEANS and the LSMEANS statements.

Table 32.3. Multiple Comparisons Procedures for All Pairwise Comparison

	Strength of	Syntax	
Method	Inference	MEANS	LSMEANS
Student's t	Individual	T	PDIFF ADJUST=T
Duncan	Individual	DUNCAN	
Student-Newman-Keuls	Inhomogeneity	SNK	
REGWQ	Inequalities	REGWQ	
Tukey-Kramer	Intervals	TUKEY	PDIFF ADJUST=TUKEY
Bonferroni	Intervals	BON	PDIFF ADJUST=BON
Sidak	Intervals	SIDAK	PDIFF ADJUST=SIDAK
Scheffé	Intervals	SCHEFFE	PDIFF ADJUST=SCHEFFE
SMM	Intervals	SMM	PDIFF ADJUST=SMM
Gabriel	Intervals	GABRIEL	
Simulation	Intervals		PDIFF ADJUST=SIMULATE

Table 32.4. Multiple Comparisons Procedures for Comparisons with a Control

	Strength of	Syntax	
Method	Inference	MEANS	LSMEANS
Student's t	Individual		PDIFF=CONTROL ADJUST=T
Dunnett	Intervals	DUNNETT	PDIFF=CONTROL ADJUST=DUNNETT
Bonferroni	Intervals		PDIFF=CONTROL ADJUST=BON
Sidak	Intervals		PDIFF=CONTROL ADJUST=SIDAK
Scheffé	Intervals		PDIFF=CONTROL ADJUST=SCHEFFE
SMM	Intervals		PDIFF=CONTROL ADJUST=SMM
Simulation	Intervals		PDIFF=CONTROL ADJUST=SIMULATE

Note: One-sided Dunnett's tests are also available from the MEANS statement with the DUNNETTL and DUNNETTU options and from the LSMEANS statement with PDIFF=CONTROLL and PDIFF=CONTROLU.

Details of these multiple comparison methods are given in the following sections.

Pairwise Comparisons

All the methods discussed in this section depend on the standardized pairwise differences $t_{ij} = (\bar{y}_i - \bar{y}_j)/\hat{\sigma}_{ij}$, where

*The Duncan-Waller method does not fit into the preceding scheme, since it is based on the Bayes risk rather than any particular error rate.

- i and j are the indices of two groups
- $\bar{y}_i$ and $\bar{y}_j$ are the means or LS-means for groups i and j
- $\hat{\sigma}_{ij}$ is the square-root of the estimated variance of $\bar{y}_i - \bar{y}_j$. For simple arithmetic means, $\hat{\sigma}_{ij}^2 = s^2(1/n_i + 1/n_j)$, where n_i and n_j are the sizes of groups i and j, respectively, and s^2 is the mean square for error, with ν degrees of freedom. For weighted arithmetic means, $\hat{\sigma}_{ij}^2 = s^2(1/w_i + 1/w_j)$, where w_i and w_j are the sums of the weights in groups i and j, respectively. Finally, for LS-means defined by the linear combinations $l_i'b$ and $l_j'b$ of the parameter estimates, $\hat{\sigma}_{ij}^2 = s^2 l_i'(\mathbf{X}'\mathbf{X})^- l_j$.

Furthermore, all of the methods are discussed in terms of significance tests of the form

$$|t_{ij}| \quad \geq \quad c(\alpha)$$

where $c(\alpha)$ is some constant depending on the significance level. Such tests can be inverted to form confidence intervals of the form

$$(\bar{y}_i - \bar{y}_j) - \hat{\sigma}_{ij} c(\alpha) \quad \leq \quad \mu_i - \mu_j \quad \leq \quad (\bar{y}_i - \bar{y}_j) + \hat{\sigma}_{ij} c(\alpha)$$

The simplest approach to multiple comparisons is to do a t test on every pair of means (the T option in the MEANS statement, ADJUST=T in the LSMEANS statement). For the ith and jth means, you can reject the null hypothesis that the population means are equal if

$$|t_{ij}| \quad \geq \quad t(\alpha; \nu)$$

where α is the significance level, ν is the number of error degrees of freedom, and $t(\alpha; \nu)$ is the two-tailed critical value from a Student's t distribution. If the cell sizes are all equal to, say, n, the preceding formula can be rearranged to give

$$|\bar{y}_i - \bar{y}_j| \quad \geq \quad t(\alpha; \nu) s \sqrt{\frac{2}{n}}$$

the value of the right-hand side being Fisher's least significant difference (LSD).

There is a problem with repeated t tests, however. Suppose there are ten means and each t test is performed at the 0.05 level. There are 10(10-1)/2=45 pairs of means to compare, each with a 0.05 probability of a type 1 error (a false rejection of the null hypothesis). The chance of making at least one type 1 error is much higher than 0.05. It is difficult to calculate the exact probability, but you can derive a pessimistic approximation by assuming that the comparisons are independent, giving an upper bound to the probability of making at least one type 1 error (the experimentwise error rate) of

$$1 - (1 - 0.05)^{45} \quad = \quad 0.90$$

The actual probability is somewhat less than 0.90, but as the number of means increases, the chance of making at least one type 1 error approaches 1.

If you decide to control the individual type 1 error rates for each comparison, you are controlling the individual or comparisonwise error rate. On the other hand, if you want to control the overall type 1 error rate for all the comparisons, you are controlling the experimentwise error rate. It is up to you to decide whether to control the comparisonwise error rate or the experimentwise error rate, but there are many situations in which the experimentwise error rate should be held to a small value. Statistical methods for comparing three or more means while controlling the probability of making at least one type 1 error are called *multiple comparisons procedures*.

It has been suggested that the experimentwise error rate can be held to the α level by performing the overall ANOVA F-test at the α level and making further comparisons only if the F-test is significant, as in Fisher's protected LSD. This assertion is false if there are more than three means (Einot and Gabriel 1975). Consider again the situation with ten means. Suppose that one population mean differs from the others by such a sufficiently large amount that the power (probability of correctly rejecting the null hypothesis) of the F-test is near 1 but that all the other population means are equal to each other. There will be $9(9-1)/2 = 36$ t tests of true null hypotheses, with an upper limit of 0.84 on the probability of at least one type 1 error. Thus, you must distinguish between the experimentwise error rate under the complete null hypothesis, in which all population means are equal, and the experimentwise error rate under a partial null hypothesis, in which some means are equal but others differ. The following abbreviations are used in the discussion:

CER	comparisonwise error rate
EERC	experimentwise error rate under the complete null hypothesis
MEER	maximum experimentwise error rate under any complete or partial null hypothesis

These error rates are associated with the different strengths of inference discussed on page 1806: individual tests control the CER; tests for inhomogeneity of means control the EERC; tests that yield confidence inequalities or confidence intervals control the MEER. A preliminary F-test controls the EERC but not the MEER.

You can control the MEER at the α level by setting the CER to a sufficiently small value. The Bonferroni inequality (Miller 1981) has been widely used for this purpose. If

$$\text{CER} = \frac{\alpha}{c}$$

where c is the total number of comparisons, then the MEER is less than α. Bonferroni t tests (the BON option in the MEANS statement, ADJUST=BON in the LSMEANS statement) with $\text{MEER} < \alpha$ declare two means to be significantly different if

$$|t_{ij}| \geq t(\epsilon;\nu)$$

where

$$\epsilon \quad = \quad \frac{2\alpha}{k(k-1)}$$

for comparison of k means.

Sidak (1967) has provided a tighter bound, showing that

$$\text{CER} \quad = \quad 1-(1-\alpha)^{1/c}$$

also ensures that $\text{MEER} \leq \alpha$ for any set of c comparisons. A Sidak t test (Games 1977), provided by the SIDAK option, is thus given by

$$|t_{ij}| \quad \geq \quad t(\epsilon;\nu)$$

where

$$\epsilon \quad = \quad 1-(1-\alpha)^{\frac{2}{k(k-1)}}$$

for comparison of k means.

You can use the Bonferroni additive inequality and the Sidak multiplicative inequality to control the MEER for any set of contrasts or other hypothesis tests, not just pairwise comparisons. The Bonferroni inequality can provide simultaneous inferences in any statistical application requiring tests of more than one hypothesis. Other methods discussed in this section for pairwise comparisons can also be adapted for general contrasts (Miller 1981).

Scheffé (1953; 1959) proposes another method to control the MEER for any set of contrasts or other linear hypotheses in the analysis of linear models, including pairwise comparisons, obtained with the SCHEFFE option. Two means are declared significantly different if

$$|t_{ij}| \quad \geq \quad \sqrt{(k-1)F(\alpha;k-1,\nu)}$$

where $F(\alpha;k-1,\nu)$ is the α-level critical value of an F distribution with $k-1$ numerator degrees of freedom and ν denominator degrees of freedom.

Scheffé's test is compatible with the overall ANOVA F-test in that Scheffé's method never declares a contrast significant if the overall F-test is nonsignificant. Most other multiple comparison methods can find significant contrasts when the overall F-test is nonsignificant and, therefore, suffer a loss of power when used with a preliminary F-test.

Scheffé's method may be more powerful than the Bonferroni or Sidak methods if the number of comparisons is large relative to the number of means. For pairwise comparisons, Sidak t tests are generally more powerful.

Tukey (1952; 1953) proposes a test designed specifically for pairwise comparisons based on the studentized range, sometimes called the "honestly significant difference test," that controls the MEER when the sample sizes are equal. Tukey (1953) and Kramer (1956) independently propose a modification for unequal cell sizes. The Tukey or Tukey-Kramer method is provided by the TUKEY option in the MEANS statement and the ADJUST=TUKEY option in the LSMEANS statement. This method has fared extremely well in Monte Carlo studies (Dunnett 1980). In addition, Hayter (1984) gives a proof that the Tukey-Kramer procedure controls the MEER for means comparisons, and Hayter (1989) describes the extent to which the Tukey-Kramer procedure has been proven to control the MEER for LS-means comparisons. The Tukey-Kramer method is more powerful than the Bonferroni, Sidak, or Scheffé methods for pairwise comparisons. Two means are considered significantly different by the Tukey-Kramer criterion if

$$|t_{ij}| \quad \geq \quad q(\alpha; k, \nu)$$

where $q(\alpha; k, \nu)$ is the α-level critical value of a studentized range distribution of k independent normal random variables with ν degrees of freedom.

Hochberg (1974) devised a method (the GT2 or SMM option) similar to Tukey's, but it uses the studentized maximum modulus instead of the studentized range and employs Sidak (1967) uncorrelated t inequality. It is proven to hold the MEER at a level not exceeding α with unequal sample sizes. It is generally less powerful than the Tukey-Kramer method and always less powerful than Tukey's test for equal cell sizes. Two means are declared significantly different if

$$|t_{ij}| \quad \geq \quad m(\alpha; c, \nu)$$

where $m(\alpha; c, \nu)$ is the α-level critical value of the studentized maximum modulus distribution of c independent normal random variables with ν degrees of freedom and $c = k(k-1)/2$.

Gabriel (1978) proposes another method (the GABRIEL option) based on the studentized maximum modulus. This method is applicable only to arithmetic means. It rejects if

$$\frac{|\bar{y}_i - \bar{y}_j|}{s\left(\frac{1}{\sqrt{2n_i}} + \frac{1}{\sqrt{2n_j}}\right)} \quad \geq \quad m(\alpha; k, \nu)$$

For equal cell sizes, Gabriel's test is equivalent to Hochberg's GT2 method. For unequal cell sizes, Gabriel's method is more powerful than GT2 but may become liberal with highly disparate cell sizes (refer also to Dunnett (1980)). Gabriel's test is the only method for unequal sample sizes that lends itself to a graphical representation as intervals around the means. Assuming $\bar{y}_i > \bar{y}_j$, you can rewrite the preceding inequality as

$$\bar{y}_i - m(\alpha; k, \nu)\frac{s}{\sqrt{2n_i}} \quad \geq \quad \bar{y}_j + m(\alpha; k, \nu)\frac{s}{\sqrt{2n_j}}$$

The expression on the left does not depend on j, nor does the expression on the right depend on i. Hence, you can form what Gabriel calls an (l, u)-interval around each sample mean and declare two means to be significantly different if their (l, u)-intervals do not overlap. See Hsu (1996, section 5.2.1.1) for a discussion of other methods of graphically representing all pair-wise comparisons.

Comparing All Treatments to a Control

One special case of means comparison is that in which the only comparisons that need to be tested are between a set of new treatments and a single control. In this case, you can achieve better power by using a method that is restricted to test only comparisons to the single control mean. Dunnett (1955) proposes a test for this situation that declares a mean significantly different from the control if

$$|t_{i0}| \quad \geq \quad d(\alpha; k, \nu, \rho_1, \ldots, \rho_{k-1})$$

where $\bar{y}_0$ is the control mean and $d(\alpha; k, \nu, \rho_1, \ldots, \rho_{k-1})$ is the critical value of the "many-to-one t statistic" (Miller 1981; Krishnaiah and Armitage 1966) for k means to be compared to a control, with ν error degrees of freedom and correlations $\rho_1, \ldots, \rho_{k-1}$, $\rho_i = n_i/(n_0 + n_i)$. The correlation terms arise because each of the treatment means is being compared to the same control. Dunnett's test holds the MEER to a level not exceeding the stated α.

Approximate and Simulation-based Methods

Both Tukey's and Dunnett's tests are based on the same general quantile calculation:

$$q^t(\alpha, \nu, R) \quad = \quad \{q \ni P(\max(|t_1|, \ldots, |t_n|) > q) = \alpha\}$$

where the t_i have a joint multivariate t distribution with ν degrees of freedom and correlation matrix R. In general, evaluating $q^t(\alpha, \nu, R)$ requires repeated numerical calculation of an $(n + 1)$-fold integral. This is usually intractable, but the problem reduces to a feasible 2-fold integral when R has a certain symmetry in the case of Tukey's test, and a *factor analytic structure* (cf. Hsu 1992) in the case of Dunnett's test. The R matrix has the required symmetry for exact computation of Tukey's test if the t_is are studentized differences between

- $k(k-1)/2$ pairs of k uncorrelated means with equal variances—that is, equal sample sizes
- $k(k-1)/2$ pairs of k LS-means from a *variance-balanced* design (for example, a balanced incomplete block design)

Refer to Hsu (1992; 1996) for more information. The R matrix has the factor analytic structure for exact computation of Dunnett's test if the t_is are studentized differences between

- $k-1$ means and a control mean, all uncorrelated. (Dunnett's one-sided methods depend on a similar probability calculation, without the absolute values.) Note that it is not required that the variances of the means (that is, the sample sizes) be equal.

- $k-1$ LS-means and a control LS-mean from either a *variance-balanced* design, or a design in which the other factors are *orthogonal* to the treatment factor (for example, a randomized block design with proportional cell frequencies).

However, other important situations that do **not** result in a correlation matrix R that has the structure for exact computation include

- all pairwise differences with unequal sample sizes
- differences between LS-means in many unbalanced designs

In these situations, exact calculation of $q^t(\alpha, \nu, R)$ is intractable in general. Most of the preceding methods can be viewed as using various approximations for $q^t(\alpha, \nu, R)$. When the sample sizes are unequal, the Tukey-Kramer test is equivalent to another approximation. For comparisons with a control when the correlation R does not have a factor analytic structure, Hsu (1992) suggests approximating R with a matrix R^* that does have such a structure and correspondingly approximating $q^t(\alpha, \nu, R)$ with $q^t(\alpha, \nu, R^*)$. When you request Dunnett's test for LS-means (the PDIFF=CONTROL and ADJUST=DUNNETT options), the GLM procedure automatically uses Hsu's approximation when appropriate.

Finally, Edwards and Berry (1987) suggest calculating $q^t(\alpha, \nu, R)$ by simulation. Multivariate t vectors are sampled from a distribution with the appropriate ν and R parameters, and Edwards and Berry (1987) suggest estimating $q^t(\alpha, \nu, R)$ by $\hat{q}$, the α percentile of the observed values of $\max(|t_1|, \ldots, |t_n|)$. Sufficient samples are generated for the true $P(\max(|t_1|, \ldots, |t_n|) > \hat{q})$ to be within a certain accuracy radius γ of α with accuracy confidence $100(1-\epsilon)$. You can approximate $q^t(\alpha, \nu, R)$ by simulation for comparisons between LS-means by specifying ADJUST=SIM (with either PDIFF=ALL or PDIFF=CONTROL). By default, $\gamma = 0.005$ and $\epsilon = 0.01$, so that the tail area of $\hat{q}$ is within 0.005 of α with 99% confidence. You can use the ACC= and EPS= options with ADJUST=SIM to reset γ and ϵ, or you can use the NSAMP= option to set the sample size directly. You can also control the random number sequence with the SEED= option.

Hsu and Nelson (1998) suggest a more accurate simulation method for estimating $q^t(\alpha, \nu, R)$, using a control variate adjustment technique. The same independent, standardized normal variates that are used to generate multivariate t vectors from a distribution with the appropriate ν and R parameters are also used to generate multivariate t vectors from a distribution for which the exact value of $q^t(\alpha, \nu, R)$ is known. $\max(|t_1|, \ldots, |t_n|)$ for the second sample is used as a control variate for adjusting the quantile estimate based on the first sample; refer to Hsu and Nelson (1998) for more details. The control variate adjustment has the drawback that it takes somewhat longer than the crude technique of Edwards and Berry (1987), but it typically yields an estimate that is many times more accurate. In most cases, if you are using ADJUST=SIM, then you should specify ADJUST=SIM(CVADJUST). You can also specify ADJUST=SIM(CVADJUST REPORT) to display a summary of the simulation that includes, among other things, the actual accuracy radius γ, which should be substantially smaller than the target accuracy radius (0.005 by default).

Multiple-Stage Tests

You can use all of the methods discussed so far to obtain simultaneous confidence intervals (Miller 1981). By sacrificing the facility for simultaneous estimation, you can obtain simultaneous tests with greater power using multiple-stage tests (MSTs). MSTs come in both step-up and step-down varieties (Welsch 1977). The step-down methods, which have been more widely used, are available in SAS/STAT software.

Step-down MSTs first test the homogeneity of all of the means at a level γ_k. If the test results in a rejection, then each subset of $k-1$ means is tested at level γ_{k-1}; otherwise, the procedure stops. In general, if the hypothesis of homogeneity of a set of p means is rejected at the γ_p level, then each subset of $p-1$ means is tested at the γ_{p-1} level; otherwise, the set of p means is considered not to differ significantly and none of its subsets are tested. The many varieties of MSTs that have been proposed differ in the levels γ_p and the statistics on which the subset tests are based. Clearly, the EERC of a step-down MST is not greater than γ_k, and the CER is not greater than γ_2, but the MEER is a complicated function of γ_p, $p = 2, \ldots, k$.

With unequal cell sizes, PROC GLM uses the harmonic mean of the cell sizes as the common sample size. However, since the resulting operating characteristics can be undesirable, MSTs are recommended only for the balanced case. When the sample sizes are equal and if the range statistic is used, you can arrange the means in ascending or descending order and test only contiguous subsets. But if you specify the F statistic, this shortcut cannot be taken. For this reason, only range-based MSTs are implemented. It is common practice to report the results of an MST by writing the means in such an order and drawing lines parallel to the list of means spanning the homogeneous subsets. This form of presentation is also convenient for pairwise comparisons with equal cell sizes.

The best known MSTs are the Duncan (the DUNCAN option) and Student-Newman-Keuls (the SNK option) methods (Miller 1981). Both use the studentized range statistic and, hence, are called *multiple range tests*. Duncan's method is often called the "new" multiple range test despite the fact that it is one of the oldest MSTs in current use.

The Duncan and SNK methods differ in the γ_p values used. For Duncan's method, they are

$$\gamma_p \quad = \quad 1-(1-\alpha)^{p-1}$$

whereas the SNK method uses

$$\gamma_p \quad = \quad \alpha$$

Duncan's method controls the CER at the α level. Its operating characteristics appear similar to those of Fisher's unprotected LSD or repeated t tests at level α (Petrinovich and Hardyck 1969). Since repeated t tests are easier to compute, easier to explain, and applicable to unequal sample sizes, Duncan's method is not recommended. Several published studies (for example, Carmer and Swanson (1973)) have claimed that Duncan's method is superior to Tukey's because of greater power without considering that the greater power of Duncan's method is due to its higher type 1 error rate (Einot and Gabriel 1975).

The SNK method holds the EERC to the α level but does not control the MEER (Einot and Gabriel 1975). Consider ten population means that occur in five pairs such that means within a pair are equal, but there are large differences between pairs. If you make the usual sampling assumptions and also assume that the sample sizes are very large, all subset homogeneity hypotheses for three or more means are rejected. The SNK method then comes down to five independent tests, one for each pair, each at the α level. Letting α be 0.05, the probability of at least one false rejection is

$$1 - (1 - 0.05)^5 \quad = \quad 0.23$$

As the number of means increases, the MEER approaches 1. Therefore, the SNK method cannot be recommended.

A variety of MSTs that control the MEER have been proposed, but these methods are not as well known as those of Duncan and SNK. An approach developed by Ryan (1959; 1960), Einot and Gabriel (1975), and Welsch (1977) sets

$$\gamma_p \quad = \quad \begin{cases} 1 - (1 - \alpha)^{p/k} & \text{for } p < k - 1 \\ \alpha & \text{for } p \geq k - 1 \end{cases}$$

You can use range statistics, leading to what is called the REGWQ method after the authors' initials. If you assume that the sample means have been arranged in descending order from $\bar{y}_1$ through $\bar{y}_k$, the homogeneity of means $\bar{y}_i, \ldots, \bar{y}_j, i < j$, is rejected by REGWQ if

$$\bar{y}_i - \bar{y}_j \quad \geq \quad q(\gamma_p; p, \nu) \frac{s}{\sqrt{n}}$$

where $p = j - i + 1$ and the summations are over $u = i, \ldots, j$ (Einot and Gabriel 1975). To ensure that the MEER is controlled, the current implementation checks whether $q(\gamma_p; p, \nu)$ is monotonically increasing in p. If not, then a set of critical values that are increasing in p is substituted instead.

REGWQ appears to be the most powerful step-down MST in the current literature (for example, Ramsey 1978). Use of a preliminary F-test decreases the power of all the other multiple comparison methods discussed previously except for Scheffé's test.

Bayesian Approach

Waller and Duncan (1969) and Duncan (1975) take an approach to multiple comparisons that differs from all the methods previously discussed in minimizing the Bayes risk under additive loss rather than controlling type 1 error rates. For each pair of population means μ_i and μ_j, null (H_0^{ij}) and alternative (H_a^{ij}) hypotheses are defined:

$$\begin{aligned} H_0^{ij}: &\quad \mu_i - \mu_j \leq 0 \\ H_a^{ij}: &\quad \mu_i - \mu_j > 0 \end{aligned}$$

For any i, j pair, let d_0 indicate a decision in favor of H_0^{ij} and d_a indicate a decision in favor of H_a^{ij}, and let $\delta = \mu_i - \mu_j$. The loss function for the decision on the i, j pair is

$$L(d_0 \mid \delta) \quad = \quad \begin{cases} 0 & \text{if } \delta \leq 0 \\ \delta & \text{if } \delta > 0 \end{cases}$$

$$L(d_a \mid \delta) \quad = \quad \begin{cases} -k\delta & \text{if } \delta \leq 0 \\ 0 & \text{if } \delta > 0 \end{cases}$$

where k represents a constant that you specify rather than the number of means. The loss for the joint decision involving all pairs of means is the sum of the losses for each individual decision. The population means are assumed to have a normal prior distribution with unknown variance, the logarithm of the variance of the means having a uniform prior distribution. For the i, j pair, the null hypothesis is rejected if

$$\bar{y}_i - \bar{y}_j \quad \geq \quad t_B s \sqrt{\frac{2}{n}}$$

where t_B is the Bayesian t value (Waller and Kemp 1976) depending on k, the F statistic for the one-way ANOVA, and the degrees of freedom for F. The value of t_B is a decreasing function of F, so the Waller-Duncan test (specified by the WALLER option) becomes more liberal as F increases.

Recommendations

In summary, if you are interested in several individual comparisons and are not concerned about the effects of multiple inferences, you can use repeated t tests or Fisher's unprotected LSD. If you are interested in all pairwise comparisons or all comparisons with a control, you should use Tukey's or Dunnett's test, respectively, in order to make the strongest possible inferences. If you have weaker inferential requirements and, in particular, if you don't want confidence intervals for the mean differences, you should use the REGWQ method. Finally, if you agree with the Bayesian approach and Waller and Duncan's assumptions, you should use the Waller-Duncan test.

Interpretation of Multiple Comparisons

When you interpret multiple comparisons, remember that failure to reject the hypothesis that two or more means are equal should not lead you to conclude that the population means are, in fact, equal. Failure to reject the null hypothesis implies only that the difference between population means, if any, is not large enough to be detected with the given sample size. A related point is that nonsignificance is nontransitive:

that is, given three sample means, the largest and smallest may be significantly different from each other, while neither is significantly different from the middle one. Nontransitive results of this type occur frequently in multiple comparisons.

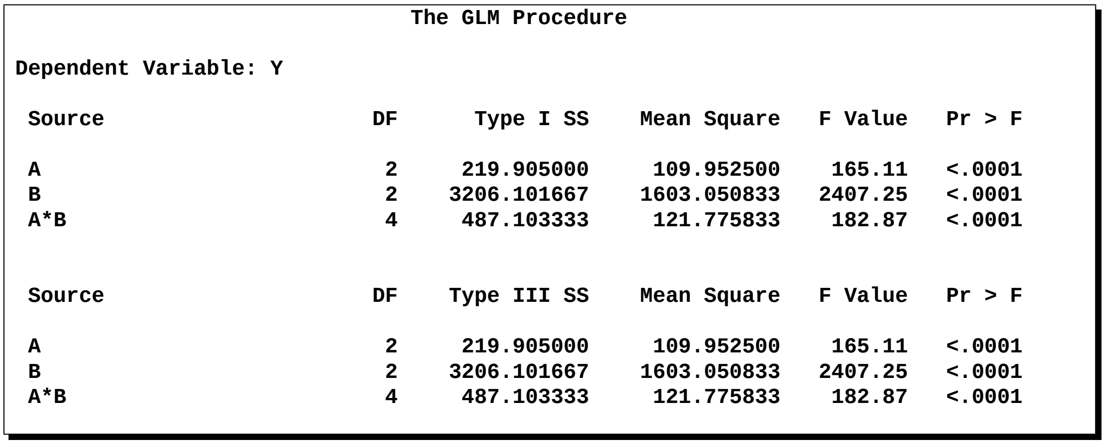
Multiple comparisons can also lead to counter-intuitive results when the cell sizes are unequal. Consider four cells labeled A, B, C, and D, with sample means in the order A>B>C>D. If A and D each have two observations, and B and C each have 10,000 observations, then the difference between B and C may be significant, while the difference between A and D is not.

Simple Effects

Suppose you use the following statements to fit a full factorial model to a two-way design:

```
data twoway;
   input A B Y @@;
   datalines;
1 1 10.6   1 1 11.0   1 1 10.6   1 1 11.3
1 2 -0.2   1 2  1.3   1 2 -0.2   1 2  0.2
1 3  0.1   1 3  0.4   1 3 -0.4   1 3  1.0
2 1 19.7   2 1 19.3   2 1 18.5   2 1 20.4
2 2 -0.2   2 2  0.5   2 2  0.8   2 2 -0.4
2 3 -0.9   2 3 -0.1   2 3 -0.2   2 3 -1.7
3 1 29.7   3 1 29.6   3 1 29.0   3 1 30.2
3 2  1.5   3 2  0.2   3 2 -1.5   3 2  1.3
3 3  0.2   3 3  0.4   3 3 -0.4   3 3 -2.2
;
proc glm data=twoway;
   class A B;
   model Y = A B A*B;
run;
```

Partial results for the analysis of variance are shown in Figure 32.17. The Type I and Type III results are the same because this is a balanced design.



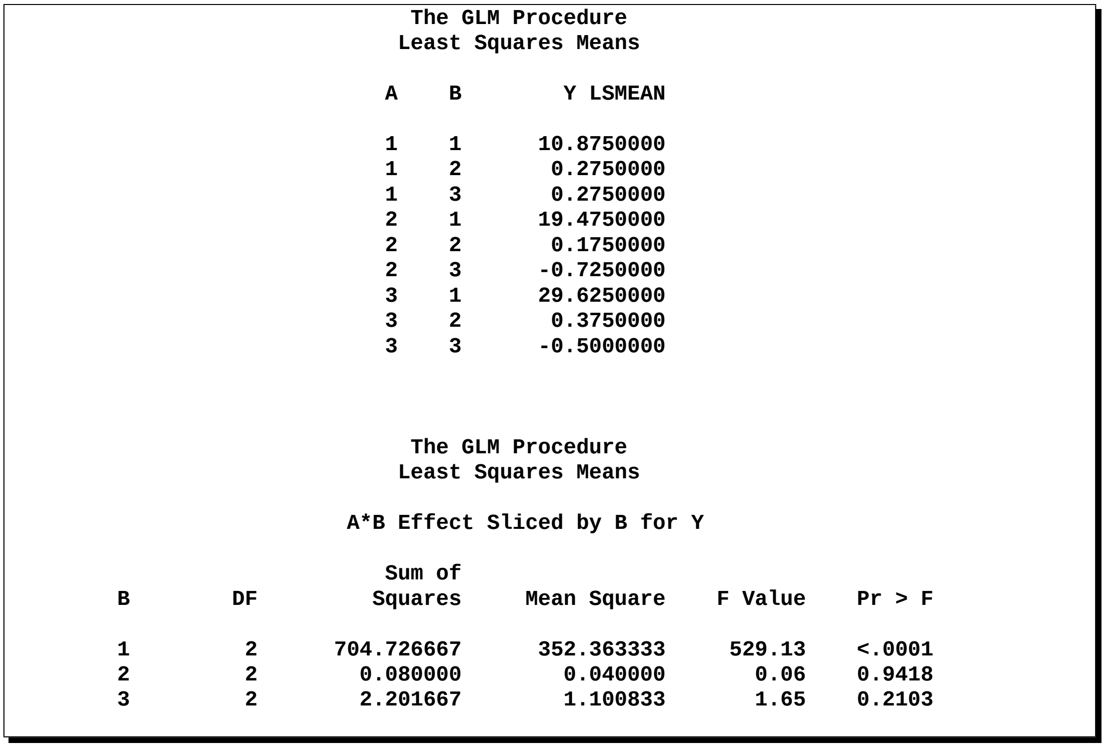
```
                               The GLM Procedure

Dependent Variable: Y

 Source                     DF       Type I SS     Mean Square   F Value   Pr > F

 A                           2      219.905000      109.952500    165.11   <.0001
 B                           2     3206.101667     1603.050833   2407.25   <.0001
 A*B                         4      487.103333      121.775833    182.87   <.0001


 Source                     DF     Type III SS     Mean Square   F Value   Pr > F

 A                           2      219.905000      109.952500    165.11   <.0001
 B                           2     3206.101667     1603.050833   2407.25   <.0001
 A*B                         4      487.103333      121.775833    182.87   <.0001
```

Figure 32.17. Two-way Design with Significant Interaction

The interaction A*B is significant, indicating that the effect of A depends on the level of B. In some cases, you may be interested in looking at the differences between predicted values across A for different levels of B. Winer (1971) calls this the *simple effects* of A. You can compute simple effects with the LSMEAN statement by specifying the SLICE= option. In this case, since the GLM procedure is interactive, you can compute the simple effects of A by submitting the following statements after the preceding statements.

```
lsmeans A*B / slice=B;
run;
```

The results are shown Figure 32.18. Note that A has a significant effect for B=1 but not for B=2 and B=3.



```
                          The GLM Procedure
                         Least Squares Means

                       A    B        Y LSMEAN

                       1    1      10.8750000
                       1    2       0.2750000
                       1    3       0.2750000
                       2    1      19.4750000
                       2    2       0.1750000
                       2    3      -0.7250000
                       3    1      29.6250000
                       3    2       0.3750000
                       3    3      -0.5000000



                          The GLM Procedure
                         Least Squares Means

                     A*B Effect Sliced by B for Y

                            Sum of
      B        DF          Squares     Mean Square    F Value    Pr > F

      1         2       704.726667      352.363333     529.13    <.0001
      2         2         0.080000        0.040000       0.06    0.9418
      3         2         2.201667        1.100833       1.65    0.2103
```

Figure 32.18. Interaction LS-means and Simple Effects

Homogeneity of Variance in One-Way Models

One of the usual assumptions for the GLM procedure is that the underlying errors are all uncorrelated with homogeneous variances (see page 1783). You can test this assumption in PROC GLM by using the HOVTEST option in the MEANS statement, requesting a *homogeneity of variance* test. This section discusses the computational details behind these tests. Note that the GLM procedure allows homogeneity of variance testing for simple one-way models only. Homogeneity of variance testing for more complex models is a subject of current research.

Bartlett (1937) proposes a test for equal variances that is a modification of the normal-theory likelihood ratio test (the HOVTEST=BARTLETT option). While Bartlett's test has accurate Type I error rates and optimal power when the underlying distribution of the data is normal, it can be very inaccurate if that distribution is even slightly nonnormal (Box 1953). Therefore, Bartlett's test is not recommended for routine use.

An approach that leads to tests that are much more robust to the underlying distribution is to transform the original values of the dependent variable to derive a *dispersion variable* and then to perform analysis of variance on this variable. The significance level for the test of homogeneity of variance is the *p*-value for the ANOVA *F*-test on the dispersion variable. All of the homogeneity of variance tests available in PROC GLM except Bartlett's use this approach.

Levene's test (Levene 1960) is widely considered to be the standard homogeneity of variance test (the HOVTEST=LEVENE option). Levene's test is of the dispersion-variable-ANOVA form discussed previously, where the dispersion variable is either

$$\begin{aligned} z_{ij}^2 &= (y_{ij} - \bar{y}_i)^2 && \text{(TYPE=SQUARE, the default)} \\ z_{ij} &= |y_{ij} - \bar{y}_i| && \text{(TYPE=ABS)} \end{aligned}$$

O'Brien (1979) proposes a test (HOVTEST=OBRIEN) that is basically a modification of Levene's z_{ij}^2, using the dispersion variable

$$z_{ij}^W = \frac{(W + n_i - 2)n_i(y_{ij} - \bar{y}_i)^2 - W(n_i - 1)\sigma_i^2}{(n_i - 1)(n_i - 2)}$$

where n_i is the size of the i^{th} group and σ_i^2 is its sample variance. You can use the W= option in parentheses to tune O'Brien's z_{ij}^W dispersion variable to match the suspected kurtosis of the underlying distribution. The choice of the value of the W= option is rarely critical. By default, W=0.5, as suggested by O'Brien (1979; 1981).

Finally, Brown and Forsythe (1974) suggest using the absolute deviations from the group *medians*:

$$z_{ij}^{\text{BF}} = |y_{ij} - m_i|$$

where m_i is the median of the i^{th} group. You can use the HOVTEST=BF option to specify this test.

Simulation results (Conover et al. 1981; Olejnik and Algina 1987) show that, while all of these ANOVA-based tests are reasonably robust to the underlying distribution, the Brown-Forsythe test seems best at providing power to detect variance differences while protecting the Type I error probability. However, since the within-group medians are required for the Brown-Forsythe test, it can be resource intensive if there are very many groups or if some groups are very large.

If one of these tests rejects the assumption of homogeneity of variance, you should use Welch's ANOVA instead of the usual ANOVA to test for differences between group means. However, this conclusion holds only if you use one of the robust homogeneity of variance tests (that is, not for HOVTEST=BARTLETT); even then, any homogeneity of variance test has too little power to be relied upon always to detect when Welch's ANOVA is appropriate. Unless the group variances are extremely different or the number of groups is large, the usual ANOVA test is relatively robust when the groups are all about the same size. As Box (1953) notes, "To make the preliminary test on variances is rather like putting to sea in a rowing boat to find out whether conditions are sufficiently calm for an ocean liner to leave port!"

Example 32.10 on page 1892 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

Weighted Means

In previous releases, if you specified a WEIGHT statement and one or more of the multiple comparisons options, PROC GLM estimated the variance of the difference between weighted group means for group i and j as

$$MSE \times \left(\frac{1}{n_i} + \frac{1}{n_j}\right)$$

where MSE is the (weighted) mean square for error and n_i is the size of group i. This variance is involved in all of the multiple comparison methods. Beginning with Release 6.12, the variance of the difference between weighted group means for group i and j is computed as

$$MSE \times \left(\frac{1}{w_i} + \frac{1}{w_j}\right)$$

where w_i is the sum of the weights for the observations in group i.

Construction of Least-Squares Means

To construct a least-squares mean (LS-mean) for a given level of a given effect, construct a row vector L according to the following rules and use it in an ESTIMATE statement to compute the value of the LS-mean:

1. Set all L_i corresponding to covariates (continuous variables) to their mean value.
2. Consider effects contained by the given effect. Set the L_i corresponding to levels associated with the given level equal to 1. Set all other L_i in these effects equal to 0. (See Chapter 11, "The Four Types of Estimable Functions," for a definition of *containing*.)
3. Consider the given effect. Set the L_i corresponding to the given level equal to 1. Set the L_i corresponding to other levels equal to 0.
4. Consider the effects that contain the given effect. If these effects are not nested within the given effect, then set the L_i corresponding to the given level to $1/k$, where k is the number of such columns. If these effects are nested within the given effect, then set the L_i corresponding to the given level to $1/(k_1 k_2)$, where

k_1 is the number of nested levels within this combination of nested effects, and k_2 is the number of such combinations. For L_i corresponding to other levels, use 0.

5. Consider the other effects not yet considered. If there are no nested factors, then set all L_i corresponding to this effect to $1/j$, where j is the number of levels in the effect. If there are nested factors, then set all L_i corresponding to this effect to $1/(j_1 j_2)$, where j_1 is the number of nested levels within a given combination of nested effects and j_2 is the number of such combinations.

The consequence of these rules is that the sum of the Xs within any classification effect is 1. This set of Xs forms a linear combination of the parameters that is checked for estimability before it is evaluated.

For example, consider the following model:

```
proc glm;
   class A B C;
   model Y=A B A*B C Z;
   lsmeans A B A*B C;
run;
```

Assume A has 3 levels, B has 2 levels, and C has 2 levels, and assume that every combination of levels of A and B exists in the data. Assume also that Z is a continuous variable with an average of 12.5. Then the least-squares means are computed by the following linear combinations of the parameter estimates:

		A			B		A*B						C		
	μ	1	2	3	1	2	11	12	21	22	31	32	1	2	Z
LSM( )	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	1/2	1/2	12.5
LSM(A1)	1	1	0	0	1/2	1/2	1/2	1/2	0	0	0	0	1/2	1/2	12.5
LSM(A2)	1	0	1	0	1/2	1/2	0	0	1/2	1/2	0	0	1/2	1/2	12.5
LSM(A3)	1	0	0	1	1/2	1/2	0	0	0	0	1/2	1/2	1/2	1/2	12.5
LSM(B1)	1	1/3	1/3	1/3	1	0	1/3	0	1/3	0	1/3	0	1/2	1/2	12.5
LSM(B2)	1	1/3	1/3	1/3	0	1	0	1/3	0	1/3	0	1/3	1/2	1/2	12.5
LSM(AB11)	1	1	0	0	1	0	1	0	0	0	0	0	1/2	1/2	12.5
LSM(AB12)	1	1	0	0	0	1	0	1	0	0	0	0	1/2	1/2	12.5
LSM(AB21)	1	0	1	0	1	0	0	0	1	0	0	0	1/2	1/2	12.5
LSM(AB22)	1	0	1	0	0	1	0	0	0	1	0	0	1/2	1/2	12.5
LSM(AB31)	1	0	0	1	1	0	0	0	0	0	1	0	1/2	1/2	12.5
LSM(AB32)	1	0	0	1	0	1	0	0	0	0	0	1	1/2	1/2	12.5
LSM(C1)	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	1	0	12.5
LSM(C2)	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	0	1	12.5

Setting Covariate Values

By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option in the LSMEANS statement enables you to set the covariates to whatever values you consider interesting.

If there is an effect containing two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option leaves covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to crossproducts of covariates.

As an example, the following is a model with a classification variable A and two continuous variables, x1 and x2:

```
class A;
model y = A x1 x2 x1*x2;
```

The coefficients for the continuous effects with various AT specifications are shown in the following table.

Syntax	x1	x2	x1*x2
`lsmeans A;`	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1 x_2}$
`lsmeans A / at means;`	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} \cdot \overline{x_2}$
`lsmeans A / at x1=1.2;`	1.2	$\overline{x_2}$	$1.2 \cdot \overline{x_2}$
`lsmeans A / at (x1 x2)=(1.2 0.3);`	1.2	0.3	$1.2 \cdot 0.3$

For the first two LSMEANS statements, the A LS-mean coefficient for x1 is $\overline{x_1}$ (the mean of x1) and for x2 is $\overline{x_2}$ (the mean of x2). However, for the first LSMEANS statement, the coefficient for x1*x2 is $\overline{x_1 x_2}$, but for the second LSMEANS statement the coefficient is $\overline{x_1} \cdot \overline{x_2}$. The third LSMEANS statement sets the coefficient for x1 equal to 1.2 and leaves that for x2 at $\overline{x_2}$, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

If you specify a WEIGHT variable, then weighted means are used for the covariate values. Also, observations with missing dependent variables are included in computing the covariate means, unless these observations form a missing cell. You can use the E option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you desire.

The AT option is disabled if you specify the BYLEVEL option, in which case the coefficients for the covariates are set equal to their means within each level of the LS-mean effect in question.

Changing the Weighting Scheme

The standard LS-means have equal coefficients across classification effects; however, the OM option in the LSMEANS statement changes these coefficients to be proportional to those found in the input data set. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the original data set.

In computing the observed margins, PROC GLM uses all observations for which there are no missing independent variables, including those for which there are missing dependent variables. Also, if there is a WEIGHT variable, PROC GLM uses weighted margins to construct the LS-means coefficients. If the analysis data set is balanced or if you specify a simple one-way model, the LS-means will be unchanged by the OM option.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across the entire data set, PROC GLM computes separate margins for each level of the LS-mean effect in question. The resulting LS-means are actually equal to raw means in this case. The BYLEVEL option disables the AT option if it is specified.

Note that the MIXED procedure implements a more versatile form of the OM option, enabling you to specifying an alternative data set over which to compute observed margins. If you use the BYLEVEL option, too, then this data set is effectively the "population" over which the population marginal means are computed. See Chapter 46, "The MIXED Procedure," for more information.

You may want to use the E option in conjunction with either the OM or BYLEVEL option to check that the modified LS-means coefficients are the ones you desire. It is possible that the modified LS-means are not estimable when the standard ones are, or vice versa.

Multivariate Analysis of Variance

If you fit several dependent variables to the same effects, you may want to make tests jointly involving parameters of several dependent variables. Suppose you have p dependent variables, k parameters for each dependent variable, and n observations. The models can be collected into one equation:

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon$$

where $\mathbf{Y}$ is $n \times p$, $\mathbf{X}$ is $n \times k$, $\boldsymbol{\beta}$ is $k \times p$, and ϵ is $n \times p$. Each of the p models can be estimated and tested separately. However, you may also want to consider the joint distribution and test the p models simultaneously.

For multivariate tests, you need to make some assumptions about the errors. With p dependent variables, there are $n \times p$ errors that are independent across observations but not across dependent variables. Assume

$$\text{vec}(\epsilon) \sim N(\mathbf{0}, \mathbf{I}_n \otimes \boldsymbol{\Sigma})$$

where vec(ϵ) strings ϵ out by rows, $\otimes$ denotes Kronecker product multiplication, and $\boldsymbol{\Sigma}$ is $p \times p$. $\boldsymbol{\Sigma}$ can be estimated by

$$\mathbf{S} = \frac{\mathbf{e}'\mathbf{e}}{n-r} = \frac{(\mathbf{Y} - \mathbf{Xb})'(\mathbf{Y} - \mathbf{Xb})}{n-r}$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$, r is the rank of the $\mathbf{X}$ matrix, and $\mathbf{e}$ is the matrix of residuals.

If $\mathbf{S}$ is scaled to unit diagonals, the values in $\mathbf{S}$ are called *partial correlations of the Ys adjusting for the Xs*. This matrix can be displayed by PROC GLM if PRINTE is specified as a MANOVA option.

The multivariate general linear hypothesis is written

$$\mathbf{L}\beta\mathbf{M} = 0$$

You can form hypotheses for linear combinations across columns, as well as across rows of $\boldsymbol{\beta}$.

The MANOVA statement of the GLM procedure tests special cases where $\mathbf{L}$ corresponds to Type I, Type II, Type III, or Type IV tests, and $\mathbf{M}$ is the $p \times p$ identity matrix. These tests are joint tests that the given type of hypothesis holds for all dependent variables in the model, and they are often sufficient to test all hypotheses of interest.

Finally, when these special cases are not appropriate, you can specify your own $\mathbf{L}$ and $\mathbf{M}$ matrices by using the CONTRAST statement before the MANOVA statement and the M= specification in the MANOVA statement, respectively. Another alternative is to use a REPEATED statement, which automatically generates a variety of $\mathbf{M}$ matrices useful in repeated measures analysis of variance. See the "REPEATED Statement" section on page 1777 and the "Repeated Measures Analysis of Variance" section on page 1825 for more information.

One useful way to think of a MANOVA analysis with an $\mathbf{M}$ matrix other than the identity is as an analysis of a set of transformed variables defined by the columns of the $\mathbf{M}$ matrix. You should note, however, that PROC GLM always displays the $\mathbf{M}$ matrix in such a way that the transformed variables are defined by the rows, not the columns, of the displayed $\mathbf{M}$ matrix.

All multivariate tests carried out by the GLM procedure first construct the matrices $\mathbf{H}$ and $\mathbf{E}$ corresponding to the numerator and denominator, respectively, of a univariate F-test.

$$\begin{aligned}\mathbf{H} &= \mathbf{M}'(\mathbf{L}\mathbf{b})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{L}\mathbf{b})\mathbf{M} \\ \mathbf{E} &= \mathbf{M}'(\mathbf{Y}'\mathbf{Y} - \mathbf{b}'(\mathbf{X}'\mathbf{X})\mathbf{b})\mathbf{M}\end{aligned}$$

The diagonal elements of $\mathbf{H}$ and $\mathbf{E}$ correspond to the hypothesis and error SS for univariate tests. When the $\mathbf{M}$ matrix is the identity matrix (the default), these tests are for the original dependent variables on the left-hand side of the MODEL statement. When an $\mathbf{M}$ matrix other than the identity is specified, the tests are for transformed variables defined by the columns of the $\mathbf{M}$ matrix. These tests can be studied by requesting the SUMMARY option, which produces univariate analyses for each original or transformed variable.

Four multivariate test statistics, all functions of the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ (or $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$), are constructed:

- Wilks' lambda = $\det(\mathbf{E})/\det(\mathbf{H}+\mathbf{E})$
- Pillai's trace = $\mathrm{trace}(\mathbf{H}(\mathbf{H}+\mathbf{E})^{-1})$
- Hotelling-Lawley trace = $\mathrm{trace}(\mathbf{E}^{-1}\mathbf{H})$
- Roy's maximum root = λ, largest eigenvalue of $\mathbf{E}^{-1}\mathbf{H}$

By default, all four are reported with *p*-values based on F approximations, as discussed in the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures." Alternatively, if you specify MSTAT=EXACT on the associated MANOVA or REPEATED statement, *p*-values for three of the four tests are computed exactly (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's Greatest Root), and the *p*-values for the fourth (Pillai's trace) are based on an F-approximation that is more accurate than the default. See the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures," for more details on the exact calculations.

Repeated Measures Analysis of Variance

When several measurements are taken on the same experimental unit (person, plant, machine, and so on), the measurements tend to be correlated with each other. When the measurements represent qualitatively different things, such as weight, length, and width, this correlation is best taken into account by use of multivariate methods, such as multivariate analysis of variance. When the measurements can be thought of as responses to levels of an experimental factor of interest, such as time, treatment, or dose, the correlation can be taken into account by performing a repeated measures analysis of variance.

PROC GLM provides both univariate and multivariate tests for repeated measures for one response. For an overall reference on univariate repeated measures, refer to Winer (1971). The multivariate approach is covered in Cole and Grizzle (1966). For a discussion of the relative merits of the two approaches, see LaTour and Miniard (1983).

Another approach to analysis of repeated measures is via general mixed models. This approach can handle balanced as well as unbalanced or missing within-subject data, and it offers more options for modeling the within-subject covariance. The main drawback of the mixed models approach is that it generally requires iteration and, thus, may be less computationally efficient. For further details on this approach, see Chapter 46, "The MIXED Procedure," and Wolfinger and Chang (1995).

Organization of Data for Repeated Measures Analysis

In order to deal efficiently with the correlation of repeated measures, the GLM procedure uses the multivariate method of specifying the model, even if only a univariate analysis is desired. In some cases, data may already be entered in the univariate mode, with each repeated measure listed as a separate observation along with a variable that represents the experimental unit (subject) on which measurement is taken. Consider the following data set old:

```
SUBJ   GROUP    TIME       Y
   1       1       1      15
   1       1       2      19
   1       1       3      25
   2       1       1      21
   2       1       2      18
   2       1       3      17
   1       2       1      14
   1       2       2      12
   1       2       3      16
   2       2       1      11
   2       2       2      20
               .
               .
               .
  10       3       1      14
  10       3       2      18
  10       3       3      16
```

There are three observations for each subject, corresponding to measurements taken at times 1, 2, and 3. These data could be analyzed using the following statements:

```
proc glm data=old;
   class group subj time;
   model y=group subj(group) time group*time;
   test h=group e=subj(group);
run;
```

However, this analysis assumes subjects' measurements are uncorrelated across time. A repeated measures analysis does not make this assumption. It uses a data set new:

```
GROUP        Y1     Y2     Y3
    1        15     19     25
    1        21     18     17
    2        14     12     16
    2        11     20     21
                 .
                 .
                 .
    3        14     18     16
```

In the data set new, the three measurements for a subject are all in one observation. For example, the measurements for subject 1 for times 1, 2, and 3 are 15, 19, and 25. For these data, the statements for a repeated measures analysis (assuming default options) are

```
proc glm data=new;
   class group;
   model y1-y3=group / nouni;
   repeated time;
run;
```

To convert the univariate form of repeated measures data to the multivariate form, you can use a program like the following:

```
proc sort data=old;
   by group subj;
run;

data new(keep=y1-y3 group);
   array yy(3)  y1-y3;
   do time=1 to 3;
      set old;
      by group subj;
      yy(time)=y;
      if last.subj then return;
   end;
run;
```

Alternatively, you could use PROC TRANSPOSE to achieve the same results with a program like this one:

```
proc sort data=old;
   by group subj;
run;

proc transpose out=new(rename=(_1=y1 _2=y2 _3=y3));
   by group subj;
   id time;
run;
```

Refer to the discussions in *SAS Language Reference: Concepts* for more information on rearrangement of data sets.

Hypothesis Testing in Repeated Measures Analysis

In repeated measures analysis of variance, the effects of interest are

- between-subject effects (such as GROUP in the previous example)
- within-subject effects (such as TIME in the previous example)
- interactions between the two types of effects (such as GROUP*TIME in the previous example)

Repeated measures analyses are distinguished from MANOVA because of interest in testing hypotheses about the within-subject effects and the within-subject-by-between-subject interactions.

For tests that involve only between-subjects effects, both the multivariate and univariate approaches give rise to the same tests. These tests are provided for all effects in the MODEL statement, as well as for any CONTRASTs specified. The ANOVA table for these tests is labeled "Tests of Hypotheses for Between Subjects Effects" on the PROC GLM results. These tests are constructed by first adding together the dependent variables in the model. Then an analysis of variance is performed on the sum divided by the square root of the number of dependent variables. For example, the statements

```
model y1-y3=group;
repeated time;
```

give a one-way analysis of variance using $(Y1 + Y2 + Y3)/\sqrt{3}$ as the dependent variable for performing tests of hypothesis on the between-subject effect GROUP. Tests for between-subject effects are equivalent to tests of the hypothesis $\mathbf{L}\beta\mathbf{M} = 0$, where $\mathbf{M}$ is simply a vector of 1s.

For within-subject effects and for within-subject-by-between-subject interaction effects, the univariate and multivariate approaches yield different tests. These tests are provided for the within-subject effects and for the interactions between these effects and the other effects in the MODEL statement, as well as for any CONTRASTs specified. The univariate tests are displayed in a table labeled "Univariate Tests of Hypotheses for Within Subject Effects." Results for multivariate tests are displayed in a table labeled "Repeated Measures Analysis of Variance."

The multivariate tests provided for within-subjects effects and interactions involving these effects are Wilks' Lambda, Pillai's Trace, Hotelling-Lawley Trace, and Roy's maximum root. For further details on these four statistics, see the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures." As an example, the statements

```
model y1-y3=group;
repeated time;
```

produce multivariate tests for the within-subject effect TIME and the interaction TIME*GROUP.

The multivariate tests for within-subject effects are produced by testing the hypothesis $\mathbf{L}\beta\mathbf{M} = 0$, where the $\mathbf{L}$ matrix is the usual matrix corresponding to Type I, Type II, Type III, or Type IV hypotheses tests, and the $\mathbf{M}$ matrix is one of several matrices depending on the transformation that you specify in the REPEATED statement. The only assumption required for valid tests is that the dependent variables in the model have a multivariate normal distribution with a common covariance matrix across the between-subject effects.

The univariate tests for within-subject effects and interactions involving these effects require some assumptions for the probabilities provided by the ordinary F-tests to be correct. Specifically, these tests require certain patterns of covariance matrices, known as Type H covariances (Huynh and Feldt 1970). Data with these patterns in the covariance matrices are said to satisfy the Huynh-Feldt condition. You can test this assumption (and the Huynh-Feldt condition) by applying a sphericity test (Anderson 1958) to any set of variables defined by an orthogonal contrast transformation. Such a set of variables is known as a set of orthogonal components. When you use the PRINTE option in the REPEATED statement, this sphericity test is applied both to the transformed variables defined by the REPEATED statement and to a set of orthogonal components if the specified transformation is not orthogonal. It is the test applied to the orthogonal components that is important in determining whether your data have Type H covariance structure. When there are only two levels of the within-subject effect, there is only one transformed variable, and a sphericity test is not needed. The sphericity test is labeled "Test for Sphericity" on the output.

If your data satisfy the preceding assumptions, use the usual F-tests to test univariate hypotheses for the within-subject effects and associated interactions.

If your data do not satisfy the assumption of Type H covariance, an adjustment to numerator and denominator degrees of freedom can be used. Two such adjustments, based on a degrees of freedom adjustment factor known as ϵ (epsilon) (Box 1954), are provided in PROC GLM. Both adjustments estimate ϵ and then multiply the numerator and denominator degrees of freedom by this estimate before determining significance levels for the F-tests. Significance levels associated with the adjusted tests are labeled "Adj Pr > F" on the output. The first adjustment, initially proposed for use in data analysis by Greenhouse and Geisser (1959), is labeled "Greenhouse-Geisser Epsilon" and represents the maximum-likelihood estimate of Box's ϵ factor. Significance levels associated with adjusted F-tests are labeled "G-G" on the output. Huynh and Feldt (1976) have shown that the G-G estimate tends to be biased downward (that is, too conservative), especially for small samples, and they have proposed an alternative estimator that is constructed using unbiased estimators of the numerator and denominator of Box's ϵ. Huynh and Feldt's estimator is labeled "Huynh-Feldt Epsilon" on the PROC GLM output, and the significance levels associated with adjusted F-tests are labeled "H-F." Although ϵ must be in the range of 0 to 1, the H-F estimator can be outside this range. When the H-F estimator is greater than 1, a value of 1 is used in all calculations for probabilities, and the H-F probabilities are not adjusted. In summary, if your data do not meet the assumptions, use adjusted F-tests. However, when you strongly suspect that your data may not have Type H covariance, all these univariate tests should be interpreted cautiously. In such cases, you should consider using the multivariate tests instead.

The univariate sums of squares for hypotheses involving within-subject effects can be easily calculated from the $\mathbf{H}$ and $\mathbf{E}$ matrices corresponding to the multivariate tests described in the "Multivariate Analysis of Variance" section on page 1823. If the $\mathbf{M}$ matrix is orthogonal, the univariate sums of squares is calculated as the trace (sum of diagonal elements) of the appropriate $\mathbf{H}$ matrix; if it is not orthogonal, PROC GLM calculates the trace of the $\mathbf{H}$ matrix that results from an orthogonal $\mathbf{M}$ matrix transformation. The appropriate error term for the univariate F-tests is constructed in a similar way from the error SSCP matrix and is labeled Error(*factorname*), where *factorname* indicates the $\mathbf{M}$ matrix that is used in the transformation.

When the design specifies more than one repeated measures factor, PROC GLM computes the $\mathbf{M}$ matrix for a given effect as the direct (Kronecker) product of the $\mathbf{M}$ matrices defined by the REPEATED statement if the factor is involved in the effect or as a vector of 1s if the factor is not involved. The test for the main effect of a repeated-measures factor is constructed using an $\mathbf{L}$ matrix that corresponds to a test that the mean of the observation is zero. Thus, the main effect test for repeated measures is a test that the means of the variables defined by the $\mathbf{M}$ matrix are all equal to zero, while interactions involving repeated-measures effects are tests that the between-subjects factors involved in the interaction have no effect on the means of the transformed variables defined by the $\mathbf{M}$ matrix. In addition, you can specify other $\mathbf{L}$ matrices to test hypotheses of interest by using the CONTRAST statement, since hypotheses defined by CONTRAST statements are also tested in the REPEATED analysis. To see which combinations of the original variables the transformed variables represent, you can specify the PRINTM option in the REPEATED statement. This option displays the transpose of $\mathbf{M}$, which is labeled as M in the PROC GLM results. The tests produced are the same for any choice of transformation $(\mathbf{M})$ matrix specified in the REPEATED statement; however, depending on the nature of the repeated measurements being studied, a particular choice of transformation matrix, coupled with the CANONICAL or SUMMARY option, can provide additional insight into the data being studied.

Transformations Used in Repeated Measures Analysis of Variance

As mentioned in the specifications of the REPEATED statement, several different $\mathbf{M}$ matrices can be generated automatically, based on the transformation that you specify in the REPEATED statement. Remember that both the univariate and multivariate tests that PROC GLM performs are unaffected by the choice of transformation; the choice of transformation is important only when you are trying to study the nature of a repeated measures effect, particularly with the CANONICAL and SUMMARY options. If one of these matrices does not meet your needs for a particular analysis, you may want to use the M= option in the MANOVA statement to perform the tests of interest.

The following sections describe the transformations available in the REPEATED statement, provide an example of the $\mathbf{M}$ matrix that is produced, and give guidelines for the use of the transformation. As in the PROC GLM output, the displayed matrix is labeled M. This is the $\mathbf{M}'$ matrix.

CONTRAST Transformation

This is the default transformation used by the REPEATED statement. It is useful when one level of the repeated measures effect can be thought of as a control level against which the others are compared. For example, if five drugs are administered to each of several animals and the first drug is a control or placebo, the statements

```
proc glm;
   model d1-d5= / nouni;
```

```
      repeated drug 5 contrast(1) / summary printm;
   run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

When you examine the analysis of variance tables produced by the SUMMARY option, you can tell which of the drugs differed significantly from the placebo.

POLYNOMIAL Transformation

This transformation is useful when the levels of the repeated measure represent quantitative values of a treatment, such as dose or time. If the levels are unequally spaced, *level values* can be specified in parentheses after the number of levels in the REPEATED statement. For example, if five levels of a drug corresponding to 1, 2, 5, 10 and 20 milligrams are administered to different treatment groups, represented by the variable group, the statements

```
   proc glm;
      class group;
      model r1-r5=group / nouni;
      repeated dose 5 (1 2 5 10 20) polynomial / summary printm;
   run;
```

produce the following $\mathbf{M}$ matrix.

$$\mathbf{M} = \begin{bmatrix} -0.4250 & -0.3606 & -0.1674 & 0.1545 & 0.7984 \\ 0.4349 & 0.2073 & -0.3252 & -0.7116 & 0.3946 \\ -0.4331 & 0.1366 & 0.7253 & -0.5108 & 0.0821 \\ 0.4926 & -0.7800 & 0.3743 & -0.0936 & 0.0066 \end{bmatrix}$$

The SUMMARY option in this example provides univariate ANOVAs for the variables defined by the rows of this $\mathbf{M}$ matrix. In this case, they represent the linear, quadratic, cubic, and quartic trends for dose and are labeled dose_1, dose_2, dose_3, and dose_4, respectively.

HELMERT Transformation

Since the Helmert transformation compares a level of a repeated measure to the mean of subsequent levels, it is useful when interest lies in the point at which responses cease to change. For example, if four levels of a repeated measures factor represent responses to treatments administered over time to males and females, the statements

```
proc glm;
   class sex;
   model resp1-resp4=sex / nouni;
   repeated trtmnt 4 helmert / canon printm;
run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & -0.33333 & -0.33333 & -0.33333 \\ 0 & 1 & -0.50000 & -0.50000 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

MEAN Transformation

This transformation can be useful in the same types of situations in which the CONTRAST transformation is useful. If you substitute the following statement for the REPEATED statement shown in the “CONTRAST Transformation” section,

```
repeated drug 5 mean / printm;
```

the following $\mathbf{M}$ matrix is produced:

$$\mathbf{M} = \begin{bmatrix} 1 & -0.25 & -0.25 & -0.25 & -0.25 \\ -0.25 & 1 & -0.25 & -0.25 & -0.25 \\ -0.25 & -0.25 & 1 & -0.25 & -0.25 \\ -0.25 & -0.25 & -0.25 & 1 & -0.25 \end{bmatrix}$$

As with the CONTRAST transformation, if you want to omit a level other than the last, you can specify it in parentheses after the keyword MEAN in the REPEATED statement.

PROFILE Transformation

When a repeated measure represents a series of factors administered over time, but a polynomial response is unreasonable, a profile transformation may prove useful. As an example, consider a training program in which four different methods are employed to teach students at several different schools. The repeated measure is the score on tests administered after each of the methods is completed. The statements

```
proc glm;
   class school;
   model t1-t4=school / nouni;
   repeated method 4 profile / summary nom printm;
run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

To determine the point at which an improvement in test scores takes place, you can examine the analyses of variance for the transformed variables representing the differences between adjacent tests. These analyses are requested by the SUMMARY option in the REPEATED statement, and the variables are labeled METHOD.1, METHOD.2, and METHOD.3.

Random Effects Analysis

When some model effects are random (that is, assumed to be sampled from a normal population of effects), you can specify these effects in the RANDOM statement in order to compute the expected values of mean squares for various model effects and contrasts and, optionally, to perform random effects analysis of variance tests.

PROC GLM versus PROC MIXED for Random Effects Analysis

Other SAS procedures that can be used to analyze models with random effects include the MIXED and VARCOMP procedures. Note that, for these procedures, the random effects specification is an integral part of the model, affecting how both random and fixed effects are fit; for PROC GLM, the random effects are treated in a *post hoc* fashion after the complete fixed effect model is fit. This distinction affects other features in the GLM procedure, such as the results of the LSMEANS and ESTIMATE statements. These features assume that all effects are fixed, so that all tests and estimability checks for these statements are based on a fixed effects model, even when you use a RANDOM statement. Standard errors for estimates and LS-means based on the fixed effects model may be significantly smaller than those based on a true random effects model; in fact, some functions that are estimable under a true random effects model may not even be estimable under the fixed effects model. Therefore, you should use the MIXED procedure to compute tests involving these features that take the random effects into account; see Chapter 46, “The MIXED Procedure,” for more information.

Note that, for balanced data, the test statistics computed when you specify the TEST option on the RANDOM statement have an exact F distribution only when the design is balanced; for unbalanced designs, the p values for the F-tests are approximate. For balanced data, the values obtained by PROC GLM and PROC MIXED agree; for unbalanced data, they usually do not.

Computation of Expected Mean Squares for Random Effects

The RANDOM statement in PROC GLM declares one or more effects in the model to be random rather than fixed. By default, PROC GLM displays the coefficients of the expected mean squares for all terms in the model. In addition, when you specify the TEST option in the RANDOM statement, the procedure determines what tests are appropriate and provides F ratios and probabilities for these tests.

The expected mean squares are computed as follows. Consider the model

$$Y = X_0\boldsymbol{\beta}_0 + X_1\boldsymbol{\beta}_1 + \cdots + X_k\boldsymbol{\beta}_k + \epsilon$$

where $\boldsymbol{\beta}_0$ represents the fixed effects and $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \epsilon$ represent the random effects. Random effects are assumed to be normally and independently distributed. For any $\mathbf{L}$ in the row space of $\mathbf{X} = (X_0 \mid X_1 \mid X_2 \mid \cdots \mid X_k)$, the expected value of the sum of squares for $\mathbf{L}\beta$ is

$$E(\mathrm{SS}_L) = \boldsymbol{\beta}_0'\mathbf{C}_0'\mathbf{C}_0\boldsymbol{\beta}_0 + \mathrm{SSQ}(\mathbf{C}_1)\sigma_1^2 + \mathrm{SSQ}(\mathbf{C}_2)\sigma_2^2 + \cdots + \mathrm{SSQ}(\mathbf{C}_k)\sigma_k^2 + \mathrm{rank}(\mathbf{L})\sigma_\epsilon^2$$

where $\mathbf{C}$ is of the same dimensions as $\mathbf{L}$ and is partitioned as the $\mathbf{X}$ matrix. In other words,

$$\mathbf{C} = (\mathbf{C}_0 \mid \mathbf{C}_1 \mid \cdots \mid \mathbf{C}_k)$$

Furthermore, $\mathbf{C} = \mathbf{ML}$, where $\mathbf{M}$ is the inverse of the lower triangular Cholesky decomposition matrix of $\mathbf{L}(\mathbf{X}'\mathbf{X})^-\mathbf{L}'$. SSQ($\mathbf{A}$) is defined as $\mathrm{tr}(\mathbf{A}'\mathbf{A})$.

For the model in the following MODEL statement

```
model Y=A B(A) C A*C;
random B(A);
```

with B(A) declared as random, the expected mean square of each effect is displayed as

$$\mathrm{Var(Error)} + \textit{constant} \times \mathrm{Var}(\mathsf{B(A)}) + Q(\mathsf{A}, \mathsf{C}, \mathsf{A} * \mathsf{C})$$

If any fixed effects appear in the expected mean square of an effect, the letter Q followed by the list of fixed effects in the expected value is displayed. The actual numeric values of the quadratic form ($\mathbf{Q}$ matrix) can be displayed using the Q option.

To determine appropriate means squares for testing the effects in the model, the TEST option in the RANDOM statement performs the following.

1. First, it forms a matrix of coefficients of the expected mean squares of those effects that were declared to be random.
2. Next, for each effect in the model, it determines the combination of these expected mean squares that produce an expectation that includes all the terms in the expected mean square of the effect of interest except the one corresponding to the effect of interest. For example, if the expected mean square of an effect A*B is

   $$\mathrm{Var(Error)} + 3 \times \mathrm{Var}(\mathsf{A}) + \mathrm{Var}(\mathsf{A} * \mathsf{B})$$

PROC GLM determines the combination of other expected mean squares in the model that has expectation

$$\text{Var(Error)} + 3 \times \text{Var}(\mathsf{A})$$

3. If the preceding criterion is met by the expected mean square of a single effect in the model (as is often the case in balanced designs), the F test is formed directly. In this case, the mean square of the effect of interest is used as the numerator, the mean square of the single effect with an expected mean square that satisfies the criterion is used as the denominator, and the degrees of freedom for the test are simply the usual model degrees of freedom.
4. When more than one mean square must be combined to achieve the appropriate expectation, an approximation is employed to determine the appropriate degrees of freedom (Satterthwaite 1946). When effects other than the effect of interest are listed after the Q in the output, tests of hypotheses involving the effect of interest are not valid unless all other fixed effects involved in it are assumed to be zero. When tests such as these are performed by using the TEST option in the RANDOM statement, a note is displayed reminding you that further assumptions are necessary for the validity of these tests. Remember that although the tests are not valid unless these assumptions are made, this does not provide a basis for these assumptions to be true. The particulars of a given experiment must be examined to determine whether the assumption is reasonable.

Refer to Goodnight and Speed (1978), Milliken and Johnson (1984, Chapters 22 and 23), and Hocking (1985) for further theoretical discussion.

Sum-to-Zero Assumptions

The formulation and parameterization of the expected mean squares for random effects in mixed models is an ongoing item of controversy in the statistical literature. Confusion arises over whether or not to assume that terms involving fixed effects sum to zero. Cornfield and Tukey (1956), Winer (1971), and others assume that they do sum to zero; Searle (1971), Hocking (1973), and others (including PROC GLM) do not.

Different assumptions about these sum-to-zero constraints can lead to different expected mean squares for certain terms, and hence to different F and p values.

For arguments in favor of not assuming that terms involving fixed effects sum to zero, see Section 9.7 of Searle (1971) and Sections 1 and 4 of McLean et al. (1991). Other references are Hartley and Searle (1969) and Searle et al. (1992).

Computing Type I, II, and IV Expected Mean Squares

When you use the RANDOM statement, by default the GLM procedure produces the Type III expected mean squares for model effects and for contrasts specified before the RANDOM statement. In order to obtain expected values for other types of mean squares, you need to specify which types of mean squares are of interest in the MODEL statement. For example, in order to obtain the Type IV expected mean squares for effects in the RANDOM and CONTRAST statements, specify the SS4 option in the MODEL statement. If you want both Type III and Type IV expected mean squares, specify both the SS3 and SS4 options in the MODEL statement. Since the estimable function basis is not automatically calculated for Type I and Type II SS, the E1 (for Type I) or E2 (for Type II) option must be specified in the MODEL statement in order for the RANDOM statement to produce the expected mean squares for the Type I or Type II sums of squares. Note that it is important to list the fixed effects first in the MODEL statement when requesting the Type I expected mean squares.

For example, suppose you have a two-way design with factors A and B in which the main effect for B and the interaction are random. In order to compute the Type III expected mean squares (in addition to the fixed-effect analysis), you can use the following statements:

```
proc glm;
   class A B;
   model Y = A B A*B;
   random B A*B;
run;
```

If you use the SS4 option in the MODEL statement,

```
proc glm;
   class A B;
   model Y = A B A*B / ss4;
   random B A*B;
run;
```

then only the Type IV expected mean squares are computed (as well as the Type IV fixed-effect tests). For the Type I expected mean squares, you can use the following statements:

```
proc glm;
   class A B;
   model Y = A B A*B / e1;
   random B A*B;
run;
```

For each of these cases, in order to perform random effect analysis of variance tests for each effect specified in the model, you need to specify the TEST option in the RANDOM statement, as follows:

```
proc glm;
   class A B;
   model Y = A B A*B;
   random B A*B / test;
run;
```

The GLM procedure automatically determines the appropriate error term for each test, based on the expected mean squares.

Missing Values

For an analysis involving one dependent variable, PROC GLM uses an observation if values are nonmissing for that dependent variable and all the class variables.

For an analysis involving multiple dependent variables without the MANOVA or REPEATED statement, or without the MANOVA option in the PROC GLM statement, a missing value in one dependent variable does not eliminate the observation from the analysis of other nonmissing dependent variables. On the other hand, for an analysis with the MANOVA or REPEATED statement, or with the MANOVA option in the PROC GLM statement, PROC GLM uses an observation if values are nonmissing for all dependent variables and all the variables used in independent effects.

During processing, the GLM procedure groups the dependent variables by their pattern of missing values across observations so that sums and crossproducts can be collected in the most efficient manner.

If your data have different patterns of missing values among the dependent variables, interactivity is disabled. This can occur when some of the variables in your data set have missing values and

- you do not use the MANOVA option in the PROC GLM statement
- you do not use a MANOVA or REPEATED statement before the first RUN statement

Note that the REG procedure handles missing values differently in this case; see Chapter 61, "The REG Procedure," for more information.

Computational Resources

Memory

For large problems, most of the memory resources are required for holding the $\mathbf{X}'\mathbf{X}$ matrix of the sums and crossproducts. The section "Parameterization of PROC GLM Models" on page 1787 describes how columns of the $\mathbf{X}$ matrix are allocated for various types of effects. For each level that occurs in the data for a combination of class variables in a given effect, a row and column for $\mathbf{X}'\mathbf{X}$ is needed.

The following example illustrates the calculation. Suppose A has 20 levels, B has 4 levels, and C has 3 levels. Then consider the model

```
proc glm;
  class A B C;
  model Y1 Y2 Y3=A B A*B C A*C B*C A*B*C X1 X2;
run;
```

The $\mathbf{X}'\mathbf{X}$ matrix (bordered by $\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}'\mathbf{Y}$) can have as many as 425 rows and columns:

1	for the intercept term
20	for A
4	for B
80	for A*B
3	for C
60	for A*C
12	for B*C
240	for A*B*C
2	for X1 and X2 (continuous variables)
3	for Y1, Y2, and Y3 (dependent variables)

The matrix has 425 rows and columns only if all combinations of levels occur for each effect in the model. For m rows and columns, $8m^2$ bytes are needed for crossproducts. In this case, $8 \cdot 425^2 = 1,445,000$ bytes, or about $1,445,000/1024 = 1411K$.

The required memory grows as the square of the number of columns of $\mathbf{X}$; most of the memory is for the A*B*C interaction. Without A*B*C, you have 185 columns and need 268K for $\mathbf{X}'\mathbf{X}$. Without either A*B*C or A*B, you need 86K. If A is recoded to have ten levels, then the full model has only 220 columns and requires 378K.

The second time that a large amount of memory is needed is when Type III, Type IV, or contrast sums of squares are being calculated. This memory requirement is a function of the number of degrees of freedom of the model being analyzed and the maximum degrees of freedom for any single source. Let Rank equal the sum of the model degrees of freedom, MaxDF be the maximum number of degrees of freedom for any single source, and N_y be the number of dependent variables in the model. Then the memory requirement in bytes is

$$\begin{aligned}\left(8 \times \left(\frac{\text{Rank } \times (\text{Rank } + 1)}{2}\right)\right) &+ (N_y \times \text{Rank })\\ &+ \left(\frac{\text{MaxDF} \times (\text{MaxDF} + 1)}{2}\right)\\ &+ (N_y \times \text{MaxDF})\end{aligned}$$

Unfortunately, these quantities are not available when the $\mathbf{X}'\mathbf{X}$ matrix is being constructed, so PROC GLM may occasionally request additional memory even after you have increased the memory allocation available to the program.

If you have a large model that exceeds the memory capacity of your computer, these are your options:

- Eliminate terms, especially high-level interactions.
- Reduce the number of levels for variables with many levels.
- Use the ABSORB statement for parts of the model that are large.

- Use the REPEATED statement for repeated measures variables.
- Use PROC ANOVA or PROC REG rather than PROC GLM, if your design allows.

A related limitation is that for any model effect involving classification variables (interactions as well as main effects), the number of levels can not exceed 32,767. This is because GLM internally indexes effect levels using signed short (16-bit) integers, for which the maximum value is $2^{15} - 1 = 32,767$.

CPU Time

Typically, if the GLM procedure requires a lot of CPU time, it will be for one of several reasons. Suppose that the input data has n rows (observations) and the model has E effects which together produce a design matrix $\mathbf{X}$ with m columns. Then if m or n is relatively large, the procedure may spend a lot of time in any of the following areas:

- collecting the sums of squares and crossproducts
- solving the normal equations
- computing the Type III tests

The time required for collecting sums and crossproducts is difficult to calculate because it is a complicated function of the model. The worst case occurs if all columns are continuous variables, involving $nm^2/2$ multiplications and additions. If the columns are levels of a classification, then only m sums may be needed, but a significant amount of time may be spent in look-up operations. Solving the normal equations requires time for approximately $m^3/2$ multiplications and additions, and the number of operations required to compute the Type III tests is also proportional to both E and m^3.

Suppose that you know that Type IV sums of squares are appropriate for the model you are analyzing (for example, if your design has no missing cells). You can specify the SS4 option in your MODEL statement, which saves CPU time by requesting the Type IV sums of squares instead of the more computationally burdensome Type III sums of squares. This proves especially useful if you have a factor in your model that has many levels and is involved in several interactions.

If the operating system enables SAS to run parallel computational threads on multiple CPUs, then both the solution of the normal equations and the computation of Type III tests can take advantage of this to reduce the computational time for large models. In solving the normal equations, the fundamental row sweep operations (Goodnight 1979) are performed in parallel. In computing the Type III tests, both the orthogonalization for the estimable functions and the sums of squares calculation have been parallelized.

The reduction in computational time due to parallel processing depends on the size of the model, the number of processors, and the parallel architecture of the operating system. If the model is large enough that the overwhelming proportion of CPU time for the procedure is accounted for in solving the normal equations and/or computing the Type III tests, then you can expect a reduction in computational time approximately inversely proportional to the number of CPUs. However, as you increase the number of processors, the efficiency of this scaling can be reduced by several effects. One mitigating factor is a purely mathematical one known as “Amdahl’s Law”, which is related to the fact that only part of the processing time for the procedure can be parallelized. Even taking Amdahl’s Law into account, the parallelization efficiency can be reduced by cache effects related to how fast the multiple processors can access memory. See Cohen (2002) for a discussion of these issues. For additional information on parallel processing in SAS, refer to the chapter on “Support for Parallel Processing” in *SAS Language Reference: Concepts*.

Computational Method

Let $\mathbf{X}$ represent the $n \times p$ design matrix and $\mathbf{Y}$ the $n \times 1$ vector of dependent variables. (See the section “Parameterization of PROC GLM Models” on page 1787 for information on how $\mathbf{X}$ is formed from your model specification.)

The normal equations $\mathbf{X}'\mathbf{X}\beta = \mathbf{X}'\mathbf{Y}$ are solved using a modified sweep routine that produces a generalized (g2) inverse $(\mathbf{X}'\mathbf{X})^-$ and a solution $\mathbf{b} = (\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{y}$ (Pringle and Raynor 1971).

For each effect in the model, a matrix $\mathbf{L}$ is computed such that the rows of $\mathbf{L}$ are estimable. Tests of the hypothesis $\mathbf{L}\beta = 0$ are then made by first computing

$$\mathrm{SS}(\mathbf{L}\beta = 0) = (\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^-\mathbf{L}')^{-1}(\mathbf{Lb})$$

and then computing the associated F value using the mean squared error.

Output Data Sets

OUT= Data Set Created by the OUTPUT Statement

The OUTPUT statement produces an output data set that contains the following:

- all original data from the SAS data set input to PROC GLM
- the new variables corresponding to the diagnostic measures specified with statistics keywords in the OUTPUT statement (PREDICTED=, RESIDUAL=, and so on).

With multiple dependent variables, a name can be specified for any of the diagnostic measures for each of the dependent variables in the order in which they occur in the MODEL statement.

For example, suppose that the input data set A contains the variables y1, y2, y3, x1, and x2. Then you can use the following statements:

```
proc glm data=A;
   model y1 y2 y3=x1;
   output out=out p=y1hat y2hat y3hat
                  r=y1resid lclm=y1lcl uclm=y1ucl;
run;
```

The output data set out contains y1, y2, y3, x1, x2, y1hat, y2hat, y3hat, y1resid, y1lcl, and y1ucl. The variable x2 is output even though it is not used by PROC GLM. Although predicted values are generated for all three dependent variables, residuals are output for only the first dependent variable.

When any independent variable in the analysis (including all class variables) is missing for an observation, then all new variables that correspond to diagnostic measures are missing for the observation in the output data set.

When a dependent variable in the analysis is missing for an observation, then some new variables that correspond to diagnostic measures are missing for the observation in the output data set, and some are still available. Specifically, in this case, the new variables that correspond to COOKD, COVRATIO, DFFITS, PRESS, R, RSTUDENT, STDR, and STUDENT are missing in the output data set. The variables corresponding to H, LCL, LCLM, P, STDI, STDP, UCL, and UCLM are not missing.

OUT= Data Set Created by the LSMEANS Statement

The OUT= option in the LSMEANS statement produces an output data set that contains

- the unformatted values of each classification variable specified in any effect in the LSMEANS statement
- a new variable, LSMEAN, which contains the LS-mean for the specified levels of the classification variables
- a new variable, STDERR, which contains the standard error of the LS-mean

The variances and covariances among the LS-means are also output when the COV option is specified along with the OUT= option. In this case, only one effect can be specified in the LSMEANS statement, and the following variables are included in the output data set:

- new variables, COV1, COV2, . . . , COVn, where n is the number of levels of the effect specified in the LSMEANS statement. These variables contain the covariances of each LS-mean with each other LS-mean.
- a new variable, NUMBER, which provides an index for each observation to identify the covariances that correspond to that observation. The covariances for the observation with NUMBER equal to n can be found in the variable COVn.

OUTSTAT= Data Set

The OUTSTAT= option in the PROC GLM statement produces an output data set that contains

- the BY variables, if any
- _TYPE_, a new character variable. _TYPE_ may take the values 'SS1', 'SS2', 'SS3', 'SS4', or 'CONTRAST', corresponding to the various types of sums of squares generated, or the values 'CANCORR', 'STRUCTUR', or 'SCORE', if a canonical analysis is performed through the MANOVA statement and no M= matrix is specified.
- _SOURCE_, a new character variable. For each observation in the data set, _SOURCE_ contains the name of the model effect or contrast label from which the corresponding statistics are generated.
- _NAME_, a new character variable. For each observation in the data set, _NAME_ contains the name of one of the dependent variables in the model or, in the case of canonical statistics, the name of one of the canonical variables (CAN1, CAN2, and so forth).
- four new numeric variables: SS, DF, F, and PROB, containing sums of squares, degrees of freedom, F values, and probabilities, respectively, for each model or contrast sum of squares generated in the analysis. For observations resulting from canonical analyses, these variables have missing values.
- if there is more than one dependent variable, then variables with the same names as the dependent variables represent
  - for _TYPE_=SS1, SS2, SS3, SS4, or CONTRAST, the crossproducts of the hypothesis matrices
  - for _TYPE_=CANCORR, canonical correlations for each variable
  - for _TYPE_=STRUCTUR, coefficients of the total structure matrix
  - for _TYPE_=SCORE, raw canonical score coefficients

The output data set can be used to perform special hypothesis tests (for example, with the IML procedure in SAS/IML software), to reformat output, to produce canonical variates (through the SCORE procedure), or to rotate structure matrices (through the FACTOR procedure).

Displayed Output

The GLM procedure produces the following output by default:

- The overall analysis-of-variance table breaks down the Total Sum of Squares for the dependent variable into the portion attributed to the Model and the portion attributed to Error.
- The Mean Square term is the Sum of Squares divided by the degrees of freedom (DF).

- The Mean Square for Error is an estimate of σ^2, the variance of the true errors.
- The F Value is the ratio produced by dividing the Mean Square for the Model by the Mean Square for Error. It tests how well the model as a whole (adjusted for the mean) accounts for the dependent variable's behavior. An F-test is a joint test to determine that all parameters except the intercept are zero.
- A small significance probability, Pr > F, indicates that some linear function of the parameters is significantly different from zero.
- R-Square, R^2, measures how much variation in the dependent variable can be accounted for by the model. R^2, which can range from 0 to 1, is the ratio of the sum of squares for the model divided by the sum of squares for the corrected total. In general, the larger the value of R^2, the better the model's fit.
- Coef Var, the coefficient of variation, which describes the amount of variation in the population, is 100 times the standard deviation estimate of the dependent variable, Root MSE (Mean Square for Error), divided by the Mean. The coefficient of variation is often a preferred measure because it is unitless.
- Root MSE estimates the standard deviation of the dependent variable (or equivalently, the error term) and equals the square root of the Mean Square for Error.
- Mean is the sample mean of the dependent variable.

These tests are used primarily in analysis-of-variance applications:

- The Type I SS (sum of squares) measures incremental sums of squares for the model as each variable is added.
- The Type III SS is the sum of squares for a balanced test of each effect, adjusted for every other effect.

These items are used primarily in regression applications:

- The Estimates for the model Parameters (the intercept and the coefficients)
- t Value is the Student's t value for testing the null hypothesis that the parameter (if it is estimable) equals zero.
- The significance level, Pr > |t|, is the probability of getting a larger value of t if the parameter is truly equal to zero. A very small value for this probability leads to the conclusion that the independent variable contributes significantly to the model.
- The Standard Error is the square root of the estimated variance of the estimate of the true value of the parameter.

Other portions of output are discussed in the following examples.

ODS Table Names

PROC GLM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 32.5. ODS Tables Produced in PROC GLM

ODS Table Name	Description	Statement / Option
Aliasing	Type 1,2,3,4 aliasing structure	MODEL / (E1 E2 E3 or E4) and ALIASING
AltErrContrasts	ANOVA table for contrasts with alternative error	CONTRAST / E=
AltErrTests	ANOVA table for tests with alternative error	TEST / E=
Bartlett	Bartlett's homogeneity of variance test	MEANS / HOVTEST=BARTLETT
CLDiffs	Multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLDiffsInfo	Information for multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLMeans	Multiple comparisons of means with confidence/comparison interval	MEANS / CLM
CLMeansInfo	Information for multiple comparison of means with confidence/comparison interval	MEANS / CLM
CanAnalysis	Canonical analysis	(MANOVA or REPEATED) / CANONICAL
CanCoef	Canonical coefficients	(MANOVA or REPEATED) / CANONICAL
CanStructure	Canonical structure	(MANOVA or REPEATED) / CANONICAL
CharStruct	Characteristic roots and vectors	(MANOVA / not CANONICAL) or (REPEATED / PRINTRV)
ClassLevels	Classification variable levels	CLASS statement
ContrastCoef	**L** matrix for contrast	CONTRAST / EST
Contrasts	ANOVA table for contrasts	CONTRAST statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables with different patterns of missing values
Diff	PDiff matrix of Least-Squares Means	LSMEANS / PDIFF
Epsilons	Greenhouse-Geisser and Huynh-Feldt epsilons	REPEATED statement
ErrorSSCP	Error SSCP matrix	(MANOVA or REPEATED) / PRINTE
EstFunc	Type 1,2,3,4 estimable functions	MODEL / (E1 E2 E3 or E4)

Table 32.5. (continued)

ODS Table Name	**Description**	**Statement / Option**
Estimates	Estimate statement results	ESTIMATE statement
ExpectedMeanSquares	Expected mean squares	RANDOM statement
FitStatistics	R-Square, C.V., Root MSE, and dependent mean	default
GAliasing	General form of aliasing structure	MODEL / E and ALIASING
GEstFunc	General form of estimable functions	MODEL / E
HOVFTest	Homogeneity of variance ANOVA	MEANS / HOVTEST
HypothesisSSCP	Hypothesis SSCP matrix	(MANOVA or REPEATED) / PRINTH
InvXPX	inv(**X**'**X**) matrix	MODEL / INVERSE
LSMeanCL	Confidence interval for LS-means	LSMEANS / CL
LSMeanCoef	Coefficients of Least-Squares Means	LSMEANS / E
LSMeanDiffCL	Confidence interval for LS-mean differences	LSMEANS / PDIFF and CL
LSMeans	Least-Squares means	LSMEANS statement
MANOVATransform	Multivariate transformation matrix	MANOVA / M=
MCLines	Multiple comparisons LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesInfo	Information for multiple comparison LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesRange	Ranges for multiple range MC tests	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MatrixRepresentation	**X** matrix element representation	as needed for other options
Means	Group means	MEANS statement
ModelANOVA	ANOVA for model terms	default
MultStat	Multivariate tests	MANOVA statement
NObs	Number of observations	default
OverallANOVA	Over-all ANOVA	default
ParameterEstimates	Estimated linear model coefficients	MODEL / SOLUTION
PartialCorr	Partial correlation matrix	(MANOVA or REPEATED) / PRINTE

Table 32.5. (continued)

ODS Table Name	Description	Statement / Option
PredictedInfo	Predicted values info	MODEL / PREDICTED or CLM or CLI
PredictedValues	Predicted values	MODEL / PREDICTED or CLM or CLI
QForm	Quadratic form for expected mean squares	RANDOM / Q
RandomModelANOVA	Random effect tests	RANDOM / TEST
RepeatedLevelInfo	Correspondence between dependents and repeated measures levels	REPEATED statement
RepeatedTransform	Repeated Measures Transformation Matrix	REPEATED / PRINTM
SimDetails	Details of difference quantile simulation	LSMEANS / ADJUST=SIMULATE(REPORT)
SimResults	Evaluation of difference quantile simulation	LSMEANS / ADJUST=SIMULATE(REPORT)
SlicedANOVA	Sliced effect ANOVA table	LSMEANS / SLICE
Sphericity	Sphericity tests	REPEATED / PRINTE
Tests	Summary ANOVA for specified MANOVA H= effects	MANOVA / H= SUMMARY
Tolerances	**X'X** Tolerances	MODEL / TOLERANCE
Welch	Welch's ANOVA	MEANS / WELCH
XPX	**X'X** matrix	MODEL / XPX

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the GLM procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request these graphs you must specify the ODS GRAPHICS statement with an appropriate model, as discussed in the following. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

When the ODS GRAPHICS are in effect, then for particular models the GLM procedure will produce default graphics.

- If you specify an analysis of covariance model, with one classification variable and one continuous variable, the GLM procedure will produce an analysis of covariance plot of the response values versus the covariate values, with lines representing the fitted relationship within each classification level. For an example of the analysis of covariance plot, see Example 32.4 on page 1860.
- If you specify a one-way analysis of variance model, with just one independent classification variable, the GLM procedure will produce a grouped box plot of the response values versus the classification levels. For an example of the box plot, see the "One-Way Layout with Means Comparisons" section on page 424.

ODS Graph Names

PROC GLM assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 32.6.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 32.6. ODS Graphics Produced by PROC GLM

ODS Graph Name	Plot Description
ANCOVAPlot	Analysis of covariance plot
BoxPlot	Box plot

Examples

Example 32.1. Balanced Data from Randomized Complete Block with Means Comparisons and Contrasts

The following example, reported by Stenstrom (1940), analyzes an experiment to investigate how snapdragons grow in various soils. To eliminate the effect of local fertility variations, the experiment is run in blocks, with each soil type sampled in each block. Since these data are balanced, the Type I and Type III SS are the same and are equal to the traditional ANOVA SS.

First, the standard analysis is shown followed by an analysis that uses the SOLUTION option and includes MEANS and CONTRAST statements. The ORDER=DATA option in the second PROC GLM statement is used so that the ordering of coefficients in the CONTRAST statement can correspond to the ordering in the input data. The SOLUTION option requests a display of the parameter estimates, which are only produced by default if there are no CLASS variables. A MEANS statement is used to request a table of the means with two multiple comparison procedures requested. In experiments with focused treatment questions, CONTRAST statements are preferable to general means comparison methods. The following statements produce Output 32.1.1 through Output 32.1.6:

```
title 'Balanced Data from Randomized Complete Block';
data plants;
   input Type $ @;
   do Block = 1 to 3;
      input StemLength @;
      output;
      end;
   datalines;
Clarion  32.7 32.3 31.5
Clinton  32.1 29.7 29.1
Knox     35.7 35.9 33.1
O'Neill  36.0 34.2 31.2
Compost  31.8 28.0 29.2
Wabash   38.2 37.8 31.9
Webster  32.5 31.1 29.7
;
```

```
proc glm;
   class Block Type;
   model StemLength = Block Type;
run;

proc glm order=data;
   class Block Type;
   model StemLength = Block Type / solution;

/*------------------------------------clrn-cltn-knox-onel-cpst-wbsh-wstr */
contrast 'Compost vs. others'  Type   -1   -1   -1   -1    6   -1   -1;
contrast 'River soils vs. non' Type   -1   -1   -1   -1    0    5   -1,
                               Type   -1    4   -1   -1    0    0   -1;
contrast 'Glacial vs. drift'   Type   -1    0    1    1    0    0   -1;
contrast 'Clarion vs. Webster' Type   -1    0    0    0    0    0    1;
contrast ''Knox vs. O'Neill''  Type    0    0    1   -1    0    0    0;
run;

    means Type / waller regwq;
run;
```

Output 32.1.1. Classes and Levels for Randomized Complete Blocks

```
                  Balanced Data from Randomized Complete Block

                               The GLM Procedure

                            Class Level Information

Class         Levels    Values

Block              3    1 2 3

Type               7    Clarion Clinton Compost Knox O'Neill Wabash Webster


                    Number of Observations Read          21
                    Number of Observations Used          21
```

Output 32.1.2. Analysis of Variance for Randomized Complete Blocks

```
                  Balanced Data from Randomized Complete Block

                               The GLM Procedure

Dependent Variable: StemLength

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                        8    142.1885714     17.7735714     10.80   0.0002

 Error                       12     19.7428571      1.6452381

 Corrected Total             20    161.9314286


            R-Square     Coeff Var      Root MSE    StemLength Mean

            0.878079      3.939745      1.282668           32.55714


 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 Block                        2     39.0371429     19.5185714     11.86   0.0014
 Type                         6    103.1514286     17.1919048     10.45   0.0004


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 Block                        2     39.0371429     19.5185714     11.86   0.0014
 Type                         6    103.1514286     17.1919048     10.45   0.0004
```

This analysis shows that the stem length is significantly different for the different soil types. In addition, there are significant differences in stem length between the three blocks in the experiment.

Output 32.1.3. Standard Analysis Again

```
                  Balanced Data from Randomized Complete Block

                               The GLM Procedure

                            Class Level Information

  Class         Levels    Values

  Block              3    1 2 3

  Type               7    Clarion Clinton Compost Knox O'Neill Wabash Webster


                     Number of Observations Read          21
                     Number of Observations Used          21
```

The GLM procedure is invoked again, this time with the ORDER=DATA option. This enables you to write accurate contrast statements more easily because you know the order SAS is using for the levels of the variable Type. The standard analysis is displayed again.

Output 32.1.4. Contrasts and Solutions

```
                  Balanced Data from Randomized Complete Block

                               The GLM Procedure

Dependent Variable: StemLength

 Contrast                  DF    Contrast SS    Mean Square   F Value   Pr > F

 Compost vs. others         1    29.24198413    29.24198413     17.77   0.0012
 River soils vs. non        2    48.24694444    24.12347222     14.66   0.0006
 Glacial vs. drift          1    22.14083333    22.14083333     13.46   0.0032
 Clarion vs. Webster        1     1.70666667     1.70666667      1.04   0.3285
 Knox vs. O'Neill           1     1.81500000     1.81500000      1.10   0.3143


                                                  Standard
  Parameter                 Estimate                 Error    t Value    Pr > |t|

  Intercept              29.35714286 B      0.83970354      34.96      <.0001
  Block     1             3.32857143 B      0.68561507       4.85      0.0004
  Block     2             1.90000000 B      0.68561507       2.77      0.0169
  Block     3             0.00000000 B       .                .          .
  Type      Clarion       1.06666667 B      1.04729432       1.02      0.3285
  Type      Clinton      -0.80000000 B      1.04729432      -0.76      0.4597
  Type      Knox          3.80000000 B      1.04729432       3.63      0.0035
  Type      O'Neill       2.70000000 B      1.04729432       2.58      0.0242
  Type      Compost      -1.43333333 B      1.04729432      -1.37      0.1962
  Type      Wabash        4.86666667 B      1.04729432       4.65      0.0006
  Type      Webster       0.00000000 B       .                .          .

NOTE: The X'X matrix has been found to be singular, and a generalized inverse
      was used to solve the normal equations.  Terms whose estimates are
      followed by the letter 'B' are not uniquely estimable.
```

Output 32.1.4 shows the tests for contrasts that you specified as well as the estimated parameters. The contrast label, degrees of freedom, sum of squares, Mean Square, F Value, and Pr > F are shown for each contrast requested. In this example, the contrast results show that at the 5% significance level,

- the stem length of plants grown in compost soil is significantly different from the average stem length of plants grown in other soils
- the stem length of plants grown in river soils is significantly different from the average stem length of those grown in nonriver soils
- the average stem length of plants grown in glacial soils (Clarion and Webster) is significantly different from the average stem length of those grown in drift soils (Knox and O'Neill)
- stem lengths for Clarion and Webster are not significantly different
- stem lengths for Knox and O'Neill are not significantly different

In addition to the estimates for the parameters of the model, the results of t tests about the parameters are also displayed. The ‘B’ following the parameter estimates indicates that the estimates are biased and do not represent a unique solution to the normal equations.

Output 32.1.5. Waller-Duncan tests

```
                  Balanced Data from Randomized Complete Block

                               The GLM Procedure

                  Waller-Duncan K-ratio t Test for StemLength

NOTE: This test minimizes the Bayes risk under additive loss and certain other
                                  assumptions.


                    Kratio                                 100
                    Error Degrees of Freedom                12
                    Error Mean Square                 1.645238
                    F Value                              10.45
                    Critical Value of t                2.12034
                    Minimum Significant Difference      2.2206


           Means with the same letter are not significantly different.


            Waller Grouping          Mean      N    Type

                          A        35.967      3    Wabash
                          A
                          A        34.900      3    Knox
                          A
                     B    A        33.800      3    O’Neill
                     B
                     B    C        32.167      3    Clarion
                          C
                     D    C        31.100      3    Webster
                     D    C
                     D    C        30.300      3    Clinton
                     D
                     D             29.667      3    Compost
```

Output 32.1.6. Ryan-Einot-Gabriel-Welsch Multiple Range Test

```
                Balanced Data from Randomized Complete Block

                             The GLM Procedure

        Ryan-Einot-Gabriel-Welsch Multiple Range Test for StemLength

       NOTE: This test controls the Type I experimentwise error rate.


                     Alpha                        0.05
                     Error Degrees of Freedom       12
                     Error Mean Square        1.645238


Number of Means         2         3         4         5         6         7
Critical Range  2.9876649 3.2838329 3.4396257 3.5402242 3.5402242 3.6653734


        Means with the same letter are not significantly different.


             REGWQ Grouping          Mean      N    Type

                     A             35.967      3    Wabash
                     A
                B    A             34.900      3    Knox
                B    A
                B    A    C        33.800      3    O’Neill
                B         C
                B    D    C        32.167      3    Clarion
                     D    C
                     D    C        31.100      3    Webster
                     D
                     D             30.300      3    Clinton
                     D
                     D             29.667      3    Compost
```

The final two pages of output (Output 32.1.5 and Output 32.1.6) present results of the Waller-Duncan and REGWQ multiple comparison procedures. For each test, notes and information pertinent to the test are given on the output. The Type means are arranged from highest to lowest. Means with the same letter are not significantly different. For this example, while some pairs of means are significantly different, there are no clear equivalence classes among the different soils.

Example 32.2. Regression with Mileage Data

A car is tested for gas mileage at various speeds to determine at what speed the car achieves the greatest gas mileage. A quadratic model is fit to the experimental data. The following statements produce Output 32.2.1 through Output 32.2.5:

```
title 'Gasoline Mileage Experiment';
data mileage;
   input mph mpg @@;
   datalines;
20 15.4
30 20.2
40 25.7
50 26.2  50 26.6  50 27.4
55   .
60 24.8
;

proc glm;
   model mpg=mph mph*mph / p clm;
   output out=pp p=mpgpred r=resid;

axis1 minor=none major=(number=5);
axis2 minor=none major=(number=8);
symbol1 c=black i=none   v=plus;
symbol2 c=black i=spline v=none;
proc gplot data=pp;
   plot mpg*mph=1 mpgpred*mph=2 / overlay haxis=axis1
        vaxis=axis2;
run;
```

Output 32.2.1. Observations for Standard Regression Analysis

```
                         Gasoline Mileage Experiment

                              The GLM Procedure

                   Number of Observations Read           8
```

Output 32.2.2. Standard Analysis of Variance for Regression

```
                          Gasoline Mileage Experiment

                               The GLM Procedure

Dependent Variable: mpg

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                        2    111.8086183     55.9043091     77.96   0.0006

 Error                        4      2.8685246      0.7171311

 Corrected Total              6    114.6771429


               R-Square     Coeff Var      Root MSE      mpg Mean

               0.974986      3.564553      0.846836      23.75714


 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 mph                          1    85.64464286    85.64464286    119.43   0.0004
 mph*mph                      1    26.16397541    26.16397541     36.48   0.0038


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 mph                          1    41.01171219    41.01171219     57.19   0.0016
 mph*mph                      1    26.16397541    26.16397541     36.48   0.0038


                                              Standard
       Parameter         Estimate                Error    t Value    Pr > |t|

       Intercept     -5.985245902       3.18522249      -1.88      0.1334
       mph            1.305245902       0.17259876       7.56      0.0016
       mph*mph       -0.013098361       0.00216852      -6.04      0.0038
```

The overall F statistic is significant. The tests of mph and mph*mph in the Type I sums of squares show that both the linear and quadratic terms in the regression model are significant. The model fits well, with an R^2 of 0.97. The table of parameter estimates indicates that the estimated regression equation is

$$\text{mpg} \quad = \quad -5.9852 + 1.3052 \times \text{mph} - 0.0131 \times \text{mph}^2$$

Output 32.2.3. Results of Requesting the P and CLM Options

```
                 1          15.40000000        14.88032787         0.51967213
                 2          20.20000000        21.38360656        -1.18360656
                 3          25.70000000        25.26721311         0.43278689
                 4          26.20000000        26.53114754        -0.33114754
                 5          26.60000000        26.53114754         0.06885246
                 6          27.40000000        26.53114754         0.86885246
                 7 *          .                26.18073770          .
                 8          24.80000000        25.17540984        -0.37540984

                                  95% Confidence Limits for
                  Observation        Mean Predicted Value

                            1          12.69701317     17.06364257
                            2          20.01727192     22.74994119
                            3          23.87460041     26.65982582
                            4          25.44573423     27.61656085
                            5          25.44573423     27.61656085
                            6          25.44573423     27.61656085
                            7 *        24.88679308     27.47468233
                            8          23.05954977     27.29126990


* Observation was not used in this analysis
```

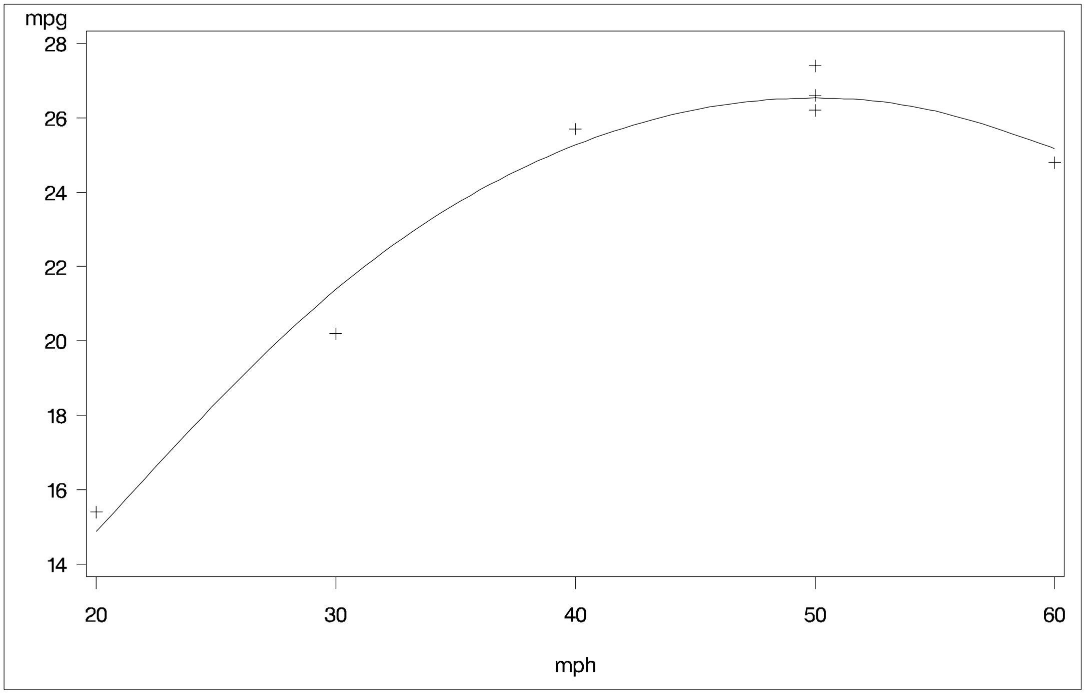
The P and CLM options in the MODEL statement produce the table shown in Output 32.2.3. For each observation, the observed, predicted, and residual values are shown. In addition, the 95% confidence limits for a mean predicted value are shown for each observation. Note that the observation with a missing value for mph is not used in the analysis, but predicted and confidence limit values are shown.

Output 32.2.4. Additional Results of Requesting the P and CLM Options

```
                           Gasoline Mileage Experiment

                                The GLM Procedure

               Sum of Residuals                            0.00000000
               Sum of Squared Residuals                    2.86852459
               Sum of Squared Residuals - Error SS        -0.00000000
               PRESS Statistic                            23.18107335
               First Order Autocorrelation                -0.54376613
               Durbin-Watson D                             2.94425592
```

The final portion of output gives some additional information on the residuals. The Press statistic gives the sum of squares of predicted residual errors, as described in Chapter 2, “Introduction to Regression Procedures.” The First Order Autocorrelation and the Durbin-Watson D statistic, which measures first-order autocorrelation, are also given.

Output 32.2.5. Plot of Mileage Data



Output 32.2.5 shows the actual and predicted values for the data. The quadratic relationship between mpg and mph is evident.

Example 32.3. Unbalanced ANOVA for Two-Way Design with Interaction

This example uses data from Kutner (1974, p. 98) to illustrate a two-way analysis of variance. The original data source is Afifi and Azen (1972, p. 166). These statements produce Output 32.3.1 and Output 32.3.2.

```
/*---------------------------------------------------------*/
/* Note: Kutner’s 24 for drug 2, disease 1 changed to 34.  */
/*---------------------------------------------------------*/
title ’Unbalanced Two-Way Analysis of Variance’;
data a;
   input drug disease @;
   do i=1 to 6;
      input y @;
      output;
   end;
   datalines;
1 1 42 44 36 13 19 22
1 2 33  . 26  . 33 21
1 3 31 -3  . 25 25 24
2 1 28  . 23 34 42 13
2 2  . 34 33 31  . 36
2 3  3 26 28 32  4 16
3 1  .  .  1 29  . 19
3 2  . 11  9  7  1 -6
3 3 21  1  .  9  3  .
4 1 24  .  9 22 -2 15
4 2 27 12 12 -5 16 15
4 3 22  7 25  5 12  .
;

proc glm;
   class drug disease;
   model y=drug disease drug*disease / ss1 ss2 ss3 ss4;
run;
```

Output 32.3.1. Classes and Levels for Unbalanced Two-Way Design

```
                    Unbalanced Two-Way Analysis of Variance

                               The GLM Procedure

                            Class Level Information

                        Class         Levels    Values

                        drug               4    1 2 3 4

                        disease            3    1 2 3


                    Number of Observations Read          72
                    Number of Observations Used          58
```

Output 32.3.2. Analysis of Variance for Unbalanced Two-Way Design

```
                    Unbalanced Two-Way Analysis of Variance

                               The GLM Procedure

Dependent Variable: y

                                            Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                       11      4259.338506     387.212591      3.51   0.0013

 Error                       46      5080.816667     110.452536

 Corrected Total             57      9340.155172


              R-Square     Coeff Var      Root MSE        y Mean

              0.456024      55.66750      10.50964      18.87931


 Source                      DF        Type I SS    Mean Square   F Value   Pr > F

 drug                         3      3133.238506    1044.412835      9.46   <.0001
 disease                      2       418.833741     209.416870      1.90   0.1617
 drug*disease                 6       707.266259     117.877710      1.07   0.3958


 Source                      DF       Type II SS    Mean Square   F Value   Pr > F

 drug                         3      3063.432863    1021.144288      9.25   <.0001
 disease                      2       418.833741     209.416870      1.90   0.1617
 drug*disease                 6       707.266259     117.877710      1.07   0.3958


 Source                      DF      Type III SS    Mean Square   F Value   Pr > F

 drug                         3      2997.471860     999.157287      9.05   <.0001
 disease                      2       415.873046     207.936523      1.88   0.1637
 drug*disease                 6       707.266259     117.877710      1.07   0.3958


 Source                      DF       Type IV SS    Mean Square   F Value   Pr > F

 drug                         3      2997.471860     999.157287      9.05   <.0001
 disease                      2       415.873046     207.936523      1.88   0.1637
 drug*disease                 6       707.266259     117.877710      1.07   0.3958
```

Note the differences between the four types of sums of squares. The Type I sum of squares for drug essentially tests for differences between the expected values of the arithmetic mean response for different drugs, unadjusted for the effect of disease. By contrast, the Type II sum of squares for drug measure the differences between arithmetic means for each drug after adjusting for disease. The Type III sum of squares measures the differences between predicted drug means over a balanced drug×disease population—that is, between the LS-means for drug. Finally, the Type IV sum of squares is the same as the Type III sum of squares in this case, since there is data for every drug-by-disease combination.

No matter which sum of squares you prefer to use, this analysis shows a significant difference among the four drugs, while the disease effect and the drug-by-disease interaction are not significant. As the previous discussion indicates, Type III sums of squares correspond to differences between LS-means, so you can follow up the Type III tests with a multiple comparisons analysis of the drug LS-means. Since the GLM procedure is interactive, you can accomplish this by submitting the following statements after the previous ones that performed the ANOVA.

```
      lsmeans drug / pdiff=all adjust=tukey;
   run;
```

Both the LS-means themselves and a matrix of adjusted p-values for pairwise differences between them are displayed; see Output 32.3.3.

Output 32.3.3. LS-Means for Unbalanced ANOVA

```
                    Unbalanced Two-Way Analysis of Variance

                               The GLM Procedure
                              Least Squares Means
               Adjustment for Multiple Comparisons: Tukey-Kramer

                                                   LSMEAN
                        drug         y LSMEAN      Number

                        1          25.9944444           1
                        2          26.5555556           2
                        3           9.7444444           3
                        4          13.5444444           4
```

```
                    Unbalanced Two-Way Analysis of Variance

                               The GLM Procedure
                              Least Squares Means
               Adjustment for Multiple Comparisons: Tukey-Kramer

                         Least Squares Means for effect drug
                         Pr > |t| for H0: LSMean(i)=LSMean(j)

                              Dependent Variable: y

            i/j              1             2             3             4

               1                      0.9989        0.0016        0.0107
               2        0.9989                      0.0011        0.0071
               3        0.0016        0.0011                      0.7870
               4        0.0107        0.0071        0.7870
```

The multiple comparisons analysis shows that drugs 1 and 2 have very similar effects, and that drugs 3 and 4 are also insignificantly different from each other. Evidently, the main contribution to the significant drug effect is the difference between the 1/2 pair and the 3/4 pair.

Example 32.4. Analysis of Covariance

Analysis of covariance combines some of the features of both regression and analysis of variance. Typically, a continuous variable (the covariate) is introduced into the model of an analysis-of-variance experiment.

Data in the following example are selected from a larger experiment on the use of drugs in the treatment of leprosy (Snedecor and Cochran 1967, p. 422).

Variables in the study are

Drug	- two antibiotics (A and D) and a control (F)
PreTreatment	- a pre-treatment score of leprosy bacilli
PostTreatment	- a post-treatment score of leprosy bacilli

Ten patients are selected for each treatment (Drug), and six sites on each patient are measured for leprosy bacilli.

The covariate (a pretreatment score) is included in the model for increased precision in determining the effect of drug treatments on the posttreatment count of bacilli.

The following code creates the data set, performs a parallel-slopes analysis of covariance with PROC GLM, and computes Drug LS-means. These statements produce Output 32.4.1.

```
data drugtest;
   input Drug $ PreTreatment PostTreatment @@;
   datalines;
A 11  6   A  8  0   A  5  2   A 14  8   A 19 11
A  6  4   A 10 13   A  6  1   A 11  8   A  3  0
D  6  0   D  6  2   D  7  3   D  8  1   D 18 18
D  8  4   D 19 14   D  8  9   D  5  1   D 15  9
F 16 13   F 13 10   F 11 18   F  9  5   F 21 23
F 16 12   F 12  5   F 12 16   F  7  1   F 12 20
;

proc glm;
   class Drug;
   model PostTreatment = Drug PreTreatment / solution;
   lsmeans Drug / stderr pdiff cov out=adjmeans;
run;

proc print data=adjmeans;
run;
```

Output 32.4.1. Overall Analysis of Variance

```
                                The GLM Procedure

                             Class Level Information

                          Class         Levels    Values

                          Drug               3    A D F


                     Number of Observations Read          30
                     Number of Observations Used          30
```

```
                                The GLM Procedure

Dependent Variable: PostTreatment

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                        3     871.497403     290.499134     18.10   <.0001

 Error                       26     417.202597      16.046254

 Corrected Total             29    1288.700000


           R-Square     Coeff Var      Root MSE    PostTreatment Mean

           0.676261      50.70604      4.005778              7.900000
```

This model assumes that the slopes relating posttreatment scores to pretreatment scores are parallel for all drugs. You can check this assumption by including the class-by-covariate interaction, Drug*PreTreatment, in the model and examining the ANOVA test for the significance of this effect. This extra test is omitted in this example, but it is insignificant, justifying the equal-slopes assumption.

In Output 32.4.2, the Type I SS for Drug (293.6) gives the between-drug sums of squares that are obtained for the analysis-of-variance model PostTreatment=Drug. This measures the difference between arithmetic means of posttreatment scores for different drugs, disregarding the covariate. The Type III SS for Drug (68.5537) gives the Drug sum of squares adjusted for the covariate. This measures the differences between Drug LS-means, controlling for the covariate. The Type I test is highly significant ($p = 0.001$), but the Type III test is not. This indicates that, while there is a statistically significant difference between the arithmetic drug means, this difference is reduced to below the level of background noise when you take the pretreatment scores into account. From the table of parameter estimates, you can derive the least-squares predictive formula model for estimating posttreatment score based on pretreatment score and drug.

$$\text{post} = \begin{cases} (-0.435 + -3.446) & + \quad 0.987 \cdot \text{pre}, \quad \text{if Drug=A} \\ (-0.435 + -3.337) & + \quad 0.987 \cdot \text{pre}, \quad \text{if Drug=D} \\ \;-0.435 & + \quad 0.987 \cdot \text{pre}, \quad \text{if Drug=F} \end{cases}$$

Output 32.4.2. Tests and Parameter Estimates

```
                                 The GLM Procedure

Dependent Variable: PostTreatment

 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 Drug                         2    293.6000000    146.8000000      9.15   0.0010
 PreTreatment                 1    577.8974030    577.8974030     36.01   <.0001


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 Drug                         2     68.5537106     34.2768553      2.14   0.1384
 PreTreatment                 1    577.8974030    577.8974030     36.01   <.0001


                                              Standard
    Parameter              Estimate              Error    t Value    Pr > |t|

    Intercept          -0.434671164 B      2.47135356      -0.18      0.8617
    Drug         A     -3.446138280 B      1.88678065      -1.83      0.0793
    Drug         D     -3.337166948 B      1.85386642      -1.80      0.0835
    Drug         F      0.000000000 B       .                .         .
    PreTreatment        0.987183811        0.16449757       6.00      <.0001

NOTE: The X'X matrix has been found to be singular, and a generalized inverse
      was used to solve the normal equations.  Terms whose estimates are
      followed by the letter 'B' are not uniquely estimable.
```

Output 32.4.3 displays the LS-means, which are, in a sense, the means adjusted for the covariate. The STDERR option in the LSMEANS statement causes the standard error of the LS-means and the probability of getting a larger t value under the hypothesis H_0: LS-mean $= 0$ to be included in this table as well. Specifying the PDIFF option causes all probability values for the hypothesis H_0: LS-mean(i) $=$ LS-mean(j) to be displayed, where the indexes i and j are numbered treatment levels.

Output 32.4.3. LS-means

```
                              The GLM Procedure
                             Least Squares Means

                       Post
                  Treatment        Standard                        LSMEAN
           Drug      LSMEAN           Error    Pr > |t|            Number

           A      6.7149635       1.2884943      <.0001                 1
           D      6.8239348       1.2724690      <.0001                 2
           F     10.1611017       1.3159234      <.0001                 3


                     Least Squares Means for effect Drug
                     Pr > |t| for H0: LSMean(i)=LSMean(j)

                      Dependent Variable: PostTreatment

                 i/j              1              2              3

                    1                       0.9521         0.0793
                    2        0.9521                        0.0835
                    3        0.0793         0.0835


NOTE: To ensure overall protection level, only probabilities associated with
      pre-planned comparisons should be used.
```

The OUT= and COV options in the LSMEANS statement create a data set of the estimates, their standard errors, and the variances and covariances of the LS-means, which is displayed in Output 32.4.4

Output 32.4.4. LS-means Output Data Set

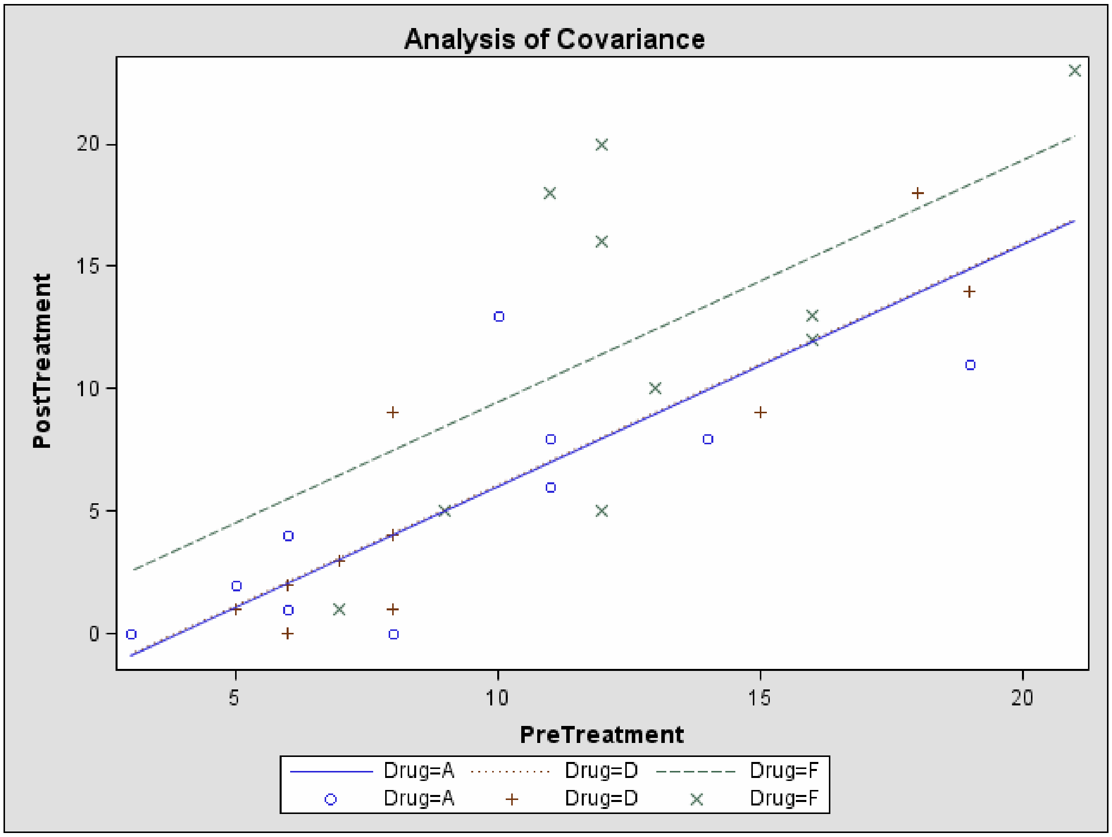
```
Obs     _NAME_       Drug    LSMEAN   STDERR  NUMBER     COV1       COV2       COV3

 1   PostTreatment    A      6.7150  1.28849     1      1.66022    0.02844   -0.08403
 2   PostTreatment    D      6.8239  1.27247     2      0.02844    1.61918   -0.04299
 3   PostTreatment    F     10.1611  1.31592     3     -0.08403   -0.04299    1.73165
```

The experimental graphics features of PROC GLM enable you to visualize the fitted analysis of covariance model.

```
ods html;
ods graphics on;

proc glm;
   class Drug;
   model PostTreatment = Drug PreTreatment;
run;

ods graphics off;
ods html close;
```

When you specify the experimental ODS GRAPHICS statement and fit an analysis of covariance model, the GLM procedure output includes an analysis of covariance plot, as in Output 32.4.5. For general information about ODS graphics see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the GLM procedure, see the section "ODS Graphics" on page 1846.

Output 32.4.5. Analysis of Covariance Plot (Experimental)



The plot makes it clear that the control (drug F) has higher post-treatment scores across the range of pre-treatment scores, while the fitted models for the two antibiotics (drugs A and D) nearly coincide.

Example 32.5. Three-Way Analysis of Variance with Contrasts

This example uses data from Cochran and Cox (1957, p. 176) to illustrate the analysis of a three-way factorial design with replication, including the use of the CONTRAST statement with interactions, the OUTSTAT= data set, and the SLICE= option in the LSMEANS statement.

The object of the study is to determine the effects of electric current on denervated muscle. The variables are

Rep the replicate number, 1 or 2

Time the length of time the current is applied to the muscle, ranging from 1 to 4

Current the level of electric current applied, ranging from 1 to 4

Number the number of treatments per day, ranging from 1 to 3

MuscleWeight the weight of the denervated muscle

The following code produces Output 32.5.1 through Output 32.5.4.

```
data muscles;
   do Rep=1 to 2;
      do Time=1 to 4;
         do Current=1 to 4;
            do Number=1 to 3;
               input MuscleWeight @@;
               output;
            end;
         end;
      end;
   end;
   datalines;
72 74 69 61 61 65 62 65 70 85 76 61
67 52 62 60 55 59 64 65 64 67 72 60
57 66 72 72 43 43 63 66 72 56 75 92
57 56 78 60 63 58 61 79 68 73 86 71
46 74 58 60 64 52 71 64 71 53 65 66
44 58 54 57 55 51 62 61 79 60 78 82
53 50 61 56 57 56 56 56 71 56 58 69
46 55 64 56 55 57 64 66 62 59 58 88
;

proc glm outstat=summary;
   class Rep Current Time Number;
   model MuscleWeight = Rep Current|Time|Number;
   contrast 'Time in Current 3'
      Time 1 0 0 -1 Current*Time 0 0 0 0 0 0 0 0 1 0 0 -1,
      Time 0 1 0 -1 Current*Time 0 0 0 0 0 0 0 0 0 1 0 -1,
      Time 0 0 1 -1 Current*Time 0 0 0 0 0 0 0 0 0 0 1 -1;
   contrast 'Current 1 versus 2' Current 1 -1;
   lsmeans Current*Time / slice=Current;
run;

proc print data=summary;
run;
```

The first CONTRAST statement examines the effects of Time within level 3 of Current. This is also called the *simple effect* of Time within Current*Time. Note that, since there are three degrees of freedom, it is necessary to specify three rows in the CONTRAST statement, separated by commas. Since the parameterization that PROC GLM uses is determined in part by the ordering of the variables in the CLASS statement, Current is specified before Time so that the Time parameters are nested within the Current*Time parameters; thus, the Current*Time contrast coefficients in each row are simply the Time coefficients of that row within the appropriate level of Current.

The second CONTRAST statement isolates a single degree of freedom effect corresponding to the difference between the first two levels of Current. You can use such a contrast in a large experiment where certain preplanned comparisons are important, but you want to take advantage of the additional error degrees of freedom available when all levels of the factors are considered.

The LSMEANS statement with the SLICE= option is an alternative way to test for the simple effect of Time within Current*Time. In addition to listing the LS-means for each current strength and length of time, it gives a table of F-tests for differences between the LS-means across Time within each Current level. In some cases, this can be a way to disentangle a complex interaction.

Output 32.5.1. Overall Analysis

```
                               The GLM Procedure

                            Class Level Information

                        Class         Levels    Values

                        Rep                2    1 2

                        Current            4    1 2 3 4

                        Time               4    1 2 3 4

                        Number             3    1 2 3


                    Number of Observations Read          96
                    Number of Observations Used          96


                               The GLM Procedure

Dependent Variable: MuscleWeight

                                       Sum of
 Source                      DF       Squares    Mean Square   F Value   Pr > F

 Model                       48   5782.916667     120.477431      1.77   0.0261

 Error                       47   3199.489583      68.074246

 Corrected Total             95   8982.406250


           R-Square     Coeff Var      Root MSE    MuscleWeight Mean

           0.643805      13.05105      8.250712             63.21875
```

The output, shown in Output 32.5.2 and Output 32.5.3, indicates that the main effects for Rep, Current, and Number are significant (with p-values of 0.0045, <0.0001, and 0.0461, respectively), but Time is not significant, indicating that, in general, it doesn't matter how long the current is applied. None of the interaction terms are significant, nor are the contrasts significant. Notice that the row in the sliced ANOVA table corresponding to level 3 of current matches the "Time in Current 3" contrast.

Output 32.5.2. Individual Effects and Contrasts

```
                                The GLM Procedure

Dependent Variable: MuscleWeight

 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 Rep                          1     605.010417     605.010417      8.89   0.0045
 Current                      3    2145.447917     715.149306     10.51   <.0001
 Time                         3     223.114583      74.371528      1.09   0.3616
 Current*Time                 9     298.677083      33.186343      0.49   0.8756
 Number                       2     447.437500     223.718750      3.29   0.0461
 Current*Number               6     644.395833     107.399306      1.58   0.1747
 Time*Number                  6     367.979167      61.329861      0.90   0.5023
 Current*Time*Number         18    1050.854167      58.380787      0.86   0.6276


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 Rep                          1     605.010417     605.010417      8.89   0.0045
 Current                      3    2145.447917     715.149306     10.51   <.0001
 Time                         3     223.114583      74.371528      1.09   0.3616
 Current*Time                 9     298.677083      33.186343      0.49   0.8756
 Number                       2     447.437500     223.718750      3.29   0.0461
 Current*Number               6     644.395833     107.399306      1.58   0.1747
 Time*Number                  6     367.979167      61.329861      0.90   0.5023
 Current*Time*Number         18    1050.854167      58.380787      0.86   0.6276


 Contrast                    DF    Contrast SS    Mean Square   F Value   Pr > F

 Time in Current 3            3    34.83333333    11.61111111      0.17   0.9157
 Current 1 versus 2           1    99.18750000    99.18750000      1.46   0.2334
```

Output 32.5.3. Simple Effects of Time

```
                                The GLM Procedure
                               Least Squares Means

              Current*Time Effect Sliced by Current for MuscleWeight

                                     Sum of
     Current          DF            Squares     Mean Square    F Value    Pr > F

     1                 3         271.458333       90.486111       1.33    0.2761
     2                 3         120.666667       40.222222       0.59    0.6241
     3                 3          34.833333       11.611111       0.17    0.9157
     4                 3          94.833333       31.611111       0.46    0.7085
```

The SS, F statistics, and p-values can be stored in an OUTSTAT= data set, as shown in Output 32.5.4.

Output 32.5.4. Contents of the OUTSTAT= Data Set

```
Obs      _NAME_      _SOURCE_               _TYPE_     DF      SS         F        PROB

  1   MuscleWeight  ERROR                  ERROR      47   3199.49      .          .
  2   MuscleWeight  Rep                    SS1         1    605.01    8.8875   0.00454
  3   MuscleWeight  Current                SS1         3   2145.45   10.5054   0.00002
  4   MuscleWeight  Time                   SS1         3    223.11    1.0925   0.36159
  5   MuscleWeight  Current*Time           SS1         9    298.68    0.4875   0.87562
  6   MuscleWeight  Number                 SS1         2    447.44    3.2864   0.04614
  7   MuscleWeight  Current*Number         SS1         6    644.40    1.5777   0.17468
  8   MuscleWeight  Time*Number            SS1         6    367.98    0.9009   0.50231
  9   MuscleWeight  Current*Time*Number    SS1        18   1050.85    0.8576   0.62757
 10   MuscleWeight  Rep                    SS3         1    605.01    8.8875   0.00454
 11   MuscleWeight  Current                SS3         3   2145.45   10.5054   0.00002
 12   MuscleWeight  Time                   SS3         3    223.11    1.0925   0.36159
 13   MuscleWeight  Current*Time           SS3         9    298.68    0.4875   0.87562
 14   MuscleWeight  Number                 SS3         2    447.44    3.2864   0.04614
 15   MuscleWeight  Current*Number         SS3         6    644.40    1.5777   0.17468
 16   MuscleWeight  Time*Number            SS3         6    367.98    0.9009   0.50231
 17   MuscleWeight  Current*Time*Number    SS3        18   1050.85    0.8576   0.62757
 18   MuscleWeight  Time in Current 3      CONTRAST    3     34.83    0.1706   0.91574
 19   MuscleWeight  Current 1 versus 2     CONTRAST    1     99.19    1.4570   0.23344
```

Example 32.6. Multivariate Analysis of Variance

The following example employs multivariate analysis of variance (MANOVA) to measure differences in the chemical characteristics of ancient pottery found at four kiln sites in Great Britain. The data are from Tubb et al. (1980), as reported in Hand et al. (1994).

For each of 26 samples of pottery, the percentages of oxides of five metals are measured. The following statements create the data set and invoke the GLM procedure to perform a one-way MANOVA. Additionally, it is of interest to know whether the pottery from one site in Wales (Llanederyn) differs from the samples from other sites; a CONTRAST statement is used to test this hypothesis.

```
data pottery;
   title1 "Romano-British Pottery";
   input Site $12. Al Fe Mg Ca Na;
   datalines;
Llanederyn   14.4 7.00 4.30 0.15 0.51
Llanederyn   13.8 7.08 3.43 0.12 0.17
Llanederyn   14.6 7.09 3.88 0.13 0.20
Llanederyn   11.5 6.37 5.64 0.16 0.14
Llanederyn   13.8 7.06 5.34 0.20 0.20
Llanederyn   10.9 6.26 3.47 0.17 0.22
Llanederyn   10.1 4.26 4.26 0.20 0.18
Llanederyn   11.6 5.78 5.91 0.18 0.16
Llanederyn   11.1 5.49 4.52 0.29 0.30
Llanederyn   13.4 6.92 7.23 0.28 0.20
Llanederyn   12.4 6.13 5.69 0.22 0.54
Llanederyn   13.1 6.64 5.51 0.31 0.24
Llanederyn   12.7 6.69 4.45 0.20 0.22
Llanederyn   12.5 6.44 3.94 0.22 0.23
Caldicot     11.8 5.44 3.94 0.30 0.04
Caldicot     11.6 5.39 3.77 0.29 0.06
IslandThorns 18.3 1.28 0.67 0.03 0.03
IslandThorns 15.8 2.39 0.63 0.01 0.04
IslandThorns 18.0 1.50 0.67 0.01 0.06
IslandThorns 18.0 1.88 0.68 0.01 0.04
IslandThorns 20.8 1.51 0.72 0.07 0.10
AshleyRails  17.7 1.12 0.56 0.06 0.06
AshleyRails  18.3 1.14 0.67 0.06 0.05
AshleyRails  16.7 0.92 0.53 0.01 0.05
AshleyRails  14.8 2.74 0.67 0.03 0.05
AshleyRails  19.1 1.64 0.60 0.10 0.03
;
proc glm data=pottery;
   class Site;
   model Al Fe Mg Ca Na = Site;
   contrast 'Llanederyn vs. the rest' Site 1 1 1 -3;
   manova h=_all_ / printe printh;
run;
```

After the summary information, displayed in Output 32.6.1, PROC GLM produces the univariate analyses for each of the dependent variables, as shown in Output 32.6.2 to Output 32.6.6. These analyses show that sites are significantly different for all oxides individually. You can suppress these univariate analyses by specifying the NOUNI option in the MODEL statement.

Output 32.6.1. Summary Information on Groups

```
                              Romano-British Pottery

                                The GLM Procedure

                             Class Level Information

     Class          Levels    Values

     Site                4    AshleyRails Caldicot IslandThorns Llanederyn


                   Number of Observations Read          26
                   Number of Observations Used          26
```

Output 32.6.2. Univariate Analysis of Variance for Aluminum Oxide

```
                             Romano-British Pottery

                               The GLM Procedure

Dependent Variable: Al

                                         Sum of
 Source                      DF         Squares    Mean Square   F Value   Pr > F

 Model                        3     175.6103187     58.5367729     26.67   <.0001

 Error                       22      48.2881429      2.1949156

 Corrected Total             25     223.8984615


               R-Square     Coeff Var      Root MSE       Al Mean

               0.784330      10.22284      1.481525      14.49231


 Source                      DF       Type I SS    Mean Square   F Value   Pr > F

 Site                         3     175.6103187     58.5367729     26.67   <.0001


 Source                      DF     Type III SS    Mean Square   F Value   Pr > F

 Site                         3     175.6103187     58.5367729     26.67   <.0001


  Contrast                       DF    Contrast SS    Mean Square  F Value  Pr > F

  Llanederyn vs. the rest         1    58.58336640    58.58336640    26.69  <.0001
```

Output 32.6.3. Univariate Analysis of Variance for Iron Oxide

```
                              Romano-British Pottery

                                The GLM Procedure

Dependent Variable: Fe

                                             Sum of
 Source                      DF             Squares     Mean Square    F Value    Pr > F

 Model                        3         134.2216158      44.7405386      89.88    <.0001

 Error                       22          10.9508457       0.4977657

 Corrected Total             25         145.1724615


               R-Square     Coeff Var      Root MSE       Fe Mean

               0.924567      15.79171      0.705525      4.467692


 Source                      DF           Type I SS     Mean Square    F Value    Pr > F

 Site                         3         134.2216158      44.7405386      89.88    <.0001


 Source                      DF         Type III SS     Mean Square    F Value    Pr > F

 Site                         3         134.2216158      44.7405386      89.88    <.0001


  Contrast                       DF    Contrast SS    Mean Square  F Value  Pr > F

  Llanederyn vs. the rest         1    71.15144132    71.15144132   142.94  <.0001
```

Output 32.6.4. Univariate Analysis of Variance for Calcium Oxide

```
                              Romano-British Pottery

                                The GLM Procedure

Dependent Variable: Mg

                                             Sum of
 Source                      DF         Squares     Mean Square    F Value    Pr > F

 Model                        3     103.3505270      34.4501757      49.12    <.0001

 Error                       22      15.4296114       0.7013460

 Corrected Total             25     118.7801385


               R-Square     Coeff Var      Root MSE       Mg Mean

               0.870099      26.65777      0.837464      3.141538


 Source                      DF       Type I SS     Mean Square    F Value    Pr > F

 Site                         3     103.3505270      34.4501757      49.12    <.0001


 Source                      DF     Type III SS     Mean Square    F Value    Pr > F

 Site                         3     103.3505270      34.4501757      49.12    <.0001


  Contrast                      DF    Contrast SS    Mean Square   F Value   Pr > F

  Llanederyn vs. the rest        1    56.59349339    56.59349339     80.69   <.0001
```

Output 32.6.5. Univariate Analysis of Variance for Magnesium Oxide

```
                              Romano-British Pottery

                                The GLM Procedure

Dependent Variable: Ca

                                          Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                        3       0.20470275     0.06823425     29.16   <.0001

 Error                       22       0.05148571     0.00234026

 Corrected Total             25       0.25618846


               R-Square     Coeff Var      Root MSE       Ca Mean

               0.799032      33.01265      0.048376      0.146538


 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 Site                         3     0.20470275     0.06823425     29.16   <.0001


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 Site                         3     0.20470275     0.06823425     29.16   <.0001


  Contrast                      DF   Contrast SS   Mean Square  F Value  Pr > F

  Llanederyn vs. the rest        1    0.03531688    0.03531688    15.09  0.0008
```

Output 32.6.6. Univariate Analysis of Variance for Sodium Oxide

```
                              Romano-British Pottery

                                The GLM Procedure

Dependent Variable: Na

                                              Sum of
 Source                      DF          Squares     Mean Square    F Value    Pr > F

 Model                        3       0.25824560      0.08608187       9.50    0.0003

 Error                       22       0.19929286      0.00905877

 Corrected Total             25       0.45753846


               R-Square     Coeff Var      Root MSE       Na Mean

               0.564424      60.06350      0.095178      0.158462


 Source                      DF        Type I SS     Mean Square    F Value    Pr > F

 Site                         3       0.25824560      0.08608187       9.50    0.0003


 Source                      DF      Type III SS     Mean Square    F Value    Pr > F

 Site                         3       0.25824560      0.08608187       9.50    0.0003


  Contrast                      DF    Contrast SS    Mean Square   F Value   Pr > F

  Llanederyn vs. the rest        1     0.23344446     0.23344446     25.77   <.0001
```

The PRINTE option in the MANOVA statement displays the elements of the error matrix, also called the Error Sums of Squares and Crossproducts matrix. See Output 32.6.7. The diagonal elements of this matrix are the error sums of squares from the corresponding univariate analyses.

The PRINTE option also displays the partial correlation matrix associated with the E matrix. In this example, none of the oxides are very strongly correlated; the strongest correlation ($r = 0.488$) is between magnesium oxide and calcium oxide.

Output 32.6.7. Error SSCP Matrix and Partial Correlations

```
                              Romano-British Pottery

                                The GLM Procedure
                          Multivariate Analysis of Variance

                              E = Error SSCP Matrix

                Al              Fe              Mg              Ca              Na

Al    48.288142857    7.0800714286    0.6080142857    0.1064714286    0.5889571429
Fe    7.0800714286    10.950845714    0.5270571429    -0.155194286    0.0667585714
Mg    0.6080142857    0.5270571429    15.429611429    0.4353771429    0.0276157143
Ca    0.1064714286    -0.155194286    0.4353771429    0.0514857143    0.0100785714
Na    0.5889571429    0.0667585714    0.0276157143    0.0100785714    0.1992928571


  Partial Correlation Coefficients from the Error SSCP Matrix / Prob > |r|

DF = 22               Al             Fe             Mg             Ca             Na

Al              1.000000       0.307889       0.022275       0.067526       0.189853
                                 0.1529         0.9196         0.7595         0.3856

Fe              0.307889       1.000000       0.040547      -0.206685       0.045189
                  0.1529                        0.8543         0.3440         0.8378

Mg              0.022275       0.040547       1.000000       0.488478       0.015748
                  0.9196         0.8543                        0.0180         0.9431

Ca              0.067526      -0.206685       0.488478       1.000000       0.099497
                  0.7595         0.3440         0.0180                        0.6515

Na              0.189853       0.045189       0.015748       0.099497       1.000000
                  0.3856         0.8378         0.9431         0.6515
```

The PRINTH option produces the SSCP matrix for the hypotheses being tested (Site and the contrast); see Output 32.6.8 and Output 32.6.9. Since the Type III SS are the highest level SS produced by PROC GLM by default, and since the HTYPE= option is not specified, the SSCP matrix for Site gives the Type III $\mathbf{H}$ matrix. The diagonal elements of this matrix are the model sums of squares from the corresponding univariate analyses.

Four multivariate tests are computed, all based on the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$. These roots and vectors are displayed along with the tests. All four tests can be transformed to variates that have F distributions under the null hypothesis. Note that the four tests all give the same results for the contrast, since it has only one degree of freedom. In this case, the multivariate analysis matches the univariate results: there is an overall difference between the chemical composition of samples from different sites, and the samples from Llanederyn are different from the average of the other sites.

Output 32.6.8. Hypothesis SSCP Matrix and Multivariate Tests for Overall Site Effect

```
                                  Romano-British Pottery

                                    The GLM Procedure
                              Multivariate Analysis of Variance

                              H = Type III SSCP Matrix for Site

                    Al                 Fe                 Mg                 Ca                 Na

  Al      175.61031868       -149.295533       -130.8097066       -5.889163736       -5.372264835
  Fe       -149.295533      134.22161582       117.74503516       4.8217865934       5.3259491209
  Mg      -130.8097066      117.74503516       103.35052703       4.2091613187       4.7105458242
  Ca      -5.889163736      4.8217865934       4.2091613187       0.2047027473        0.154782967
  Na      -5.372264835      5.3259491209       4.7105458242        0.154782967       0.2582456044


                   Characteristic Roots and Vectors of: E Inverse * H, where
                              H = Type III SSCP Matrix for Site
                                    E = Error SSCP Matrix

 Characteristic           Characteristic Vector  V'EV=1
           Root  Percent             Al             Fe             Mg             Ca             Na

     34.1611140    96.39     0.09562211    -0.26330469    -0.05305978    -1.87982100    -0.47071123
      1.2500994     3.53     0.02651891    -0.01239715     0.17564390    -4.25929785     1.23727668
      0.0275396     0.08     0.09082220     0.13159869     0.03508901    -0.15701602    -1.39364544
      0.0000000     0.00     0.03673984    -0.15129712     0.20455529     0.54624873    -0.17402107
      0.0000000     0.00     0.06862324     0.03056912    -0.10662399     2.51151978     1.23668841


    MANOVA Test Criteria and F Approximations for the Hypothesis of No Overall Site Effect
                              H = Type III SSCP Matrix for Site
                                    E = Error SSCP Matrix

                                     S=3    M=0.5    N=8

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.01230091      13.09        15    50.091    <.0001
        Pillai's Trace              1.55393619       4.30        15        60    <.0001
        Hotelling-Lawley Trace     35.43875302      40.59        15     29.13    <.0001
        Roy's Greatest Root        34.16111399     136.64         5        20    <.0001

                 NOTE: F Statistic for Roy's Greatest Root is an upper bound.
```

Output 32.6.9. Hypothesis SSCP Matrix and Multivariate Tests for Differences Between Llanederyn and the Rest

```
                                    Romano-British Pottery

                                      The GLM Procedure
                              Multivariate Analysis of Variance

                     H = Contrast SSCP Matrix for Llanederyn vs. the rest

                   Al                Fe                Mg                Ca                Na

 Al      58.583366402      -64.56230291      -57.57983466      -1.438395503      -3.698102513
 Fe      -64.56230291      71.151441323      63.456352116      1.5851961376      4.0755256878
 Mg      -57.57983466      63.456352116      56.593493386      1.4137558201      3.6347541005
 Ca      -1.438395503      1.5851961376      1.4137558201      0.0353168783      0.0907993915
 Na      -3.698102513      4.0755256878      3.6347541005      0.0907993915      0.2334444577


                  Characteristic Roots and Vectors of: E Inverse * H, where
                     H = Contrast SSCP Matrix for Llanederyn vs. the rest
                                    E = Error SSCP Matrix

Characteristic           Characteristic Vector  V’EV=1
          Root  Percent            Al            Fe            Mg            Ca            Na

    16.1251646   100.00   -0.08883488    0.25458141    0.08723574    0.98158668    0.71925759
     0.0000000     0.00   -0.00503538    0.03825743   -0.17632854    5.16256699   -0.01022754
     0.0000000     0.00    0.00162771   -0.08885364   -0.01774069   -0.83096817    2.17644566
     0.0000000     0.00    0.04450136   -0.15722494    0.22156791    0.00000000    0.00000000
     0.0000000     0.00    0.11939206    0.10833549    0.00000000    0.00000000    0.00000000


                MANOVA Test Criteria and Exact F Statistics for the Hypothesis
                         of No Overall Llanederyn vs. the rest Effect
                     H = Contrast SSCP Matrix for Llanederyn vs. the rest
                                    E = Error SSCP Matrix

                                     S=1    M=1.5    N=8

       Statistic                        Value    F Value    Num DF    Den DF    Pr > F

       Wilks’ Lambda               0.05839360      58.05         5        18    <.0001
       Pillai’s Trace              0.94160640      58.05         5        18    <.0001
       Hotelling-Lawley Trace     16.12516462      58.05         5        18    <.0001
       Roy’s Greatest Root        16.12516462      58.05         5        18    <.0001
```

Example 32.7. Repeated Measures Analysis of Variance

This example uses data from Cole and Grizzle (1966) to illustrate a commonly occurring repeated measures ANOVA design. Sixteen dogs are randomly assigned to four groups. (One animal is removed from the analysis due to a missing value for one dependent variable.) Dogs in each group receive either morphine or trimethaphan (variable Drug) and have either depleted or intact histamine levels (variable Depleted) before receiving the drugs. The dependent variable is the blood concentration of histamine at 0, 1, 3, and 5 minutes after injection of the drug. Logarithms are applied to these concentrations to minimize correlation between the mean and the variance of the data.

The following SAS statements perform both univariate and multivariate repeated measures analyses and produce Output 32.7.1 through Output 32.7.7:

```
data dogs;
   input Drug $12. Depleted $ Histamine0 Histamine1
         Histamine3 Histamine5;
   LogHistamine0=log(Histamine0);
   LogHistamine1=log(Histamine1);
   LogHistamine3=log(Histamine3);
   LogHistamine5=log(Histamine5);
   datalines;
Morphine      N  .04  .20  .10  .08
Morphine      N  .02  .06  .02  .02
Morphine      N  .07 1.40  .48  .24
Morphine      N  .17  .57  .35  .24
Morphine      Y  .10  .09  .13  .14
Morphine      Y  .12  .11  .10   .
Morphine      Y  .07  .07  .06  .07
Morphine      Y  .05  .07  .06  .07
Trimethaphan  N  .03  .62  .31  .22
Trimethaphan  N  .03 1.05  .73  .60
Trimethaphan  N  .07  .83 1.07  .80
Trimethaphan  N  .09 3.13 2.06 1.23
Trimethaphan  Y  .10  .09  .09  .08
Trimethaphan  Y  .08  .09  .09  .10
Trimethaphan  Y  .13  .10  .12  .12
Trimethaphan  Y  .06  .05  .05  .05
;
proc glm;
   class Drug Depleted;
   model LogHistamine0--LogHistamine5 =
         Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
run;
```

The NOUNI option in the MODEL statement suppresses the individual ANOVA tables for the original dependent variables. These analyses are usually of no interest in a repeated measures analysis. The POLYNOMIAL option in the REPEATED statement indicates that the transformation used to implement the repeated measures analysis is an orthogonal polynomial transformation, and the SUMMARY option requests that the univariate analyses for the orthogonal polynomial contrast variables be displayed. The parenthetical numbers (0 1 3 5) determine the spacing of the orthogonal polynomials used in the analysis. The output is displayed in Output 32.7.1 through Output 32.7.7.

Output 32.7.1. Summary Information on Groups

```
                                       The GLM Procedure

                                    Class Level Information

                         Class         Levels    Values

                         Drug               2    Morphine Trimethaphan

                         Depleted           2    N Y


                            Number of Observations Read          16
                            Number of Observations Used          15



                                       The GLM Procedure
                             Repeated Measures Analysis of Variance
                           Analysis of Variance of Contrast Variables

Time_N represents the nth degree polynomial contrast for Time

Contrast Variable: Time_1


Contrast Variable: Time_2


Contrast Variable: Time_3
```

The “Repeated Measures Level Information” table gives information on the repeated measures effect; it is displayed in Output 32.7.2. In this example, the within-subject (within-dog) effect is Time, which has the levels 0, 1, 3, and 5.

Output 32.7.2. Repeated Measures Levels

```
                                       The GLM Procedure
                             Repeated Measures Analysis of Variance

                              Repeated Measures Level Information

                                            Log        Log        Log        Log
               Dependent Variable    Histamine0 Histamine1 Histamine3 Histamine5

                    Level of Time             0          1          3          5
```

The multivariate analyses for within-subject effects and related interactions are displayed in Output 32.7.3. For the example, the first table displayed shows that the TIME effect is significant. In addition, the Time*Drug*Depleted interaction is significant, as shown in the fourth table. This means that the effect of Time on the blood concentration of histamine is different for the four Drug*Depleted combinations studied.

Output 32.7.3. Multivariate Tests of Within-Subject Effects

```
                                 The GLM Procedure
                       Repeated Measures Analysis of Variance

         MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time Effect
                             H = Type III SSCP Matrix for Time
                                   E = Error SSCP Matrix

                                   S=1    M=0.5    N=3.5

         Statistic                        Value    F Value    Num DF    Den DF    Pr > F

         Wilks' Lambda               0.11097706      24.03         3         9    0.0001
         Pillai's Trace              0.88902294      24.03         3         9    0.0001
         Hotelling-Lawley Trace      8.01087137      24.03         3         9    0.0001
         Roy's Greatest Root         8.01087137      24.03         3         9    0.0001


      MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time*Drug Effect
                          H = Type III SSCP Matrix for Time*Drug
                                   E = Error SSCP Matrix

                                   S=1    M=0.5    N=3.5

         Statistic                        Value    F Value    Num DF    Den DF    Pr > F

         Wilks' Lambda               0.34155984       5.78         3         9    0.0175
         Pillai's Trace              0.65844016       5.78         3         9    0.0175
         Hotelling-Lawley Trace      1.92774470       5.78         3         9    0.0175
         Roy's Greatest Root         1.92774470       5.78         3         9    0.0175


    MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time*Depleted Effect
                        H = Type III SSCP Matrix for Time*Depleted
                                   E = Error SSCP Matrix

                                   S=1    M=0.5    N=3.5

         Statistic                        Value    F Value    Num DF    Den DF    Pr > F

         Wilks' Lambda               0.12339988      21.31         3         9    0.0002
         Pillai's Trace              0.87660012      21.31         3         9    0.0002
         Hotelling-Lawley Trace      7.10373567      21.31         3         9    0.0002
         Roy's Greatest Root         7.10373567      21.31         3         9    0.0002


  MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time*Drug*Depleted Effect
                      H = Type III SSCP Matrix for Time*Drug*Depleted
                                   E = Error SSCP Matrix

                                   S=1    M=0.5    N=3.5

         Statistic                        Value    F Value    Num DF    Den DF    Pr > F

         Wilks' Lambda               0.19383010      12.48         3         9    0.0015
         Pillai's Trace              0.80616990      12.48         3         9    0.0015
         Hotelling-Lawley Trace      4.15915732      12.48         3         9    0.0015
         Roy's Greatest Root         4.15915732      12.48         3         9    0.0015
```

Output 32.7.4 displays tests of hypotheses for between-subject (between-dog) effects. This section tests the hypotheses that the different Drugs, Depleteds, and their interactions have no effects on the dependent variables, while ignoring the within-dog effects. From this analysis, there is a significant between-dog effect for Depleted (p-value=0.0229). The interaction and the main effect for Drug are not significant (p-values=0.1734 and 0.1281, respectively).

Output 32.7.4. Tests of Between-Subject Effects

```
                                The GLM Procedure
                       Repeated Measures Analysis of Variance
                  Tests of Hypotheses for Between Subjects Effects

       Source                       DF     Type III SS     Mean Square    F Value    Pr > F

       Drug                          1      5.99336243      5.99336243       2.71    0.1281
       Depleted                      1     15.44840703     15.44840703       6.98    0.0229
       Drug*Depleted                 1      4.69087508      4.69087508       2.12    0.1734
       Error                        11     24.34683348      2.21334850
```

Univariate analyses for within-subject (within-dog) effects and related interactions are displayed in Output 32.7.6. The results for this example are the same as for the multivariate analyses; this is not always the case. In addition, before the univariate analyses are used to make conclusions about the data, the result of the sphericity test (requested with the PRINTE option in the REPEATED statement and displayed in Output 32.7.5) should be examined. If the sphericity test is rejected, use the adjusted G-G or H-F probabilities. See the “Repeated Measures Analysis of Variance” section on page 1825 for more information.

Output 32.7.5. Sphericity Test

```
                                The GLM Procedure
                       Repeated Measures Analysis of Variance

                                Sphericity Tests

                                        Mauchly’s
             Variables                DF    Criterion    Chi-Square    Pr > ChiSq

             Transformed Variates      5    0.1752641     16.930873        0.0046
             Orthogonal Components     5    0.1752641     16.930873        0.0046
```

Output 32.7.6. Univariate Tests of Within-Subject Effects

```
                                The GLM Procedure
                       Repeated Measures Analysis of Variance
               Univariate Tests of Hypotheses for Within Subject Effects

                                                                                  Adj Pr > F
 Source                      DF    Type III SS    Mean Square   F Value   Pr > F    G - G    H - F

 Time                         3    12.05898677     4.01966226     53.44   <.0001   <.0001   <.0001
 Time*Drug                    3     1.84429514     0.61476505      8.17   0.0003   0.0039   0.0008
 Time*Depleted                3    12.08978557     4.02992852     53.57   <.0001   <.0001   <.0001
 Time*Drug*Depleted           3     2.93077939     0.97692646     12.99   <.0001   0.0005   <.0001
 Error(Time)                 33     2.48238887     0.07522391


                              Greenhouse-Geisser Epsilon    0.5694
                              Huynh-Feldt Epsilon           0.8475
```

Output 32.7.7 is produced by the SUMMARY option in the REPEATED statement. If the POLYNOMIAL option is not used, a similar table is displayed using the default CONTRAST transformation. The linear, quadratic, and cubic trends for Time, labeled as ‘Time_1’, ‘Time_2’, and ‘Time_3’, are displayed, and in each case, the Source labeled ‘Mean’ gives a test for the respective trend.

Output 32.7.7. Tests of Between-Subject Effects for Transformed Variables

```
                              The GLM Procedure
                    Repeated Measures Analysis of Variance
                  Analysis of Variance of Contrast Variables

Time_N represents the nth degree polynomial contrast for Time

Contrast Variable: Time_1


      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Mean                         1      2.00963483      2.00963483      34.99    0.0001
      Drug                         1      1.18069076      1.18069076      20.56    0.0009
      Depleted                     1      1.36172504      1.36172504      23.71    0.0005
      Drug*Depleted                1      2.04346848      2.04346848      35.58    <.0001
      Error                       11      0.63171161      0.05742833


Contrast Variable: Time_2


      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Mean                         1      5.40988418      5.40988418      57.15    <.0001
      Drug                         1      0.59173192      0.59173192       6.25    0.0295
      Depleted                     1      5.94945506      5.94945506      62.86    <.0001
      Drug*Depleted                1      0.67031587      0.67031587       7.08    0.0221
      Error                       11      1.04118707      0.09465337


Contrast Variable: Time_3


      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Mean                         1      4.63946776      4.63946776      63.04    <.0001
      Drug                         1      0.07187246      0.07187246       0.98    0.3443
      Depleted                     1      4.77860547      4.77860547      64.94    <.0001
      Drug*Depleted                1      0.21699504      0.21699504       2.95    0.1139
      Error                       11      0.80949018      0.07359002
```

Example 32.8. Mixed Model Analysis of Variance Using the RANDOM Statement

Milliken and Johnson (1984) present an example of an unbalanced mixed model. Three machines, which are considered as a fixed effect, and six employees, which are considered a random effect, are studied. Each employee operates each machine for either one, two, or three different times. The dependent variable is an overall rating, which takes into account the number and quality of components produced.

The following statements form the data set and perform a mixed model analysis of variance by requesting the TEST option in the RANDOM statement. Note that the machine*person interaction is declared as a random effect; in general, when an interaction involves a random effect, it too should be declared as random. The results of the analysis are shown in Output 32.8.1 through Output 32.8.4.

```
data machine;
   input machine person rating @@;
   datalines;
1 1 52.0  1 2 51.8  1 2 52.8  1 3 60.0  1 4 51.1  1 4 52.3
1 5 50.9  1 5 51.8  1 5 51.4  1 6 46.4  1 6 44.8  1 6 49.2
2 1 64.0  2 2 59.7  2 2 60.0  2 2 59.0  2 3 68.6  2 3 65.8
2 4 63.2  2 4 62.8  2 4 62.2  2 5 64.8  2 5 65.0  2 6 43.7
2 6 44.2  2 6 43.0  3 1 67.5  3 1 67.2  3 1 66.9  3 2 61.5
3 2 61.7  3 2 62.3  3 3 70.8  3 3 70.6  3 3 71.0  3 4 64.1
3 4 66.2  3 4 64.0  3 5 72.1  3 5 72.0  3 5 71.1  3 6 62.0
3 6 61.4  3 6 60.5
;

proc glm data=machine;
   class machine person;
   model rating=machine person machine*person;
   random person machine*person / test;
run;
```

The TEST option in the RANDOM statement requests that PROC GLM determine the appropriate F-tests based on person and machine*person being treated as random effects. As you can see in Output 32.8.4, this requires that a linear combination of mean squares be constructed to test both the machine and person hypotheses; thus, F-tests using Satterthwaite approximations are used.

Output 32.8.1. Summary Information on Groups

```
                          The GLM Procedure

                       Class Level Information

                  Class         Levels    Values

                  machine            3    1 2 3

                  person             6    1 2 3 4 5 6


                Number of Observations Read          44
                Number of Observations Used          44
```

Output 32.8.2. Fixed-Effect Model Analysis of Variance

```
                               The GLM Procedure

Dependent Variable: rating

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                       17    3061.743333     180.102549    206.41   <.0001

 Error                       26      22.686667       0.872564

 Corrected Total             43    3084.430000


              R-Square     Coeff Var      Root MSE    rating Mean

              0.992645      1.560754      0.934111       59.85000


 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 machine                      2    1648.664722     824.332361    944.72   <.0001
 person                       5    1008.763583     201.752717    231.22   <.0001
 machine*person              10     404.315028      40.431503     46.34   <.0001


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 machine                      2    1238.197626     619.098813    709.52   <.0001
 person                       5    1011.053834     202.210767    231.74   <.0001
 machine*person              10     404.315028      40.431503     46.34   <.0001
```

Output 32.8.3. Expected Values of Type III Mean Squares

```
                               The GLM Procedure

Source                  Type III Expected Mean Square

machine                 Var(Error) + 2.137 Var(machine*person) + Q(machine)

person                  Var(Error) + 2.2408 Var(machine*person) + 6.7224
                        Var(person)

machine*person          Var(Error) + 2.3162 Var(machine*person)
```

Output 32.8.4. Mixed Model Analysis of Variance

```
                              The GLM Procedure
            Tests of Hypotheses for Mixed Model Analysis of Variance

Dependent Variable: rating

 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 machine                     2    1238.197626     619.098813     16.57   0.0007

 Error                  10.036     375.057436      37.370384
 Error: 0.9226*MS(machine*person) + 0.0774*MS(Error)


 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 person                      5    1011.053834     202.210767      5.17   0.0133

 Error                  10.015     392.005726      39.143708
 Error: 0.9674*MS(machine*person) + 0.0326*MS(Error)


 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 machine*person             10     404.315028      40.431503     46.34   <.0001

 Error: MS(Error)           26      22.686667       0.872564
```

Note that you can also use the MIXED procedure to analyze mixed models. The following statements use PROC MIXED to reproduce the mixed model analysis of variance; the relevant part of the PROC MIXED results is shown in Output 32.8.5

```
proc mixed data=machine method=type3;
   class machine person;
   model rating = machine;
   random person machine*person;
run;
```

Output 32.8.5. PROC MIXED Mixed Model Analysis of Variance (Partial Output)

```
                              The Mixed Procedure

                          Type 3 Analysis of Variance

                                                Sum of
             Source                 DF         Squares     Mean Square

             machine                 2     1238.197626      619.098813
             person                  5     1011.053834      202.210767
             machine*person         10      404.315028       40.431503
             Residual               26       22.686667        0.872564

                          Type 3 Analysis of Variance

Source          Expected Mean Square

machine         Var(Residual) + 2.137 Var(machine*person) + Q(machine)
person          Var(Residual) + 2.2408 Var(machine*person) + 6.7224 Var(person)
machine*person  Var(Residual) + 2.3162 Var(machine*person)
Residual        Var(Residual)

                          Type 3 Analysis of Variance

                                                         Error
Source          Error Term                                  DF  F Value  Pr > F

machine         0.9226 MS(machine*person)               10.036    16.57  0.0007
                + 0.0774 MS(Residual)
person          0.9674 MS(machine*person)               10.015     5.17  0.0133
                + 0.0326 MS(Residual)
machine*person  MS(Residual)                                26    46.34  <.0001
Residual        .                                            .        .       .
```

The advantage of PROC MIXED is that it offers more versatility for mixed models; the disadvantage is that it can be less computationally efficient for large data sets. See Chapter 46, “The MIXED Procedure,” for more details.

Example 32.9. Analyzing a Doubly-multivariate Repeated Measures Design

This example shows how to analyze a doubly-multivariate repeated measures design by using PROC GLM with an IDENTITY factor in the REPEATED statement. Note that this differs from previous releases of PROC GLM, in which you had to use a MANOVA statement to get a doubly repeated measures analysis.

Two responses, Y1 and Y2, are each measured three times for each subject (pretreatment, posttreatment, and in a later follow-up). Each subject receives one of three treatments; A, B, or the control. In PROC GLM, you use a REPEATED factor of type IDENTITY to identify the different responses and another repeated factor to identify the different measurement times. The repeated measures analysis includes multivariate tests for time and treatment main effects, as well as their interactions, across responses. The following statements produce Output 32.9.1 through Output 32.9.3.

```
data Trial;
   input Treatment $ Repetition PreY1 PostY1 FollowY1
                                PreY2 PostY2 FollowY2;
   datalines;
A        1  3  13  9  0  0  9
A        2  0  14 10  6  6  3
A        3  4   6 17  8  2  6
A        4  7   7 13  7  6  4
A        5  3  12 11  6 12  6
A        6 10  14  8 13  3  8
B        1  9  11 17  8 11 27
B        2  4  16 13  9  3 26
B        3  8  10  9 12  0 18
B        4  5   9 13  3  0 14
B        5  0  15 11  3  0 25
B        6  4  11 14  4  2  9
Control  1 10  12 15  4  3  7
Control  2  2   8 12  8  7 20
Control  3  4   9 10  2  0 10
Control  4 10   8  8  5  8 14
Control  5 11  11 11  1  0 11
Control  6  1  5  15  8  9 10
;

proc glm data=Trial;
   class Treatment;
   model PreY1 PostY1 FollowY1
         PreY2 PostY2 FollowY2 = Treatment / nouni;
   repeated Response 2 identity, Time 3;
run;
```

Output 32.9.1. A Doubly-multivariate Repeated Measures Design

```
                                       The GLM Procedure

                                    Class Level Information

                             Class          Levels    Values

                             Treatment           3    A B Control


                            Number of Observations Read          18
                            Number of Observations Used          18
```

The levels of the repeated factors are displayed in Output 32.9.2. Note that RESPONSE is 1 for all the Y1 measurements and 2 for all the Y2 measurements, while the three levels of Time identify the pretreatment, posttreatment, and follow-up measurements within each response. The multivariate tests for within-subject effects are displayed in Output 32.9.3.

Output 32.9.2. Repeated Factor Levels

```
                                       The GLM Procedure
                             Repeated Measures Analysis of Variance

                              Repeated Measures Level Information

          Dependent Variable       PreY1   PostY1 FollowY1    PreY2   PostY2 FollowY2

           Level of Response           1        1        1        2        2        2
               Level of Time           1        2        3        1        2        3
```

Output 32.9.3. Within-subject Tests

```
                                The GLM Procedure
                       Repeated Measures Analysis of Variance

      MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Response Effect
                       H = Type III SSCP Matrix for Response
                               E = Error SSCP Matrix

                                 S=1    M=0    N=6

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.02165587     316.24         2        14    <.0001
        Pillai's Trace              0.97834413     316.24         2        14    <.0001
        Hotelling-Lawley Trace     45.17686368     316.24         2        14    <.0001
        Roy's Greatest Root        45.17686368     316.24         2        14    <.0001


  MANOVA Test Criteria and F Approximations for the Hypothesis of no Response*Treatment Effect
                  H = Type III SSCP Matrix for Response*Treatment
                               E = Error SSCP Matrix

                                S=2    M=-0.5    N=6

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.72215797       1.24         4        28    0.3178
        Pillai's Trace              0.27937444       1.22         4        30    0.3240
        Hotelling-Lawley Trace      0.38261660       1.31         4    15.818    0.3074
        Roy's Greatest Root         0.37698780       2.83         2        15    0.0908

                 NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                         NOTE: F Statistic for Wilks' Lambda is exact.


   MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Response*Time Effect
                     H = Type III SSCP Matrix for Response*Time
                               E = Error SSCP Matrix

                                 S=1    M=1    N=5

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.14071380      18.32         4        12    <.0001
        Pillai's Trace              0.85928620      18.32         4        12    <.0001
        Hotelling-Lawley Trace      6.10662362      18.32         4        12    <.0001
        Roy's Greatest Root         6.10662362      18.32         4        12    <.0001


                        MANOVA Test Criteria and F Approximations for the
                         Hypothesis of no Response*Time*Treatment Effect
                       H = Type III SSCP Matrix for Response*Time*Treatment
                               E = Error SSCP Matrix

                                S=2    M=0.5    N=5

        Statistic                        Value    F Value    Num DF    Den DF    Pr > F

        Wilks' Lambda               0.22861451       3.27         8        24    0.0115
        Pillai's Trace              0.96538785       3.03         8        26    0.0151
        Hotelling-Lawley Trace      2.52557514       3.64         8        15    0.0149
        Roy's Greatest Root         2.12651905       6.91         4        13    0.0033

                 NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                         NOTE: F Statistic for Wilks' Lambda is exact.
```

The table for Response*Treatment tests for an overall treatment effect across the two responses; likewise, the tables for Response*Time and Response*Treatment*Time test for time and the treatment-by-time interaction, respectively. In this case, there is a strong main effect for time and possibly for the interaction, but not for treatment.

In previous releases (before the IDENTITY transformation was introduced), in order to perform a doubly repeated measures analysis, you had to use a MANOVA statement with a customized transformation matrix M. You might still want to use this approach to see details of the analysis, such as the univariate ANOVA for each transformed variate. The following statements demonstrate this approach by using the MANOVA statement to test for the overall main effect of time and specifying the SUMMARY option.

```
proc glm data=Trial;
   class Treatment;
   model PreY1 PostY1 FollowY1
         PreY2 PostY2 FollowY2 = Treatment / nouni;
   manova  h=intercept  m=prey1 - posty1,
                          prey1 - followy1,
                          prey2 - posty2,
                          prey2 - followy2 / summary;
run;
```

The M matrix used to perform the test for time effects is displayed in Output 32.9.4, while the results of the multivariate test are given in Output 32.9.5. Note that the test results are the same as for the Response*Time effect in Output 32.9.3.

Output 32.9.4. M Matrix to Test for Time Effect (Repeated Measure)

```
                                     The GLM Procedure
                             Multivariate Analysis of Variance

                          M Matrix Describing Transformed Variables

                 PreY1         PostY1       FollowY1          PreY2         PostY2       FollowY2

MVAR1                1             -1              0              0              0              0
MVAR2                1              0             -1              0              0              0
MVAR3                0              0              0              1             -1              0
MVAR4                0              0              0              1              0             -1
```

Output 32.9.5. Tests for Time Effect (Repeated Measure)

```
                                 The GLM Procedure
                            Multivariate Analysis of Variance

                Characteristic Roots and Vectors of: E Inverse * H, where
                          H = Type III SSCP Matrix for Intercept
                                  E = Error SSCP Matrix

                     Variables have been transformed by the M Matrix

 Characteristic               Characteristic Vector  V'EV=1
           Root     Percent          MVAR1           MVAR2           MVAR3           MVAR4

     6.10662362      100.00     -0.00157729      0.04081620     -0.04210209      0.03519437
     0.00000000        0.00      0.00796367      0.00493217      0.05185236      0.00377940
     0.00000000        0.00     -0.03534089     -0.01502146     -0.00283074      0.04259372
     0.00000000        0.00     -0.05672137      0.04500208      0.00000000      0.00000000


MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Intercept Effect
                  on the Variables Defined by the M Matrix Transformation
                         H = Type III SSCP Matrix for Intercept
                                  E = Error SSCP Matrix

                                  S=1    M=1    N=5

       Statistic                        Value    F Value    Num DF    Den DF    Pr > F

       Wilks' Lambda               0.14071380      18.32         4        12    <.0001
       Pillai's Trace              0.85928620      18.32         4        12    <.0001
       Hotelling-Lawley Trace      6.10662362      18.32         4        12    <.0001
       Roy's Greatest Root         6.10662362      18.32         4        12    <.0001
```

The SUMMARY option in the MANOVA statement creates an ANOVA table for each transformed variable as defined by the M matrix. MVAR1 and MVAR2 contrast the pretreatment measurement for Y1 with the posttreatment and follow-up measurements for Y1, respectively; MVAR3 and MVAR4 are the same contrasts for Y2. Output 32.9.6 displays these univariate ANOVA tables and shows that the contrasts are all strongly significant except for the pre-versus-post difference for Y2.

Output 32.9.6. Summary Output for the Test for Time Effect

```
                              The GLM Procedure
                        Multivariate Analysis of Variance

Dependent Variable: MVAR1

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     512.0000000     512.0000000      22.65    0.0003
      Error                       15     339.0000000      22.6000000


                              The GLM Procedure
                        Multivariate Analysis of Variance

Dependent Variable: MVAR2

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     813.3888889     813.3888889      32.87    <.0001
      Error                       15     371.1666667      24.7444444


                              The GLM Procedure
                        Multivariate Analysis of Variance

Dependent Variable: MVAR3

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1      68.0555556      68.0555556       3.49    0.0814
      Error                       15     292.5000000      19.5000000


                              The GLM Procedure
                        Multivariate Analysis of Variance

Dependent Variable: MVAR4

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     800.0000000     800.0000000      26.43    0.0001
      Error                       15     454.0000000      30.2666667
```

Example 32.10. Testing for Equal Group Variances

This example demonstrates how you can test for equal group variances in a one-way design. The data come from the University of Pennsylvania Smell Identification Test (UPSIT), reported in O’Brien and Heft (1995). The study is undertaken to explore how age and gender are related to sense of smell. A total of 180 subjects 20 to 89 years old are exposed to 40 different odors: for each odor, subjects are asked to choose which of four words best describes the odor. The Freeman-Tukey modified arcsine transformation (Bishop et al. 1975) is applied to the proportion of correctly identified odors to arrive at an olfactory index. For the following analysis, subjects are divided into five age groups:

$$\text{agegroup} = \left\{ \begin{array}{llcccc} 1 & \text{if} & & & \text{age} & \leq 25 \\ 2 & \text{if} & 25 & < & \text{age} & \leq 40 \\ 3 & \text{if} & 40 & < & \text{age} & \leq 55 \\ 4 & \text{if} & 55 & < & \text{age} & \leq 70 \\ 5 & \text{if} & 70 & < & \text{age} & \end{array} \right.$$

The following statements create a data set named upsit, containing the age group and olfactory index for each subject.

```
data upsit;
   input agegroup smell @@;
   datalines;
1 1.381  1 1.322  1 1.162  1 1.275  1 1.381  1 1.275  1 1.322
1 1.492  1 1.322  1 1.381  1 1.162  1 1.013  1 1.322  1 1.322
1 1.275  1 1.492  1 1.322  1 1.322  1 1.492  1 1.322  1 1.381
1 1.234  1 1.162  1 1.381  1 1.381  1 1.381  1 1.322  1 1.381
1 1.322  1 1.381  1 1.275  1 1.492  1 1.275  1 1.322  1 1.275
1 1.381  1 1.234  1 1.105
2 1.234  2 1.234  2 1.381  2 1.322  2 1.492  2 1.234  2 1.381
2 1.381  2 1.492  2 1.492  2 1.275  2 1.492  2 1.381  2 1.492
2 1.322  2 1.275  2 1.275  2 1.275  2 1.322  2 1.492  2 1.381
2 1.322  2 1.492  2 1.196  2 1.322  2 1.275  2 1.234  2 1.322
2 1.098  2 1.322  2 1.381  2 1.275  2 1.492  2 1.492  2 1.381
2 1.196
3 1.381  3 1.381  3 1.492  3 1.492  3 1.492  3 1.098  3 1.492
3 1.381  3 1.234  3 1.234  3 1.129  3 1.069  3 1.234  3 1.322
3 1.275  3 1.230  3 1.234  3 1.234  3 1.322  3 1.322  3 1.381
4 1.322  4 1.381  4 1.381  4 1.322  4 1.234  4 1.234  4 1.234
4 1.381  4 1.322  4 1.275  4 1.275  4 1.492  4 1.234  4 1.098
4 1.322  4 1.129  4 0.687  4 1.322  4 1.322  4 1.234  4 1.129
4 1.492  4 0.810  4 1.234  4 1.381  4 1.040  4 1.381  4 1.381
4 1.129  4 1.492  4 1.129  4 1.098  4 1.275  4 1.322  4 1.234
4 1.196  4 1.234  4 0.585  4 0.785  4 1.275  4 1.322  4 0.712
4 0.810
5 1.322  5 1.234  5 1.381  5 1.275  5 1.275  5 1.322  5 1.162
5 0.909  5 0.502  5 1.234  5 1.322  5 1.196  5 0.859  5 1.196
5 1.381  5 1.322  5 1.234  5 1.275  5 1.162  5 1.162  5 0.585
5 1.013  5 0.960  5 0.662  5 1.129  5 0.531  5 1.162  5 0.737
5 1.098  5 1.162  5 1.040  5 0.558  5 0.960  5 1.098  5 0.884
5 1.162  5 1.098  5 0.859  5 1.275  5 1.162  5 0.785  5 0.859
;
```

Older people are more at risk for problems with their sense of smell, and this should be reflected in significant differences in the mean of the olfactory index across the different age groups. However, many older people also have an excellent sense of smell, which implies that the older age groups should have greater variability. In order to test this hypothesis and to compute a one-way ANOVA for the olfactory index that is robust to the possibility of unequal group variances, you can use the HOVTEST and WELCH options in the MEANS statement for the GLM procedure, as shown in the following code.

```
proc glm data=upsit;
   class agegroup;
   model smell = agegroup;
   means agegroup / hovtest welch;
run;
```

Output 32.10.1, Output 32.10.2, and Output 32.10.3 display the usual ANOVA test for equal age group means, Levene's test for equal age group variances, and Welch's test for equal age group means, respectively. The hypotheses of age effects for mean and variance of the olfactory index are both confirmed.

Output 32.10.1. Usual ANOVA Test for Age Group Differences in Mean Olfactory Index

```
                               The GLM Procedure

Dependent Variable: smell

 Source                     DF      Type I SS    Mean Square   F Value   Pr > F

 agegroup                    4     2.13878141     0.53469535     16.65   <.0001
```

Output 32.10.2. Levene's Test for Age Group Differences in Olfactory Variability

```
                               The GLM Procedure

                 Levene's Test for Homogeneity of smell Variance
                  ANOVA of Squared Deviations from Group Means

                               Sum of        Mean
        Source           DF   Squares      Square    F Value    Pr > F

        agegroup          4    0.0799      0.0200       6.35    <.0001
        Error           175    0.5503     0.00314
```

Output 32.10.3. Welch's Test for Age Group Differences in Mean Olfactory Index

```
                               The GLM Procedure

                            Welch's ANOVA for smell

                   Source            DF    F Value    Pr > F

                   agegroup      4.0000      13.72    <.0001
                   Error        78.7489
```

Example 32.11. Analysis of a Screening Design

Yin and Jillie (1987) describe an experiment on a nitride etch process for a single wafer plasma etcher. The experiment is run using four factors: cathode power (power), gas flow (flow), reactor chamber pressure (pressure), and electrode gap (gap). Of interest are the main effects and interaction effects of the factors on the nitride etch rate (rate). The following statements create a SAS data set named HalfFraction, containing the factor settings and the observed etch rate for each of eight experimental runs.

```
data HalfFraction;
   input power flow pressure gap rate;
   datalines;
0.8   4.5 125 275     550
0.8   4.5 200 325     650
0.8 550.0 125 325     642
0.8 550.0 200 275     601
1.2   4.5 125 325     749
1.2   4.5 200 275    1052
1.2 550.0 125 275    1075
1.2 550.0 200 325     729
;
```

Notice that each of the factors has just two values. This is a common experimental design when the intent is to screen from the many factors that *might* affect the response the few that actually *do*. Since there are $2^4 = 16$ different possible settings of four two-level factors, this design with only eight runs is called a “half fraction.” The eight runs are chosen specifically to provide unambiguous information on main effects at the cost of confounding interaction effects with each other.

One way to analyze this data is simply to use PROC GLM to compute an analysis of variance, including both main effects and interactions in the model. The following statements demonstrate this approach.

```
proc glm data=HalfFraction;
   class power flow pressure gap;
   model rate=power|flow|pressure|gap@2;
run;
```

The ‘@2’ notation on the model statement includes all main effects and two-factor interactions between the factors. The output is shown in Output 32.11.1.

Output 32.11.1. Analysis of Variance for Nitride Etch Process Half Fraction

```
                                The GLM Procedure

                             Class Level Information

                         Class         Levels    Values

                         power              2    0.8 1.2

                         flow               2    4.5 550

                         pressure           2    125 200

                         gap                2    275 325


                     Number of Observations Read           8
                     Number of Observations Used           8



                                The GLM Procedure

Dependent Variable: rate

                                        Sum of
 Source                      DF        Squares    Mean Square   F Value   Pr > F

 Model                        7    280848.0000     40121.1429       .       .

 Error                        0         0.0000          .

 Corrected Total              7    280848.0000


               R-Square     Coeff Var      Root MSE     rate Mean

               1.000000             .             .      756.0000


 Source                      DF      Type I SS    Mean Square   F Value   Pr > F

 power                        1    168780.5000    168780.5000       .       .
 flow                         1       264.5000       264.5000       .       .
 power*flow                   1       200.0000       200.0000       .       .
 pressure                     1        32.0000        32.0000       .       .
 power*pressure               1      1300.5000      1300.5000       .       .
 flow*pressure                1     78012.5000     78012.5000       .       .
 gap                          1     32258.0000     32258.0000       .       .
 power*gap                    0         0.0000          .           .       .
 flow*gap                     0         0.0000          .           .       .
 pressure*gap                 0         0.0000          .           .       .


 Source                      DF    Type III SS    Mean Square   F Value   Pr > F

 power                        1    168780.5000    168780.5000       .       .
 flow                         1       264.5000       264.5000       .       .
 power*flow                   0         0.0000          .           .       .
 pressure                     1        32.0000        32.0000       .       .
 power*pressure               0         0.0000          .           .       .
 flow*pressure                0         0.0000          .           .       .
 gap                          1     32258.0000     32258.0000       .       .
 power*gap                    0         0.0000          .           .       .
 flow*gap                     0         0.0000          .           .       .
 pressure*gap                 0         0.0000          .           .       .
```

Notice that there are no error degrees of freedom. This is because there are 10 effects in the model (4 main effects plus 6 interactions) but only 8 observations in the data set. This is another cost of using a fractional design: not only is it impossible to estimate all the main effects and interactions, but there is also no information left to estimate the underlying error rate in order to measure the significance of the effects that are estimable.

Another thing to notice in Output 32.11.1 is the difference between the Type I and Type III ANOVA tables. The rows corresponding to main effects in each are the same, but no Type III interaction tests are estimable, while some Type I interaction tests are estimable. This indicates that there is *aliasing* in the design: some interactions are completely confounded with each other.

In order to analyze this confounding, you should examine the aliasing structure of the design using the ALIASING option in the MODEL statement. Before doing so, however, it is advisable to *code* the design, replacing low and high levels of each factor with the values -1 and +1, respectively. This puts each factor on an equal footing in the model and makes the aliasing structure much more interpretable. The following statements code the data, creating a new data set named Coded.

```
data Coded; set HalfFraction;
   power    = -1*(power   =0.80) + 1*(power   =1.20);
   flow     = -1*(flow    =4.50) + 1*(flow    =550 );
   pressure = -1*(pressure=125 ) + 1*(pressure=200 );
   gap      = -1*(gap     =275 ) + 1*(gap     =325 );
run;
```

The following statements use the GLM procedure to reanalyze the coded design, displaying the parameter estimates as well as the functions of the parameters that they each estimate.

```
proc glm data=Coded;
   model rate=power|flow|pressure|gap@2 / solution aliasing;
run;
```

The parameter estimates table is shown in Output 32.11.2.

Output 32.11.2. Parameter Estimates and Aliases for Nitride Etch Process Half Fraction

```
                                       The GLM Procedure

Dependent Variable: rate

                                       Standard
Parameter             Estimate            Error    t Value    Pr > |t|    Expected Value

Intercept          756.0000000                .       .        .       Intercept
power              145.2500000                .       .        .       power
flow                 5.7500000                .       .        .       flow
power*flow          -5.0000000 B              .       .        .       power*flow + pressure*gap
pressure             2.0000000                .       .        .       pressure
power*pressure     -12.7500000 B              .       .        .       power*pressure + flow*gap
flow*pressure      -98.7500000 B              .       .        .       flow*pressure + power*gap
gap                -63.5000000                .       .        .       gap
power*gap            0.0000000 B              .       .        .
flow*gap             0.0000000 B              .       .        .
pressure*gap         0.0000000 B              .       .        .

NOTE: The X'X matrix has been found to be singular, and a generalized inverse was used to solve
      the normal equations.  Terms whose estimates are followed by the letter 'B' are not
      uniquely estimable.
```

Looking at the "Expected Value" column, notice that, while each of the main effects is unambiguously estimated by its associated term in the model, the expected values of the interaction estimates are more complicated. For example, the relatively large effect (-98.75) corresponding to flow*pressure actually estimates the combined effect of flow*pressure and power*gap. Without further information, it is impossible to disentangle these aliased interactions; however, since the main effects of both power and gap are large and those for flow and pressure are small, it is reasonable to suspect that power*gap is the more "active" of the two interactions.

Fortunately, eight more runs are available for this experiment (the other half fraction.) The following statements create a data set containing these extra runs and add it to the previous eight, resulting in a full $2^4 = 16$ run replicate. Then PROC GLM displays the analysis of variance again.

```
data OtherHalf;
   input power flow pressure gap rate;
   datalines;
0.8   4.5 125 325     669
0.8   4.5 200 275     604
0.8 550.0 125 275     633
0.8 550.0 200 325     635
1.2   4.5 125 275    1037
1.2   4.5 200 325     868
1.2 550.0 125 325     860
1.2 550.0 200 275    1063
;
data FullRep;
   set HalfFraction OtherHalf;
run;

proc glm data=FullRep;
   class power flow pressure gap;
   model rate=power|flow|pressure|gap@2;
run;
```

The results are displayed in Output 32.11.3.

Output 32.11.3. Analysis of Variance for Nitride Etch Process Full Replicate

```
                                The GLM Procedure

                             Class Level Information

                         Class         Levels    Values

                         power              2    0.8 1.2

                         flow               2    4.5 550

                         pressure           2    125 200

                         gap                2    275 325


                     Number of Observations Read          16
                     Number of Observations Used          16



                                The GLM Procedure

Dependent Variable: rate

                                           Sum of
 Source                      DF          Squares    Mean Square   F Value   Pr > F

 Model                       10      521234.1250     52123.4125     25.58   0.0011

 Error                        5       10186.8125      2037.3625

 Corrected Total             15      531420.9375


               R-Square     Coeff Var      Root MSE     rate Mean

               0.980831      5.816175      45.13715      776.0625


 Source                      DF        Type I SS    Mean Square   F Value   Pr > F

 power                        1      374850.0625    374850.0625    183.99   <.0001
 flow                         1         217.5625       217.5625      0.11   0.7571
 power*flow                   1          18.0625        18.0625      0.01   0.9286
 pressure                     1          10.5625        10.5625      0.01   0.9454
 power*pressure               1           1.5625         1.5625      0.00   0.9790
 flow*pressure                1        7700.0625      7700.0625      3.78   0.1095
 gap                          1       41310.5625     41310.5625     20.28   0.0064
 power*gap                    1       94402.5625     94402.5625     46.34   0.0010
 flow*gap                     1        2475.0625      2475.0625      1.21   0.3206
 pressure*gap                 1         248.0625       248.0625      0.12   0.7414


 Source                      DF      Type III SS    Mean Square   F Value   Pr > F

 power                        1      374850.0625    374850.0625    183.99   <.0001
 flow                         1         217.5625       217.5625      0.11   0.7571
 power*flow                   1          18.0625        18.0625      0.01   0.9286
 pressure                     1          10.5625        10.5625      0.01   0.9454
 power*pressure               1           1.5625         1.5625      0.00   0.9790
 flow*pressure                1        7700.0625      7700.0625      3.78   0.1095
 gap                          1       41310.5625     41310.5625     20.28   0.0064
 power*gap                    1       94402.5625     94402.5625     46.34   0.0010
 flow*gap                     1        2475.0625      2475.0625      1.21   0.3206
 pressure*gap                 1         248.0625       248.0625      0.12   0.7414
```

With sixteen runs, the analysis of variance tells the whole story: all effects are estimable and there are five degrees of freedom left over to estimate the underlying error. The main effects of power and gap and their interaction are all significant, and no other effects are. Notice that the Type I and Type III ANOVA tables are the same; this is because the design is orthogonal and all effects are estimable.

This example illustrates the use of the GLM procedure for the model analysis of a screening experiment. Typically, there is much more involved in performing an experiment of this type, from selecting the design points to be studied to graphically assessing significant effects, optimizing the final model, and performing subsequent experimentation. Specialized tools for this are available in SAS/QC software, in particular the ADX Interface and the FACTEX and OPTEX procedures. Refer to *SAS/QC User's Guide* for more information.

References

Afifi, A. A. and Azen, S. P. (1972), *Statistical Analysis: A Computer-Oriented Approach*, New York: Academic Press, Inc.

Anderson, T. W. (1958), *An Introduction to Multivariate Statistical Analysis*, New York: John Wiley & Sons, Inc.

Bartlett, M. S. (1937), "Properties of Sufficiency and Statistical Tests," *Proceedings of the Royal Society of London, Series A*, 160, 268–282.

Begun, J. M. and Gabriel, K. R. (1981), "Closure of the Newman-Keuls Multiple Comparisons Procedure," *Journal of the American Statistical Association*, 76, 374.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons, Inc.

Bishop, Y., Feinberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Box, G. E. (1953), "Non-normality and Tests on Variance," *Biometrika*, 40, 318–335.

Box, G. E. (1954), "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, II. Effects of Inequality of Variance and of Correlation Between Errors in the Two-Way Classification," *Annals of Mathematical Statistics*, 25, 484–498.

Brown, M. B. and Forsythe, A. B. (1974), "Robust Tests for Equality of Variances," *Journal of the American Statistical Association*, 69, 364–367.

Carmer, S. G. and Swanson, M. R. (1973), "Evaluation of Ten Pairwise Multiple Comparison Procedures by Monte-Carlo Methods," *Journal of the American Statistical Association*, 68, 66–74.

Cochran, W. G. and Cox, G. M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons, Inc.

Cohen, R. (2002), “SAS Meets Big Iron: High Performance Computing in SAS Analytical Procedures,” in “Proceedings of the Twenty-seventh Annual SAS Users Group International Conference,” SAS.

Cole, J. W. and Grizzle, J. E. (1966), “Applications of Multivariate Analysis of Variance to Repeated Measures Experiments,” *Biometrics*, 22, 810–828.

Conover, W. J., Johnson, M. E., and Johnson, M. M. (1981), “A Comparative Study of Tests for Homogeneity of Variances, with Applications to the Outer Continental Shelf Bidding Data,” *Technometrics*, 23, 351–361.

Cornfield, J. and Tukey, J. W. (1956), “Average Values of Mean Squares in Factorials,” *Annals of Mathematical Statistics*, 27, 907–949.

Draper, N. R. and Smith, H. (1966), *Applied Regression Analysis*, New York: John Wiley & Sons, Inc.

Duncan, D. B. (1975), “t Tests and Intervals for Comparisons Suggested by the Data,” *Biometrics*, 31, 339–359.

Dunnett, C. W. (1955), “A Multiple Comparisons Procedure for Comparing Several Treatments with a Control,” *Journal of the American Statistical Association*, 50, 1096–1121.

Dunnett, C. W. (1980), “Pairwise Multiple Comparisons in the Homogeneous Variance, Unequal Sample Size Case,” *Journal of the American Statistical Association*, 75, 789–795.

Edwards, D. and Berry, J. J. (1987), “The Efficiency of Simulation-Based Multiple Comparisons,” *Biometrics*, 43, 913–928.

Einot, I. and Gabriel, K. R. (1975), “A Study of the Powers of Several Methods of Multiple Comparisons,” *Journal of the American Statistical Association*, 70, 351.

Freund, R. J., Littell, R. C., and Spector, P. C. (1986), *SAS System for Linear Models, 1986 Edition*, Cary, NC: SAS Institute Inc.

Gabriel, K. R. (1978), “A Simple Method of Multiple Comparisons of Means,” *Journal of the American Statistical Association*, 73, 364.

Games, P. A. (1977), “An Improved t Table for Simultaneous Control on g Contrasts,” *Journal of the American Statistical Association*, 72, 531–534.

Goodnight, J. H. (1976), “The New General Linear Models Procedure,” *Proceedings of the First International SAS Users’ Conference*.

Goodnight, J. H. (1978a), “The SWEEP Operator: Its Importance in Statistical Computing,” Technical Report R-106, SAS Institute Inc., Cary, NC.

Goodnight, J. H. (1978b), “Tests of the Hypotheses in Fixed-Effects Linear Models,” Technical Report R-101, SAS Institute Inc., Cary, NC.

Goodnight, J. H. (1979), “A Tutorial on the Sweep Operator,” *The American Statistician*, 33, 149–158.

Goodnight, J. H. and Harvey, W. R. (1978), “Least-Squares Means in the Fixed-Effects General Linear Models,” Technical Report R-103, SAS Institute Inc., Cary, NC.

Goodnight, J. H. and Speed, F. M. (1978), “Computing Expected Mean Squares,” Technical Report R-102, SAS Institute Inc., Cary, NC.

Graybill, F. A. (1961), *An Introduction to Linear Statistical Models, Volume I*, New York: McGraw-Hill Book Co.

Greenhouse, S. W. and Geisser, S. (1959), “On Methods in the Analysis of Profile Data,” *Psychometrika*, 32, 95–112.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and E., O. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall, 297–298.

Hartley, H. O. and Searle, S. R. (1969), “On Interaction Variance Components in Mixed Models,” *Biometrics*, 25, 573–576.

Harvey, W. R. (1975), *Least-squares Analysis of Data with Unequal Subclass Numbers*, USDA Report ARS H-4.

Hayter, A. J. (1984), “A Proof of the Conjecture that the Tukey-Kramer Method is Conservative,” *The Annals of Statistics*, 12, 61–75.

Hayter, A. J. (1989), “Pairwise Comparisons of Generally Correlated Means,” *Journal of the American Statistical Association*, 84, 208–213.

Heck, D. L. (1960), “Charts of Some Upper Percentage Points of the Distribution of the Largest Characteristic Root,” *Annals of Mathematical Statistics*, 31, 625–642.

Hochberg, Y. (1974), “Some Conservative Generalizations of the T-Method in Simultaneous Inference,” *Journal of Multivariate Analysis*, 4, 224–234.

Hocking, R. R. (1973), “A Discussion of the Two-Way Mixed Model,” *The American Statistician*, 27, 148–152.

Hocking, R. R. (1976), “The Analysis and Selection of Variables in a Linear Regression,” *Biometrics*, 32, 1–50.

Hocking, R. R. (1985), *The Analysis of Linear Models*, Belmont, CA: Brooks/Cole Publishing Co.

Hsu, J. C. (1992), “The Factor Analytic Approach to Simultaneous Inference in the General Linear Model,” *Journal of Computational and Graphical Statistics*, 1, 151–168.

Hsu, J. C. (1996), *Multiple Comparisons: Theory and Methods*, London: Chapman & Hall.

Hsu, J. C. and Nelson, B. (1998), “Multiple Comparisons in the General Linear Model,” *Journal of Computational and Graphical Statistics*, 7, 23–41.

Huynh, H. and Feldt, L. S. (1970), “Conditions under Which Mean Square Ratios in Repeated Measurements Designs Have Exact F-Distributions,” *Journal of the American Statistical Association*, 65, 1582–1589.

Huynh, H. and Feldt, L. S. (1976), “Estimation of the Box Correction for Degrees of Freedom from Sample Data in the Randomized Block and Split Plot Designs,” *Journal of Educational Statistics*, 1, 69–82.

Kennedy, W. J., Jr., and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker, Inc.

Kramer, C. Y. (1956), “Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications,” *Biometrics*, 12, 307–310.

Krishnaiah, P. R. and Armitage, J. V. (1966), “Tables for Multivariate t Distribution,” *Sankhya*, *Series B*, 31–56.

Kutner, M. H. (1974), “Hypothesis Testing in Linear Models (Eisenhart Model),” *American Statistician*, 28, 98–100.

LaTour, S. A. and Miniard, P. W. (1983), “The Misuse of Repeated Measures Analysis in Marketing Research,” *Journal of Marketing Research*, 45–57.

Levene, H. (1960), “Robust Tests for the Equality of Variance,” in I. Olkin, ed., “Contributions to Probability and Statistics,” 278–292, Palo Alto, CA: Stanford University Press.

Marcus, R., Peritz, E., and Gabriel, K. R. (1976), “On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance,” *Biometrika*, 63, 655–660.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), “A Unified Approach to Mixed Linear Models,” *The American Statistician*, 45, 54–64.

Miller, R. G. J. (1981), *Simultaneous Statistical Inference*, New York: Springer-Verlag.

Milliken, G. A. and Johnson, D. E. (1984), *Analysis of Messy Data, Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill Book Co.

Nelder, J. A. (1994), “The Statistics of Linear Models: Back to Basics,” *Statistics and Computing*, 4.

O’Brien, R. G. (1979), “A General ANOVA Method for Robust Tests of Additive Models for Variances,” *Journal of the American Statistical Association*, 74, 877–880.

O’Brien, R. G. (1981), “A Simple Test for Variance Effects in Experimental Designs,” *Psychological Bulletin*, 89, 570–574.

O’Brien, R. G. and Heft, M. W. (1995), “New Discrimination Indexes and Models for Studying Sensory Functioning in Aging,” *Journal of Applied Statistics*, 22, 9–27.

Olejnik, S. F. and Algina, J. (1987), “Type I Error Rates and Power Estimates of Selected Parametric and Non-parametric Tests of Scale,” *Journal of Educational Statistics*, 12, 45–61.

Petrinovich, L. F. and Hardyck, C. D. (1969), “Error Rates for Multiple Comparison Methods: Some Evidence Concerning the Frequency of Erroneous Conclusions,” *Psychological Bulletin*, 71, 43–54.

Pillai, K. C. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of Philippines.

Pringle, R. M. and Raynor, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Ramsey, P. H. (1978), “Power Differences Between Pairwise Multiple Comparisons,” *Journal of the American Statistical Association*, 73, 363.

Rao, C. R. (1965), *Linear Statistical Inference and Its Applications*, New York: John Wiley & Sons, Inc.

Rodriguez, R., Tobias, R., and Wolfinger, R. (1995), “Comments on J.A. Nelder ‘The Statistics of Linear Models: Back to Basics’,” *Statistics and Computing*, 5, 97–101.

Ryan, T. A. (1959), “Multiple Comparisons in Psychological Research,” *Psychological Bulletin*, 56, 26–47.

Ryan, T. A. (1960), “Significance Tests for Multiple Comparison of Proportions, Variances, and Other Statistics,” *Psychological Bulletin*, 57, 318–328.

Satterthwaite, F. E. (1946), “An Approximate Distribution of Estimates of Variance Components,” *Biometrics Bulletin*, 2, 110–114.

Schatzoff, M. (1966), “Exact Distributions of Wilks’ Likelihood Ratio Criterion,” *Biometrika*, 53, 347–358.

Scheffé, H. (1953), “A Method for Judging All Contrasts in the Analysis of Variance,” *Biometrika*, 40, 87–104.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons, Inc.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Searle, S. R. (1987), *Linear Models for Unbalanced Data*, New York: John Wiley & Sons, Inc.

Searle, S. R. (1995), “Comments on J.A. Nelder ‘The Statistics of Linear Models: Back to Basics’,” *Statistics and Computing*, 5, 103–107.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley and Sons, Inc.

Searle, S. R., Speed, F. M., and Milliken, G. A. (1980), “Population Marginal Means in the Linear Model: An Alternative to Least Squares Means,” *The American Statistician*, 34, 216–221.

Sidak, Z. (1967), “Rectangular Confidence Regions for the Means of Multivariate Normal Distributions,” *Journal of the American Statistical Association*, 62, 626–633.

Snedecor, G. W. and Cochran, W. G. (1967), *Statistical Methods*, Sixth Edition, Ames, IA: Iowa State University Press.

Steel, R. G. and Torrie, J. H. (1960), *Principles and Procedures of Statistics*, New York: McGraw-Hill Book Co.

Stenstrom, F. H. (1940), *The Growth of Snapdragons, Stocks, Cinerarias and Carnations on Six Iowa Soils*, Master’s thesis, Iowa State College.

Tubb, A., Parker, A. J., and Nickless, G. (1980), “The Analysis of Romano-British Pottery by Atomic Absorption Spectrophotometry,” *Archaeometry*, 22, 153–171.

Tukey, J. W. (1952), “Allowances for Various Types of Error Rates,” .

Tukey, J. W. (1953), “The Problem of Multiple Comparisons,” .

Waller, R. A. and Duncan, D. B. (1969), “A Bayes Rule for the Symmetric Multiple Comparison Problem,” *Journal of the American Statistical Association*, 64, 1484–1499.

Waller, R. A. and Duncan, D. B. (1972), “Corrigenda to ‘A Bayes Rule for the Symmetric Multiple Comparison Problem’,” *Journal of the American Statistical Association*, 67, 253–255.

Waller, R. A. and Kemp, K. E. (1976), “Computations of Bayesian t-Values for Multiple Comparisons,” *Journal of Statistical Computation and Simulation*, 75, 169–172.

Welch, B. L. (1951), “On the Comparison of Several Mean Values: An Alternative Approach,” *Biometrika*, 38, 330–336.

Welsch, R. E. (1977), “Stepwise Multiple Comparison Procedures,” *Journal of the American Statistical Association*, 72, 359.

Westfall, P. J. and Young, S. S. (1993), *Resampling-based Multiple Testing*, New York: John Wiley & Sons, Inc.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill, Inc.

Wolfinger, R. D. and Chang, M. (1995), “Comparing the SAS GLM and MIXED Procedures for Repeated Measures,” *Proceedings of the Twentieth Annual SAS Users Group Conference*.

Yin, G. Z. and Jillie, D. W. (1987), “Orthogonal Design for Process Optimization and Its Application in Plasma Etching,” *Solid State Technology*, May, 127–132.

Chapter 33
The GLMMOD Procedure

Chapter Contents

Chapter 33
The GLMMOD Procedure

Overview

The GLMMOD procedure constructs the design matrix for a general linear model; it essentially constitutes the model-building front end for the GLM procedure. You can use the GLMMOD procedure in conjunction with other SAS/STAT software regression procedures or with SAS/IML software to obtain specialized analyses for general linear models that you cannot obtain with the GLM procedure.

While some of the regression procedures in SAS/STAT software provide for general linear effects modeling with classification variables and interaction or polynomial effects, many others do not. For such procedures, you must specify the model directly in terms of distinct variables. For example, if you want to use the REG procedure to fit a polynomial model, you must first create the crossproduct and power terms as new variables, usually in a DATA step. Alternatively, you can use the GLMMOD procedure to create a data set that contains the design matrix for a model as specified using the effects modeling facilities of the GLM procedure.

Note that the TRANSREG procedure provides alternative methods to construct design matrices for full-rank and less-than-full-rank models, polynomials, and splines. See Chapter 75, "The TRANSREG Procedure," for more information.

Getting Started

A One-Way Design

A one-way analysis of variance considers one treatment factor with two or more treatment levels. This example employs PROC GLMMOD together with PROC REG to perform a one-way analysis of variance to study the effect of bacteria on the nitrogen content of red clover plants. The treatment factor is bacteria strain, and it has six levels. Red clover plants are inoculated with the treatments, and nitrogen content is later measured in milligrams. The data are derived from an experiment by Erdman (1946) and are analyzed in Chapters 7 and 8 of Steel and Torrie (1980). PROC GLMMOD is used to create the design matrix. The following DATA step creates the SAS data set Clover.

```
title 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;
```

The variable Strain contains the treatment levels, and the variable Nitrogen contains the response. The following statements produce the design matrix:

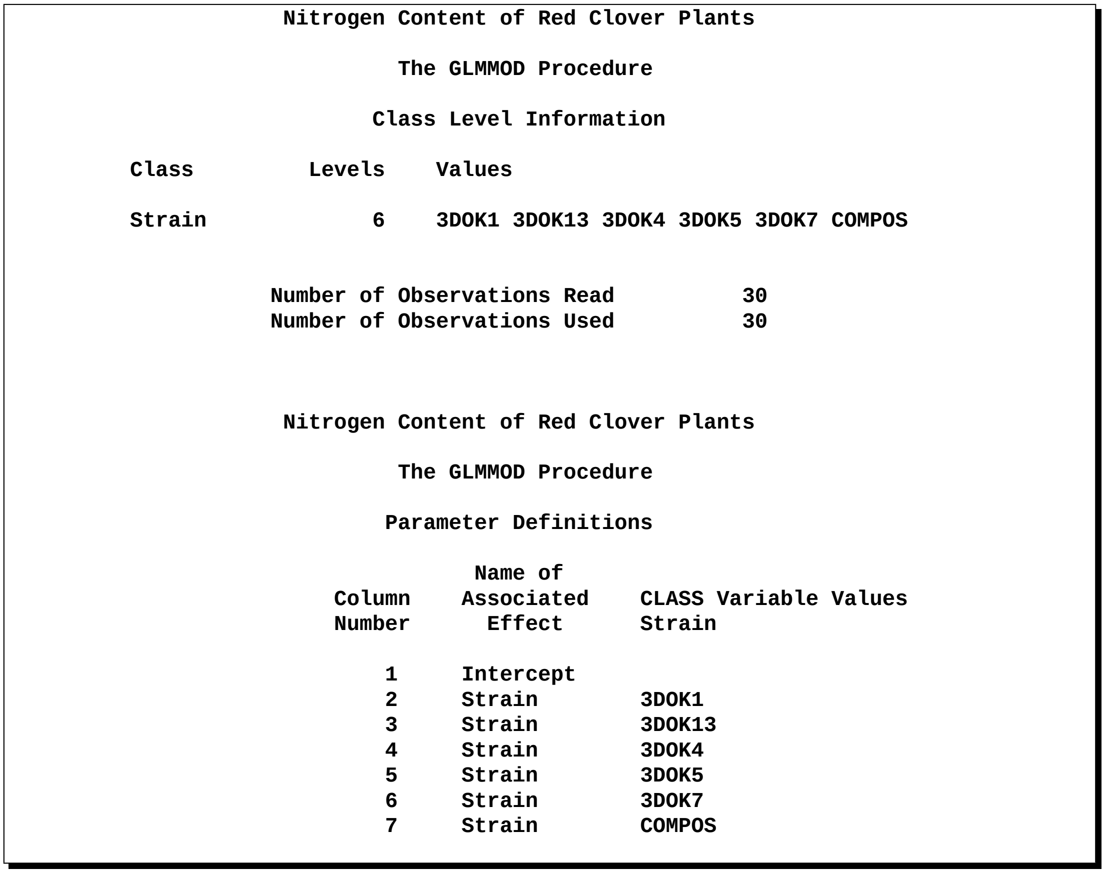
```
proc glmmod data=Clover;
   class Strain;
   model Nitrogen = Strain;
run;
```

The classification variable, or treatment factor, is specified in the CLASS statement. The MODEL statement defines the response and independent variables. The design matrix produced corresponds to the model

$$Y_{i,j} = \mu + \alpha_i + \epsilon_{i,j}$$

where $i = 1, \ldots, 6$, and $j = 1, \ldots, 5$.

Figure 33.1 and Figure 33.2 display the output produced by these statements. Figure 33.1 displays information about the data set, which is useful for checking your data.



```
                      Nitrogen Content of Red Clover Plants

                               The GLMMOD Procedure

                             Class Level Information

          Class         Levels    Values

          Strain             6    3DOK1 3DOK13 3DOK4 3DOK5 3DOK7 COMPOS


                     Number of Observations Read          30
                     Number of Observations Used          30



                      Nitrogen Content of Red Clover Plants

                               The GLMMOD Procedure

                              Parameter Definitions

                                     Name of
                          Column    Associated    CLASS Variable Values
                          Number      Effect      Strain

                              1     Intercept
                              2     Strain        3DOK1
                              3     Strain        3DOK13
                              4     Strain        3DOK4
                              5     Strain        3DOK5
                              6     Strain        3DOK7
                              7     Strain        COMPOS
```

Figure 33.1. Class Level Information and Parameter Definitions

The design matrix, shown in Figure 33.2, consists of seven columns: one for the mean and six for the treatment levels. The vector of responses, Nitrogen, is also displayed.

```
                      Nitrogen Content of Red Clover Plants

                              The GLMMOD Procedure

                                 Design Points

      Observation                             Column Number
        Number       Nitrogen    1    2    3    4    5    6    7

             1         19.4      1    1    0    0    0    0    0
             2         32.6      1    1    0    0    0    0    0
             3         27.0      1    1    0    0    0    0    0
             4         32.1      1    1    0    0    0    0    0
             5         33.0      1    1    0    0    0    0    0
             6         17.7      1    0    0    0    1    0    0
             7         24.8      1    0    0    0    1    0    0
             8         27.9      1    0    0    0    1    0    0
             9         25.2      1    0    0    0    1    0    0
            10         24.3      1    0    0    0    1    0    0
            11         17.0      1    0    0    1    0    0    0
            12         19.4      1    0    0    1    0    0    0
            13          9.1      1    0    0    1    0    0    0
            14         11.9      1    0    0    1    0    0    0
            15         15.8      1    0    0    1    0    0    0
            16         20.7      1    0    0    0    0    1    0
            17         21.0      1    0    0    0    0    1    0
            18         20.5      1    0    0    0    0    1    0
            19         18.8      1    0    0    0    0    1    0
            20         18.6      1    0    0    0    0    1    0
            21         14.3      1    0    1    0    0    0    0
            22         14.4      1    0    1    0    0    0    0
            23         11.8      1    0    1    0    0    0    0
            24         11.6      1    0    1    0    0    0    0
            25         14.2      1    0    1    0    0    0    0
            26         17.3      1    0    0    0    0    0    1
            27         19.4      1    0    0    0    0    0    1
            28         19.1      1    0    0    0    0    0    1
            29         16.9      1    0    0    0    0    0    1
            30         20.8      1    0    0    0    0    0    1
```

Figure 33.2. Design Matrix

Usually, you will find PROC GLMMOD most useful for the data sets it can create rather than for its displayed output. For example, the following statements use PROC GLMMOD to save the design matrix for the clover study to the data set CloverDesign instead of displaying it.

```
proc glmmod data=Clover outdesign=CloverDesign noprint;
   class Strain;
   model Nitrogen = Strain;
run;
```

Now you can use the REG procedure to analyze the data, as the following statements demonstrate:

```
proc reg data=CloverDesign;
   model Nitrogen = Col2-Col7;
run;
```

The results are shown in Figure 33.3.

```
                           Nitrogen Content of Red Clover Plants

                                     The REG Procedure
                                      Model: MODEL1
                               Dependent Variable: Nitrogen

                                   Analysis of Variance

                                            Sum of           Mean
 Source                     DF             Squares         Square    F Value    Pr > F

 Model                       5           847.04667      169.40933      14.37    <.0001
 Error                      24           282.92800       11.78867
 Corrected Total            29          1129.97467


              Root MSE                 3.43346    R-Square     0.7496
              Dependent Mean          19.88667    Adj R-Sq     0.6975
              Coeff Var               17.26515

NOTE: Model is not full rank. Least-squares solutions for the parameters are
      not unique. Some statistics will be misleading. A reported DF of 0 or B
      means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a
      linear combination of other variables as shown.


              Col7 =  Intercept - Col2 - Col3 - Col4 - Col5 - Col6


                                   Parameter Estimates

                                       Parameter       Standard
Variable     Label             DF       Estimate          Error    t Value    Pr > |t|

Intercept    Intercept          B       18.70000        1.53549      12.18      <.0001
Col2         Strain 3DOK1       B       10.12000        2.17151       4.66      <.0001
Col3         Strain 3DOK13      B       -5.44000        2.17151      -2.51      0.0194
Col4         Strain 3DOK4       B       -4.06000        2.17151      -1.87      0.0738
Col5         Strain 3DOK5       B        5.28000        2.17151       2.43      0.0229
Col6         Strain 3DOK7       B        1.22000        2.17151       0.56      0.5794
Col7         Strain COMPOS      0              0              .          .           .
```

Figure 33.3. Regression Analysis

Syntax

The following statements are available in PROC GLMMOD.

PROC GLMMOD < *options* > **;**
  BY *variables* **;**
  CLASS *variables* **;**
  FREQ *variable* **;**
  MODEL *dependents=independents* / < *options* > **;**
  WEIGHT *variable* **;**

The PROC GLMMOD and MODEL statements are required. If classification effects are used, the class variables must be declared in a CLASS statement, and the CLASS statement must appear before the MODEL statement.

PROC GLMMOD Statement

PROC GLMMOD $<$ *options* $>$ **;**

The PROC GLMMOD statement invokes the GLMMOD procedure. It has the following options:

DATA=*SAS-data-set*

specifies the SAS data set to be used by the GLMMOD procedure. If you do not specify the DATA= option, the most recently created SAS data set is used.

NAMELEN=*n*

specifies the maximum length for an effect name. Effect names are listed in the table of parameter definitions and stored in the EFFNAME variable in the OUTPARM= data set. By default, $n = 20$. You can specify $20 < n \leq 200$ if 20 characters are not enough to distinguish between effects, which may be the case if the model includes a high-order interaction between variables with relatively long, similar names.

NOPRINT

suppresses the normal display of results. This option is generally useful only when one or more output data sets are being produced by the GLMMOD procedure. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which you want the levels of the classification variables (specified in the CLASS statement) to be sorted. This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal (numeric) value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The ORDER= option can take the following values.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

If you omit the ORDER= option, PROC GLMMOD orders by the external formatted value.

OUTPARM=*SAS-data-set*

names an output data set to contain the information regarding the association between model effects and design matrix columns.

OUTDESIGN=*SAS-data-set*

names an output data set to contain the columns of the design matrix.

PREFIX=*name*

specifies a prefix to use in naming the columns of the design matrix in the OUTDESIGN= data set. The default prefix is Col and the column name is formed by appending the column number to the prefix, so that by default the columns are named Col1, Col2, and so on. If you specify the ZEROBASED option, the column numbering starts at zero, so that with the default value of PREFIX= the columns of the design matrix in the OUTDESIGN= data set are named Col0, Col1, and so on.

ZEROBASED

specifies that the numbering for the columns of the design matrix in the OUTDESIGN= data set should begin at 0. By default it begins at 1, so that with the default value of PREFIX= the columns of the design matrix in the OUTDESIGN= data set are named Col1, Col2, and so on. If you use the ZEROBASED option, the column names are instead Col0, Col1, and so on.

BY Statement

BY *variables* **;**

You can specify a BY statement with the GLMMOD procedure to obtain separate designs for observations in groups defined by the BY variables. When you specify a BY statement, the procedure expects the input DATA= data set to be sorted in the order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLMMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < *option* > **;**

The CLASS statement names the classification variables to be used in the analysis. Typical classification variables are Treatment, Sex, Race, Group, and Replication. If you specify the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

FREQ and WEIGHT Statements

FREQ *variable* **;**

WEIGHT *variable* **;**

FREQ and WEIGHT variables are transferred to the output data sets without change.

MODEL Statement

MODEL *dependents=independents* / < *options* > **;**

The MODEL statement names the dependent variables and independent effects. For the syntax of effects, see the "Specification of Effects" section on page 1784 in Chapter 32, "The GLM Procedure."

You can specify the following option in the MODEL statement after a slash (/).

NOINT
requests that the intercept parameter not be included in the model.

Details

Displayed Output

For each pass of the data (that is, for each BY group and for each pass required by the pattern of missing values for the dependent variables), the GLMMOD procedure displays the definitions of the columns of the design matrix along with the following:

- the number of the column
- the name of the associated effect
- the values that the class variables take for this level of the effect

The design matrix itself is also displayed, along with the following:

- the observation number
- the dependent variable values
- the FREQ and WEIGHT values, if any
- the columns of the design matrix

Missing Values

If some variables have missing values for some observations, then PROC GLMMOD handles missing values in the same way as PROC GLM; see the "Missing Values" section on page 1837 in Chapter 32, "The GLM Procedure," for further details.

OUTPARM= Data Set

An output data set containing information regarding the association between model effects and design matrix columns is created whenever you specify the OUTPARM= option in the PROC GLMMOD statement. The OUTPARM= data set contains an observation for each column of the design matrix with the following variables:

- a numeric variable, _COLNUM_, identifying the number of the column of the design matrix corresponding to this observation

- a character variable, EFFNAME, containing the name of the effect that generates the column of the design matrix corresponding to this observation
- the CLASS variables, with the values they have for the column corresponding to this observation, or blanks if they are not involved with the effect associated with this column

If there are BY-group variables or if the pattern of missing values for the dependent variables requires it, the single data set defines several design matrices. In this case, for each of these design matrices, the OUTPARM= data set also contains the following:

- the current values of the BY variables, if you specify a BY statement
- a numeric variable, _YPASS_, containing the current pass of the data, if the pattern of missing values for the dependent variables requires multiple passes

OUTDESIGN= Data Set

An output data set containing the design matrix is created whenever you specify the OUTDESIGN= option in the PROC GLMMOD statement. The OUTDESIGN= data set contains an observation for each observation in the DATA= data set, with the following variables:

- the dependent variables
- the FREQ variable, if any
- the WEIGHT variable, if any
- a variable for each column of the design matrix, with names COL1, COL2, and so forth

If there are BY-group variables or if the pattern of missing values for the dependent variables requires it, the single data set contains several design matrices. In this case, for each of these, the OUTDESIGN= data set also contains the following:

- the current values of the BY variables, if you specify a BY statement
- a numeric variable, _YPASS_, containing the current pass of the data, if the pattern of missing values for the dependent variables requires multiple passes

ODS Table Names

PROC GLMMOD assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 33.1. ODS Tables Produced in PROC GLMMOD

ODS Table Name	Description	Statement
ClassLevels	Table of class levels	CLASS statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables
DesignPoints	Design matrix	default
NObs	Number of observations	default
Parameters	Parameters and associated column numbers	default

Examples

Example 33.1. A Two-Way Design

The following program uses the GLMMOD procedure to produce the design matrix for a two-way design. The two classification factors have seven and three levels, respectively, so the design matrix contains $1 + 7 + 3 + 21 = 32$ columns in all.

```
data Plants;
   input Type $ @;
   do Block=1 to 3;
      input StemLength @;
      output;
      end;
   datalines;
Clarion  32.7 32.3 31.5
Clinton  32.1 29.7 29.1
Knox     35.7 35.9 33.1
O'Neill  36.0 34.2 31.2
Compost  31.8 28.0 29.2
Wabash   38.2 37.8 31.9
Webster  32.5 31.1 29.7
;
proc glmmod outparm=Parm outdesign=Design;
   class Type Block;
   model StemLength = Type|Block;
run;

proc print data=Parm;
run;

proc print data=Design;
run;
```

Output 33.1.1. A Two-Way Design

```
                            The GLMMOD Procedure

                          Class Level Information

Class         Levels    Values

Type               7    Clarion Clinton Compost Knox O’Neill Wabash Webster

Block              3    1 2 3


                  Number of Observations Read          21
                  Number of Observations Used          21
```

```
                            The GLMMOD Procedure

                           Parameter Definitions

                             Name of
                  Column    Associated    CLASS Variable Values
                  Number      Effect       Type      Block

                      1     Intercept
                      2     Type          Clarion
                      3     Type          Clinton
                      4     Type          Compost
                      5     Type          Knox
                      6     Type          O’Neill
                      7     Type          Wabash
                      8     Type          Webster
                      9     Block                      1
                     10     Block                      2
                     11     Block                      3
                     12     Type*Block    Clarion      1
                     13     Type*Block    Clarion      2
                     14     Type*Block    Clarion      3
                     15     Type*Block    Clinton      1
                     16     Type*Block    Clinton      2
                     17     Type*Block    Clinton      3
                     18     Type*Block    Compost      1
                     19     Type*Block    Compost      2
                     20     Type*Block    Compost      3
                     21     Type*Block    Knox         1
                     22     Type*Block    Knox         2
                     23     Type*Block    Knox         3
                     24     Type*Block    O’Neill      1
                     25     Type*Block    O’Neill      2
                     26     Type*Block    O’Neill      3
                     27     Type*Block    Wabash       1
                     28     Type*Block    Wabash       2
                     29     Type*Block    Wabash       3
                     30     Type*Block    Webster      1
                     31     Type*Block    Webster      2
                     32     Type*Block    Webster      3
```

```
                              The GLMMOD Procedure

                                 Design Points

Observation   Stem                       Column Number
  Number     Length  1  2  3  4  5  6  7  8  9  10   11   12   13   14   15   16   17

       1      32.7   1  1  0  0  0  0  0  0  1  0    0    1    0    0    0    0    0
       2      32.3   1  1  0  0  0  0  0  0  0  1    0    0    1    0    0    0    0
       3      31.5   1  1  0  0  0  0  0  0  0  0    1    0    0    1    0    0    0
       4      32.1   1  0  1  0  0  0  0  0  1  0    0    0    0    0    1    0    0
       5      29.7   1  0  1  0  0  0  0  0  0  1    0    0    0    0    0    1    0
       6      29.1   1  0  1  0  0  0  0  0  0  0    1    0    0    0    0    0    1
       7      35.7   1  0  0  0  1  0  0  0  1  0    0    0    0    0    0    0    0
       8      35.9   1  0  0  0  1  0  0  0  0  1    0    0    0    0    0    0    0
       9      33.1   1  0  0  0  1  0  0  0  0  0    1    0    0    0    0    0    0
      10      36.0   1  0  0  0  0  1  0  0  1  0    0    0    0    0    0    0    0
      11      34.2   1  0  0  0  0  1  0  0  0  1    0    0    0    0    0    0    0
      12      31.2   1  0  0  0  0  1  0  0  0  0    1    0    0    0    0    0    0
      13      31.8   1  0  0  1  0  0  0  0  1  0    0    0    0    0    0    0    0
      14      28.0   1  0  0  1  0  0  0  0  0  1    0    0    0    0    0    0    0
      15      29.2   1  0  0  1  0  0  0  0  0  0    1    0    0    0    0    0    0
      16      38.2   1  0  0  0  0  0  1  0  1  0    0    0    0    0    0    0    0
      17      37.8   1  0  0  0  0  0  1  0  0  1    0    0    0    0    0    0    0
      18      31.9   1  0  0  0  0  0  1  0  0  0    1    0    0    0    0    0    0
      19      32.5   1  0  0  0  0  0  0  1  1  0    0    0    0    0    0    0    0
      20      31.1   1  0  0  0  0  0  0  1  0  1    0    0    0    0    0    0    0
      21      29.7   1  0  0  0  0  0  0  1  0  0    1    0    0    0    0    0    0
```

```
                                 Design Points

   Observation                         Column Number
     Number     18   19   20   21   22   23   24   25   26   27   28   29   30   31   32

          1     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          2     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          3     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          4     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          5     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          6     0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
          7     0    0    0    1    0    0    0    0    0    0    0    0    0    0    0
          8     0    0    0    0    1    0    0    0    0    0    0    0    0    0    0
          9     0    0    0    0    0    1    0    0    0    0    0    0    0    0    0
         10     0    0    0    0    0    0    1    0    0    0    0    0    0    0    0
         11     0    0    0    0    0    0    0    1    0    0    0    0    0    0    0
         12     0    0    0    0    0    0    0    0    1    0    0    0    0    0    0
         13     1    0    0    0    0    0    0    0    0    0    0    0    0    0    0
         14     0    1    0    0    0    0    0    0    0    0    0    0    0    0    0
         15     0    0    1    0    0    0    0    0    0    0    0    0    0    0    0
         16     0    0    0    0    0    0    0    0    0    1    0    0    0    0    0
         17     0    0    0    0    0    0    0    0    0    0    1    0    0    0    0
         18     0    0    0    0    0    0    0    0    0    0    0    1    0    0    0
         19     0    0    0    0    0    0    0    0    0    0    0    0    1    0    0
         20     0    0    0    0    0    0    0    0    0    0    0    0    0    1    0
         21     0    0    0    0    0    0    0    0    0    0    0    0    0    0    1
```

Output 33.1.2. The OUTPARM= Data Set

```
Obs    _COLNUM_    EFFNAME       Type       Block

  1        1       Intercept
  2        2       Type          Clarion
  3        3       Type          Clinton
  4        4       Type          Compost
  5        5       Type          Knox
  6        6       Type          O’Neill
  7        7       Type          Wabash
  8        8       Type          Webster
  9        9       Block                      1
 10       10       Block                      2
 11       11       Block                      3
 12       12       Type*Block    Clarion      1
 13       13       Type*Block    Clarion      2
 14       14       Type*Block    Clarion      3
 15       15       Type*Block    Clinton      1
 16       16       Type*Block    Clinton      2
 17       17       Type*Block    Clinton      3
 18       18       Type*Block    Compost      1
 19       19       Type*Block    Compost      2
 20       20       Type*Block    Compost      3
 21       21       Type*Block    Knox         1
 22       22       Type*Block    Knox         2
 23       23       Type*Block    Knox         3
 24       24       Type*Block    O’Neill      1
 25       25       Type*Block    O’Neill      2
 26       26       Type*Block    O’Neill      3
 27       27       Type*Block    Wabash       1
 28       28       Type*Block    Wabash       2
 29       29       Type*Block    Wabash       3
 30       30       Type*Block    Webster      1
 31       31       Type*Block    Webster      2
 32       32       Type*Block    Webster      3
```

Output 33.1.3. The OUTDESIGN= Data Set

```
     S
     t
     e
     m
     L
     e                    C C C C C C C C C C C C C C C C C C C C C C C
     n  C C C C C C C C C o o o o o o o o o o o o o o o o o o o o o o o
 O   g  o o o o o o o o o l l l l l l l l l l l l l l l l l l l l l l l
 b   t  l l l l l l l l l 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3
 s   h  1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2

 1 32.7 1 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 2 32.3 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 3 31.5 1 1 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 4 32.1 1 0 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 5 29.7 1 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 6 29.1 1 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 7 35.7 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 8 35.9 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 9 33.1 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
10 36.0 1 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
11 34.2 1 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
12 31.2 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
13 31.8 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
14 28.0 1 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
15 29.2 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
16 38.2 1 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
17 37.8 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
18 31.9 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
19 32.5 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
20 31.1 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
21 29.7 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

Example 33.2. Factorial Screening

Screening experiments are undertaken to select from among the many possible factors that might affect a response the few that actually do, either simply (main effects) or in conjunction with other factors (interactions). One method of selecting significant factors is forward model selection, in which the model is built by successively adding the most statistically significant effects. Forward selection is an option in the REG procedure, but the REG procedure does not allow you to specify interactions directly (as the GLM procedure does, for example). You can use the GLMMOD procedure to create the screening model for a design and then use the REG procedure on the results to perform the screening.

The following statements create the SAS data set Screening, which contains the results of a screening experiment:

```
title 'PROC GLMMOD and PROC REG for Forward Selection Screening';
data Screening;
   input a b c d e y;
   datalines;
-1 -1 -1 -1  1  -6.688
-1 -1 -1  1 -1 -10.664
-1 -1  1 -1 -1  -1.459
-1 -1  1  1  1   2.042
-1  1 -1 -1 -1  -8.561
-1  1 -1  1  1  -7.095
-1  1  1 -1  1   0.553
-1  1  1  1 -1  -2.352
 1 -1 -1 -1 -1  -4.802
 1 -1 -1  1  1   5.705
 1 -1  1 -1  1  14.639
 1 -1  1  1 -1   2.151
 1  1 -1 -1  1   5.884
 1  1 -1  1 -1  -3.317
 1  1  1 -1 -1   4.048
 1  1  1  1  1  15.248
;
run;
```

The data set contains a single dependent variable (y) and five independent factors (a, b, c, d, and e). The design is a half-fraction of the full 2^5 factorial, the precise half-fraction having been chosen to provide uncorrelated estimates of all main effects and two-factor interactions.

The following statements use the GLMMOD procedure to create a design matrix data set containing all the main effects and two factor interactions for the preceding screening design.

```
ods output DesignPoints = DesignMatrix;
proc glmmod data=Screening;
   model y = a|b|c|d|e@2;
run;
```

Notice that the preceding statements use ODS to create the design matrix data set, instead of the OUTDESIGN= option in the PROC GLMMOD statement. The results are equivalent, but the columns of the data set produced by ODS have names that are directly related to the names of their corresponding effects.

Finally, the following statements use the REG procedure to perform forward model selection for the screening design. Two MODEL statements are used, one without the selection options (which produces the regression analysis for the full model) and one with the selection options.

```
proc reg data=DesignMatrix;
   model y = a--d_e;
   model y = a--d_e / selection = forward
                      details   = summary
                      slentry   = 0.05;
run;
```

Output 33.2.1. PROC REG Full Model Fit

```
           PROC GLMMOD and PROC REG for Forward Selection Screening

                              The REG Procedure
                                Model: MODEL1
                            Dependent Variable: y

                             Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                    15      861.48436       57.43229          .       .
Error                     0              0              .
Corrected Total          15      861.48436


             Root MSE                    .    R-Square     1.0000
             Dependent Mean        0.33325    Adj R-Sq          .
             Coeff Var                   .


                             Parameter Estimates

                                Parameter      Standard
 Variable    Label       DF      Estimate         Error    t Value   Pr > |t|

 Intercept   Intercept    1       0.33325             .          .          .
 a                        1       4.61125             .          .          .
 b                        1       0.21775             .          .          .
 a_b         a*b          1       0.30350             .          .          .
 c                        1       4.02550             .          .          .
 a_c         a*c          1       0.05150             .          .          .
 b_c         b*c          1      -0.20225             .          .          .
 d                        1      -0.11850             .          .          .
 a_d         a*d          1       0.12075             .          .          .
 b_d         b*d          1       0.18850             .          .          .
 c_d         c*d          1       0.03200             .          .          .
 e                        1       3.45275             .          .          .
 a_e         a*e          1       1.97175             .          .          .
 b_e         b*e          1      -0.35625             .          .          .
 c_e         c*e          1       0.30900             .          .          .
 d_e         d*e          1       0.30750             .          .          .
```

Output 33.2.2. PROC REG Screening Results

```
             PROC GLMMOD and PROC REG for Forward Selection Screening

                                  The REG Procedure
                                    Model: MODEL2
                                Dependent Variable: y


                             Summary of Forward Selection

      Variable                 Number  Partial   Model
 Step Entered   Label          Vars In R-Square R-Square  C(p)   F Value Pr > F

   1  a                            1    0.3949   0.3949    .        9.14 0.0091
   2  c                            2    0.3010   0.6959    .       12.87 0.0033
   3  e                            3    0.2214   0.9173    .       32.13 0.0001
   4  a_e       a*e                4    0.0722   0.9895    .       75.66 <.0001
```

Output 33.2.1 and Output 33.2.2 contain the results of the REG analysis. The full model has 16 parameters (the intercept + 5 main effects + 10 interactions). These are all estimable, but since there are only 16 observations in the design, there are no degrees of freedom left to estimate error; consequently, there is no way to use the full model to test for the statistical significance of effects. However, the forward selection method chooses only four effects for the model: the main effects of factors a, c, and e, and the interaction between a and e. Using this reduced model enables you to estimate the underlying level of noise, although note that the selection method biases this estimate somewhat.

References

Erdman, L. W. (1946), “Studies to Determine if Antibiosis Occurs among Rhizobia,” *Journal of the American Society of Agronomy*, 38, 251–258.

Steel, R. G. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill Book Company.

Chapter 34
The GLMPOWER Procedure

Chapter Contents

Chapter 34
The GLMPOWER Procedure

Overview

Power and sample size analysis optimizes the resource usage and design of a study, improving chances of conclusive results with maximum efficiency. The GLMPOWER procedure performs prospective power analysis for linear models, with a variety of goals:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

Here *prospective* indicates that the analysis pertains to planning for a future study. This is in contrast to *retrospective* analysis for a past study, which is not supported by this procedure.

The statistical analyses that are covered include Type III tests and contrasts of fixed class effects in univariate linear models, optionally with covariates. The covariates can be continuous or categorical. Tests and contrasts involving random effects are not supported. For power and sample size analyses in a variety of other statistical situations, see Chapter 57, “The POWER Procedure.”

Input for PROC GLMPOWER includes the components considered in study planning:

- design (including subject profiles and their allocation weights)
- statistical model
- contrasts of class effects
- significance level (alpha)
- surmised response means for subject profiles (often called “cell means”)
- surmised variability
- power
- sample size

In order to identify power or sample size as the result parameter, you designate it by a missing value in the input. The procedure calculates this result value over one or more scenarios of input values for all other components.

You specify the design and the cell means using an *exemplary data set*, a data set of artificial values constructed to represent the intended sampling design and the surmised response means in the underlying population. You specify the model and contrasts using MODEL and CONTRAST statements similar to those in the GLM, ANOVA, and MIXED procedures. You specify the remaining parameters with the POWER statement, which is similar to analysis statements in the POWER procedure.

In addition to tabular results, PROC GLMPOWER produces graphs. You can produce the most common types of plots easily with default settings and use a variety of options for more customized graphics. For example, you can control the choice of axis variables, axis ranges, number of plotted points, mapping of graphical features (such as color, line style, symbol, and panel) to analysis parameters, and legend appearance.

The GLMPOWER procedure is one of several tools available in SAS/STAT software for power and sample size analysis. PROC POWER covers a variety of more basic analyses such as *t* tests, equivalence tests, confidence intervals, binomial proportions, multiple regression, and one-way ANOVA. The Power and Sample Size application provides a user interface and implements many of the analyses supported in the procedures.

The following sections of this chapter describe how to use PROC GLMPOWER and discuss the underlying statistical methodology. The "Getting Started" section on page 1930 introduces PROC GLMPOWER with examples of power computation for a two-way analysis of variance. The "Syntax" section on page 1935 describes the syntax of the procedure. The "Details" section on page 1945 summarizes the methods employed by PROC GLMPOWER and provides details on several special topics. The "Examples" section on page 1951 illustrates the use of the GLMPOWER procedure with several applications.

For more discussion and examples on power and sample size analysis for linear models, refer to Castelloe and O'Brien (2001), O'Brien and Shieh (1992), Muller et al. (1992), and O'Brien and Muller (1993). For additional discussion on general power and sample size concepts, refer to Castelloe (2000), Muller and Benignus (1992), and Lenth (2001).

Getting Started

Simple Two-Way ANOVA

This example demonstrates how to use PROC GLMPOWER to compute and plot power for each effect test in a two-way analysis of variance (ANOVA).

Suppose you are planning an experiment to study the effect of light exposure on the growth of two varieties of flowers. The planned data analysis is a two-way ANOVA with flower height (measured at two weeks) as the response and a model consisting of the effects of light exposure, flower variety, and their interaction. You want to calculate the power of each effect test using a balanced design with a total of 60 specimens (10 for each combination of exposure and variety) and $\alpha = 0.05$ for each test.

As a first step, create an *exemplary data set* describing your conjectures about the underlying population means. You believe that the mean flower height for each combination of variety and exposure level (i.e., for each design profile, or for each *cell* in the design) roughly follows Table 34.1.

Table 34.1. Mean Flower Height (in cm) by Variety and Exposure

	Exposure		
Variety	1	2	3
1	14	16	21
2	10	15	16

The following statements create a data set Exemplary containing these cell means.

```
data Exemplary;
   do Variety = 1 to 2;
      do Exposure = 1 to 3;
         input Height @@;
         output;
      end;
   end;
   datalines;
      14 16 21
      10 15 16
   ;
run;
```

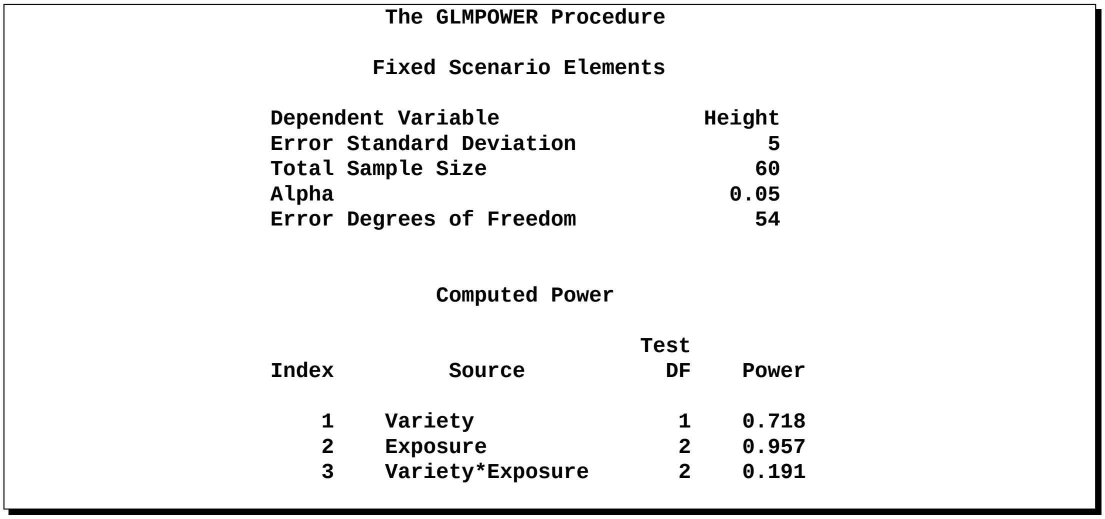
You also conjecture that the error standard deviation is about 5 cm.

Use the DATA= option in the PROC GLMPOWER statement to specify Exemplary as the exemplary data set. Identify the class variables (Variety and Exposure) using the CLASS statement. Specify the model using the MODEL statement. Use the POWER statement to specify power as the result parameter and provide values for the other analysis parameters, error standard deviation and total sample size.

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model Height = Variety | Exposure;
   power
      stddev = 5
      ntotal = 60
      power  = .;
run;
```

The MODEL statement defines the full model including both main effects and the interaction. The POWER= option in the POWER statement identifies power as the result parameter with a missing value (POWER=.). The STDDEV= option specifies an error standard deviation of 5, and the NTOTAL= option specifies a total sample size of 60. The default value for the ALPHA= option sets the significance level to $\alpha = 0.05$.

Figure 34.1 shows the output.



```
                         The GLMPOWER Procedure

                        Fixed Scenario Elements

              Dependent Variable                Height
              Error Standard Deviation               5
              Total Sample Size                     60
              Alpha                               0.05
              Error Degrees of Freedom              54


                             Computed Power

                                           Test
              Index          Source          DF     Power

                  1    Variety                1    0.718
                  2    Exposure               2    0.957
                  3    Variety*Exposure       2    0.191
```

Figure 34.1. Sample Size Analysis for Two-Way ANOVA

The power is about 0.72 for the test of the Variety effect. In other words, there is a probability of 0.72 that the test of the Variety effect will produce a significant result (given the assumptions for the means and error standard deviation). The power is 0.96 for the test of the Exposure effect and 0.19 for the interaction test.

Now, suppose you want to account for some of your uncertainty in conjecturing the true error standard deviation by evaluating the power at reasonable low and high values, 4 and 6.5. You also want to plot power for sample sizes between 30 and 90. The following statements perform the analysis:

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model Height = Variety | Exposure;
   power
      stddev = 4 6.5
      ntotal = 60
      power  = .;
   plot x=n min=30 max=90;
run;
```

The PLOT statement with the X=N option requests a plot with sample size on the x-axis. (The result parameter, here power, is always plotted on the other axis.) The MIN= and MAX= options in the PLOT statement specify the sample size range.

Figure 34.2 shows the output, and Figure 34.3 shows the plot.

```
                     The GLMPOWER Procedure

                    Fixed Scenario Elements

          Dependent Variable                Height
          Total Sample Size                     60
          Alpha                               0.05
          Error Degrees of Freedom              54


                         Computed Power

                                      Std    Test
      Index         Source            Dev      DF    Power

          1    Variety                4.0       1    0.887
          2    Variety                6.5       1    0.496
          3    Exposure               4.0       2    0.996
          4    Exposure               6.5       2    0.793
          5    Variety*Exposure       4.0       2    0.280
          6    Variety*Exposure       6.5       2    0.130
```

Figure 34.2. Sample Size Analysis for Two-Way ANOVA with Input Ranges

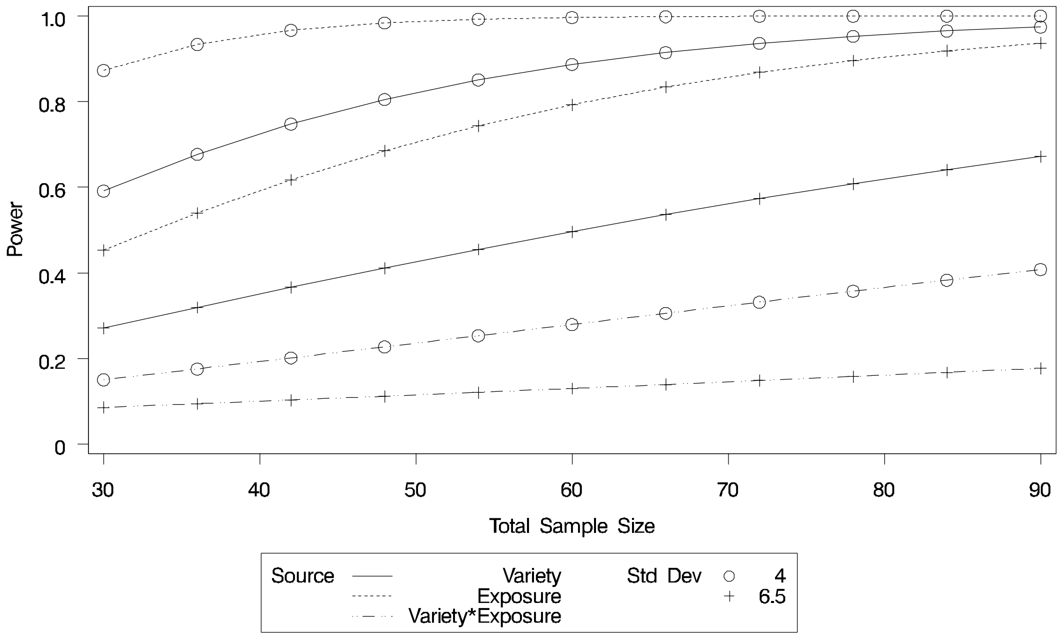


Figure 34.3. Plot of Power versus Sample Size for Two-Way ANOVA with Input Ranges

Figure 34.2 reveals that the power ranges from about 0.130 to 0.996 for the different effect tests and scenarios for standard deviation, with a sample size of 60. In Figure 34.3, the line style identifies the effect test, and the plotting symbol identifies the standard deviation. The locations of the plotting symbols identify actual computed powers; the curves are linear interpolations of these points. Note that the computed points in the plot occur at sample size multiples of 6, because there are 6 cells in the design (and by default, sample sizes are rounded to produce integer cell sizes).

Incorporating Contrasts, Unbalanced Designs, and Multiple Means Scenarios

Suppose you want to compute power for the two-way ANOVA described in "Simple Two-Way ANOVA," but you want to

- Try an unbalanced sample size allocation with respect to Exposure, using twice as many samples for levels 2 and 3 as for level 1.
- Consider an additional, less optimistic scenario for the cell means, shown in Table 34.2.
- Test a contrast of Exposure comparing levels 1 and 3.

Table 34.2. Additional Cell Means Scenario

	Exposure		
Variety	1	2	3
1	15	16	20
2	11	14	15

To specify the unbalanced design and the additional cell means scenario, you can add two new variables to the exemplary data set (Weight for the sample size weights, and HeightNew for the new cell means scenario). Rename the original cell means scenario to HeightOrig.

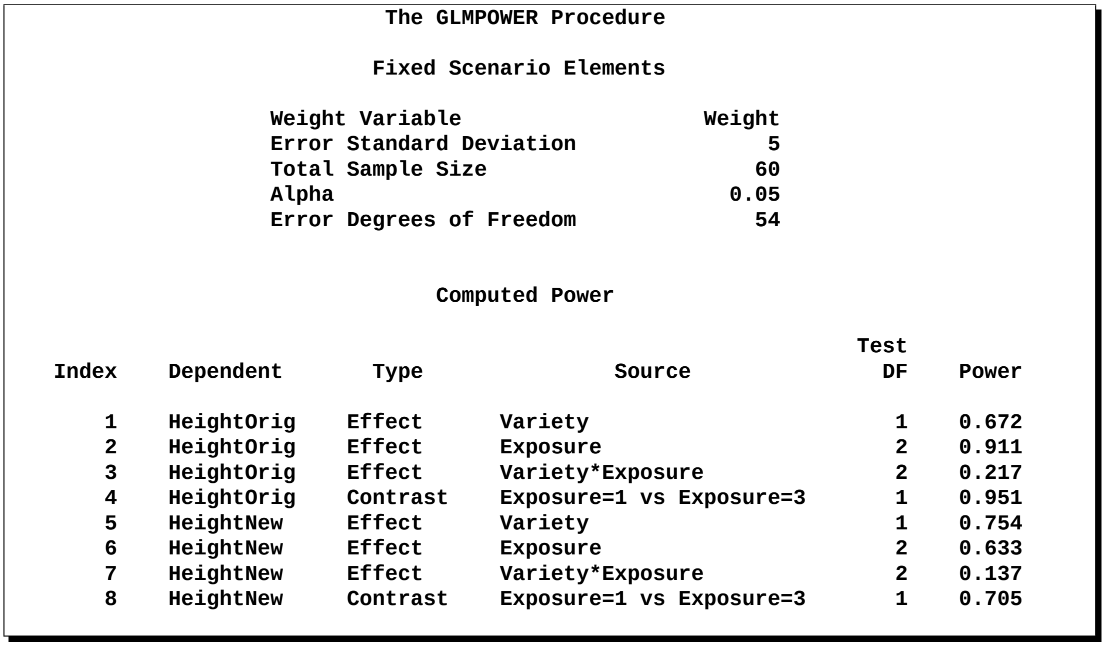
```
data Exemplary;
   input Variety $ Exposure $ HeightOrig HeightNew Weight;
   datalines;
         1   1   14  15  1
         1   2   16  16  2
         1   3   21  20  2
         2   1   10  11  1
         2   2   15  14  2
         2   3   16  15  2
   ;
run;
```

In PROC GLMPOWER, specify the name of the weight variable using the WEIGHT statement, and specify the name of the cell means variables as dependent variables in the MODEL statement. Use the CONTRAST statement to specify the contrast as you would in PROC GLM. The following statements perform the sample size analysis.

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model HeightOrig HeightNew = Variety | Exposure;
   weight Weight;
   contrast 'Exposure=1 vs Exposure=3' Exposure 1 0 -1;
```

```
      power
         stddev = 5
         ntotal = 60
         power  = .;
   run;
```

Figure 34.4 shows the output.



```
                              The GLMPOWER Procedure

                             Fixed Scenario Elements

                 Weight Variable                  Weight
                 Error Standard Deviation              5
                 Total Sample Size                    60
                 Alpha                              0.05
                 Error Degrees of Freedom             54


                                  Computed Power

                                                                 Test
  Index     Dependent        Type               Source             DF     Power

      1     HeightOrig    Effect      Variety                       1     0.672
      2     HeightOrig    Effect      Exposure                      2     0.911
      3     HeightOrig    Effect      Variety*Exposure              2     0.217
      4     HeightOrig    Contrast    Exposure=1 vs Exposure=3      1     0.951
      5     HeightNew     Effect      Variety                       1     0.754
      6     HeightNew     Effect      Exposure                      2     0.633
      7     HeightNew     Effect      Variety*Exposure              2     0.137
      8     HeightNew     Contrast    Exposure=1 vs Exposure=3      1     0.705
```

Figure 34.4. Sample Size Analysis for More Complex Two-Way ANOVA

The power of the contrast of Exposure levels 1 and 3 is about 0.95 for the original cell means scenario (HeightOrig) and only 0.71 for the new one (HeightNew). The power is higher for the test of Variety, but lower for the tests of Exposure and of Variety*Exposure for the new cell means scenario compared to the original one. Note also for the HeightOrig scenario that the power for the unbalanced design (Figure 34.4) compared to the balanced design (Figure 34.1) is slightly lower for the tests of Variety and Exposure, but slightly higher for the test of Variety*Exposure.

Syntax

The following statements are available in PROC GLMPOWER.

PROC GLMPOWER < *options* > **;**
CLASS *variables* **;**
MODEL *dependent-variables* = *effects* **;**
WEIGHT *variable* **;**
CONTRAST *'label' effect values* < . . . *effect values* > < **/** *options* > **;**
POWER < *options* > **;**
PLOT < *plot-options* > < **/** *graph-options* > **;**

The PROC GLMPOWER statement, the MODEL statement, and the POWER statement are required. If your model contains classification effects, the classification variables must be listed in a CLASS statement, and the CLASS statement must appear before the MODEL statement. In addition, CONTRAST statements must appear after the MODEL statement. PLOT statements must appear after the POWER statement defining the analysis for the plot.

You can use multiple CONTRAST, POWER, and PLOT statements. Each CONTRAST statement defines a separate contrast. Each POWER statement produces a separate analysis and uses the information contained in the CLASS, MODEL, WEIGHT, and all CONTRAST statements. Each PLOT statement refers to the previous POWER statement and generates a separate graph (or set of graphs).

Table 34.3 summarizes the basic functions of each statement in PROC GLMPOWER. The syntax of each statement in Table 34.3 is described in the following pages.

Table 34.3. Statements in the GLMPOWER Procedure

Statement	**Description**
PROC GLMPOWER	invokes procedure and specifies exemplary data set
CLASS	declares classification variables
CONTRAST	defines linear tests of model parameters
MODEL	defines model and specifies dependent variable(s) used for cell means scenarios
WEIGHT	specifies variable for allocating sample sizes to different subject profiles
POWER	identifies parameter to solve for and provides one or more scenarios for values of other analysis parameters
PLOT	displays graphs for preceding POWER statement

PROC GLMPOWER Statement

PROC GLMPOWER < *options* > **;**

The PROC GLMPOWER statement invokes the GLMPOWER procedure. You can specify the following options.

DATA=*SAS-data-set*

names a SAS data set to be used as the exemplary data set, which is an artificial data set constructed to represent the intended sampling design and the conjectured response means for the underlying population.

PLOTONLY

specifies that only graphical results from the PLOT statement should be produced.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Classification variables can be either character or numeric.

CONTRAST Statement

CONTRAST *'label' effect values* $<$ . . . *effect values* $>$ $<$ **/** *options* $>$ **;**

The CONTRAST statement enables you to define custom hypothesis tests by specifying an $\mathbf{L}$ vector or matrix for testing the hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$. Thus, to use this feature you must be familiar with the details of the model parameterization used in PROC GLM. For more information, see the "Parameterization of PROC GLM Models" section on page 1787 of Chapter 32, "The GLM Procedure." All of the elements of the $\mathbf{L}$ vector may be given, or if only certain portions of the $\mathbf{L}$ vector are given, the remaining elements are constructed by PROC GLMPOWER from the context (in a manner similar to rule 4 discussed in the "Construction of Least-Squares Means" section on page 1820 of Chapter 32, "The GLM Procedure").

There is no limit to the number of CONTRAST statements you can specify. Each sample size analysis includes tests for all CONTRAST statements.

In the CONTRAST statement,

label identifies the contrast on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ vector associated with the effect.

You can specify the following option in the CONTRAST statement after a slash(/):

SINGULAR=*number*

tunes the estimability checking. If ABS$(\mathbf{L} - \mathbf{LH}) > C \times$*number* for any row in the contrast, then $\mathbf{L}$ is declared nonestimable. $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS$(\mathbf{L})$ except for rows where $\mathbf{L}$ is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

The CONTRAST statement enables you to perform custom hypothesis tests. If the hypothesis is estimable, then the sum of squares due to it, SS$(H_0 \colon \mathbf{L}\boldsymbol{\beta} = 0)$, is computed as

$$(\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$ is the estimated solution vector.

The degrees of freedom associated with the hypothesis is equal to the row rank of **L**. The sum of squares computed in this situation is equivalent to the sum of squares computed using an **L** matrix with any row deleted that is a linear combination of previous rows.

Multiple-degree-of-freedom hypotheses can be specified by separating the rows of the **L** matrix with commas.

MODEL Statement

MODEL *dependent-variables* = *classification-effects* **;**

The MODEL statement serves two basic purposes.

- The *dependent-variables* specify scenarios for the cell means.
- The *classification-effects* specify the model effects.

The *classification-effects* specification defines the model effects. You can include main effects and interactions using the effects notation of PROC GLM; see the "Specification of Effects" section on page 1784 in Chapter 32, "The GLM Procedure" for further details.

All variables in the *classification-effects* specification must be contained in the CLASS statement, since power and sample size analyses cover only tests and contrasts of class effects. You can account for covariates in the model by using the NCOVARIATES=, CORRXY=, and PROPVARREDUCTION= options in the POWER statement.

Each dependent variable refers to a set of surmised cell means in the exemplary data set (named by the DATA= option of the PROC GLMPOWER statement). These cell means are response means for all of the subject profiles. Multiple dependent variables correspond to multiple scenarios for these cell means. All models are univariate; the GLMPOWER procedure currently does not support multivariate analyses.

The MODEL statement is required. You can specify only one MODEL statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names a variable that provides a profile weight ("cell weight") for each observation in the exemplary data set specified by the DATA= option in the PROC GLMPOWER statement.

If the WEIGHT statement is not used, then a balanced design is assumed with default cell weights of 1.

POWER Statement

POWER < *options* > **;**

The POWER statement performs power and sample size analyses for the Type III test of each effect in the model defined by the MODEL statements and for the contrasts defined by all CONTRAST statements.

Summary of Options

Table 34.4 summarizes categories of options available in the POWER statement.

Table 34.4. Summary of Options in the POWER Statement

Task	**Options**
Specify significance level	ALPHA=
Specify covariates	CORRXY= NCOVARIATES= PROPVARREDUCTION=
Specify error standard deviation	STDDEV=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 34.5 summarizes the valid result parameters.

Table 34.5. Summary of Result Parameters in the POWER Statement

Solve for	**Syntax**
Power	POWER = .
Sample size	NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of each test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. Note that this is a test-wise significance level with the same value for all tests, not incorporating any corrections for multiple testing. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

CORRXY=*number-list*

specifies the multiple correlation ρ between all covariates and the response. The error standard deviation given by the STDDEV= option is consequently reduced by multiplying it by a factor of $(1 - \rho^2)^{\frac{1}{2}}$, provided that the number of covariates (as determined by the NCOVARIATES= option) is greater than zero. You cannot use the CORRXY= and the PROPVARREDUCTION= options simultaneously. See the

"Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

NCOVARIATES=*number-list*
NCOVARIATE=*number-list*
NCOVS=*number-list*
NCOV=*number-list*

specifies the number of additional degrees of freedom to accommodate covariate effects— both class and continuous— not listed in the MODEL statement. The error degrees of freedom is consequently reduced by the value of the NCOVARIATES= option, and the error standard deviation (whose unadjusted value is provided with the STDDEV= option) is reduced according to the value of the CORRXY= or PROPVARREDUCTION= option. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 1946 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). Values for the sample size must be no smaller than the model degrees of freedom (counting the covariates). See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces the following order.

- weight variable (from the WEIGHT statement)
- source (contrasts from CONTRAST statements, and model effects)
- ALPHA
- dependent variable (from the MODEL statement, representing scenarios for cell means)
- NCOVARIATES
- CORRXY
- PROPVARREDUCTION
- STDDEV
- NTOTAL
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the POWER statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the POWER statement.

POWER= *number-list*

specifies the desired power of each test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability (for example, 0.9) rather than a percentage. Note that this is a test-wise power with the same value for all tests, without any correction for multiple testing. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

PROPVARREDUCTION=*number-list*
PVRED=*number-list*

specifies the proportional reduction (ρ) in total R^2 incurred by the covariates; in other words, the amount of additional variation explained by the covariates. The error standard deviation given by the STDDEV= option is consequently reduced by multiplying it by a factor of $(1 - \rho)^{\frac{1}{2}}$, provided that the number of covariates (as determined by the NCOVARIATES= option) is greater than zero. You cannot use the PROPVARREDUCTION= and the CORRXY= options simultaneously. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

STDDEV=*number-list*

specifies the error standard deviation, or root MSE. If covariates are specified using the NCOVARIATES= option, then the STDDEV= option denotes the error standard deviation before accounting for these covariates. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

Restrictions on Option Combinations

For the relationship between covariates and response, specify either the multiple correlation (using the CORRXY= option) or the proportional reduction in total R^2 (using the PROPVARREDUCTION= option).

PLOT Statement

PLOT < *plot-options* > < **/** *graph-options* > **;**

The PLOT statement produces a graph or set of graphs for the sample size analysis defined by the previous POWER statement. The *plot-options* define the plot characteristics, and the *graph-options* are SAS/GRAPH-style options.

Options

You can specify the following *plot-options* in the PLOT statement.

INTERPOL=JOIN
INTERPOL=NONE

specifies the type of curve to draw through the computed points. The INTERPOL=JOIN option connects computed points by straight lines. The INTERPOL=NONE option leaves computed points unconnected.

KEY= BYCURVE < **(** *bycurve-options* **)** >
KEY= BYFEATURE < **(** *byfeature-options* **)** >
KEY= ONCURVES

specifies the style of key (or “legend”) for the plot. The default is KEY=BYFEATURE, which specifies a key with a column of entries for each plot feature (line style, color, and/or symbol). Each entry shows the mapping between a value of the feature and the value(s) of the analysis parameter(s) linked to that feature. The KEY=BYCURVE option specifies a key with each row identifying a distinct curve in the plot. The KEY=ONCURVES option places a curve-specific label adjacent to each curve.

You can specify the following *byfeature-options* in parentheses after the KEY=BYCURVE option.

NUMBERS=OFF

NUMBERS=ON specifies how the key should identify curves. If NUMBERS=OFF, then the key includes symbol, color, and line style samples to identify the curves. If NUMBERS=ON, then the key includes numbers matching numeric labels placed adjacent to the curves. The default is NUMBERS=ON.

POS=BOTTOM

POS=INSET specifies the position of the key. The POS=BOTTOM option places the key below the x-axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

You can specify the following *byfeature-options* in parentheses after KEY=BYFEATURE option.

POS=BOTTOM

POS=INSET specifies the position of the key. The POS=BOTTOM option places the key below the x-axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

MARKERS=ANALYSIS
MARKERS=COMPUTED
MARKERS=NICE
MARKERS=NONE

specifies the locations for plotting symbols.

The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the values of the relevant input parameter from the POWER statement preceding the PLOT statement.

The MARKERS=COMPUTED option (the default) places plotting symbols at the locations of actual computed points from the sample size analysis.

The MARKERS=NICE option places plotting symbols at tick mark locations (corresponding to the argument axis).

The MARKERS=NONE option disables plotting symbols.

MAX=*number*

specifies the maximum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is the maximum value occurring for this parameter in the POWER statement preceding the PLOT statement.

MIN=*number*

specifies the minimum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is the minimum value occurring for this parameter in the POWER statement preceding the PLOT statement.

NPOINTS=*number*
NPTS=*number*

specifies the number of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the NPOINTS= and STEP= options simultaneously. The default value for typical situations is 20.

STEP=*number*

specifies the increment between values of the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the STEP= and NPOINTS= options simultaneously. By default, the NPOINTS= option is used instead of the STEP= option.

VARY (*feature* < **BY** *parameter-list* > **...** *feature* < **BY** *parameter-list* > **)**

specifies how plot features should be linked to varying analysis parameters. Available plot *features* are COLOR, LINESTYLE, PANEL, and SYMBOL. A “panel” refers to a separate plot with a heading identifying the subset of values represented in the plot.

The *parameter-list* is a list of one or more names separated by spaces. Each name must match the name of an analysis option used in the POWER statement preceding the PLOT statement, *or* one of the following keywords: SOURCE (for the tests) and DEPENDENT (for the cell means scenarios). Also, the name must be the *primary* name for the analysis option, that is, the one listed first in the syntax description.

If you omit the < BY *parameter-list* > portion for a feature, then one or more multi-valued parameters from the analysis will be automatically selected for you.

X=N
X=POWER

specifies a plot with the requested type of parameter on the x-axis and the parameter being solved for on the y-axis. When X=N, sample size is assigned to the x-axis. When X=POWER, power is assigned to the x-axis. You cannot use the X= and Y= options simultaneously. The default is X=POWER, unless the result parameter is power, in which case the default is X=N.

XOPTS= (*x-options* **)**

specifies plot characteristics pertaining to the x-axis.

You can specify the following *x-options* in parentheses.

CROSSREF=NO

CROSSREF=YES specifies whether the reference lines defined by the REF= *x-option* should be crossed with a reference line on the y-axis that indicates the solution point on the curve.

REF=*number-list* specifies locations for reference lines extending from the x-axis across the entire plotting region. See the “Specifying Value Lists in the POWER Statement” section on page 1945 for information on specifying the *number-list*.

Y=N
Y=POWER

specifies a plot with the requested type of parameter on the y-axis and the parameter being solved for on the x-axis. When Y=N, sample size is assigned to the y-axis. When Y=POWER, power is assigned to the y-axis. You cannot use the Y= and X= options simultaneously. By default, the X= option is used instead of the Y= option.

YOPTS= (*y-options* **)**

specifies plot characteristics pertaining to the y-axis.

You can specify the following *y-options* in parentheses.

CROSSREF=NO

CROSSREF=YES specifies whether the reference lines defined by the REF= *y-option* should be crossed with a reference line on the x-axis that indicates the solution point on the curve.

REF=*number-list* specifies locations for reference lines extending from the y-axis across the entire plotting region. See the "Specifying Value Lists in the POWER Statement" section on page 1945 for information on specifying the *number-list*.

You can specify the following *graph-options* in the PLOT statement after a slash (/).

DESCRIPTION='*string* '

specifies a descriptive string of up to 40 characters that appears in the "Description" field of the graphics catalog. The description does not appear on the plots. By default, PROC GLMPOWER assigns a description either of the form "*Y* versus *X*" (for a single-panel plot) or of the form "*Y* versus *X* (*S*)," where *Y* is the parameter on the y-axis, *X* is the parameter on the x-axis, and *S* is a description of the subset represented on the current panel of a multipanel plot.

NAME='*string* '

specifies a name of up to eight characters for the catalog entry for the plot. The default name is PLOT*n*, where *n* is the number of the plot statement within the current invocation of PROC GLMPOWER. If the name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry, for example, PLOT11 and PLOT12 for the second and third panels of a multipanel plot generated in the first PLOT statement in an invocation of PROC GLMPOWER.

Details

Specifying Value Lists in the POWER Statement

To specify one or more scenarios for an analysis parameter (or set of parameters) in the POWER statement, you provide a list of values for the option that corresponds to the parameter(s). To identify the parameter you wish to solve for, you place missing values in the appropriate list.

Scenarios for scalar-valued parameters, such as power, are represented by a *number-list*.

Number-lists

A *number-list* can be one of two things: a series of one or more numbers expressed in the form of one or more DOLISTs, or a missing value indicator (.).

The DOLIST format is the same as in the DATA step language. For example, you can specify four scenarios (30, 50, 70, and 100) for a total sample size in any of the following ways.

```
NTOTAL = 30 50 70 100
NTOTAL = 30 to 70 by 20 100
```

A missing value identifies a parameter as the result parameter; it is valid only with options representing parameters you can solve for in a given analysis. For example, you can request a solution for NTOTAL:

```
NTOTAL = .
```

Sample Size Adjustment Options

By default, PROC GLMPOWER rounds sample sizes conservatively (down in the input, up in the output) so that all total sizes *and* sample sizes for individual design profiles are integers. This is generally considered conservative because it selects the closest realistic design providing *at most* the power of the (possibly fractional) input or mathematically optimized design. In addition, all design profile sizes are adjusted to be multiples of their corresponding weights. If a design profile is present more than once in the exemplary data set, then the weights for that design profile are summed. For example, if a particular design profile is present twice in the exemplary data set with weight values 2 and 6, then all sample sizes for this design profile become multiples of $2 + 6 = 8$.

With the NFRACTIONAL option, sample size input is not rounded, and sample size output is reported in two versions, a raw "fractional" version and a "ceiling" version rounded up to the nearest integer.

Whenever an input sample size is adjusted, both the original ("nominal") and adjusted ("actual") sample sizes are reported. Whenever computed output sample sizes are adjusted, both the original input ("nominal") power and the achieved ("actual") power at the adjusted sample size are reported.

Error and Information Output

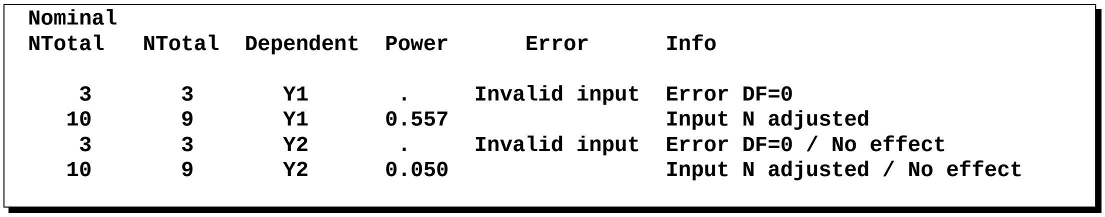
The Error column in the main output table explains reasons for missing results and flags numerical results that are bounds rather than exact answers.

The Information column provides further details about Error entries, warnings about any boundary conditions detected, and notes about any adjustments to input. Note that the Information column is hidden by default in the main output. You can view it by using the ODS OUTPUT statement to save the output as a dataset and the PRINT procedure. For example, the following SAS statements print both the Error and Info columns for a power computation in a one-way ANOVA.

```
data MyExemp;
   input A $ Y1 Y2;
   datalines;
         1   10 11
         2   12 11
         3   15 11
   ;
run;
```

```
proc glmpower data=MyExemp;
   class A;
   model Y1 Y2 = A;
   power
      stddev = 2
      ntotal = 3 10
      power  = .;
   ods output output=Power;
proc print noobs data=Power;
   var NominalNTotal NTotal Dependent Power Error Info;
run;
```

The output is shown in Figure 34.5.



```
Nominal
NTotal   NTotal  Dependent  Power      Error       Info

   3        3       Y1        .     Invalid input  Error DF=0
  10        9       Y1      0.557                  Input N adjusted
   3        3       Y2        .     Invalid input  Error DF=0 / No effect
  10        9       Y2      0.050                  Input N adjusted / No effect
```

Figure 34.5. Error and Information Columns

The sample size of 3 specified with the NTOTAL= option leads to an “Invalid input” message in the Error column and an “Error DF=0” message in the Info column, because a sample size of 3 is so small that there are no degrees of freedom left for the error term. The sample size of 10 leads to an “Input N adjusted” message in the Info column, because it is rounded down to 9 to produce integer group sizes of 3 per cell. The cell means scenario represented by the dependent variable Y2 leads to a “No effect” message to appear in the Info column, because the means in this scenario are all equal.

Displayed Output

If you use the PLOTONLY option in the PROC GLMPOWER statement, the procedure only displays graphical output. Otherwise, the displayed output of the GLMPOWER procedure includes the following:

- the “Fixed Scenario Elements” table, which shows all applicable single-valued analysis parameters, in the following order: the weight variable, the source of the test, parameters input explicitly, parameters supplied with defaults, and ancillary results
- an output table showing the following when applicable (in order): the index of the scenario, the source of the test, all multivalued input, ancillary results, the primary computed result, and error descriptions
- plots (if requested)

Ancillary results include the following:

- Actual Power, the achieved power, if it differs from the input (Nominal) power value
- fractional sample size, if the NFRACTIONAL option is used in the analysis statement

If sample size is the result parameter and the NFRACTIONAL option is used in the analysis statement, then both “Fractional” and “Ceiling” sample size results are displayed. Fractional sample sizes correspond to the “Nominal” values of power or precision probability. Ceiling sample sizes are simply the fractional sample sizes rounded up to the nearest integer; they correspond to “Actual” values of power or precision probability.

ODS Table Names

PROC GLMPOWER assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 34.6. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 34.6. ODS Tables Produced in PROC GLMPOWER

ODS Table Name	Description	Statement
FixedElements	factoid with single-valued analysis parameters	default
Output	all input and computed analysis parameters, error messages, and information messages for each scenario	default
PlotContent	data contained in plots, including analysis parameters and indices identifying plot features. (**Note:** This table is saved as a dataset and not displayed in PROC GLMPOWER output.)	PLOT

The ODS path names are created as follows:

- Glmpower.Power$< n >$.FixedElements
- Glmpower.Power$< n >$.Output
- Glmpower.Power$< n >$.PlotContent
- Glmpower.Power$< n >$.Plot$< m >$

where

- The Plot$< m >$ objects are the graphs.
- The $< n >$ indexing the Power statement is only used if there is more than one instance.

- The $< n >$ indexing the plots increases with every panel in every plot statement, resetting to 1 only at new analysis statements.

Computational Methods and Formulas

This section describes the approaches used in PROC GLMPOWER to compute power and sample size.

Contrasts in Fixed-Effect Univariate Models

The univariate linear model has the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is the $N \times 1$ vector of responses, $\mathbf{X}$ is the $N \times p$ design matrix, $\boldsymbol{\beta}$ is the $p \times 1$ vector of model parameters corresponding to the columns of $\mathbf{X}$, and $\boldsymbol{\epsilon}$ is an $N \times 1$ vector of errors with

$$\epsilon_1, \ldots, \epsilon_N \sim \mathrm{N}(0, \sigma^2) \quad \text{(i.i.d.)}$$

In PROC GLMPOWER, the model parameters $\boldsymbol{\beta}$ are not specified directly, but rather indirectly as $\mathbf{y}^\star$, which represents either conjectured response means or typical response values for each design profile. The $\mathbf{y}^\star$ values are manifested as the dependent variable in the MODEL statement. The vector $\boldsymbol{\beta}$ is obtained from $\mathbf{y}^\star$ according to the least squares equation,

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}^\star$$

Note that, in general, there is not a 1 to 1 mapping between $\mathbf{y}^\star$ and $\boldsymbol{\beta}$. Many different scenarios for $\mathbf{y}^\star$ may lead to the same $\boldsymbol{\beta}$. If you specify $\mathbf{y}^\star$ with the intention of representing cell means, keep in mind that PROC GLMPOWER allows scenarios that are *not* valid cell means according to the model specified in the MODEL statement. For example, if $\mathbf{y}^\star$ exhibits an interaction effect but the corresponding interaction term is left out of the model, then the cell means ($\mathbf{X}\boldsymbol{\beta}$) derived from $\boldsymbol{\beta}$ differ from $\mathbf{y}^\star$. In particular, the cell means thus derived are the projection of $\mathbf{y}^\star$ onto the model space.

It is convenient in power analysis to parameterize the design matrix $\mathbf{X}$ in three parts, $\{\ddot{\mathbf{X}}, \mathbf{w}, N\}$, defined as follows:

1. The $q \times p$ essence design matrix $\ddot{\mathbf{X}}$ is the collection of unique rows of $\mathbf{X}$. Its rows are sometimes referred to as “design profiles.” Here, $q \leq N$ is defined simply as the number of unique rows of $\mathbf{X}$.
2. The $q \times 1$ weight vector $\mathbf{w}$ reveals the relative proportions of design profiles. Row i of $\ddot{\mathbf{X}}$ is to be included in the design w_i times for every w_j times row j is included. The weights are assumed to be standardized (i.e., sum up to 1).

3. The total sample size is N. This is the number of rows in $\mathbf{X}$. If you gather $Nw_i = n_i$ copies of the i^{th} row of $\ddot{\mathbf{X}}$, for $i = 1, \ldots, q$, then you end up with $\mathbf{X}$.

It is useful to express the the crossproduct matrix $\mathbf{X}'\mathbf{X}$ in terms of these three parts,

$$\mathbf{X}'\mathbf{X} = N\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}$$

since this factors out the portion (N) depending on sample size and the portion ($\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}$) depending only on the design structure.

A general linear hypothesis for the univariate model has the form

$$\begin{aligned} H_0 &: \mathbf{L}\boldsymbol{\beta} = \theta_{\mathbf{0}} \\ H_A &: \mathbf{L}\boldsymbol{\beta} \neq \theta_{\mathbf{0}} \end{aligned}$$

where $\mathbf{L}$ is an $r_L \times p$ contrast matrix (assumed to be full rank), and $\theta_{\mathbf{0}}$ is the null value (usually just a vector of zeroes). Note that effect tests are just contrasts using special forms of $\mathbf{L}$. Thus, this scheme covers both effect tests and custom contrasts.

The test statistic is

$$F = \frac{\left(\frac{\text{SS}_\text{H}}{r_L}\right)}{\hat{\sigma}^2}$$

where

$$\begin{aligned} \text{SS}_\text{H} &= \frac{1}{N}\left(\mathbf{L}\hat{\boldsymbol{\beta}} - \theta_{\mathbf{0}}\right)'\left(\mathbf{L}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{L}'\right)^{-1}\left(\mathbf{L}\hat{\boldsymbol{\beta}} - \theta_{\mathbf{0}}\right) \\ \hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ \hat{\sigma}^2 &= \frac{1}{\text{DF}_\text{E}}\left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)'\left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right) \end{aligned}$$

where $\text{DF}_\text{E} = N - \text{rank}(X)$. Note that $\text{DF}_\text{E} = N - p$ if X has full rank.

Under H_0, $F \sim F(r_L, \text{DF}_\text{E})$. Under H_A, F is distributed as $F(r_L, \text{DF}_\text{E}, \lambda)$ with noncentrality

$$\lambda = N\left(\mathbf{L}\boldsymbol{\beta} - \theta_{\mathbf{0}}\right)'\left(\mathbf{L}\left(\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}\right)^{-1}\mathbf{L}'\right)^{-1}\left(\mathbf{L}\boldsymbol{\beta} - \theta_{\mathbf{0}}\right)\sigma^{-2}$$

Muller and Peterson (1984) give the exact power of the test as

$$\text{power} = P\left(F(r_L, \text{DF}_\text{E}, \lambda) \geq F_{1-\alpha}(r_L, \text{DF}_\text{E})\right)$$

Sample size is computed by inverting the power equation.

Refer to Muller et al. (1992) and O'Brien and Shieh (1992) for additional discussion.

Adjustments for Covariates

If you specify covariates in the model (whether continuous or categorical), then two adjustments are made in order to compute approximate power in the presence of the covariates. Let n_ν denote the number of covariates (counting dummy variables for categorical covariates individually). In other words, n_ν is the total degrees of freedom used by the covariates. The adjustments are the following:

1. The error degrees of freedom decreases by n_ν.
2. The error standard deviation σ shrinks by a factor of $(1-\rho^2)^{\frac{1}{2}}$ (if the CORRXY= option to specify the correlation ρ between covariates and response) or $(1-r)^{\frac{1}{2}}$ (if the PROPVARREDUCTION= option is used to specify the proportional reduction in total R^2 incurred by the covariates). Let $\sigma^\star$ represent the updated value of σ.

As a result of these changes, the power is computed as

$$\text{power} = P\left(F(r_L, \text{DF}_\text{E} - n_\nu, \lambda^\star) \geq F_{1-\alpha}(r_L, N - r_x - n_\nu)\right)$$

where $\lambda^\star$ is calculated using $\sigma^\star$ rather than σ:

$$\lambda^\star = N\left(\mathbf{L}\boldsymbol{\beta} - \theta_\mathbf{0}\right)' \left(\mathbf{L}\left(\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}\right)^{-1}\mathbf{L}'\right)^{-1}\left(\mathbf{L}\boldsymbol{\beta} - \theta_\mathbf{0}\right)(\sigma^\star)^{-2}$$

Examples

Example 34.1. One-Way ANOVA

This example deals with the same situation as in Example 57.1 on page 3536 of Chapter 57, “The POWER Procedure.”

Hocking (1985, p. 109) describes a study of the effectiveness of electrolytes in reducing lactic acid buildup for long-distance runners. You are planning a similar study in which you will allocate five different fluids to runners on a 10-mile course and measure lactic acid buildup immediately after the race. The fluids consist of water and two commercial electrolyte drinks, EZDure and LactoZap, each prepared at two concentrations, low (EZD1 and LZ1) and high (EZD2 and LZ2).

You conjecture that the standard deviation of lactic acid measurements given any particular fluid is about 3.75, and that the expected lactic acid values will correspond roughly to Table 34.7. You are least familiar with the LZ1 drink and hence decide to consider a range of reasonable values for that mean.

Table 34.7. Mean Lactic Acid Buildup by Fluid

Water	EZD1	EZD2	LZ1	LZ2
35.6	33.7	30.2	29 or 28	25.9

You are interested in four different comparisons, shown in Table 34.8 with appropriate contrast coefficients.

Table 34.8. Planned Comparisons

	Contrast Coefficients				
Comparison	Water	EZD1	EZD2	LZ1	LZ2
Water versus electrolytes	4	-1	-1	-1	-1
EZD versus LZ	0	1	1	-1	-1
EZD1 versus EZD2	0	1	-1	0	0
LZ1 versus LZ2	0	0	0	1	-1

For each of these contrasts you want to determine the sample size required to achieve a power of 0.9 for detecting an effect with magnitude in accord with Table 34.7. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed for individual contrasts. You plan to test each contrast at $\alpha = 0.025$. In the interests of reducing costs, you will provide twice as many runners with water as with any of the electrolytes; that is, you will use a sample size weighting scheme of 2:1:1:1:1.

Before calling PROC GLMPOWER, you need to create the *exemplary data set* to specify means and weights for the design profiles:

```
data Fluids;
   input Fluid $ LacticAcid1 LacticAcid2 CellWgt;
   datalines;
         Water      35.6        35.6        2
         EZD1       33.7        33.7        1
         EZD2       30.2        30.2        1
         LZ1        29          28          1
         LZ2        25.9        25.9        1
   ;
run;
```

The variable LacticAcid1 represents the cell means scenario with the larger LZ1 mean (29), and LacticAcid2 represents the scenario with the smaller LZ1 mean (28). The variable CellWgt contains the sample size allocation weights.

Use the DATA= option in the PROC GLMPOWER statement to specify Fluids as the exemplary data set. The following statements perform the sample size analysis:

```
proc glmpower data=Fluids;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = .
      power  = 0.9;
run;
```

The CLASS statement identifies Fluid as a classification variable. The MODEL statement specifies the model and the two cell means scenarios LacticAcid1 and LacticAcid2. The WEIGHT statement identifies CellWgt as the weight variable. The CONTRAST statement specifies the contrasts. Since PROC GLMPOWER processes class levels in order of formatted values, the contrast coefficients correspond to the following order: EZD1, EZD2, LZ1, LZ2, Water. The POWER statement specifies total sample size as the result parameter and provides values for the other analysis parameters (error standard deviation, alpha, and power).

Output 34.1.1 displays the results.

Output 34.1.1. Sample Sizes for One-Way ANOVA Contrasts

```
                               The GLMPOWER Procedure

                              Fixed Scenario Elements

                     Weight Variable                   CellWgt
                     Alpha                               0.025
                     Error Standard Deviation             3.75
                     Nominal Power                         0.9


                                  Computed N Total

                                                          Test   Error   Actual       N
  Index    Dependent      Type          Source              DF      DF    Power   Total

      1  LacticAcid1  Effect    Fluid                        4      25    0.958      30
      2  LacticAcid1  Contrast  Water vs. others             1      25    0.947      30
      3  LacticAcid1  Contrast  EZD vs. LZ                   1      55    0.929      60
      4  LacticAcid1  Contrast  EZD1 vs. EZD2                1     169    0.901     174
      5  LacticAcid1  Contrast  LZ1 vs. LZ2                  1     217    0.902     222
      6  LacticAcid2  Effect    Fluid                        4      25    0.972      30
      7  LacticAcid2  Contrast  Water vs. others             1      19    0.901      24
      8  LacticAcid2  Contrast  EZD vs. LZ                   1      43    0.922      48
      9  LacticAcid2  Contrast  EZD1 vs. EZD2                1     169    0.901     174
     10  LacticAcid2  Contrast  LZ1 vs. LZ2                  1     475    0.902     480
```

The sample sizes range from 24 for the comparison of water versus electrolytes to 480 for the comparison of LZ1 versus LZ2, both assuming the smaller LZ1 mean. The sample size for the latter comparison is relatively large because the small mean difference of $28 - 25.9 = 2.1$ is hard to detect. PROC GLMPOWER also includes the effect test for Fluid. Note that, in this case, it is equivalent to TEST=OVERALL_F in the ONEWAYANOVA statement of PROC POWER, since there is only one effect in the model.

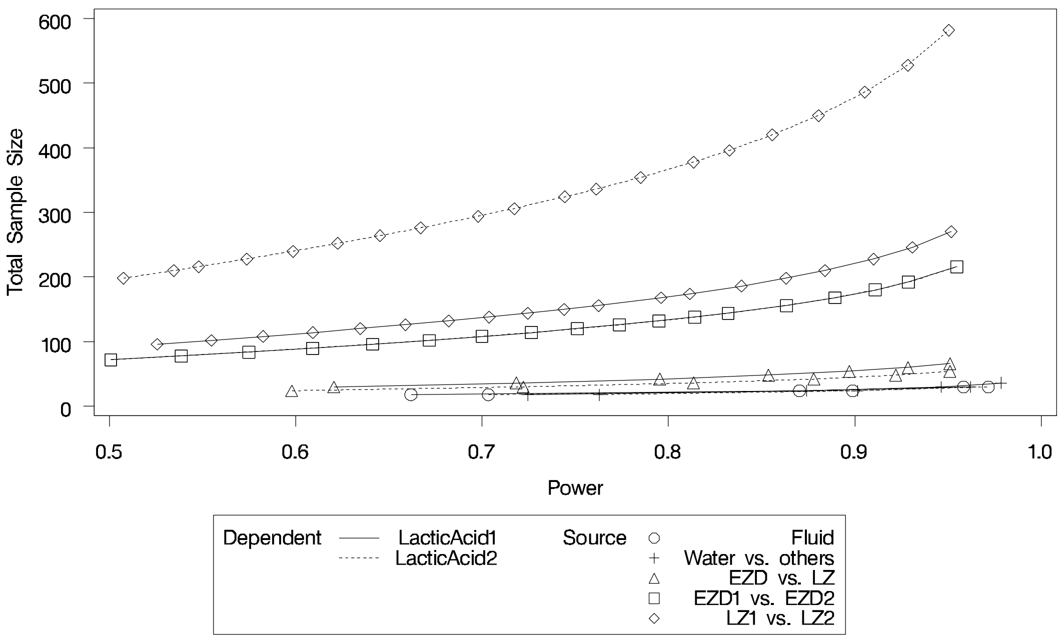
The Nominal Power of 0.9 in the Fixed Scenario Elements table in Output 34.1.1 represents the input target power, and the Actual Power column in the Computed N Total table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting. Note that all of the sample sizes are rounded up to multiples of 6 to preserve integer group sizes (since the group weights add up to 6). You can use the NFRACTIONAL option in the POWER statement to compute raw fractional sample sizes.

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC GLMPOWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the x-axis. (The result parameter, here sample size, is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the power range.

```
proc glmpower data=Fluids plotonly;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = .
      power  = 0.9;
   plot x=power min=.5 max=.95;
run;
```

See Output 34.1.2 for the resulting plot.

Output 34.1.2. Plot of Sample Size versus Power for One-Way ANOVA Contrasts



In Output 34.1.2, the line style identifies the test, and the plotting symbol identifies the cell means scenario. The plotting symbol locations identify actual computed powers; the curves are linear interpolations of these points. The plot shows that the required sample size is highest for the test of LZ1 versus LZ2 that was previously found to require the most resources, in either cell means scenario.

Note that some of the plotted points in Output 34.1.2 are unevenly spaced. This is because the plotted points are the *rounded* sample size results at their corresponding *actual* power levels. The range specified with the MIN= and MAX= values in the PLOT statement correspond to *nominal* power levels. In some cases, actual power is substantially higher than nominal power. To obtain plots with evenly spaced points (but with *fractional* sample sizes at the computed points), you can use the NFRACTIONAL option in the POWER statement preceding the PLOT statement.

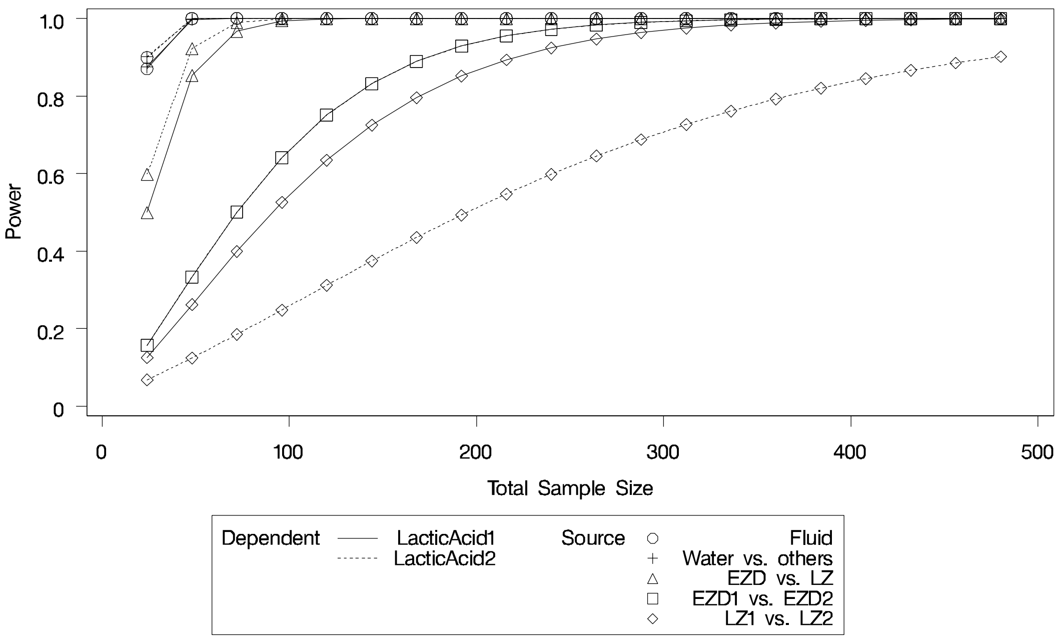
Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 24 to 480 that achieves 0.9 power for different comparisons). In the POWER statement, identify power as the result (POWER=.), and specify NTOTAL=24. Specify the PLOT statement with X=N to request a plot with sample size on the x-axis.

```
proc glmpower data=Fluids plotonly;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = 24
      power  = .;
   plot x=n min=24 max=480;
run;
```

Note that the value specified with the NTOTAL=24 option is not used. It is overridden in the plot by the MIN= and MAX= options in the PLOT statement, and the PLOTONLY option in the PROC GLMPOWER statement disables nongraphical results. But the NTOTAL= option (along with a value) is still needed in the POWER statement as a placeholder, to identify the desired parameterization for sample size.

See Output 34.1.3 for the plot.

Output 34.1.3. Plot of Power versus Sample Size for One-Way ANOVA Contrasts



Although Output 34.1.2 and Output 34.1.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment. Output 34.1.2 reveals the range of required sample sizes for powers of interest, and Output 34.1.3 reveals the range of achieved powers for sample sizes of interest.

Example 34.2. Two-Way ANOVA with Covariate

Suppose you can enhance the planned study discussed in Example 34.1 on page 1951 in two ways:

- Incorporate results from races at two different altitudes (“high” and “low”).
- Measure the body mass index of each runner before the race.

This is equivalent to adding a second fixed effect and a continuous covariate to your model.

Since lactic acid buildup is more pronounced at higher altitudes, you will include altitude as a factor in the model along with fluid, extending the one-way ANOVA to a two-way ANOVA. In doing so, you expect to lower the residual standard deviation from about 3.75 to 3.5 (in addition to generalizing the study results). You assume there is negligible interaction between fluid and altitude and plan to use a main-effects-only model. You conjecture that the mean lactic acid buildup follows Table 34.9.

Table 34.9. Mean Lactic Acid Buildup by Fluid and Altitude

	Fluid				
Altitude	Water	EZD1	EZD2	LZ1	LZ2
High	36.9	35.0	31.5	30	27.1
Low	34.3	32.4	28.9	27	24.7

By including a measurement of body mass index as a covariate in the study, you hope to further reduce the error variability. The extent of this reduction in variability is commonly expressed in two alternative ways: (1) the correlation between the covariates and the response or (2) the proportional reduction in total R^2 incurred by the covariates. You prefer the former and guess that the correlation between body mass index and lactic acid buildup is between 0.2 and 0.3. You specify these estimates with the NCOVARIATES= and CORRXY= options in the POWER statement. The covariate is not included in the MODEL statement.

You are interested in the same four fluid comparisons as in Example 34.1, shown in Table 34.8 on page 1951, except this time you want to marginalize over the effect of altitude.

For each of these contrasts, you want to determine the sample size required to achieve a power of 0.9 to detect an effect with magnitude according to Table 34.9. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed by individual contrasts. You plan to test each contrast at $\alpha = 0.025$. You will provide twice as many runners with water as with any of the electrolytes, and you predict that you can study approximately 2/3 as many runners at the high altitude than at the low altitude. The resulting planned sample size weighting scheme is shown in Table 34.10. Since the scheme is only approximate, you use the NFRACTIONAL option in the POWER statement to disable the rounding of sample sizes up to integers satisfying the weights exactly.

Table 34.10. Approximate Sample Size Allocation Weights

	Fluid				
Altitude	Water	EZD1	EZD2	LZ1	LZ2
High	4	2	2	2	2
Low	6	3	3	3	3

First, you create the exemplary data set to specify means and weights for the design profiles:

```
data Fluids2;
   input Altitude $ Fluid $ LacticAcid CellWgt;
   datalines;
         High       Water      36.9       4
         High       EZD1       35.0       2
         High       EZD2       31.5       2
         High       LZ1        30         2
         High       LZ2        27.1       2
         Low        Water      34.3       6
         Low        EZD1       32.4       3
         Low        EZD2       28.9       3
         Low        LZ1        27         3
         Low        LZ2        24.7       3
   ;
run;
```

The variables Altitude, Fluid, and LacticAcid specify the factors and cell means in Table 34.9. The variable CellWgt contains the sample size allocation weights in Table 34.10.

Use the DATA= option in the PROC GLMPOWER statement to specify Fluids2 as the exemplary data set. The following statements perform the sample size analysis:

```
proc glmpower data=Fluids2;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = .
      power       = 0.9;
run;
```

The CLASS statement identifies Altitude and Fluid as classification variables. The MODEL statement specifies the model, and the WEIGHT statement identifies CellWgt as the weight variable. The CONTRAST statement specifies the contrasts in Table 34.8 on page 1951. As in Example 34.1, the order of the contrast coefficients corresponds to the formatted class levels (EZD1, EZD2, LZ1, LZ2, Water). The POWER statement specifies total sample size as the result parameter and provides values for the other analysis parameters. The NCOVARIATES= option specifies the single covariate (body mass index), and the CORRXY= option specifies the two scenarios for its correlation with lactic acid buildup (0.2 and 0.3). Output 34.2.1 displays the results.

Output 34.2.1. Sample Sizes for Two-Way ANOVA Contrasts

```
                            The GLMPOWER Procedure

                           Fixed Scenario Elements

               Dependent Variable                          LacticAcid
               Weight Variable                                CellWgt
               Alpha                                            0.025
               Number of Covariates                                 1
               Std Dev Without Covariate Adjustment               3.5
               Nominal Power                                      0.9


                           Computed Ceiling N Total

                                                Adj
                                        Corr    Std   Test   Error     Fractional
      Index     Type         Source       XY    Dev     DF      DF        N Total

          1  Effect    Altitude          0.2   3.43      1      84      90.418451
          2  Effect    Altitude          0.3   3.34      1      79      85.862649
          3  Effect    Altitude          0.0   3.50      1      88      94.063984
          4  Effect    Fluid             0.2   3.43      4      16      22.446173
          5  Effect    Fluid             0.3   3.34      4      15      21.687544
          6  Effect    Fluid             0.0   3.50      4      17      23.055716
          7  Contrast  Water vs. others  0.2   3.43      1      15      21.720195
          8  Contrast  Water vs. others  0.3   3.34      1      14      20.848805
          9  Contrast  Water vs. others  0.0   3.50      1      16      22.422381
         10  Contrast  EZD vs. LZ        0.2   3.43      1      35      41.657424
         11  Contrast  EZD vs. LZ        0.3   3.34      1      33      39.674037
         12  Contrast  EZD vs. LZ        0.0   3.50      1      37      43.246415
         13  Contrast  EZD1 vs. EZD2     0.2   3.43      1     139     145.613657
         14  Contrast  EZD1 vs. EZD2     0.3   3.34      1     132     138.173983
         15  Contrast  EZD1 vs. EZD2     0.0   3.50      1     145     151.565917
         16  Contrast  LZ1 vs. LZ2       0.2   3.43      1     268     274.055008
         17  Contrast  LZ1 vs. LZ2       0.3   3.34      1     253     259.919126
         18  Contrast  LZ1 vs. LZ2       0.0   3.50      1     279     285.363976

                           Computed Ceiling N Total

                                     Actual    Ceiling
                             Index    Power    N Total

                                 1    0.902         91
                                 2    0.901         86
                                 3    0.903         95
                                 4    0.912         23
                                 5    0.908         22
                                 6    0.919         24
                                 7    0.905         22
                                 8    0.903         21
                                 9    0.910         23
                                10    0.903         42
                                11    0.903         40
                                12    0.906         44
                                13    0.901        146
                                14    0.902        139
                                15    0.901        152
                                16    0.901        275
                                17    0.900        260
                                18    0.901        286
```

The sample sizes in Output 34.2.1 range from 21 for the comparison of water versus electrolytes (assuming a correlation of 0.3 between body mass and lactic acid buildup) to 275 for the comparison of LZ1 versus LZ2 (assuming a correlation of 0.2). PROC GLMPOWER also includes the effect tests for Altitude and Fluid. Note that the required sample sizes for this study are lower than those for the study in Example 34.1.

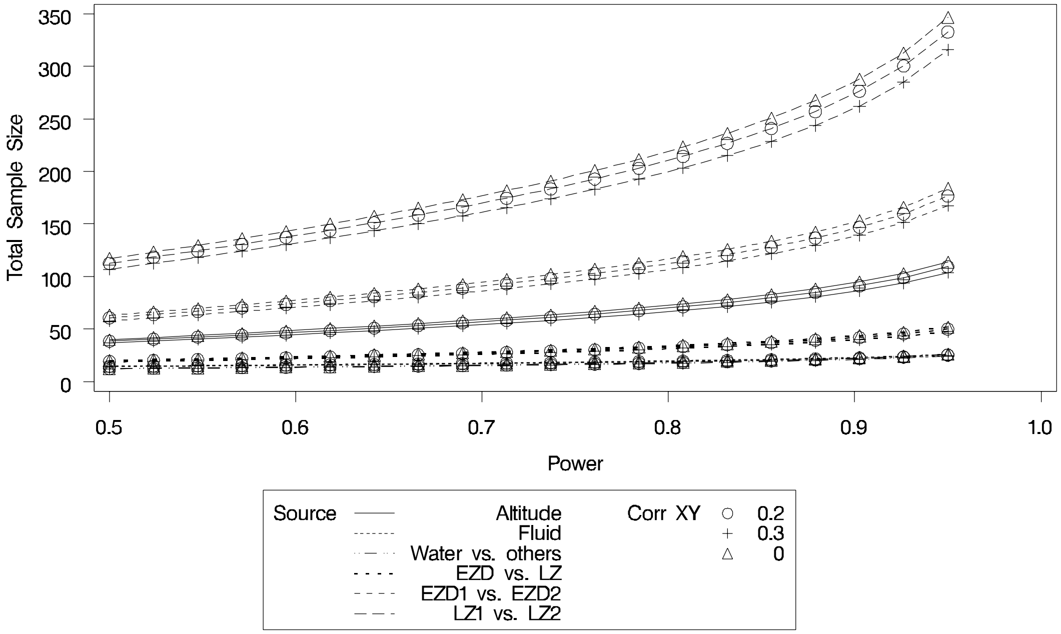
Note that the error standard deviation has been reduced from 3.5 to 3.43 (when correlation is 0.2) or 3.34 (when correlation is 0.3) in the approximation of the effect of the body mass index covariate. The error degrees of freedom has also been automatically adjusted, lowered by 1 (the number of covariates).

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC GLMPOWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the x-axis. Sample size is automatically placed on the y-axis. Use the MIN= and MAX= options in the PLOT statement to specify the power range.

```
proc glmpower data=Fluids2 plotonly;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = .
      power       = 0.9;
   plot x=power min=.5 max=.95;
run;
```

See Output 34.2.2 for the plot.

Output 34.2.2. Plot of Sample Size versus Power for Two-Way ANOVA Contrasts



In Output 34.1.2, the line style identifies the test, and the plotting symbol identifies the scenario for the correlation between covariate and response. The plotting symbol locations identify actual computed powers; the curves are linear interpolations of these points. As in Example 34.1, the required sample size is highest for the test of LZ1 versus LZ2.

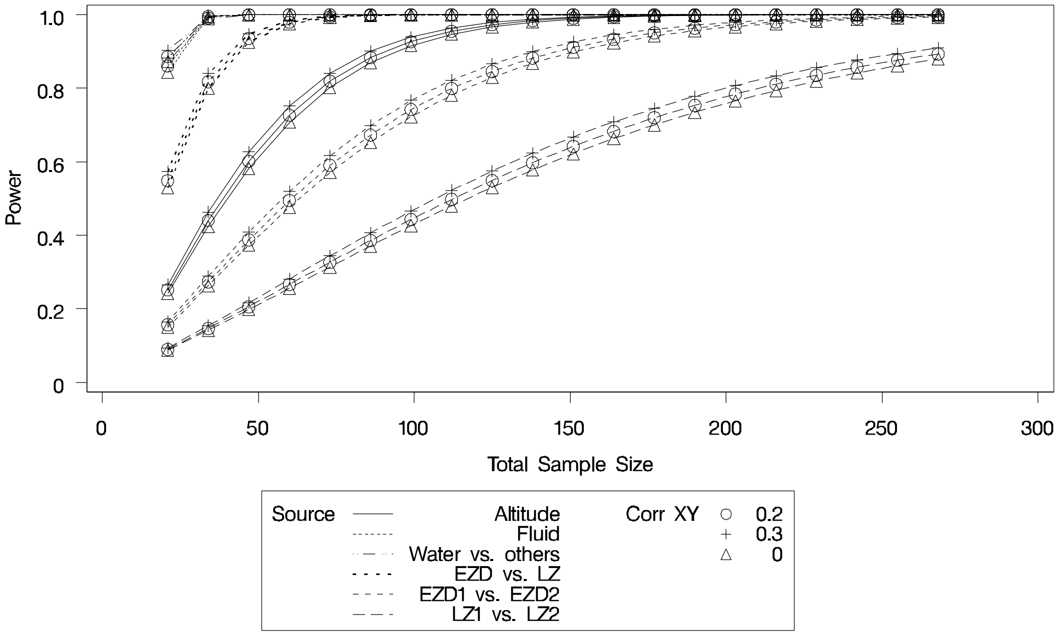
Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 21 to 275 that achieves 0.9 power for different comparisons). In the POWER statement, identify power as the result (POWER=.), and specify NTOTAL=21. Specify the PLOT statement with X=N to request a plot with sample size on the x-axis.

```
proc glmpower data=Fluids2 plotonly;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = 21
      power       = .;
   plot x=n min=21 max=275;
run;
```

The MAX=275 option in the PLOT statement sets the maximum sample size value. The MIN= option automatically defaults to the value of 21 from the NTOTAL= option in the POWER statement.

See Output 34.2.3 for the plot.

Output 34.2.3. Plot of Power versus Sample Size for Two-Way ANOVA Contrasts



Although Output 34.2.2 and Output 34.2.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment.

Output 34.2.2 reveals the range of required sample sizes for powers of interest, and Output 34.2.3 reveals the range of powers achieved for sample sizes of interest.

References

Castelloe, J.M. (2000), "Sample Size Computations and Power Analysis with the SAS ® System," *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference,* Paper 265-25, Cary, NC: SAS Institute Inc.

Castelloe, J.M. and O'Brien, R.G. (2001), "Power and Sample Size Determination for Linear Models," *Proceedings of the Twenty-sixth Annual SAS Users Group International Conference,* Paper 240-26. Cary, NC: SAS Institute Inc.

Hocking, R.R. (1985), *The Analysis of Linear Models*, Monterey, CA: Brooks/Cole Publishing Company.

Lenth, R.V. (2001), "Some Practical Guidelines for Effective Sample Size Determination," *The American Statistician*, 55, 187–193.

Muller, K.E. and Benignus, V.A. (1992), "Increasing Scientific Power with Statistical Power," *Neurotoxicology and Teratology*, 14, 211–219.

Muller, K.E., LaVange, L.M., Ramey, S.L. and Ramey, C.T. (1992), "Power Calculations for General Linear Multivariate Models Including Repeated Measures Applications," *Journal of the American Statistical Association*, 87 (420), 1209–1226.

Muller, K.E. and Peterson, B.L. (1984), "Practical Methods for Computing Power in Testing the Multivariate General Linear Hypothesis," *Computational Statistics & Data Analysis*, 2, 143–158.

O'Brien, R.G. and Muller, K.E. (1993), "Unified Power Analysis for *t*-Tests Through Multivariate Hypotheses," in *Applied Analysis of Variance in Behavioral Science*, ed. L.K. Edwards, New York: Marcel Dekker, Chapter 8, 297–344.

O'Brien, R.G. and Shieh, G. (1992), "Pragmatic, Unifying Algorithm Gives Power Probabilities for Common F Tests of the Multivariate General Linear Hypothesis." Poster presented at the American Statistical Association Meetings, Boston, Statistical Computing Section. Also, paper in review, downloadable in PDF form from www.bio.ri.ccf.org/UnifyPow.

Chapter 35
The INBREED Procedure

Chapter Contents

Chapter 35
The INBREED Procedure

Overview

The INBREED procedure calculates the covariance or inbreeding coefficients for a pedigree. PROC INBREED is unique in that it handles very large populations.

The INBREED procedure has two modes of operation. One mode carries out analysis on the assumption that all the individuals belong to the same generation. The other mode divides the population into nonoverlapping generations and analyzes each generation separately, assuming that the parents of individuals in the current generation are defined in the previous generation.

PROC INBREED also computes averages of the covariance or inbreeding coefficients within sex categories if the sex of individuals is known.

Getting Started

This section demonstrates how you can use the INBREED procedure to calculate the inbreeding or covariance coefficients for a pedigree, how you can control the analysis mode if the population consists of nonoverlapping generations, and how you can obtain averages within sex categories.

For you to use PROC INBREED effectively, your input data set must have a definite format. The following sections first introduce this format for a fictitious population and then demonstrate how you can analyze this population using the INBREED procedure.

The Format of the Input Data Set

The SAS data set used as input to the INBREED procedure must contain an observation for each individual. Each observation must include one variable identifying the individual and two variables identifying the individual's parents. Optionally, an observation can contain a known covariance coefficient and a character variable defining the gender of the individual.

For example, consider the following data:

```
data Population;
   input Individual $ Parent1 $ Parent2 $
         Covariance Sex $ Generation;
   datalines;
   MARK   GEORGE LISA    .    M  1
   KELLY  SCOTT  LISA    .    F  1
   MIKE   GEORGE AMY     .    M  1
   .      MARK   KELLY  0.50  .  1
   DAVID  MARK   KELLY   .    M  2
   MERLE  MIKE   JANE    .    F  2
   JIM    MARK   KELLY  0.50  M  2
   MARK   MIKE   KELLY   .    M  2
   ;
```

It is important to order the pedigree observations so that individuals are defined before they are used as parents of other individuals. The family relationships between individuals cannot be ascertained correctly unless you observe this ordering. Also, older individuals must precede younger ones. For example, 'MARK' appears as the first parent of 'DAVID' at observation 5; therefore, his observation needs to be defined prior to observation 5. Indeed, this is the case (see observation 1). Also, 'DAVID' is older than 'JIM', whose observation appears after the observation for 'DAVID', as is appropriate.

In populations with distinct, nonoverlapping generations, the older generation (parents) must precede the younger generation. For example, the individuals defined in Generation=1 appear as parents of individuals defined in Generation=2.

PROC INBREED produces warning messages when a parent cannot be found. For example, 'JANE' appears as the second parent of the individual 'MERLE' even though there are no previous observations defining her own parents. If the population is treated as an overlapping population, that is, if the generation grouping is ignored, then the procedure inserts an observation for 'JANE' with missing parents just before the sixth observation, which defines 'MERLE' as follows:

```
JANE   .      .       .    F  2
MERLE  MIKE   JANE    .    F  2
```

However, if generation grouping is taken into consideration, then 'JANE' is defined as the last observation in Generation=1, as follows:

```
MIKE   GEORGE AMY     .    M  1
JANE   .      .       .    F  1
```

In this latter case, however, the observation for 'JANE' is inserted after the computations are reported for the first generation. Therefore, she does not appear in the covariance/inbreeding matrix, even though her observation is used in computations for the second generation (see the example on page 1970).

If the data for an individual are duplicated, only the first occurrence of the data is used by the procedure, and a warning message is displayed to note the duplication. For example, individual 'MARK' is defined twice, at observations 1 and 8. If generation grouping is ignored, then this is an error and observation 8 is skipped. However, if the population is processed with respect to two distinct generations, then 'MARK' refers to two different individuals, one in Generation=1 and the other in Generation=2.

If a covariance is to be assigned between two individuals, then those individuals must be defined prior to the assignment observation. For example, a covariance of 0.50

can be assigned between ‘MARK’ and ‘KELLY’ since they are previously defined. Note that assignment statements must have different formats depending on whether the population is processed with respect to generations (see the “DATA= Data Set” section on page 1976 for further information). For example, while observation 4 is valid for nonoverlapping generations, it is invalid for a processing mode that ignores generation grouping. In this latter case, observation 7 indicates a valid assignment, and observation 4 is skipped.

The latest covariance specification between any given two individuals overrides the previous one between the same individuals.

Performing the Analysis

To compute the covariance coefficients for the overlapping generation mode, use the following statements:

```
proc inbreed data=Population covar matrix init=0.25;
run;
```

Here, the DATA= option names the SAS data set to be analyzed, and the COVAR and MATRIX options tell the procedure to output the covariance coefficients matrix. If you omit the COVAR option, the inbreeding coefficients are output instead of the covariance coefficients.

Note that the PROC INBREED statement also contains the INIT= option. This option gives an initial covariance between any individual and unknown individuals. For example, the covariance between any individual and ‘JANE’ would be 0.25, since ‘JANE’ is unknown, except when ‘JANE’ appears as a parent (see Figure 35.1).

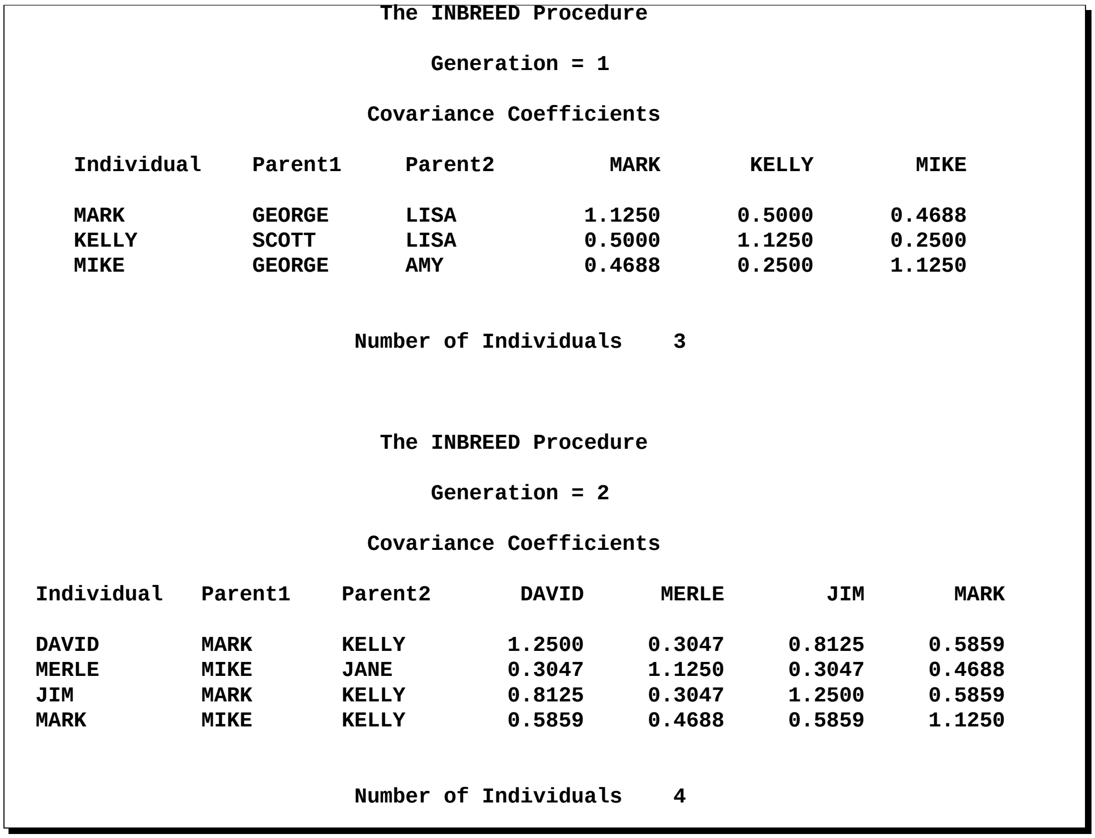
```
                              The INBREED Procedure

                             Covariance Coefficients

Individual  Parent1   Parent2     GEORGE       LISA       MARK      SCOTT      KELLY

GEORGE                            1.1250     0.2500     0.6875     0.2500     0.2500
LISA                              0.2500     1.1250     0.6875     0.2500     0.6875
MARK        GEORGE    LISA        0.6875     0.6875     1.1250     0.2500     0.5000
SCOTT                             0.2500     0.2500     0.2500     1.1250     0.6875
KELLY       SCOTT     LISA        0.2500     0.6875     0.5000     0.6875     1.1250
AMY                               0.2500     0.2500     0.2500     0.2500     0.2500
MIKE        GEORGE    AMY         0.6875     0.2500     0.4688     0.2500     0.2500
DAVID       MARK      KELLY       0.4688     0.6875     0.8125     0.4688     0.8125
JANE                              0.2500     0.2500     0.2500     0.2500     0.2500
MERLE       MIKE      JANE        0.4688     0.2500     0.3594     0.2500     0.2500
JIM         MARK      KELLY       0.4688     0.6875     0.8125     0.4688     0.8125

                             Covariance Coefficients

Individual  Parent1   Parent2        AMY       MIKE      DAVID       JANE      MERLE

GEORGE                            0.2500     0.6875     0.4688     0.2500     0.4688
LISA                              0.2500     0.2500     0.6875     0.2500     0.2500
MARK        GEORGE    LISA        0.2500     0.4688     0.8125     0.2500     0.3594
SCOTT                             0.2500     0.2500     0.4688     0.2500     0.2500
KELLY       SCOTT     LISA        0.2500     0.2500     0.8125     0.2500     0.2500
AMY                               1.1250     0.6875     0.2500     0.2500     0.4688
MIKE        GEORGE    AMY         0.6875     1.1250     0.3594     0.2500     0.6875
DAVID       MARK      KELLY       0.2500     0.3594     1.2500     0.2500     0.3047
JANE                              0.2500     0.2500     0.2500     1.1250     0.6875
MERLE       MIKE      JANE        0.4688     0.6875     0.3047     0.6875     1.1250
JIM         MARK      KELLY       0.2500     0.3594     0.8125     0.2500     0.3047

                             Covariance Coefficients

                    Individual  Parent1   Parent2        JIM

                    GEORGE                            0.4688
                    LISA                              0.6875
                    MARK        GEORGE    LISA        0.8125
                    SCOTT                             0.4688
                    KELLY       SCOTT     LISA        0.8125
                    AMY                               0.2500
                    MIKE        GEORGE    AMY         0.3594
                    DAVID       MARK      KELLY       0.8125
                    JANE                              0.2500
                    MERLE       MIKE      JANE        0.3047
                    JIM         MARK      KELLY       1.2500


                           Number of Individuals    11
```

Figure 35.1. Analysis for an Overlapping Population

In the previous example, PROC INBREED treats the population as a single generation. However, you may want to process the population with respect to distinct, nonoverlapping generations. To accomplish this, you need to identify the generation variable in a CLASS statement, as shown by the following statements.

```
proc inbreed data=Population covar matrix init=0.25;
   class Generation;
run;
```

Note that, in this case, the covariance matrix is displayed separately for each generation (see Figure 35.2).

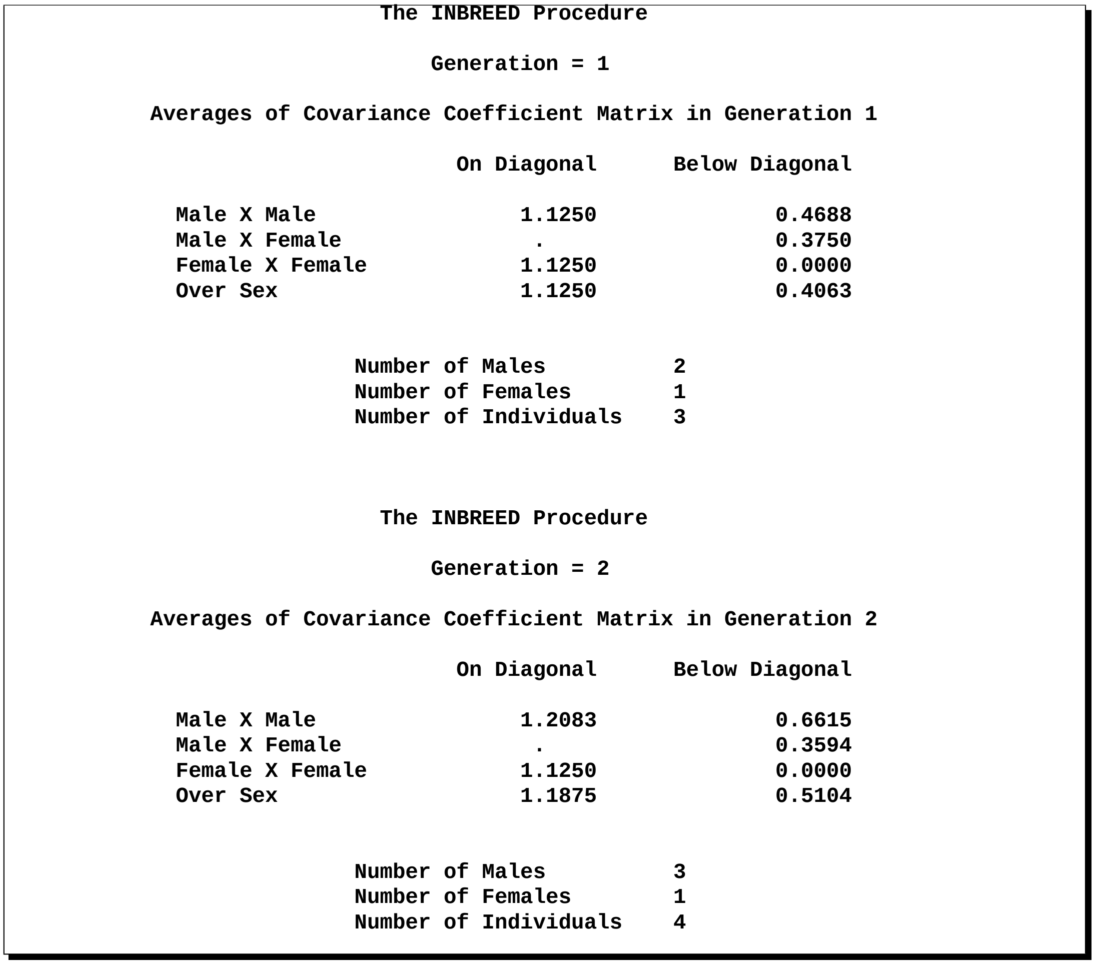


```
                              The INBREED Procedure

                                 Generation = 1

                             Covariance Coefficients

   Individual    Parent1     Parent2          MARK       KELLY        MIKE

   MARK          GEORGE      LISA           1.1250      0.5000      0.4688
   KELLY         SCOTT       LISA           0.5000      1.1250      0.2500
   MIKE          GEORGE      AMY            0.4688      0.2500      1.1250


                            Number of Individuals    3



                              The INBREED Procedure

                                 Generation = 2

                             Covariance Coefficients

Individual    Parent1    Parent2        DAVID      MERLE        JIM       MARK

DAVID         MARK       KELLY         1.2500     0.3047     0.8125     0.5859
MERLE         MIKE       JANE          0.3047     1.1250     0.3047     0.4688
JIM           MARK       KELLY         0.8125     0.3047     1.2500     0.5859
MARK          MIKE       KELLY         0.5859     0.4688     0.5859     1.1250


                            Number of Individuals    4
```

Figure 35.2. Analysis for a Nonoverlapping Population

You may also want to see covariance coefficient averages within sex categories. This is accomplished by indicating the variable defining the gender of individuals in a GENDER statement and by adding the AVERAGE option to the PROC INBREED statement. For example, the following statements produce the covariance coefficient averages shown in Figure 35.3.

```
proc inbreed data=Population covar average init=0.25;
   class Generation;
   gender Sex;
run;
```



```
                          The INBREED Procedure

                              Generation = 1

           Averages of Covariance Coefficient Matrix in Generation 1

                             On Diagonal      Below Diagonal

       Male X Male                1.1250              0.4688
       Male X Female                .                 0.3750
       Female X Female            1.1250              0.0000
       Over Sex                   1.1250              0.4063


                      Number of Males          2
                      Number of Females        1
                      Number of Individuals    3



                          The INBREED Procedure

                              Generation = 2

           Averages of Covariance Coefficient Matrix in Generation 2

                             On Diagonal      Below Diagonal

       Male X Male                1.2083              0.6615
       Male X Female                .                 0.3594
       Female X Female            1.1250              0.0000
       Over Sex                   1.1875              0.5104


                      Number of Males          3
                      Number of Females        1
                      Number of Individuals    4
```

Figure 35.3. Averages within Sex Categories for a Nonoverlapping Generation

Syntax

The following statements are available in PROC INBREED.

PROC INBREED < *options* > **;**
  BY *variables* **;**
  CLASS *variable* **;**
  GENDER *variable* **;**
  MATINGS *individual-list1 / mate-list* <, . . . > **;**
  VAR *variables* **;**

The PROC INBREED statement is required. Items within angle brackets (< >) are optional. The syntax of each statement is described in the following sections.

PROC INBREED Statement

PROC INBREED < *options* > **;**

You can specify the following options in the PROC INBREED statement.

AVERAGE
A

produces a table of averages of coefficients for each pedigree of offspring. The AVERAGE option is used together with the GENDER statement to average the inbreeding/covariance coefficients within sex categories.

COVAR
C

specifies that all coefficients output consist of covariance coefficients rather than inbreeding coefficients.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC INBREED. If you omit the DATA= option, the most recently created SAS data set is used.

IND
I

displays the individuals' inbreeding coefficients (diagonal of the inbreeding coefficients matrix) for each pedigree of offspring. If you also specify the COVAR option, the individuals' covariance coefficients (diagonal of the covariance coefficients matrix) are displayed.

INDL

displays individuals' coefficients for only the last generation of a multiparous population.

INIT=*cov*

specifies the covariance value *cov* if any of the parents are unknown; a value of 0 is assumed if you do not specify the INIT= option.

MATRIX
M

displays the inbreeding coefficient matrix for each pedigree of offspring. If you also specify the COVAR option, the covariance matrices are displayed instead of inbreeding coefficients matrices.

MATRIXL

displays coefficients for only the last generation of a multiparous population.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS).

For more information on ODS, see Chapter 14, "Using the Output Delivery System."

OUTCOV=*SAS-data-set*

names an output data set to contain the inbreeding coefficients. When the COVAR option is also specified, covariance estimates are output to the OUTCOV= data set instead of inbreeding coefficients.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC INBREED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input DATA= data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Use the SORT procedure with a similar BY statement to sort the data.
- Use the BY statement options NOTSORTED or DESCENDING in the BY statement for the INBREED procedure. As a cautionary note, the NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables), and these groups are not necessarily in alphabetical or increasing numeric order.
- Use the DATASETS procedure (in base SAS software) to create an index on the BY variables.

For more information on the BY statement, see the discussion in *SAS Language Reference: Concepts*.

CLASS Statement

CLASS *variable* **;**

To analyze the population within nonoverlapping generations, you must specify the variable that identifies generations in a CLASS statement. Values of the generation variable, called *generation numbers*, must be integers, but generations are assumed to occur in the order of their input in the input data set rather than in numerical order of the generation numbers. The name of an individual needs to be unique only within its generation.

When the MATRIXL option or the INDL option is specified, each generation requires a unique generation number in order for the specified option to work correctly. If generation numbers are not unique, all the generations with a generation number that is the same as the last generation's are output.

GENDER Statement

GENDER *variable* **;**

The GENDER statement specifies a variable that indicates the sex of the individuals. Values of the sex variable must be character beginning with ‘M’ or ‘F’, for male or female. The GENDER statement is needed only when you specify the AVERAGE option to average the inbreeding/covariance coefficients within sex categories or when you want to include a gender variable in the OUTCOV= data set.

PROC INBREED makes the following assumptions regarding the gender of individuals:

- The first parent is always assumed to be the male. See the “VAR Statement” section on page 1975.
- The second parent is always assumed to be the female. See the “VAR Statement” section on page 1975.
- If the gender of an individual is missing or invalid, this individual is assumed to be a female unless the population is overlapping and this individual appears as the first parent in a later observation.

Any contradictions to these rules are reported in the SAS log.

MATINGS Statement

MATINGS *individual-list1 / mate-list1* $<, \ldots,$*individual-listn / mate-listn* $>$**;**

You can specify the MATINGS statement with PROC INBREED to specify selected matings of individuals. Each individual given in *individual-list* is mated with each individual given in *mate-list*. You can write multiple mating specifications if you separate them by commas or asterisks. The procedure reports the inbreeding coefficients or covariances for each pair of mates. For example, you can use the following statement to specify the mating of an individual named ‘DAVID’ with an individual named ‘JANE’:

```
matings david / jane;
```

VAR Statement

VAR *individual parent1 parent2* $<$ *covariance* $>$ **;**

The VAR statement specifies three or four variables: the first variable contains an individual’s name, the second variable contains the name of the individual’s first parent, and the third variable contains the name of the individual’s second parent. An optional fourth variable assigns a known value to the covariance of the individual’s first and second parents in the current generation.

The first three variables in the VAR statement can be either numeric or character; however, only the first 12 characters of a character variable are recognized by the procedure. The fourth variable, if specified, must be numeric.

If you omit the VAR statement, then the procedure uses the first three unaddressed variables as the names of the individual and its parents. (Unaddressed variables are those that are not referenced in any other PROC INBREED statement.) If the input data set contains an unaddressed fourth variable, then it becomes the covariance variable.

Details

Missing Values

A missing value for a parent implies that the parent is unknown. Unknown parents are assumed to be unrelated and not inbred unless you specify the INIT= option (see the INIT= option on page 1973).

When the value of the variable identifying the individual is missing, the observation is not added to the list of individuals. However, for a multiparous population, an observation with a missing individual is valid and is used for assigning covariances.

Missing covariance values are determined from the INIT=*cov* option, if specified. Observations with missing generation variables are excluded.

If the gender of an individual is missing, it is determined from the order in which it is listed on the first observation defining its progeny for an overlapping population. If it appears as the first parent, it is set to 'M'; otherwise, it is set to 'F'. When the gender of an individual cannot be determined, it is assigned a default value of 'F'.

DATA= Data Set

Each observation in the input data set should contain necessary information such as the identification of an individual and the first and second parents of an individual. In addition, if a CLASS statement is specified, each observation should contain the generation identification; and, if a GENDER statement is specified, each observation should contain the gender of an individual. Optionally, each observation may also contain the covariance between the first and the second parents. Depending on how many statements are specified with the procedure, there should be enough variables in the input data set containing this information.

If you omit the VAR statement, then the procedure uses the first three *unaddressed variables* in the input data set as the names of the individual and his or her parents. Unaddressed variables in the input data set are those variables that are not referenced by the procedure in any other statements, such as CLASS, GENDER, or BY statements. If the input data set contains an unaddressed fourth variable, then the procedure uses it as the covariance variable.

If the individuals given by the variables associated with the first and second parents are not in the population, they are added to the population. However, if they are in the population, they must be defined prior to the observation that gives their progeny.

When there is a CLASS statement, the functions of defining new individuals and assigning covariances must be separated. This is necessary because the parents of any given individual are defined in the previous generation, while covariances are assigned between individuals in the current generation.

Therefore, there could be two types of observations for a multiparous population:

- one to define new individuals in the current generation whose parents have been defined in the previous generation, as in the following, where the missing value is for the covariance variable:

```
MARK   GEORGE LISA    .    M  1
KELLY  SCOTT  LISA    .    F  1
```

- one to assign covariances between two individuals in the current generation, as in the following, where the individual's name is missing, 'MARK' and 'KELLY' are in the current generation, and the covariance coefficient between these two individuals is 0.50:

```
.      MARK   KELLY  0.50  .  1
```

Note that the observations defining individuals must precede the observation assigning a covariance value between them. For example, if a covariance is to be assigned between 'MARK' and 'KELLY', then both of them should be defined prior to the assignment observation.

Computational Details

This section describes the rules that the INBREED procedure uses to compute the covariance and inbreeding coefficients. Each computational rule is explained by an example referring to the fictitious population introduced in the "Getting Started" section on page 1967.

Coancestry (or Kinship Coefficient)

To calculate the inbreeding coefficient and the covariance coefficients, use the degree of relationship by descent between the two parents, which is called *coancestry* or *kinship coefficient* (Falconer and Mackay 1996, p.85), or *coefficient of parentage* (Kempthorne 1957, p.73). Denote the coancestry between individuals X and Y by f_{XY}. For information on how to calculate the coancestries among a population, see the section "Calculation of Coancestry."

Covariance Coefficient (or Coefficient of Relationship)

The covariance coefficient between individuals X and Y is defined by

$$\mathrm{Cov}(\mathrm{X,Y}) = 2f_{\mathrm{XY}}$$

where f_{XY} is the coancestry between X and Y. The covariance coefficient is sometimes called the *coefficient of relationship* or the *theoretical correlation* (Falconer and Mackay 1996, p.153; Crow and Kimura 1970, p.134). If a covariance coefficient cannot be calculated from the individuals in the population, it is assigned to an initial value. The initial value is set to 0 if the INIT= option is not specified or to *cov* if INIT=*cov*. Therefore, the corresponding initial coancestry is set to 0 if the INIT= option is not specified or to $\frac{1}{2}cov$ if INIT=*cov*.

Inbreeding Coefficients

The inbreeding coefficient of an individual is the probability that the pair of alleles carried by the gametes that produced it are identical by descent (Falconer and Mackay 1996, Chapter 5; Kempthorne 1957, Chapter 5). For individual X, denote its inbreeding coefficient by F_{X}. The inbreeding coefficient of an individual is equal to the coancestry between its parents. For example, if X has parents A and B, then the inbreeding coefficient of X is

$$F_{\mathrm{X}} = f_{\mathrm{AB}}$$

Calculation of Coancestry

Given individuals X and Y, assume that X has parents A and B and that Y has parents C and D. For nonoverlapping generations, the basic rule to calculate the coancestry between X and Y is given by the following formula (Falconer and Mackay 1996, p.86):

$$f_{\mathrm{XY}} = \frac{1}{4}\left(f_{\mathrm{AC}} + f_{\mathrm{AD}} + f_{\mathrm{BC}} + f_{\mathrm{BD}}\right)$$

And the inbreeding coefficient for an offspring of X and Y, called Z, is the coancestry between X and Y:

$$F_{\mathrm{Z}} = f_{\mathrm{XY}}$$

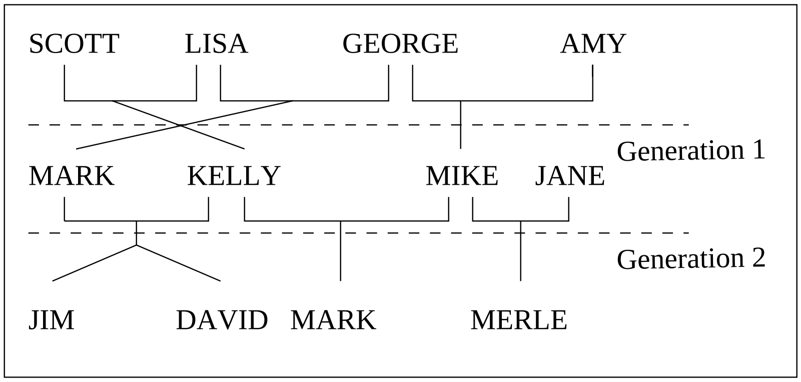


Figure 35.4. Inbreeding Relationship for Nonoverlapping Population

For example, in Figure 35.4, 'JIM' and 'MARK' from Generation 2 are progenies of 'MARK' and 'KELLY' and of 'MIKE' and 'KELLY' from Generation 1, respectively. The coancestry between 'JIM' and 'MARK' is

$$f_{\text{JIM,MARK}} = \frac{1}{4}\Big(f_{\text{MARK,MIKE}} + f_{\text{MARK, KELLY}} + f_{\text{KELLY, MIKE}} + f_{\text{KELLY, KELLY}}\Big)$$

From the covariance matrix for Generation=1 in Figure 35.2 (page 1971) and the relationship that coancestry is half of the covariance coefficient,

$$f_{\text{JIM, MARK}} = \frac{1}{4}\left(\frac{0.4688}{2} + \frac{0.5}{2} + \frac{0.25}{2} + \frac{1.125}{2}\right) = 0.29298$$

For overlapping generations, if X is older than Y, then the basic rule (on page 1978) can be simplified to

$$F_{\text{Z}} = f_{\text{XY}} = \frac{1}{2}\left(f_{\text{XC}} + f_{\text{XD}}\right)$$

That is, the coancestry between X and Y is the average of coancestries between older X with younger Y's parents. For example, in Figure 35.5, the coancestry between 'KELLY' and 'DAVID' is

$$f_{\text{KELLY,DAVID}} = \frac{1}{2}\left(f_{\text{KELLY,MARK}} + f_{\text{KELLY, KELLY}}\right)$$

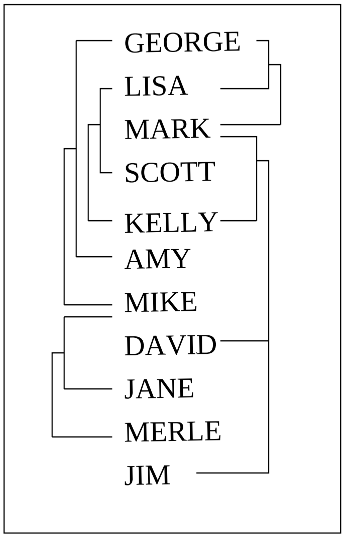


Figure 35.5. Inbreeding Relationship for Overlapping Population

This is so because 'KELLY' is defined before 'DAVID'; therefore, 'KELLY' is not younger than 'DAVID', and the parents of 'DAVID' are 'MARK' and 'KELLY'. The covariance coefficient values Cov(KELLY,MARK) and Cov(KELLY,KELLY) from the matrix in Figure 35.1 on page 1970 yield that the coancestry between ‘KELLY’ and ‘DAVID’ is

$$f_{\text{KELLY, DAVID}} = \frac{1}{2}\left(\frac{0.5}{2} + \frac{1.125}{2}\right) = 0.40625$$

The numerical values for some initial coancestries must be known in order to use these rule. Either the parents of the first generation have to be unrelated, with $f = 0$ if the INIT= option is not specified in the PROC statement, or their coancestries must have an initial value of $\frac{1}{2}cov$, where *cov* is set by the INIT= option. Then the subsequent coancestries among their progenies and the inbreeding coefficients of their progenies in the rest of the generations are calculated using these initial values.

Special rules need to be considered in the calculations of coancestries for the following cases.

Self-Mating

The coancestry for an individual X with itself, f_{XX}, is the inbreeding coefficient of a progeny that is produced by self-mating. The relationship between the inbreeding coefficient and the coancestry for self-mating is

$$f_{\text{XX}} = \frac{1}{2}\left(1 + F_{\text{X}}\right)$$

The inbreeding coefficient F_{X} can be replaced by the coancestry between X’s parents A and B, f_{AB}, if A and B are in the population:

$$f_{\text{XX}} = \frac{1}{2}\left(1 + f_{\text{AB}}\right)$$

If X’s parents are not in the population, then F_{X} is replaced by the initial value $\frac{1}{2}cov$ if *cov* is set by the INIT= option, or F_{X} is replaced by 0 if the INIT= option is not specified. For example, the coancestry of ‘JIM’ with himself is

$$f_{\text{JIM,JIM}} = \frac{1}{2}\left(1 + f_{\text{MARK, KELLY}}\right)$$

where ‘MARK’ and ‘KELLY’ are the parents of ‘JIM’. Since the covariance coefficient Cov(MARK,KELLY) is 0.5 in Figure 35.1 on page 1970 and also in the covariance matrix for GENDER=1 in Figure 35.2 on page 1971, the coancestry of ‘JIM’ with himself is

$$f_{\text{JIM,JIM}} = \frac{1}{2}\left(1 + \frac{0.5}{2}\right) = 0.625$$

When INIT=0.25, then the coancestry of ‘JANE’ with herself is

$$f_{\text{JANE,JANE}} = \frac{1}{2}\left(1 + \frac{0.25}{2}\right) = 0.5625$$

because ‘JANE’ is not an offspring in the population.

Offspring and Parent Mating

Assuming that X's parents are A and B, the coancestry between X and A is

$$f_{\mathrm{XA}} = \frac{1}{2}\left(f_{\mathrm{AB}} + f_{\mathrm{AA}}\right)$$

The inbreeding coefficient for an offspring of X and A, denoted by Z, is

$$F_{\mathrm{Z}} = f_{\mathrm{XA}} = \frac{1}{2}\left(f_{\mathrm{AB}} + f_{\mathrm{AA}}\right)$$

For example, 'MARK' is an offspring of 'GEORGE' and 'LISA', so the coancestry between 'MARK' and 'LISA' is

$$f_{\text{MARK, LISA}} = \frac{1}{2}\left(f_{\text{LISA,GEORGE}} + f_{\text{LISA, LISA}}\right)$$

From the covariance coefficient matrix in Figure 35.1 on page 1970, $f_{\text{LISA,GEORGE}} = 0.25/2 = 0.125$, $f_{\text{LISA,LISA}} = 1.125/2 = 0.5625$, so that

$$f_{\text{MARK, LISA}} = \frac{1}{2}(0.125 + 0.5625) = 0.34375$$

Thus, the inbreeding coefficient for an offspring of 'MARK' and 'LISA' is 0.34375.

Full Sibs Mating

This is a special case for the basic rule given at the beginning of the section "Calculation of Coancestry" on page 1978. If X and Y are full sibs with same parents A and B, then the coancestry between X and Y is

$$f_{\mathrm{XY}} = \frac{1}{4}\left(2f_{\mathrm{AB}} + f_{\mathrm{AA}} + f_{\mathrm{BB}}\right)$$

and the inbreeding coefficient for an offspring of A and B, denoted by Z, is

$$F_{\mathrm{Z}} = f_{\mathrm{XY}} = \frac{1}{4}\left(2f_{\mathrm{AB}} + f_{\mathrm{AA}} + f_{\mathrm{BB}}\right)$$

For example, 'DAVID' and 'JIM' are full sibs with parents 'MARK' and 'KELLY', so the coancestry between 'DAVID' and 'JIM' is

$$f_{\text{DAVID, JIM}} = \frac{1}{4}\left(2f_{\text{MARK,KELLY}} + f_{\text{MARK, MARK}} + f_{\text{KELLY, KELLY}}\right)$$

Since the coancestry is half of the covariance coefficient, from the covariance matrix in Figure 35.1 on page 1970,

$$f_{\text{DAVID,JIM}} = \frac{1}{4}\left(2 \times \frac{0.5}{2} + \frac{1.125}{2} + \frac{1.125}{2}\right) = 0.40625$$

Unknown or Missing Parents

When individuals or their parents are unknown in the population, their coancestries are assigned by the value $\frac{1}{2}cov$ if *cov* is set by the INIT= option or by the value 0 if the INIT= option is not specified. That is, if either A or B is unknown, then

$$f_{\text{AB}} = \frac{1}{2}cov$$

For example, 'JANE' is not in the population, and since 'JANE' is assumed to be defined just before the observation at which 'JANE' appears as a parent (that is, between observations 4 and 5), then 'JANE' is not older than 'SCOTT'. The coancestry between 'JANE' and 'SCOTT' is then obtained by using the simplified basic rule (see page 1979):

$$f_{\text{SCOTT,JANE}} = \frac{1}{2}\left(f_{\text{SCOTT},\cdot} + f_{\text{SCOTT},\cdot}\right)$$

Here, dots ($\cdot$) indicate JANE's unknown parents. Therefore, $f_{\text{SCOTT},\cdot}$ is replaced by $\frac{1}{2}cov$, where *cov* is set by the INIT= option. If INIT=0.25, then

$$f_{\text{SCOTT,JANE}} = \frac{1}{2}\left(\frac{0.25}{2} + \frac{0.25}{2}\right) = 0.125$$

For a more detailed discussion on the calculation of coancestries, inbreeding coefficients, and covariance coefficients, refer to Falconer and Mackay (1996), Kempthorne (1957), and Crow and Kimura (1970).

OUTCOV= Data Set

The OUTCOV= data set has the following variables:

- a list of BY variables, if there is a BY statement
- the generation variable, if there is a CLASS statement
- the gender variable, if there is a GENDER statement
- _Type_, a variable indicating the type of observation. The valid values of the _Type_ variable are 'COV' for covariance estimates and 'INBREED' for inbreeding coefficients.
- _Panel_, a variable indicating the panel number used when populations delimited by BY groups contain different numbers of individuals. If there are n individuals in the first BY group and if any subsequent BY group contains a larger population, then its covariance/inbreeding matrix is divided into panels, with each panel containing n columns of data. If you put these panels side by side in increasing _Panel_ number order, then you can reconstruct the covariance or inbreeding matrix.

- _Col_, a variable used to name columns of the inbreeding or covariance matrix. The values of this variable start with ‘COL’, followed by a number indicating the column number. The names of the individuals corresponding to any given column i can be found by reading the individual’s name across the row that has a _Col_ value of ‘COLi’. When the inbreeding or covariance matrix is divided into panels, all the rows repeat for the first n columns, all the rows repeat for the next n columns, and so on.
- the variable containing the names of the individuals, that is, the first variable listed in the VAR statement
- the variable containing the names of the first parents, that is, the second variable listed in the VAR statement
- the variable containing the names of the second parents, that is, the third variable listed in the VAR statement
- a list of covariance variables Col1-Col$n$, where n is the maximum number of individuals in the first population

The functions of the variables _Panel_ and _Col_ can best be demonstrated by an example. Assume that there are three individuals in the first BY group and that, in the current BY group (Byvar=2), there are five individuals with the following covariance matrix.

COV	**1**	**2**	**3**	**4**	**5**
1	Cov(1,1)	Cov(1,2)	Cov(1,3)	Cov(1,4)	Cov(1,5)
2	Cov(2,1)	Cov(2,2)	Cov(2,3)	Cov(2,4)	Cov(2,5)
3	Cov(3,1)	Cov(3,2)	Cov(3,3)	Cov(3,4)	Cov(3,5)
4	Cov(4,1)	Cov(4,2)	Cov(4,3)	Cov(4,4)	Cov(4,5)
5	Cov(5,1)	Cov(5,2)	Cov(5,3)	Cov(5,4)	Cov(5,5)
	Panel 1			Panel 2	

Then the OUTCOV= data set appears as follows.

Byvar	_Panel_	_Col_	Individual	Parent	Parent2	Col1	Col2	Col3
2	1	COL1	1			Cov(1,1)	Cov(1,2)	Cov(1,3)
2	1	COL2	2			Cov(2,1)	Cov(2,2)	Cov(2,3)
2	1	COL3	3			Cov(3,1)	Cov(3,2)	Cov(3,3)
2	1		4			Cov(4,1)	Cov(4,2)	Cov(4,3)
2	1		5			Cov(5,1)	Cov(5,2)	Cov(5,3)
2	2		1			Cov(1,4)	Cov(1,5)	.
2	2		2			Cov(2,4)	Cov(2,5)	.
2	2		3			Cov(3,4)	Cov(3,5)	.
2	2	COL1	4			Cov(4,4)	Cov(4,5)	.
2	2	COL2	5			Cov(5,4)	Cov(5,5)	.

Notice that the first three columns go to the first panel (_Panel_=1), and the remaining two go to the second panel (_Panel_=2). Therefore, in the first panel, ‘COL1’, ‘COL2’, and ‘COL3’ correspond to individuals 1, 2, and 3, respectively, while in the second panel, ‘COL1’ and ‘COL2’ correspond to individuals 4 and 5, respectively.

Displayed Output

The INBREED procedure can output either covariance coefficients or inbreeding coefficients. Note that the following items can be produced for each generation if generations do not overlap.

The output produced by PROC INBREED can be any or all of the following items:

- a matrix of coefficients
- coefficients of the individuals
- coefficients for selected matings

ODS Table Names

PROC INBREED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 35.1. ODS Tables Produced in PROC INBREED

ODS Table Name	**Description**	**Statement**	**Option**
AvgCovCoef	Averages of covariance coefficient matrix	GENDER	COVAR and AVERAGE
AvgInbreedingCoef	Averages of inbreeding coefficient matrix	GENDER	AVERAGE
CovarianceCoefficient	Covariance coefficient table	PROC	COVAR and MATRIX
InbreedingCoefficient	Inbreeding coefficient table	PROC	MATRIX
IndividualCovCoef	Covariance coefficients of individuals	PROC	IND and COVAR
IndividualInbreedingCoef	Inbreeding coefficients of individuals	PROC	IND
MatingCovCoef	Covariance coefficients of matings	MATINGS	COVAR
MatingInbreedingCoef	Inbreeding coefficients of matings	MATINGS	
NumberOfObservations	Number of observations	PROC	

Examples

Example 35.1. Monoecious Population Analysis

The following example shows a covariance analysis within nonoverlapping generations for a monoecious population. Parents of generation 1 are unknown and therefore assumed to be unrelated. The result appears in Output 35.1.1.

```
data Monoecious;
   input Generation Individual Parent1 Parent2 Covariance @@;
   datalines;
1 1 . .   .      1 2 . .   .       1 3 . .   .
2 1 1 1   .      2 2 1 2   .       2 3 2 3   .
3 1 1 2   .      3 2 1 3   .       3 3 2 1   .
3 4 1 3   .      3 . 2 3 0.50      3 . 4 3 1.135
;

title 'Inbreeding within Nonoverlapping Generations';
proc inbreed ind covar matrix data=Monoecious;
   class Generation;
run;
```

Output 35.1.1. Monoecious Population Analysis

```
               Inbreeding within Nonoverlapping Generations

                          The INBREED Procedure

                             Generation = 1

                         Covariance Coefficients

 Individual    Parent1    Parent2          1            2            3

 1                                    1.0000            .            .
 2                                         .       1.0000            .
 3                                         .            .       1.0000


               Inbreeding within Nonoverlapping Generations

                          The INBREED Procedure

                             Generation = 1

                  Covariance Coefficients of Individuals

           Individual    Parent1    Parent2     Coefficient

           1                                         1.0000
           2                                         1.0000
           3                                         1.0000


                       Number of Individuals    3
```

```
            Inbreeding within Nonoverlapping Generations

                       The INBREED Procedure

                           Generation = 2

                      Covariance Coefficients

Individual    Parent1    Parent2           1           2           3

1             1          1            1.5000      0.5000       .
2             1          2            0.5000      1.0000      0.2500
3             2          3             .          0.2500      1.0000



            Inbreeding within Nonoverlapping Generations

                       The INBREED Procedure

                           Generation = 2

               Covariance Coefficients of Individuals

          Individual    Parent1    Parent2     Coefficient

          1             1          1                1.5000
          2             1          2                1.0000
          3             2          3                1.0000


                     Number of Individuals    3
```

```
                   Inbreeding within Nonoverlapping Generations

                              The INBREED Procedure

                                 Generation = 3

                             Covariance Coefficients

Individual    Parent1    Parent2          1          2          3          4

1             1          2           1.2500     0.5625     0.8750     0.5625
2             1          3           0.5625     1.0000     1.1349     0.6250
3             2          1           0.8750     1.1349     1.2500     1.1349
4             1          3           0.5625     0.6250     1.1349     1.0000


                   Inbreeding within Nonoverlapping Generations

                              The INBREED Procedure

                                 Generation = 3

                      Covariance Coefficients of Individuals

                Individual    Parent1    Parent2     Coefficient

                1             1          2                1.2500
                2             1          3                1.0000
                3             2          1                1.2500
                4             1          3                1.0000


                            Number of Individuals    4
```

Note that, since the parents of the first generation are unknown, off-diagonal elements of the covariance matrix are all 0s and on-diagonal elements are all 1s. If there is an INIT=*cov* value, then the off-diagonal elements would be equal to *cov*, while on-diagonal elements would be equal to $1 + cov/2$.

In the third generation, individuals 2 and 4 are full siblings, so they belong to the same family. Since PROC INBREED computes covariance coefficients between families, the second and fourth columns of inbreeding coefficients are the same, except that their intersections with the second and fourth rows are reordered. Notice that, even though there is an observation to assign a covariance of 0.50 between individuals 2 and 3 in the third generation, the covariance between 2 and 3 is set to 1.135, the same value assigned between 4 and 3. This is because families get the same covariances, and later specifications override previous ones.

Example 35.2. Pedigree Analysis

In the following example, an inbreeding analysis is performed for a complicated pedigree. This analysis includes computing selective matings of some individuals and inbreeding coefficients of all individuals. Also, inbreeding coefficients are averaged within sex categories. The result appears in Output 35.2.1.

```
data Swine;
   input Swine_Number $ Sire $ Dam $ Sex $;
   datalines;
3504 2200 2501  M
3514 2521 3112  F
3519 2521 2501  F
2501 2200 3112  M
2789 3504 3514  F
3501 2521 3514  M
3712 3504 3514  F
3121 2200 3501  F
;

title 'Least Related Matings';
proc inbreed data=Swine ind average;
   var Swine_Number Sire Dam;
   matings 2501 / 3501 3504 ,
           3712 / 3121;
   gender Sex;
run;
```

Note the following from Output 35.2.1:

- Observation 4, which defines Swine_Number=2501, should precede the first and third observations where the progeny for 2501 are given. PROC INBREED ignores observation 4 since it is given out of order. As a result, the parents of 2501 are missing or unknown.
- The first column in the "Inbreeding Averages" table corresponds to the averages taken over the on-diagonal elements of the inbreeding coefficients matrix, and the second column gives averages over the off-diagonal elements.

Output 35.2.1. Pedigree Analysis

```
               Least Related Matings

               The INBREED Procedure

       Inbreeding Coefficients of Individuals

  Swine_
  Number      Sire        Dam          Coefficient

  2200                                       .
  2501                                       .
  3504        2200        2501               .
  2521                                       .
  3112                                       .
  3514        2521        3112               .
  3519        2521        2501               .
  2789        3504        3514               .
  3501        2521        3514            0.2500
  3712        3504        3514               .
  3121        2200        3501               .



               Least Related Matings

               The INBREED Procedure

         Inbreeding Coefficients of Matings

        Sire        Dam          Coefficient

        2501        3501               .
        2501        3504            0.2500
        3712        3121            0.1563


     Averages of Inbreeding Coefficient Matrix

                       Inbreeding        Coancestry

Male X Male                0.0625            0.1042
Male X Female               .                0.1362
Female X Female            0.0000            0.1324
Over Sex                   0.0227            0.1313


            Number of Males           4
            Number of Females         7
            Number of Individuals    11
```

Example 35.3. Pedigree Analysis with BY Groups

This example demonstrates the structure of the OUTCOV= data set created by PROC INBREED. Note that the first BY group has three individuals, while the second has five. Therefore, the covariance matrix for the second BY group is broken up into two panels, as shown in Output 35.3.1.

```
data Swine;
   input Group Swine_Number $ Sire $ Dam $ Sex $;
   datalines;
1  2789 3504 3514  F
```

```
2  2501 2200 3112  .
2  3504 2501 3782  M
;

proc inbreed data=Swine covar noprint outcov=Covariance
             init=0.4;
   var Swine_Number Sire Dam;
   gender Sex;
   by Group;
run;

title 'Printout of OUTCOV= data set';
proc print data=Covariance;
   format Col1-Col3 4.2;
run;
```

Output 35.3.1. Pedigree Analysis with BY Groups

```
                              Printout of OUTCOV= data set


                                                 Swine_
OBS  Group  Sex  _TYPE_  _PANEL_  _COL_   Number  Sire  Dam   COL1  COL2  COL3

  1    1     M    COV       1     COL1     3504               1.20  0.40  0.80
  2    1     F    COV       1     COL2     3514               0.40  1.20  0.80
  3    1     F    COV       1     COL3     2789   3504  3514  0.80  0.80  1.20
  4    2     M    COV       1     COL1     2200               1.20  0.40  0.80
  5    2     F    COV       1     COL2     3112               0.40  1.20  0.80
  6    2     M    COV       1     COL3     2501   2200  3112  0.80  0.80  1.20
  7    2     F    COV       1              3782               0.40  0.40  0.40
  8    2     M    COV       1              3504   2501  3782  0.60  0.60  0.80
  9    2     M    COV       2              2200               0.40  0.60    .
 10    2     F    COV       2              3112               0.40  0.60    .
 11    2     M    COV       2              2501   2200  3112  0.40  0.80    .
 12    2     F    COV       2     COL1     3782               1.20  0.80    .
 13    2     M    COV       2     COL2     3504   2501  3782  0.80  1.20    .
```

References

Crow, J.F. and Kimura, M. (1970), *An Introduction to Population Genetics Theory*, New York: Harper and Row.

Falconer, D. S. and Mackay, T. F. C. (1996), *Introduction to Quantitative Genetics*, Fourth Edition, London: Longman.

Kempthorne, O. (1957), *An Introduction to Genetic Statistics*, New York: John Wiley and Sons, Inc.

Chapter 36
The KDE Procedure

Chapter Contents

Chapter 36
The KDE Procedure

Overview

The KDE procedure performs either univariate or bivariate kernel density estimation. Statistical *density estimation* involves approximating a hypothesized probability density function from observed data. *Kernel density estimation* is a nonparametric technique for density estimation in which a known density function (the *kernel*) is averaged across the observed data points to create a smooth approximation. Refer to Silverman (1986) for a thorough review and discussion.

PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. PROC KDE outputs the kernel density estimate into a SAS data set, which you can then use with other procedures for plotting or analysis. PROC KDE also computes a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function.

Experimental graphics are now available with the KDE procedure. For more information, see the "ODS Graphics" section on page 2009.

Getting Started

The following example illustrates the basic features of PROC KDE. Assume that 1000 observations are simulated from a bivariate normal density with means $(0, 0)$, variances $(10, 10)$, and covariance 9. The SAS DATA step to accomplish this is as follows:

```
data bivnormal;
   seed = 1283470;
   do i = 1 to 1000;
      z1 = rannor(seed);
      z2 = rannor(seed);
      z3 = rannor(seed);
      x = 3*z1+z2;
      y = 3*z1+z3;
      output;
   end;
   drop seed;
run;
```

The following statements request a bivariate kernel density estimate for the variables x and y.

```
ods html;
ods graphics on;

proc kde data=bivnormal;
   bivar x y / plots=contour surface;
run;

ods graphics off;
ods html close;
```

A contour plot and a surface plot of the estimate are displayed in Figure 36.1 and Figure 36.2, respectively. Note that the correlation of 0.9 in the original data results in oval-shaped contours. These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental PLOTS= option in the BIVAR statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the "ODS Graphics" section on page 2009.

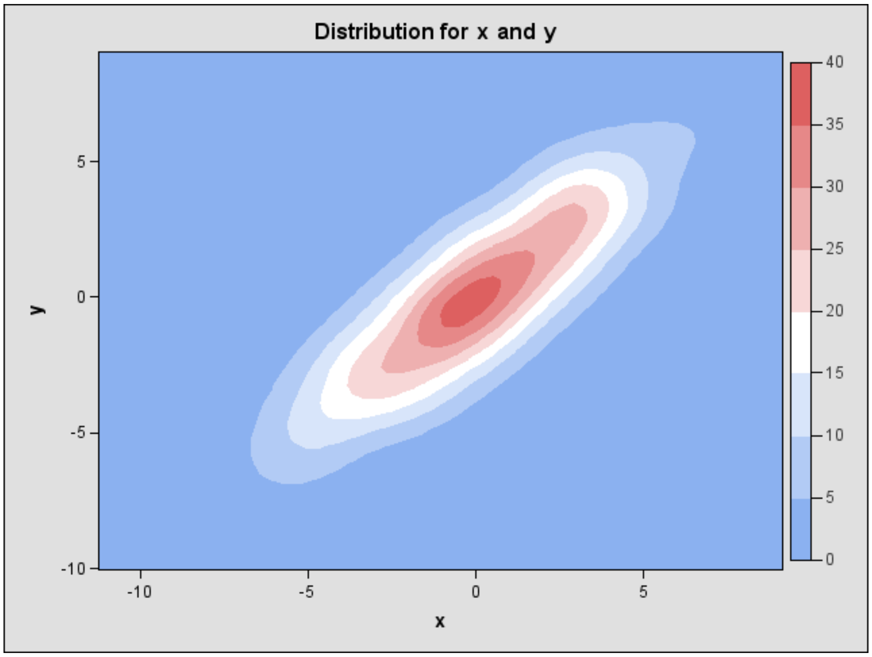


Figure 36.1. Contour Plot of Estimated Density (Experimental)

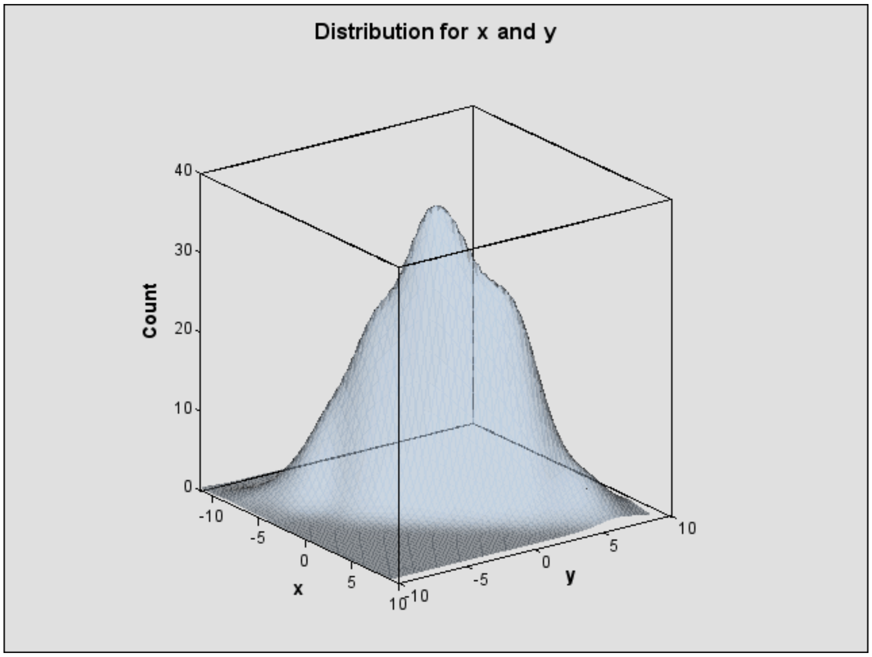


Figure 36.2. Surface Plot of Estimated Density (Experimental)

The default output tables for this analysis are as follows.

```
                                   Inputs

              Data Set                          WORK.BIVNORMAL
              Number of Observations Used       1000
              Variable 1                        x
              Variable 2                        y
              Bandwidth Method                  Simple Normal
                                                Reference
```

The “Inputs” table lists basic information about the density fit, including the input data set, the number of observations, and the variables. The bandwidth method is the technique used to select the amount of smoothing in the estimate. A simple normal reference rule is used for bivariate smoothing.

```
                                  Controls

                                                  x          y

                Grid Points                      60         60
                Lower Grid Limit             -11.25     -10.05
                Upper Grid Limit             9.1436     9.0341
                Bandwidth Multiplier              1          1
```

The "Controls" table lists the primary numbers controlling the kernel density fit. Here a 60×60 grid is fit to the entire range of the data, and no adjustment is made to the default bandwidth.

Syntax

You can use the following statements with the KDE procedure.

```
PROC KDE < options > ;
    BIVAR <(> variable < (v-options) > variable < (v-options) > <)>
        < . . . <(> variable < (v-options) > variable < (v-options) > <)>
        > < / options > ;
    UNIVAR variable < (v-options) > < . . . variable < (v-options) > >
        < / options > ;
    BY variables ;
    FREQ variable ;
    WEIGHT variable ;
```

The PROC KDE statement invokes the procedure. The BIVAR statement requests that one or more bivariate kernel density estimates be computed. The UNIVAR statement requests one or more univariate kernel density estimates.

PROC KDE Statement

PROC KDE < *options* >**;**

The PROC KDE statement invokes the procedure and specifies the input data set.

DATA=*SAS-data-set*

specifies the input SAS data set to be used by PROC KDE. The default is the most recently created data set.

Note: The following options, which were available in the PROC KDE statement in Version 8, are now obsolete. These options are now available in the UNIVAR and BIVAR statements.

Version 8	**SAS 9**	
PROC KDE option	**UNIVAR option**	**BIVAR option**
BWM=*numlist*	BWM=*number*	BWM=*number*
GRIDL=*numlist*	GRIDL=*number*	GRIDL=*number*
GRIDU=*numlist*	GRIDU=*number*	GRIDU=*number*
LEVELS		LEVELS
METHOD	METHOD	
NGRID=*numlist*	NGRID=*number*	NGRID=*number*
OUT	OUT	OUT
PERCENTILES	PERCENTILES	PERCENTILES
SJPIMAX	SJPIMAX	
SJPIMIN	SJPIMIN	
SJPINUM	SJPINUM	
SJPITOL	SJPITOL	

BIVAR Statement

The basic syntax for the BIVAR statement is

BIVAR *variable1 variable2* **;**

This statement requests a bivariate kernel density estimate for the variables variable1 and variable2.

The general form of this syntax is as follows:

BIVAR *<(> variable < (v-options) > variable < (v-options) > <)>*
< . . . <(> variable < (v-options) > variable < (v-options) > <)>
> < / options > **;**

The BIVAR statement lists variables in the input data set for which bivariate kernel density estimates are to be computed. You can specify a list of variables or a list of variable pairs, where each pair is enclosed in parentheses. If you specify a variable list, a kernel density estimate is computed for each distinct combination of two variables in the list. If you specify variable pairs, a kernel density estimate is computed for each pair.

For example, if you specify

```
bivar x y z;
```

then a bivariate kernel density estimate is computed for each of the pairs (x, y), (x, z), and (y, z). On the other hand, if you specify

```
bivar (x y) (z w);
```

then only two bivariate kernel density estimates are computed, one for (x, y) and one for (z, w).

You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* among the BIVAR statement options following a slash (/). Global *options* are applied to all the variables specified in the BIVAR statement. However, individual variable *v-options* override the global *options*.

Note: The VAR statement for PROC KDE in Version 8 is now obsolete. The VAR statement has been replaced by the UNIVAR and the BIVAR statements, which provide more control and flexibility for specifying the variables to be analyzed.

You can specify the following *options* in the BIVAR statement (as noted, some *options* can be used as *v-options*).

BIVSTATS

requests the covariance and correlation between the two variables.

BWM=*number*

specifies the bandwidth multiplier for the kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. You can specify BWM= as a *v-option*.

GRIDL=*number*

specifies the lower grid limit for the kernel density estimate. The default value equals the minimum observed values of the variables. You can specify GRIDL= as a *v-option*.

GRIDU=*number*

specifies the upper grid limit for the kernel density estimate. The default value equals the maximum observed values of the variables. You can specify GRIDU= as a *v-option*.

LEVELS
LEVELS=*numlist*

requests a table of levels for contours of the bivariate density. The contours are defined in such a way that the density has a constant level along each contour, and the volume enclosed by each contour corresponds to a specified percent. In other words, the contours correspond to slices or levels of the density surface taken along the density axis. You can use the LEVELS= option to specify the percents. By default, the percents are 1, 5, 10, 50, 90, 95, 99, and 100. The table also provides the minimum and maximum values for each contour along the directions of the two data variables.

NGRID=*number*
NG=*number*

specifies the number of grid points associated with a variable in the BIVAR statement. The default value is 60. You can specify NGRID= as a *v-option*.

OUT=*SAS-data-set*

specifies the output SAS data set containing the kernel density estimate. This output data set contains the following variables:

- var1, whose value is the name of the first variable in a bivariate kernel density estimate
- var2, whose value is the name of the second variable in a bivariate kernel density estimate
- value1, with values corresponding to grid coordinates for the first variable
- value2, with values corresponding to grid coordinates for the second variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES
PERCENTILES=*numlist*

lists percentiles to be computed for each BIVAR variable. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

UNISTATS

requests standard univariate statistics for each variable, as well as statistics associated with the density estimate.

UNIVAR Statement

UNIVAR *variable* < *(v-options)* > < . . . *variable* < *(v-options)* > >
< */ options* > **;**

The UNIVAR statement lists variables in the input data set for which univariate kernel density estimates are to be computed. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* among the UNIVAR statement options following a slash (/). Global *options* are applied to all the variables specified in the UNIVAR statement. However, individual variable *v-options* override the global *options*.

Note: The VAR statement for PROC KDE in Version 8 is now obsolete. The VAR statement has been replaced by the UNIVAR and the BIVAR statements, which provide more control and flexibility for specifying the variables to be analyzed.

You can specify the following *options* in the UNIVAR statement (as noted, some *options* can be used as *v-options*.)

BWM=*number*

specifies the bandwidth multiplier for the kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. You can specify BWM= as a *v-option*.

GRIDL=*number*

specifies the lower grid limit for the kernel density estimate. The default value equals the minimum observed values of the variables. You can specify GRIDL= as a *v-option*.

GRIDU=*number*

specifies the upper grid limit for the kernel density estimate. The default value equals the maximum observed values of the variables. You can specify GRIDU= as a *v-option*.

METHOD=SJPI | SNR | SROT | OS

specifies the method used to compute the bandwidth. Available methods are Sheather-Jones plug-in (SJPI), simple normal reference (SNR), Silverman's rule of thumb (SROT), and oversmoothed (OS). Refer to Jones, Marron, and Sheather (1996) for a description of each of these methods. SJPI is the default method.

NGRID=*number*
NG=*number*

specifies the number of grid points associated with a variable in the UNIVAR statement. The default value is 401. You can specify NGRID= as a *v-option*.

OUT=*SAS-data-set*

specifies the output SAS data set containing the kernel density estimate. This output data set contains the following variables:

- var, whose value is the name of the variable in the kernel density estimate
- value, with values corresponding to grid coordinates for the variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES
PERCENTILES=*numlist*

lists percentiles to be computed for each UNIVAR variable. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

SJPIMAX=*number*

specifies the maximum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is two times the oversmoothed estimate.

SJPIMIN=*number*

specifies the minimum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is the maximum value divided by 18.

SJPINUM=*number*

specifies the number of grid values used in determining the Sheather-Jones plug-in bandwidth. The default is 21.

SJPITOL=*number*

specifies the tolerance for termination of the bisection algorithm used in computing the Sheather-Jones plug-in bandwidth. The default value is 0.001.

UNISTATS

requests standard univariate statistics for each variable, as well as statistics associated with the density estimate.

Examples

Suppose you have the variables x1, x2, x3, x4 in the SAS data set MyData. You can request a kernel density estimate for each of these variables with the following statements.

```
proc kde data=MyData;
   univar x1 x2 x3 x4;
run;
```

You can also specify different bandwidths and other options for each variable. For example, the following statements request kernel density estimates using Silverman's rule of thumb (SROT) method for all variables. The option NGRID=200 applies to the variables x1, x3, and x4, but the *v-option* NGRID=100 is applied to x2. Bandwidth multipliers of 2 and 0.5 are specified for the variables x1 and x2, respectively.

```
proc kde data=MyData;
   univar x1 (bwm=2)
          x2 (bwm=0.5 ngrid=100)
          x3 x4 / ngrid=200 method=srot;
run;
```

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC KDE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the KDE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in the *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times. If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a variable that weights the observations in computing the kernel density estimate. Observations with higher weights have more influence in the computations. If an observation has a nonpositive or missing weight, then the entire observation is omitted from the analysis. You should be cautious in using data sets with extreme weights, as they can produce unreliable results.

Details

Computational Overview

The two main computational tasks of PROC KDE are automatic bandwidth selection and the construction of a kernel density estimate once a bandwidth has been selected. The primary computational tools used to accomplish these tasks are binning, convolutions, and the fast Fourier transform. The following sections provide analytical details on these topics, beginning with the density estimates themselves.

Kernel Density Estimates

A weighted univariate kernel density estimate involves a variable X and a weight variable W. Let $(X_i, W_i),\ i = 1, 2, \ldots, n$ denote a sample of X and W of size n. The weighted kernel density estimate of $f(x)$, the density of X, is as follows.

$$\hat{f}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} W_i \varphi_h(x - X_i)$$

where h is the bandwidth and

$$\varphi_h(x) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right)$$

is the standard normal density rescaled by the bandwidth. If $h \rightarrow 0$ and $nh \rightarrow \infty$, then the optimal bandwidth is

$$h_{\mathrm{AMISE}} = \left[\frac{1}{2\sqrt{\pi}n\int(f'')^2}\right]^{1/5}$$

This optimal value is unknown, and so approximations methods are required. For a derivation and discussion of these results, refer to Silverman (1986, Chapter 3) and Jones, Marron, and Sheather (1996).

For the bivariate case, let $\mathbf{X} = (X, Y)$ be a bivariate random element taking values in $\mathbb{R}^2$ with joint density function $f(x, y),\ (x, y) \in \mathbb{R}^2$, and let $\mathbf{X}_i = (X_i, Y_i),\ i =$

$1, 2, \ldots, n$ be a sample of size n drawn from this distribution. The kernel density estimate of $f(x, y)$ based on this sample is

$$\begin{aligned}\hat{f}(x,y) &= \frac{1}{n}\sum_{i=1}^{n}\varphi_{\mathbf{h}}(x - X_i, y - Y_i)\\ &= \frac{1}{nh_Xh_Y}\sum_{i=1}^{n}\varphi\left(\frac{x - X_i}{h_X}, \frac{y - Y_i}{h_Y}\right)\end{aligned}$$

where $(x, y) \in \mathbb{R}^2$, $h_X > 0$ and $h_Y > 0$ are the bandwidths and $\varphi_{\mathbf{h}}(x, y)$ is the rescaled normal density

$$\varphi_{\mathbf{h}}(x,y) = \frac{1}{h_Xh_Y}\varphi\left(\frac{x}{h_X}, \frac{y}{h_Y}\right)$$

where $\varphi(x, y)$ is the standard normal density function

$$\varphi(x,y) = \frac{1}{2\pi}\exp\left(-\frac{x^2 + y^2}{2}\right)$$

Under mild regularity assumptions about $f(x, y)$, the mean integrated squared error (MISE) of $\hat{f}(x, y)$ is

$$\begin{aligned}\mathrm{MISE}(h_X, h_Y) &= \mathrm{E}\int(\hat{f} - f)^2\\ &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy\\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy + O\left(h_X^4 + h_Y^4 + \frac{1}{nh_Xh_Y}\right)\end{aligned}$$

as $h_X \to 0$, $h_Y \to 0$ and $nh_Xh_Y \to \infty$.

Now set

$$\begin{aligned}\mathrm{AMISE}(h_X, h_Y) &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy\\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy\end{aligned}$$

which is the asymptotic mean integrated squared error (AMISE). For fixed n, this has minimum at $(h_{\mathrm{AMISE\_}X}, h_{\mathrm{AMISE\_}Y})$ defined as

$$h_{\mathrm{AMISE\_}X} = \left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{4n\pi}\right]^{1/6}\left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{\int(\frac{\partial^2 f}{\partial Y^2})^2}\right]^{2/3}$$

and

$$h_{\mathrm{AMISE\_}Y} = \left[\frac{\int (\frac{\partial^2 f}{\partial Y^2})^2}{4n\pi} \right]^{1/6} \left[\frac{\int (\frac{\partial^2 f}{\partial Y^2})^2}{\int (\frac{\partial^2 f}{\partial X^2})^2} \right]^{2/3}$$

These are the optimal asymptotic bandwidths in the sense that they minimize MISE. However, as in the univariate case, these expressions contain the second derivatives of the unknown density f being estimated, and so approximations are required. Refer to Wand and Jones (1993) for further details.

Binning

Binning, or assigning data to discrete categories, is an effective and fast method for large data sets (Fan and Marron 1994). When the sample size n is large, direct evaluation of the kernel estimate $\hat{f}$ at any point would involve n kernel evaluations, as shown in the preceding formulas. To evaluate the estimate at each point of a grid of size g would thus require ng kernel evaluations. When you use $g = 401$ in the univariate case or $g = 60 \times 60 = 3600$ in the bivariate case and $n \geq 1000$, the amount of computation can be prohibitively large. With binning, however, the computational order is reduced to g, resulting in a much quicker algorithm that is nearly as accurate as direct evaluation.

To bin a set of weighted univariate data $X_1, X_2, \ldots, X_n$ to a grid $x_1, x_2, \ldots, x_g$, simply assign each sample X_i, together with its weight W_i, to the nearest grid point x_j (also called the bin center). When binning is completed, each grid point x_i has an associated number c_i, which is the sum total of all the weights that correspond to sample points that have been assigned to x_i. These c_is are known as the *bin counts.*

This procedure replaces the data $(X_i, W_i),\ i = 1, 2, \ldots, n$ with the smaller set $(x_i, c_i),\ i = 1, 2, \ldots, g$, and the estimation is carried out with this new data. This is so-called *simple binning*, versus the finer *linear binning* described in Wand (1994). PROC KDE uses simple binning for the sake of faster and easier implementation. Also, it is assumed that the bin centers $x_1, x_2, \ldots, x_g$ are equally spaced and in increasing order. In addition, assume for notational convenience that $\sum_{i=1}^{n} W_i = n$ and, therefore, $\sum_{i=1}^{g} c_i = n$.

If you replace the data $(X_i, W_i), i = 1, 2, \ldots, n$ with $(x_i, c_i), i = 1, 2, \ldots, g$, the weighted estimator $\hat{f}$ then becomes

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{g} c_i \varphi_h (x - x_i)$$

with the same notation as used previously. To evaluate this estimator at the g points of the same grid vector $grid = (x_1, x_2, \ldots, x_g)'$ is to calculate

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h (x_i - x_j)$$

for $i = 1, 2, \ldots, g$. This can be rewritten as

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h(|i - j|\delta)$$

where $\delta = x_2 - x_1$ is the increment of the grid.

The same idea of binning works similarly with bivariate data, where you estimate $\hat{f}$ over the grid matrix $grid = grid_X \times grid_Y$ as follows.

$$grid = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{1,g_Y} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \ldots & \mathbf{x}_{2,g_Y} \\ \vdots & & & \\ \mathbf{x}_{g_X,1} & \mathbf{x}_{g_X,2} & \ldots & \mathbf{x}_{g_X,g_Y} \end{bmatrix}$$

where $\mathbf{x}_{i,j} = (x_i, y_i),\ i = 1, 2, \ldots, g_X,\ j = 1, 2, \ldots, g_Y$, and the estimates are

$$\hat{f}(\mathbf{x}_{i,j}) = \frac{1}{n} \sum_{k=1}^{g_X} \sum_{l=1}^{g_Y} c_{k,l} \varphi_{\mathbf{h}}(|i - k|\delta_X, |j - l|\delta_Y)$$

where $\delta_X = x_2 - x_1$ and $\delta_Y = y_2 - y_1$ are the increments of the grid.

Convolutions

The formulas for the binned estimator $\hat{f}$ in the previous subsection are in the form of a convolution product between two matrices, one of which contains the bin counts, the other of which contains the rescaled kernels evaluated at multiples of grid increments. This section defines these two matrices explicitly, and shows that $\hat{f}$ is their convolution.

Beginning with the weighted univariate case, define the following matrices:

$$\begin{aligned} K &= \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h((g-1)\delta))' \\ C &= (c_1, c_2, \ldots, c_g)' \end{aligned}$$

The first thing to note is that many terms in K are negligible. The term $\varphi_h(i\delta)$ is taken to be 0 when $|i\delta/h| \geq 5$, so you can define

$$l = \min(g - 1, \mathrm{floor}(5h/\delta))$$

as the maximum integer multiple of the grid increment to get nonzero evaluations of the rescaled kernel. Here $\mathrm{floor}(x)$ denotes the largest integer less than or equal to x.

Next, let p be the smallest power of 2 that is greater than $g + l + 1$,

$$p = 2^{\mathrm{ceil}(\log_2(g+l+1))}$$

where $\mathrm{ceil}(x)$ denotes the smallest integer greater than or equal to x.

Modify K as follows:

$$K = \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h(l\delta), \underbrace{0, \ldots, 0}_{p-2l-1}, \varphi_h(l\delta), \ldots, \varphi_h(\delta))'$$

Essentially, the negligible terms of K are omitted, and the rest are *symmetrized* (except for one term). The whole matrix is then padded to size $p \times 1$ with zeros in the middle. The dimension p is a highly composite number, that is, one that decomposes into many factors, leading to the most efficient fast Fourier transform operation (refer to Wand 1994).

The third operation is to pad the bin count matrix C with zeros to the same size as K:

$$C = (c_1, c_2, \ldots, c_g, \underbrace{0, \ldots, 0}_{p-g})'$$

The convolution $K * C$ is then a $p \times 1$ matrix, and the preceding formulas show that its first g entries are exactly the estimates $\hat{f}(x_i),\ i = 1, 2, \ldots, g$.

For bivariate smoothing, the matrix K is defined similarly as

$$K = \begin{bmatrix} \kappa_{0,0} & \kappa_{0,1} & \ldots & \kappa_{0,l_Y} & \mathbf{0} & \kappa_{0,l_Y} & \ldots & \kappa_{0,1} \\ \kappa_{1,0} & \kappa_{1,1} & \ldots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \ldots & \kappa_{1,1} \\ \vdots & & & & & & & \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \ldots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \ldots & \kappa_{l_X,1} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \ldots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \ldots & \kappa_{l_X,1} \\ \vdots & & & & & & & \\ \kappa_{1,0} & \kappa_{1,1} & \ldots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \ldots & \kappa_{1,1} \end{bmatrix}_{p_X \times p_Y}$$

where $l_X = \min(g_X - 1, \mathrm{floor}(5h_X/\delta_X)),\ p_X = 2^{\mathrm{ceil}(\log_2(g_X + l_X + 1))}$, and so forth, and $\kappa_{i,j} = \frac{1}{n}\varphi_{\mathbf{h}}(i\delta_X, j\delta_Y)\ i = 0, 1, \ldots, l_X,\ j = 0, 1, \ldots, l_Y$.

The bin count matrix C is defined as

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \ldots & c_{1,g_Y} & 0 & \ldots & 0 \\ c_{2,1} & c_{2,2} & \ldots & c_{2,g_Y} & 0 & \ldots & 0 \\ \vdots & & & & & & \\ c_{g_X,1} & c_{g_X,2} & \ldots & c_{g_X,g_Y} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & & & & & & \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \end{bmatrix}_{p_X \times p_Y}$$

As with the univariate case, the $g_X \times g_Y$ upper-left corner of the convolution $K * C$ is the matrix of the estimates $\hat{f}(grid)$.

Most of the results in this subsection are found in Wand (1994).

Fast Fourier Transform

As shown in the last subsection, kernel density estimates can be expressed as a submatrix of a certain convolution. The fast Fourier transform (FFT) is a computationally effective method for computing such convolutions. For a reference on this material, refer to Press et al. (1988).

The *discrete Fourier transform* of a complex vector $\mathbf{z} = (z_0, \ldots, z_{N-1})$ is the vector $\mathbf{Z} = (Z_0, \ldots, Z_{N-1})$, where

$$Z_j = \sum_{l=0}^{N-1} z_l e^{2\pi i l j/N}, \quad j = 0, \ldots, N-1$$

and i is the square root of -1. The vector $\mathbf{z}$ can be recovered from $\mathbf{Z}$ by applying the *inverse discrete Fourier transform* formula

$$z_l = N^{-1} \sum_{j=0}^{N-1} Z_j e^{-2\pi i l j/N}, \quad l = 0, \ldots, N-1$$

Discrete Fourier transforms and their inverses can be computed quickly using the FFT algorithm, especially when N is *highly composite*; that is, it can be decomposed into many factors, such as a power of 2. By the *Discrete Convolution Theorem*, the convolution of two vectors is the inverse Fourier transform of the element-by-element product of their Fourier transforms. This, however, requires certain periodicity assumptions, which explains why the vectors K and C require zero-padding. This is to avoid *wrap-around* effects (refer to Press et al. 1988, pp. 410–411). The vector K is actually mirror-imaged so that the convolution of C and K will be the vector of binned estimates. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of K and C, then the first g elements of S are the estimates.

The bivariate Fourier transform of an $N_1 \times N_2$ complex matrix having (l_1+1, l_2+1) entry equal to $z_{l_1 l_2}$ is the $N_1 \times N_2$ matrix with (j_1+1, j_2+1) entry given by

$$Z_{j_1 j_2} = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} z_{l_1 l_2} e^{2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

and the formula of the inverse is

$$z_{l_1 l_2} = (N_1 N_2)^{-1} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} Z_{j_1 j_2} e^{-2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

The same Discrete Convolution Theorem applies, and zero-padding is needed for matrices C and K. In the case of K, the matrix is mirror-imaged twice. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of K and C, then the upper-left $g_X \times g_Y$ corner of S contains the estimates.

Bandwidth Selection

Several different bandwidth selection methods are available in PROC KDE in the univariate case. Following the recommendations of Jones, Marron, and Sheather (1996), the default method follows a plug-in formula of Sheather and Jones.

This method solves the fixed-point equation

$$h = \left[\frac{R(\varphi)}{nR\left(\hat{f}''_{g(h)}\right)\left(\int x^2\varphi(x)dx\right)^2} \right]^{1/5}$$

where $R(\varphi) = \int \varphi^2(x)dx$.

PROC KDE solves this equation by first evaluating it on a grid of values spaced equally on a log scale. The largest two values from this grid that bound a solution are then used as starting values for a bisection algorithm.

The simple normal reference rule works by assuming $\hat{f}$ is Gaussian in the preceding fixed-point equation. This results in

$$h = \hat{\sigma}[4/(3n)]^{1/5}$$

where $\hat{\sigma}$ is the sample standard deviation.

Silverman's rule of thumb (Silverman 1986, section 3.4.2) is computed as

$$h = 0.9\min[\hat{\sigma}, (Q_3 - Q_1)/1.34]n^{-1/5}$$

where Q_3 and Q_1 are the third and first sample quartiles, respectively.

The oversmoothed bandwidth is computed as

$$h = 3\hat{\sigma}[1/(70\sqrt{\pi}n)]^{1/5}$$

When you specify a WEIGHT variable, PROC KDE uses weighted versions of Q_3, Q_1, and $\hat{\sigma}$ in the preceding expressions. The weighted quartiles are computed as weighted order statistics, and the weighted variance takes the form

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^n W_i(X_i - \bar{X})^2}{\sum_{i=1}^n W_i}$$

where $\bar{X} = (\sum_{i=1}^n W_iX_i)/(\sum_{i=1}^n W_i)$ is the weighted sample mean.

For the bivariate case, Wand and Jones (1993) note that automatic bandwidth selection is both difficult and computationally expensive. Their study of various ways of specifying a bandwidth matrix also shows that using two bandwidths, one in each coordinate's direction, is often adequate. PROC KDE enables you to adjust the two bandwidths by specifying a multiplier for the default bandwidths recommended by Bowman and Foster (1993):

$$\begin{aligned} h_X &= \hat{\sigma}_X n^{-1/6} \\ h_Y &= \hat{\sigma}_Y n^{-1/6} \end{aligned}$$

Here $\hat{\sigma}_X$ and $\hat{\sigma}_Y$ are the sample standard deviations of X and Y, respectively. These are the optimal bandwidths for two independent normal variables that have the same variances as X and Y. They are, therefore, conservative in the sense that they tend to oversmooth the surface.

You can specify the BWM= option to adjust the aforementioned bandwidths to provide the appropriate amount of smoothing for your application.

ODS Table Names

PROC KDE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 36.1. ODS Tables Produced in PROC KDE

ODS Table Name	Description	Statement	Option
BivariateStatistics	Bivariate statistics	BIVAR	BIVSTATS
Controls	Control variables	default	
Inputs	Input information	default	
Levels	Levels of density estimate	BIVAR	LEVELS
Percentiles	Percentiles of data	BIVAR / UNIVAR	PERCENTILES
UnivariateStatistics	Basic statistics	BIVAR / UNIVAR	UNISTATS

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the KDE procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the following options. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Bivariate Plots

You can specify the PLOTS= option in the BIVAR statement to request graphical displays of bivariate kernel density estimates.

PLOTS= *option1* < *option2 . . .* >

requests one or more plots of the bivariate kernel density estimate. The following table shows the available plot *options*.

Option	**Plot Description**
ALL	all available displays
CONTOUR	contour plot of bivariate density estimate
CONTOURSCATTER	contour plot of bivariate density estimate overlaid with scatter plot of data
HISTOGRAM	bivariate histogram of data
HISTSURFACE	bivariate histogram overlaid with bivariate kernel density estimate
SCATTER	scatter plot of data
SURFACE	surface plot of bivariate kernel density estimate

By default, if you enable ODS graphics and you do not specify the PLOTS= option, then the BIVAR statement creates a contour plot. If you specify the PLOTS= option, you get only the requested plots.

Univariate Plots

You can specify the PLOTS= option in the UNIVAR statement to request graphical displays of univariate kernel density estimates.

PLOTS= *option1* < *option2 . . .* >

requests one or more plots of the univariate kernel density estimate. The following table shows the available plot *options*.

Option	**Plot Description**
DENSITY	univariate kernel density estimate curve
HISTDENSITY	univariate histogram of data overlaid with kernel density estimate curve
HISTOGRAM	univariate histogram of data

By default, if you enable ODS graphics and you do not specify the PLOTS= option, then the UNIVAR statement creates a histogram overlaid with a kernel density estimate. If you specify the PLOTS= option, you get only the requested plots.

ODS Graph Names

PROC KDE assigns a name to each graph it creates using the Output Delivery System (ODS). You can use these names to reference the graphs when using ODS. The names are listed in Table 36.2.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 36.2. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 36.2. ODS Graphics Produced by PROC KDE

ODS Graph Name	Plot Description	Statement	PLOTS= Option
BivariateHistogram	Bivariate histogram of data	BIVAR	HISTOGRAM
Contour	Contour plot of bivariate kernel density estimate	BIVAR	CONTOUR
ContourScatter	Contour plot of bivariate kernel density estimate overlaid with scatter plot	BIVAR	CONTOURSCATTER
Density	Univariate kernel density estimate curve	UNIVAR	DENSITY
HistDensity	Univariate histogram overlaid with kernel density estimate curve	UNIVAR	HISTDENSITY
Histogram	Univariate histogram of data	UNIVAR	HISTOGRAM
HistSurface	Bivariate histogram overlaid with surface plot of bivariate kernel density estimate	BIVAR	HISTSURFACE
ScatterPlot	Scatter plot of data	BIVAR	SCATTER
SurfacePlot	Surface plot of bivariate kernel density estimate	BIVAR	SURFACE

Binning of Bivariate Histogram

Let $(X_i, Y_i),\ i = 1, 2, \ldots, n$ be a sample of size n drawn from a bivariate distribution. For the marginal distribution of $X_i,\ i = 1, 2, \ldots, n$, the number of bins ($\text{Nbins}_X$) in the bivariate histogram is calculated according to the formula

$$\text{Nbins}_X = \text{ceil}\left(\text{range}_X/\text{width}_X\right)$$

where $\text{ceil}(x)$ denotes the smallest integer greater than or equal to x,

$$\text{range}_X = \max_{1\le i\le n}(X_i) - \min_{1\le i\le n}(X_i)$$

and the optimal bin width is obtained, following Scott (1992, p. 84), as

$$\text{width}_X = 3.504\,\hat{\sigma}_X(1-\hat{\rho}^2)^{3/8}n^{-1/4}$$

Here, $\hat{\sigma}_X$ and $\hat{\rho}$ are the sample variance and the sample correlation coefficient, respectively. When you specify a WEIGHT variable, PROC KDE uses weighted versions of $\hat{\sigma}_X$ and $\hat{\rho}$ in the preceding expressions.

Similar formulas are used to compute the number of bins for the marginal distribution of $Y_i,\ i = 1, 2, \ldots, n$. Further details can be found in Scott (1992).

Notice that if $|\hat{\rho}| > 0.99$, then Nbins_X is calculated as in the univariate case (see Terrell and Scott 1985). In this case $\text{Nbins}_Y = \text{Nbins}_X$.

Examples

Example 36.1. Computing a Basic Kernel Density Estimate

The following example illustrates the basic functionality of the UNIVAR statement. The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable length in a SAS data set named channel; refer to the file kdex2.sas in the SAS Sample Library.

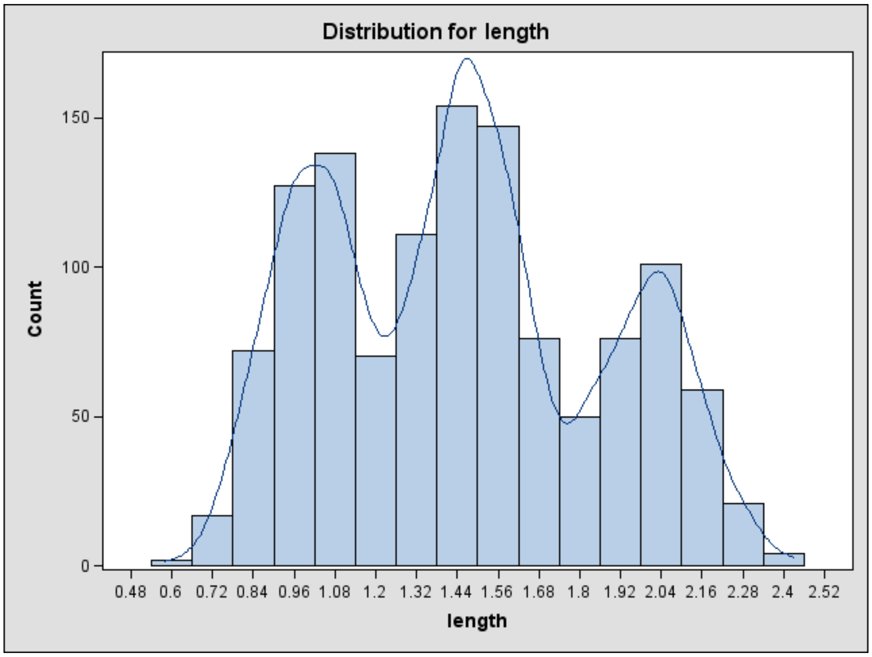
```
data channel;
   input length @@;
datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15

  ...

1.80 2.35 2.23 1.96 2.16 2.08 2.06 2.03 2.18 1.83
2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

The following statements request a kernel density estimate of the variable length.

```
ods html;
ods graphics on;

proc kde data=channel;
   univar length;
run;

ods graphics off;
ods html close;
```

You can see a histogram with an overlaid kernel density estimate in Output 36.1.1. This graph is requested by specifying the experimental ODS GRAPHICS statement prior to the PROC KDE statements. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the KDE procedure, see the “ODS Graphics” section on page 2009.

Output 36.1.1. Histogram with Overlaid Kernel Density Estimate (Experimental)



The default output tables for this analysis are the “Inputs” and the “Controls” tables.

```
                                  Inputs

              Data Set                            WORK.CHANNEL
              Number of Observations Used         1225
              Variable                            length
              Bandwidth Method                    Sheather-Jones
                                                  Plug In
```

The “Inputs” table lists basic information about the density fit, including the input data set, the number of observations used, the variable used, and the bandwidth method. The default bandwidth method is the Sheather-Jones plug-in.

```
                                 Controls

                                                length

                      Grid Points                  401
                      Lower Grid Limit            0.58
                      Upper Grid Limit            2.43
                      Bandwidth Multiplier           1
```

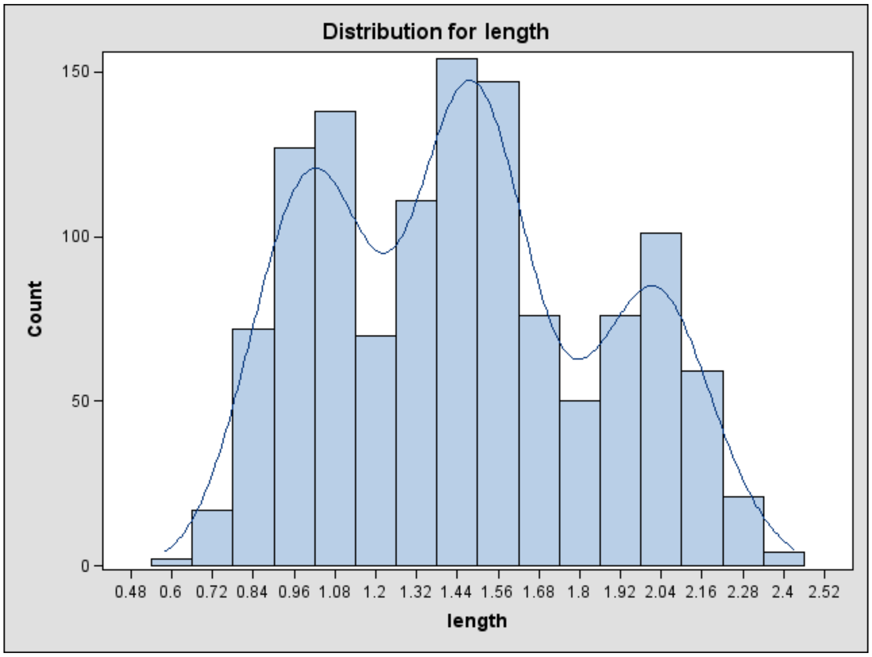
The “Controls” table lists the primary numbers controlling the kernel density fit. Here the default number of grid points is used and no adjustment is made to the default bandwidth.

Example 36.2. Changing the Bandwidth

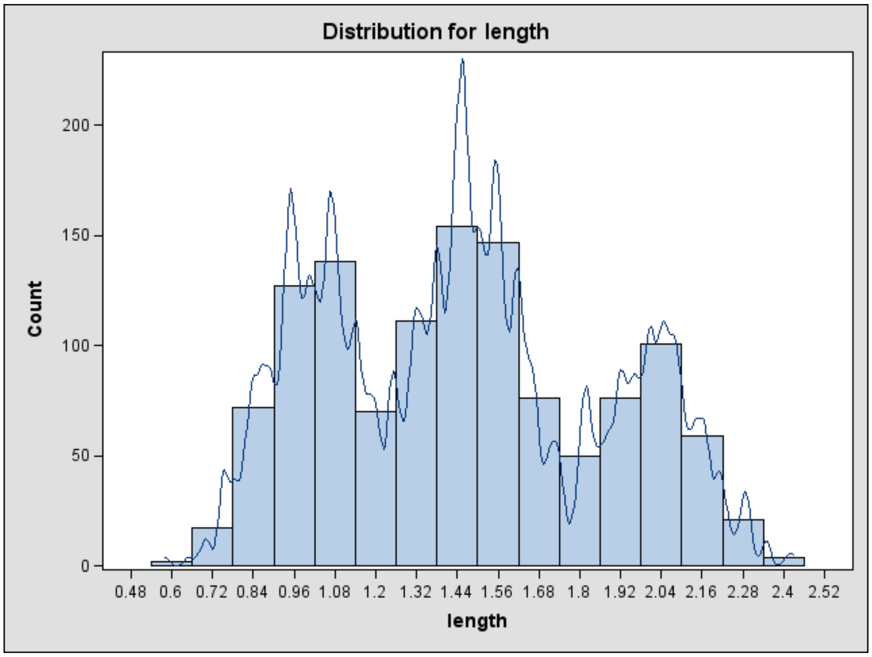
Continuing with the previous example, you can specify different bandwidth multipliers that determine the smoothness of the kernel density estimate. The following statements show kernel density estimates for the variable length by specifying two different bandwidth multipliers with the BWM= option. Output 36.2.1 shows an oversmoothed estimate because the bandwidth multiplier is 2. Output 36.2.2 is created by specifying BWM=0.25, so it is an undersmoothed estimate.

```
ods html;
ods graphics on;

proc kde data=channel;
   univar length(bwm=2) length(bwm=0.25);
run;

ods graphics off;
ods html close;
```

Output 36.2.1. Histogram with Oversmoothed Kernel Density Estimate (Experimental)



Output 36.2.2. Histogram with Undersmoothed Kernel Density Estimate (Experimental)



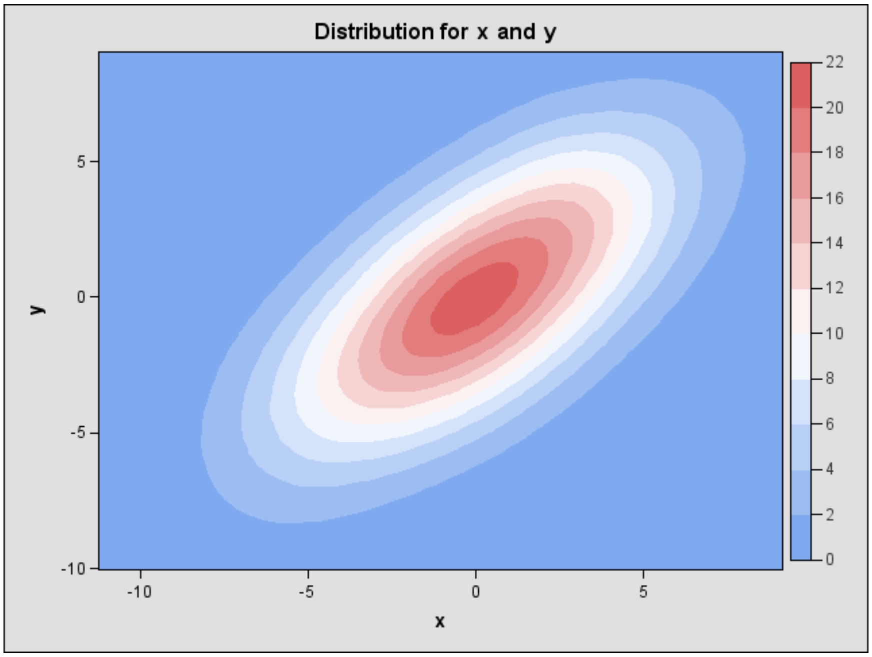
Example 36.3. Changing the Bandwidth (Bivariate)

Recall the analysis from the "Getting Started" section on page 1993. Suppose that you would like a slightly smoother estimate. You could then rerun the analysis with a larger bandwidth:

```
ods html;
ods graphics on;

proc kde data=bivnormal;
   bivar x y / bwm=2;
run;

ods graphics off;
ods html close;
```

The BWM= option requests bandwidth multipliers of 2 for both x and y. By specifying the experimental ODS GRAPHICS statement you can visualize the results of this fit immediately in a contour plot, as shown in Output 36.3.1.

Output 36.3.1. Contour Plot of Estimated Density with Additional Smoothing (Experimental)

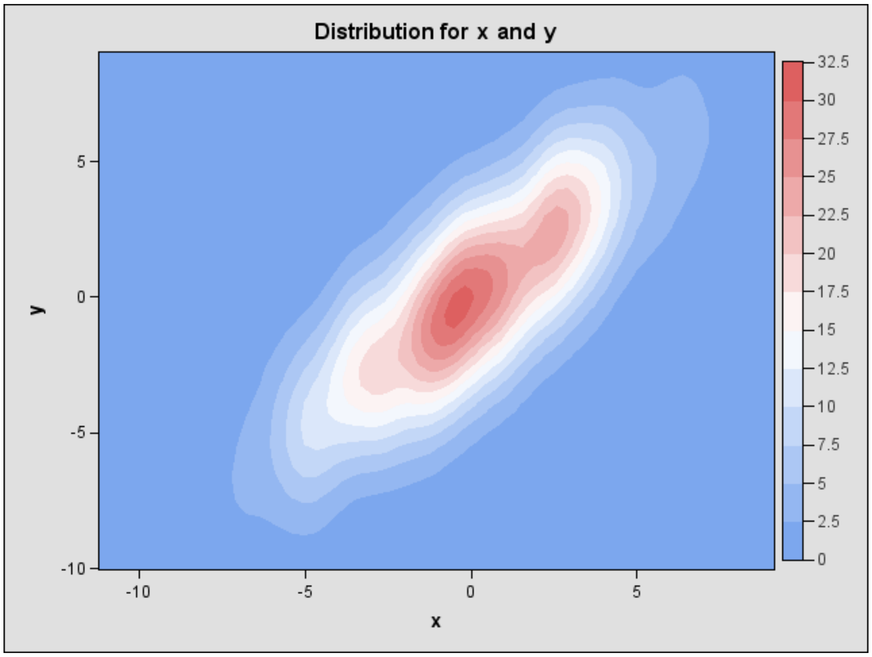


Multiple Bandwidths

You can also specify multiple bandwidths with only one run of the KDE procedure. Notice that by specifying pairs of variables inside parentheses, a kernel density estimate is computed for each pair. In the following statements the first kernel density is computed with the default bandwidth, but the second specifies a bandwidth multiplier of 0.5 for the variable x and a multiplier of 2 for the variable y. The effect of the latter options is shown in Output 36.3.2.

```
ods html;
ods graphics on;

proc kde data=bivnormal;
   bivar (x y)
         (x (bwm=0.5) y (bwm=2));
run;

ods graphics off;
ods html close;
```

Output 36.3.2. Contour Plot of Estimated Density with Different Smoothing for x and y (Experimental)



Example 36.4. Requesting Additional Output Tables

The following example illustrates how to request output tables with summary statistics in addition to the default output tables.

Using the same data as in the "Getting Started" section on page 1993, the following statements request univariate and bivariate summary statistics, percentiles, and levels of the kernel density estimate.

```
proc kde data=bivnormal;
   bivar x y / bivstats levels percentiles unistats;
run;
```

```
                         The KDE Procedure

                       Univariate Statistics

                                           x          y

                 Mean                  -0.075     -0.070
                 Variance                9.73       9.93
                 Standard Deviation      3.12       3.15
                 Range                  20.39      19.09
                 Interquartile Range     4.46       4.51
                 Bandwidth               0.99       1.00
```

The “Univariate Statistics” table contains standard univariate statistics for each variable, as well as statistics associated with the density estimate. Note that the estimated variances for both x and y are fairly close to the true values of 10.

```
                         Bivariate Statistics

                    Covariance                 8.88
                    Correlation                0.90
```

The “Bivariate Statistics” table lists the covariance and correlation between the two variables. Note that the estimated correlation is equal to its true value to two decimal places.

```
                              Percentiles

                                   x         y

                         0.5     -7.71     -8.44
                         1.0     -7.08     -7.46
                         2.5     -6.17     -6.31
                         5.0     -5.28     -5.23
                        10.0     -4.18     -4.11
                        25.0     -2.24     -2.30
                        50.0     -0.11    -0.058
                        75.0      2.22      2.21
                        90.0      3.81      3.94
                        95.0      4.88      5.22
                        97.5      6.03      5.94
                        99.0      6.90      6.77
                        99.5      7.71      7.07
```

The “Percentiles” table lists percentiles for each variable.

```
                               Levels

                               Lower     Upper     Lower     Upper
             Percent     Density     for x     for x     for y     for y

                   1    0.001181     -8.14      8.45     -8.76      8.39
                   5    0.003031     -7.10      7.07     -7.14      6.77
                  10    0.004989     -6.41      5.69     -6.49      6.12
                  50     0.01591     -3.64      3.96     -3.58      3.86
                  90     0.02388     -1.22      1.19     -1.32      0.95
                  95     0.02525     -0.88      0.50     -0.99      0.62
                  99     0.02608     -0.53      0.16     -0.67      0.30
                 100     0.02629     -0.19     -0.19     -0.35     -0.35
```

The “Levels” table lists contours of the density corresponding to percentiles of the bivariate data, and the minimum and maximum values of each variable on those contours. For example, 5% of the observed data have a density value less than 0.0030. The minimum x and y values on this contour are -7.10 and -7.14, respectively (the

Lower for x and Lower for y columns), and the maximum values are 7.07 and 6.77, respectively (the Upper for x and Upper for y columns).

You can also request "Percentiles" or "Levels" tables with specific percentiles. For example,

```
proc kde data=bivnormal;
   bivar x y / levels=2.5, 50, 97.5
               percentiles=2.5, 25, 50, 75, 97.5;
run;
```

```
                        The KDE Procedure

                           Percentiles

                                  x         y

                        2.5     -6.17     -6.31
                       25.0     -2.24     -2.30
                       50.0     -0.11    -0.058
                       75.0      2.22      2.21
                       97.5      6.03      5.94
```

```
                                   Levels

                                    Lower     Upper     Lower     Upper
             Percent      Density    for x     for x     for y     for y

                 2.5     0.001914    -7.79      8.11     -7.79      7.74
                50.0      0.01591    -3.64      3.96     -3.58      3.86
                97.5      0.02573    -0.88      0.50     -0.99      0.30
```

Example 36.5. Using Output Data Set to Produce Graphics

You can create a SAS data set containing the kernel density estimate by specifying the OUT= option. Using the same 1000 simulated observations from a bivariate normal density as in the "Getting Started" section on page 1993, you can specify

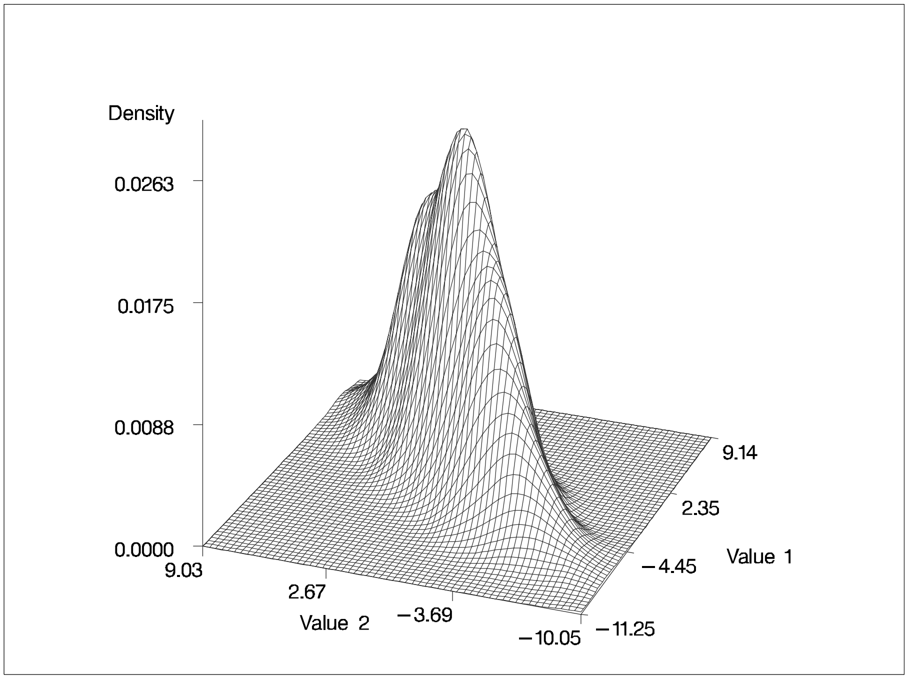
```
proc kde data=bivnormal;
   bivar x y / levels
               out=MyOut;
run;
```

The output data set MyOut from this analysis contains 3600 points containing the kernel density estimate. The variables value1 and value2 of this data set contain the grid values of the x and y variables, respectively. The variable density is the kernel density estimate. You can generate surface and contour plots of this estimate using SAS/GRAPH as follows:

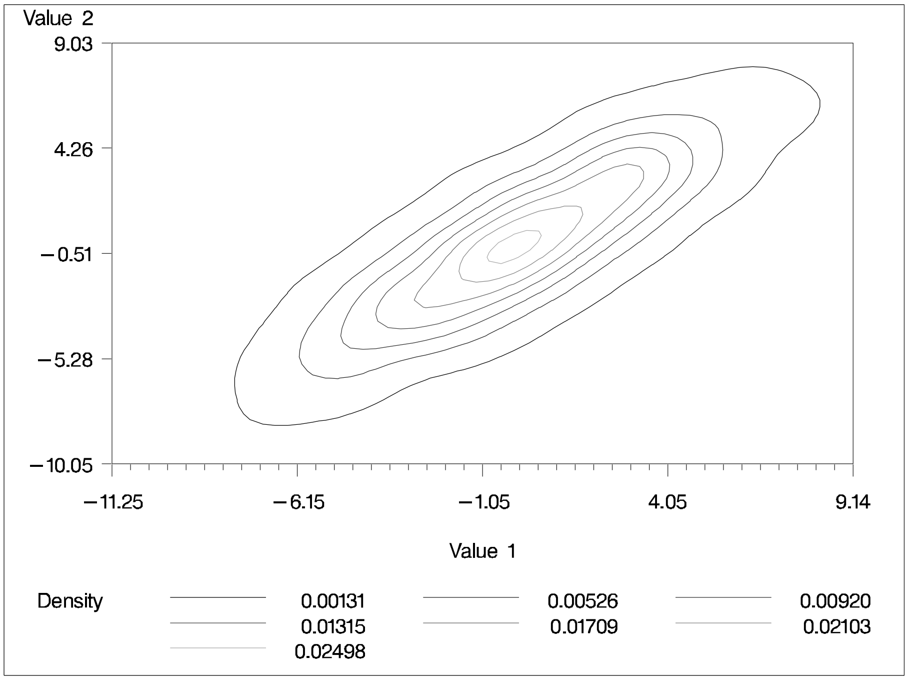
```
proc g3d data=MyOut;
   plot value2*value1=density;
run;

proc gcontour data=MyOut;
   plot value2*value1=density;
run;
```

Output 36.5.1 and Output 36.5.2 display these plots.

Output 36.5.1. Surface Plot of the Bivariate Kernel Density Estimate



Output 36.5.2. Contour Plot of the Bivariate Kernel Density Estimate

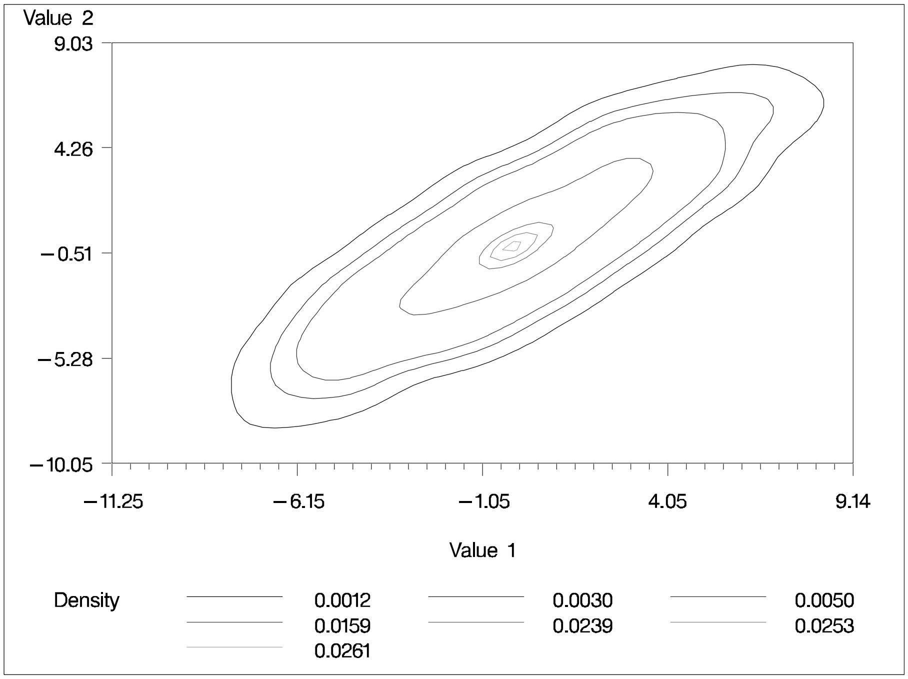


```
                                Levels

                                Lower     Upper     Lower     Upper
     Percent        Density     for x     for x     for y     for y

           1       0.001181     -8.14      8.45     -8.76      8.39
           5       0.003031     -7.10      7.07     -7.14      6.77
          10       0.004989     -6.41      5.69     -6.49      6.12
          50        0.01591     -3.64      3.96     -3.58      3.86
          90        0.02388     -1.22      1.19     -1.32      0.95
          95        0.02525     -0.88      0.50     -0.99      0.62
          99        0.02608     -0.53      0.16     -0.67      0.30
         100        0.02629     -0.19     -0.19     -0.35     -0.35
```

The “Levels” table lists contours of the density corresponding to percentiles of the bivariate data, and the minimum and maximum values of each variable on those contours. For example, 5% of the observed data have a density value less than 0.0030. You can use the results from the “Levels” table to plot specific contours corresponding to percentiles of the data. You can use the values from the Density column of this table with PROC GCONTOUR to plot the 1, 5, 10, 50, 90, 95, and 99 percent levels of the density; this plot is displayed in Output 36.5.3.

```
proc gcontour data=MyOut;
   plot value2*value1=density / levels=0.0012 0.0030 0.0050 0.0159
      0.0239 0.0253 0.0261;
run;
```

Output 36.5.3. Contour Plot of the Bivariate Kernel Density Estimate with Levels Corresponding to Percentiles



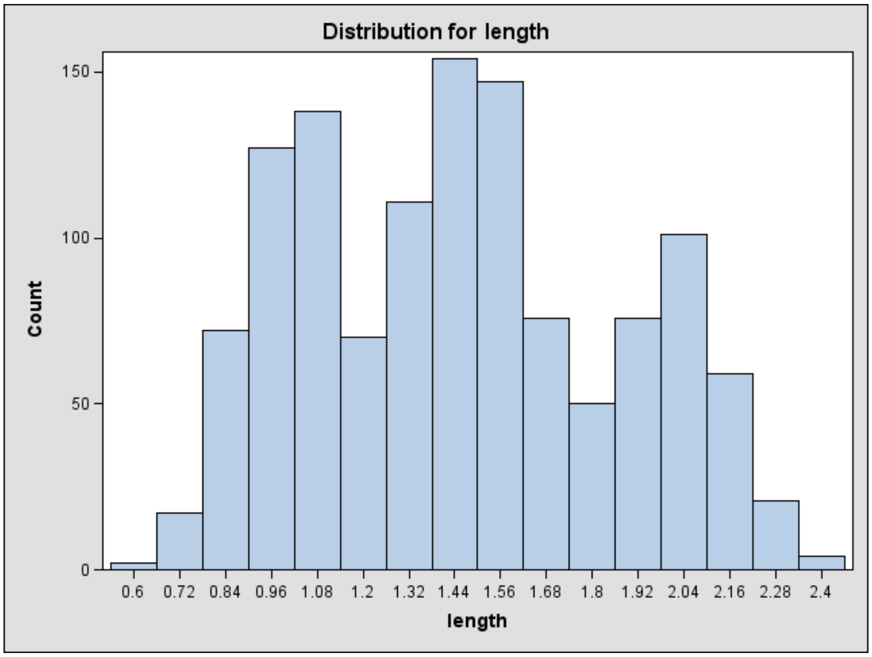
The next-to-outermost contour of Output 36.5.3 represents an approximate 95% ellipsoid for x and y.

Example 36.6. Univariate KDE Graphics (Experimental)

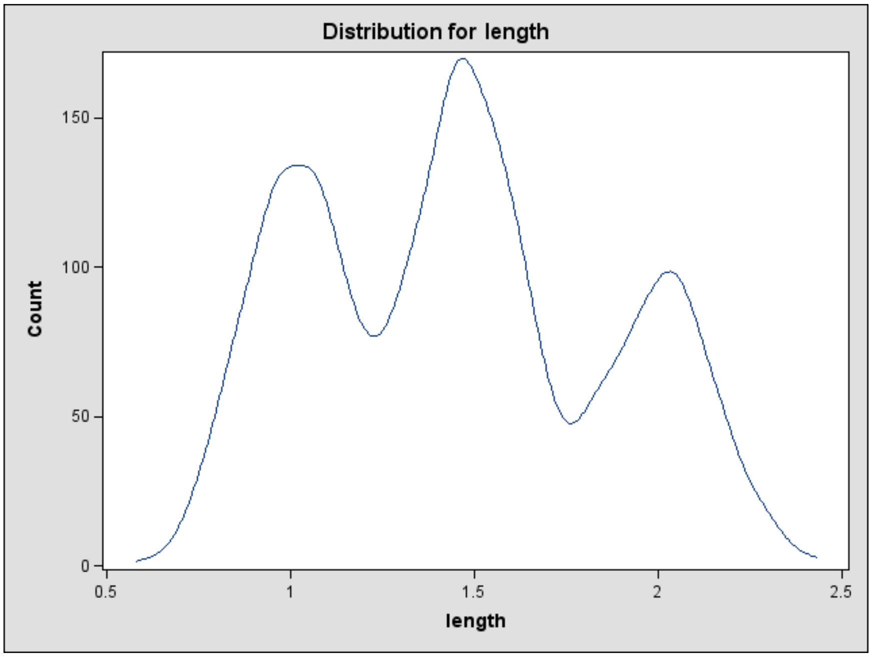
This a continuation of Example 36.1, used here to illustrate the experimental ODS graphics. The following statements request the available univariate plots in PROC KDE.

```
ods html;
ods graphics on;

proc kde data=channel;
   univar length / plots=density histogram histdensity;
run;

ods graphics off;
ods html close;
```

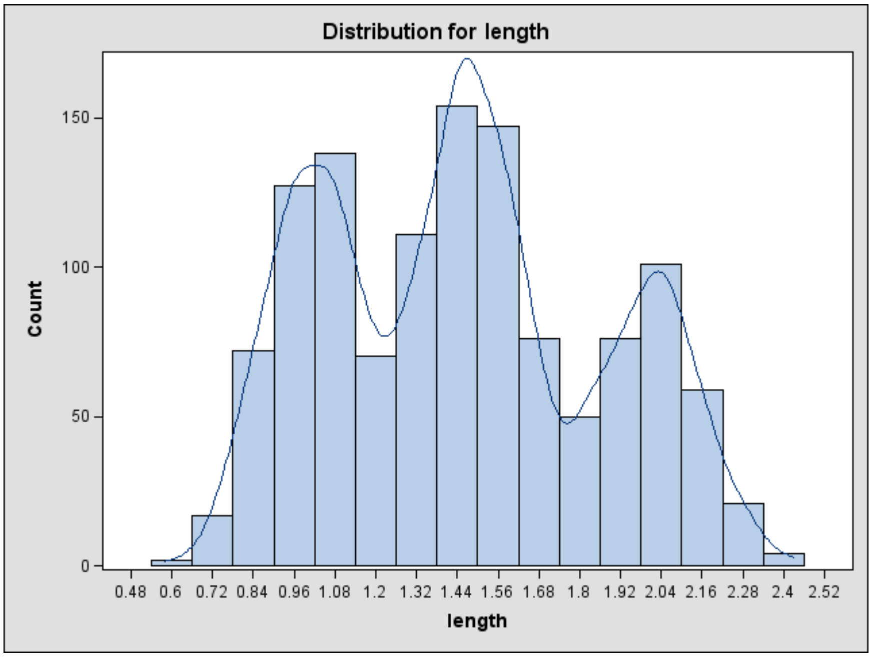
Output 36.6.1. Histogram (Experimental)



Output 36.6.2. Kernel Density Estimate (Experimental)



Output 36.6.3. Histogram with Overlaid Kernel Density Estimate (Experimental)



Output 36.6.1, Output 36.6.2, and Output 36.6.3 show a histogram, a kernel density estimate, and a histogram with an overlaid kernel density estimate, respectively. These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental PLOTS= option in the UNIVAR statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the "ODS Graphics" section on page 2009.

Example 36.7. Bivariate KDE Graphics (Experimental)

This example illustrates the available bivariate graphics in PROC KDE. The octane dataset comes from Rodriguez and Taniguchi (1980), where it is used for predicting customer octane satisfaction using trained-rater observations. The variables in this data set are Rater and Customer. Either variable may have missing values. Refer to the file kdex3.sas in the SAS Sample Library.

```
data octane;
   input Rater Customer;
   label Rater    = 'Rater'
         Customer = 'Customer';
datalines;
94.5 92.0
94.0 88.0
94.0 90.0

  ...
```

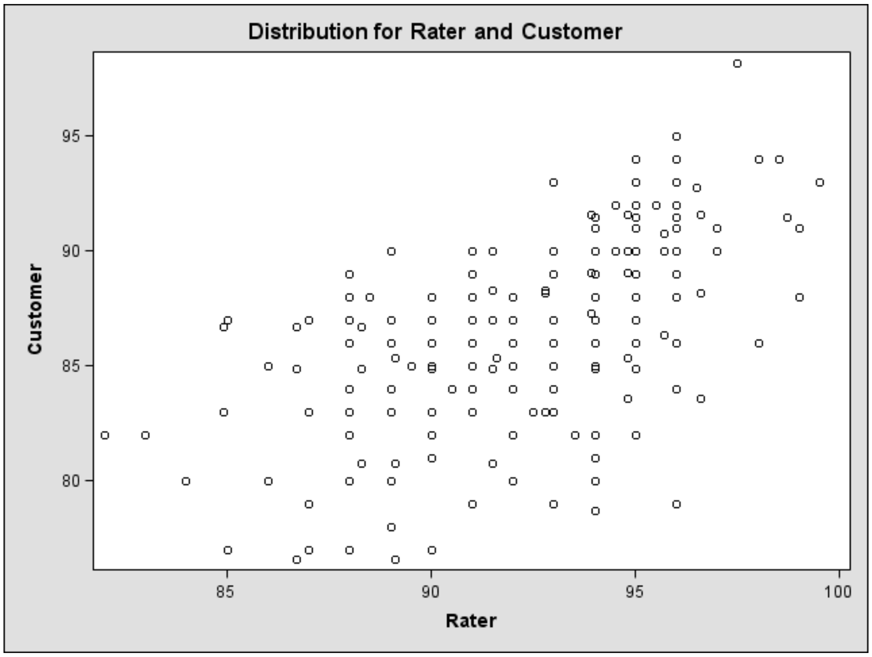
```
93.0 87.0
88.0 84.0
  .H 90.0
;
```

The following statements request all the available bivariate plots in PROC KDE.

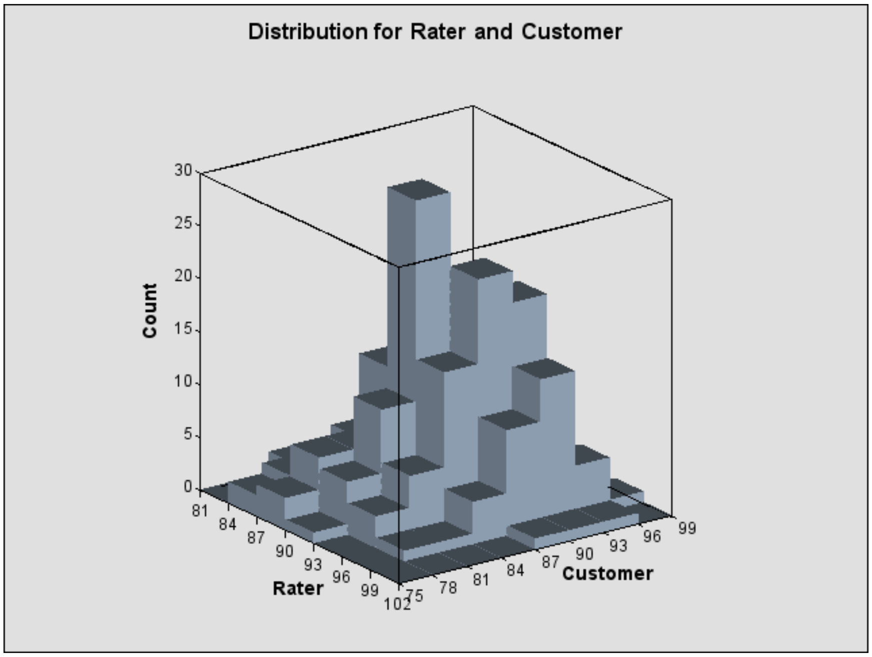
```
ods html;
ods graphics on;

proc kde data=octane;
   bivar Rater Customer / plots=all;
run;

ods graphics off;
ods html close;
```

Output 36.7.1 shows a scatter plot of the data, Output 36.7.2 shows a bivariate histogram of the data, Output 36.7.3 shows a contour plot of bivariate density estimate, Output 36.7.4 shows a contour plot of bivariate density estimate overlaid with a scatter plot of data, Output 36.7.5 shows a surface plot of bivariate kernel density estimate, and Output 36.7.6 shows a bivariate histogram overlaid with a bivariate kernel density estimate. These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental PLOTS= option in the BIVAR statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the "ODS Graphics" section on page 2009.

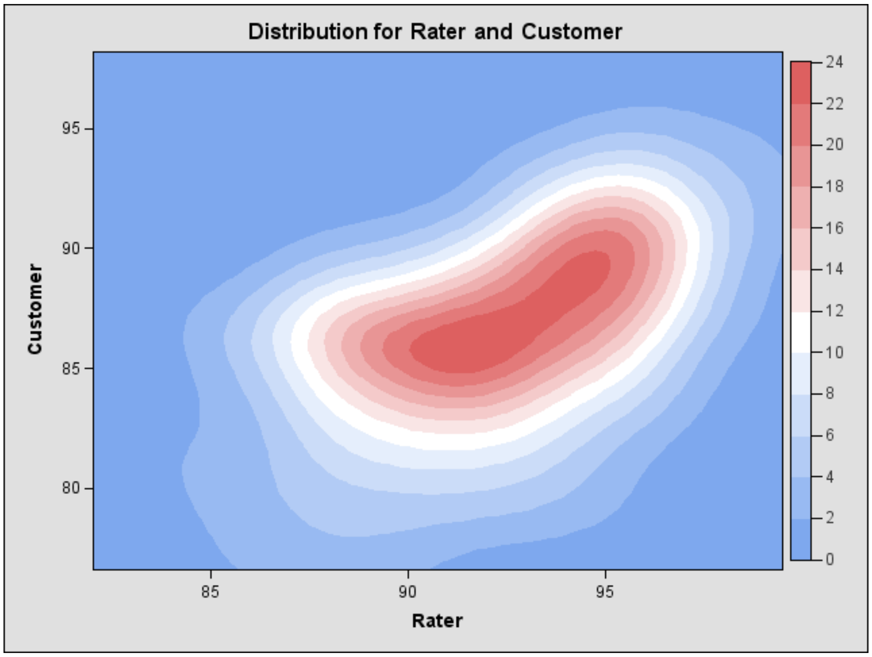
Output 36.7.1. Scatter Plot (Experimental)



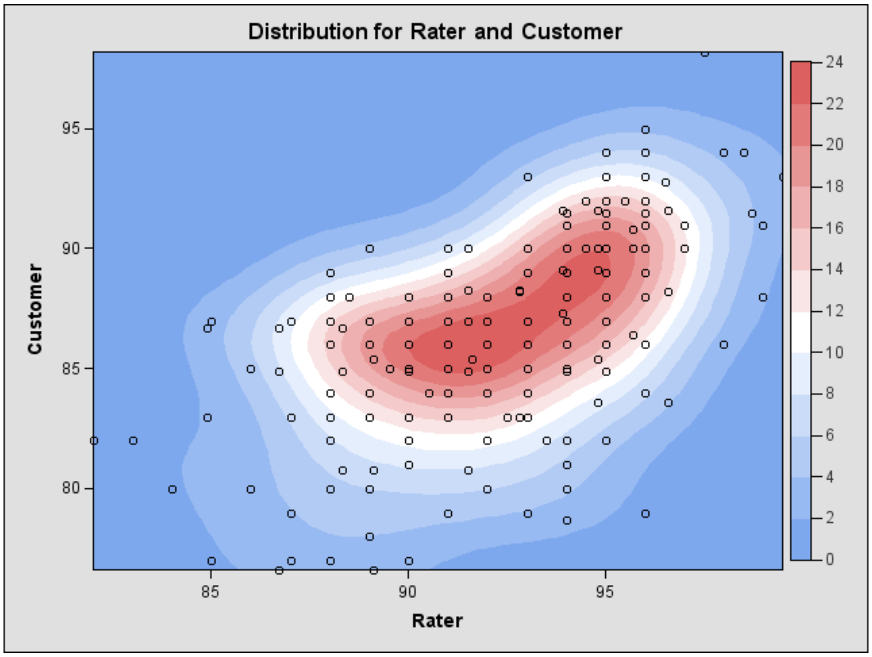
Output 36.7.2. Bivariate Histogram (Experimental)



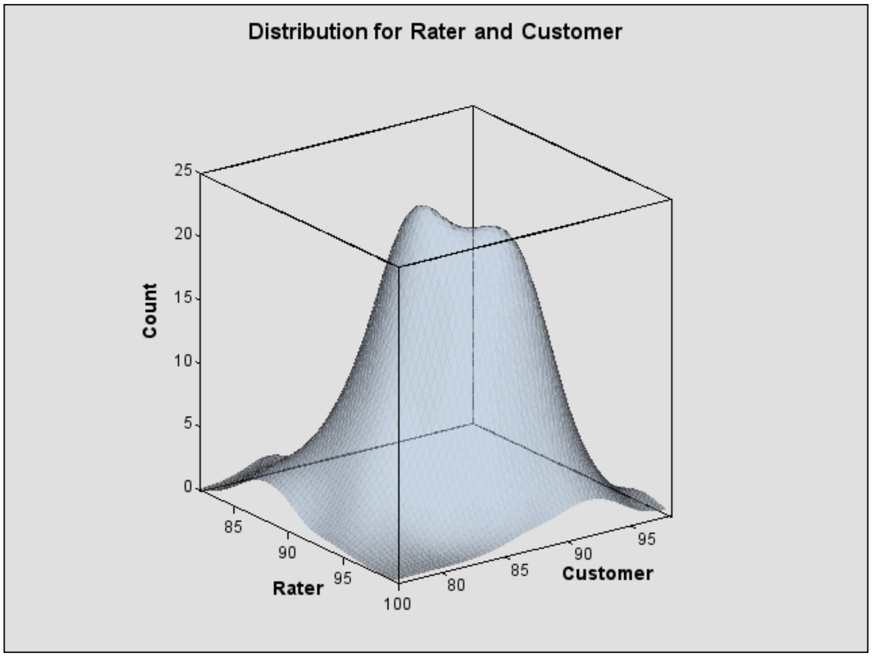
Output 36.7.3. Contour Plot (Experimental)



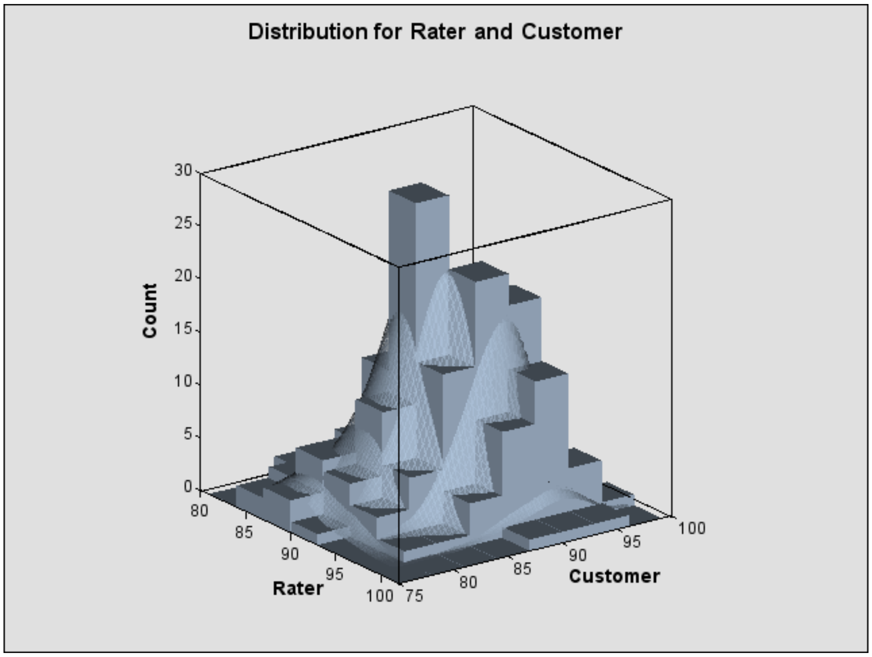
Output 36.7.4. Contour Plot with Overlaid Scatter Plot (Experimental)



Output 36.7.5. Surface Plot (Experimental)



Output 36.7.6. Bivariate Histogram with Overlaid Surface Plot (Experimental)



References

Bowman, A. W. and Foster, P. J. (1993), “Density Based Exploration of Bivariate Data,” *Statistics and Computing*, 3, 171–177.

Fan, J. and Marron, J. S. (1994), “Fast Implementations of Nonparametric Curve Estimators,” *Journal of Computational and Graphical Statistics*, 3, 35–56.

Jones, M. C., Marron, J. S., and Sheather, S. J. (1996), “A Brief Survey of Bandwidth Selection for Density Estimation,” *Journal of the American Statistical Association*, 91, 401–407.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988), *Numerical Recipes: The Art of Scientific Computing*, Cambridge: Cambridge University Press.

Rodriguez, R. N. and Taniguchi, B. Y. (1980), “A New Statistical Model for Predicting Customer Octane Satisfaction Using Trained-Rater Observations,” *Transactions of the Society of Automotive Engineers*, 4213–4235.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons, Inc.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman and Hall.

Terrell, G. R. and Scott, D. W. (1985), “Oversmoothed Nonparametric Density Estimates,” *Journal of the American Statistical Association*, 80, 209–214.

Wand, M. P. (1994), “Fast Computation of Multivariate Kernel Estimators,” *Journal of Computational and Graphical Statistics*, 3, 433–445.

Wand, M. P. and Jones, M. C. (1993), “Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation,” *Journal of the American Statistical Association*, 88, 520–528.

Chapter 37
The KRIGE2D Procedure

Chapter Contents

Chapter 37
The KRIGE2D Procedure

Overview

The KRIGE2D procedure performs ordinary kriging in two dimensions. PROC KRIGE2D can handle anisotropic and nested semivariogram models. Four semivariogram models are supported: the Gaussian, exponential, spherical, and power models. A single nugget effect is also supported.

You can specify the locations of kriging estimates in a GRID statement, or they can be read from a SAS data set. The grid specification is most suitable for a regular grid; the data set specification can handle any irregular pattern of points.

Local kriging is supported through the specification of a radius around a grid point or the specification of the number of nearest neighbors to use in the kriging system. When you perform local kriging, a separate kriging system is solved at each grid point using a neighborhood of the data point established by the radius or number specification.

The KRIGE2D procedure writes the kriging estimates and associated standard errors for each grid to an output data set. When you perform local kriging, PROC KRIGE2D writes the neighborhood information for each grid point to an additional, optional data set. The KRIGE2D procedure does not produce any displayed output.

Introduction to Spatial Prediction

Spatial prediction, in general, is any prediction method that incorporates spatial dependence. A simple and popular spatial prediction method is ordinary kriging.

Ordinary kriging requires a model of the spatial continuity, or dependence. This is typically in the form of a covariance or semivariogram.

Spatial prediction, then, involves two steps. First, you model the covariance or semivariogram of the spatial process. This involves choosing both a mathematical form and the values of the associated parameters. Second, you use this dependence model in solving the kriging system at a specified set of spatial points, resulting in predicted values and associated standard errors.

The KRIGE2D procedure performs the second of these steps using ordinary kriging of two-dimensional data.

Getting Started

Spatial Prediction Using Kriging, Contour Plots

After an appropriate variogram model is chosen, there are a number of choices involved in producing the kriging surface. In order to illustrate these choices, the variogram model in the the section "Getting Started" on page 4852 section of Chapter 80, "The VARIOGRAM Procedure," is used. This model is Gaussian,

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h^2}{a_0^2} \right) \right]$$

with a scale of $c_0 = 7.5$ and a range of $a_0 = 30$.

The first choice is whether to use local or global kriging. Local kriging uses only data points in the neighborhood of a grid point; global kriging uses all data points.

The most important consideration in this decision is the spatial covariance structure. Global kriging is appropriate when the correlation range ϵ is approximately equal to the length of the spatial domain. The correlation range ϵ is the distance r_ϵ at which the covariance is 5% of its value at zero. That is,

$$C_Z(r_\epsilon) = .05C_z(0)$$

For a Gaussian model, r_ϵ is $\sqrt{3}a_0 \approx 52$ (thousand ft). The data points are scattered uniformly throughout a 100×100 (10^6 ft^2) area. Hence, the linear dimension of the data is nearly double the ϵ range. This indicates that local kriging rather than global kriging is appropriate.

Local kriging is performed by using only data points within a specified radius of each grid point. In this example, a radius of 60 (thousand ft) is used. Other choices involved in local kriging are the minimum and maximum number of data points in each neighborhood (around a grid point). The minimum number is left at the default value of 20; the maximum number defaults to all observations in the data set.

The last step in contouring the data is to decide on the grid point locations. A convenient area that encompasses all the data points is a square of length 100 (thousand ft). The spacing of the grid points depends on the use of the contouring; a spacing of five distance units (thousand ft) is chosen for plotting purposes.

The following SAS code inputs the data and computes the kriged surface using these parameter and grid choices. The kriged surface is plotted in Figure 37.1, and the associated standard errors are plotted in Figure 37.2. The standard errors are smaller where more data are available.

```
data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;

proc krige2d data=thick outest=est;
   pred var=thick r=60;
   model scale=7.5 range=30 form=gauss;
   coord xc=east yc=north;
   grid x=0 to 100 by 5 y=0 to 100 by 5;
run;

proc g3d data=est;
   title 'Surface Plot of Kriged Coal Seam Thickness';
   scatter gyc*gxc=estimate / grid;
   label gyc      = 'North'
         gxc      = 'East'
         estimate = 'Thickness'
         ;
run;

proc g3d data=est;
   title 'Surface Plot of Standard Errors of Kriging Estimates';
   scatter gyc*gxc=stderr / grid;
   label gyc    = 'North'
         gxc    = 'East'
         stderr = 'Std Error'
         ;
run;
```

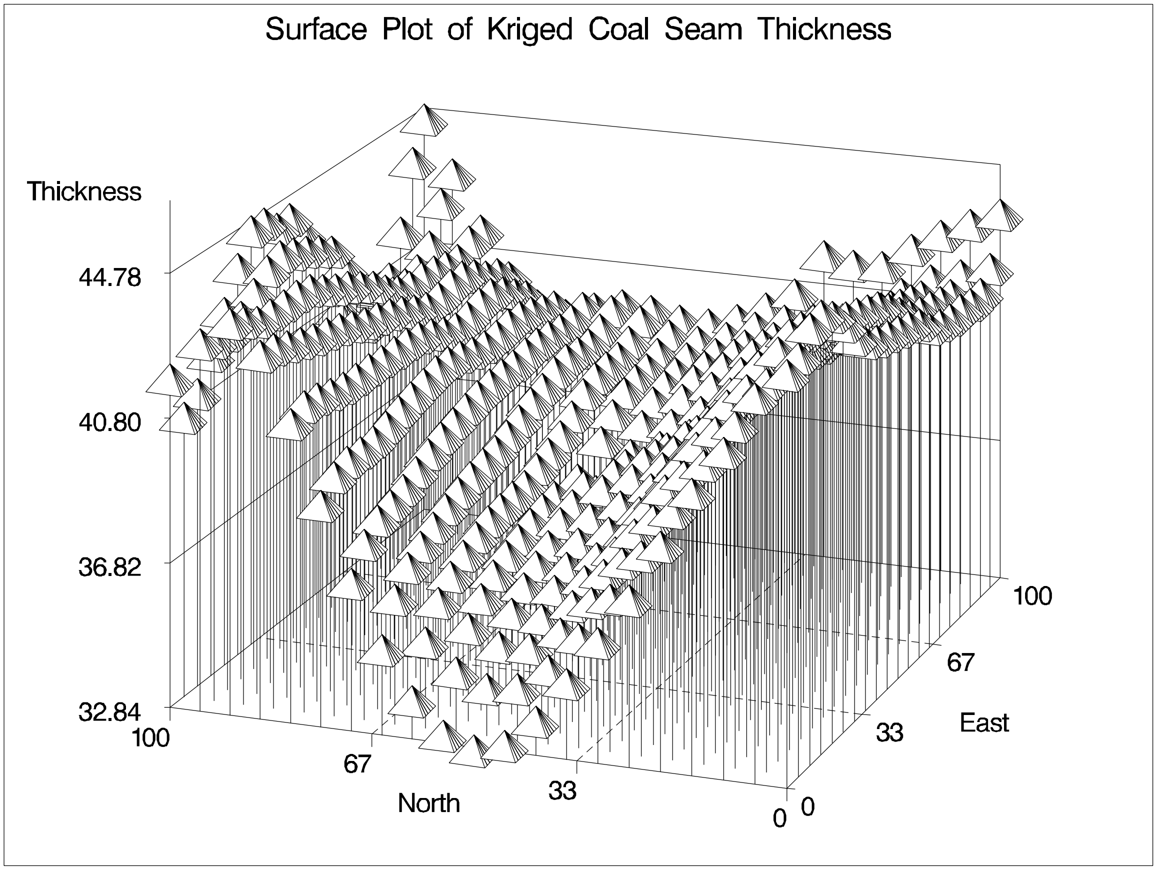


Figure 37.1. Surface Plot of Kriged Coal Seam Thickness

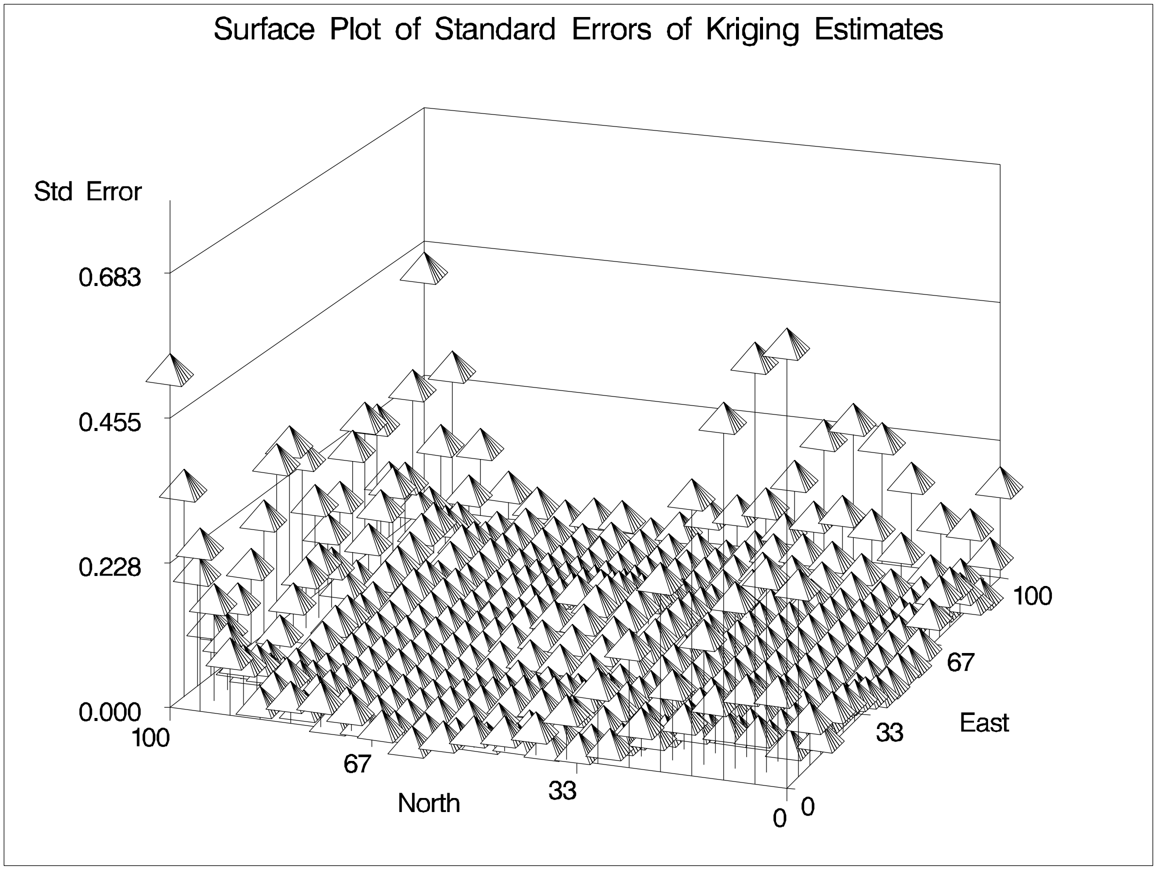


Figure 37.2. Surface Plot of Standard Errors of Kriging Estimates

Syntax

The following statements are available in PROC KRIGE2D.

> **PROC KRIGE2D** *options* **;**
> **COORDINATES | COORD** *coordinate-variables* **;**
> **GRID** *grid-options* **;**
> **PREDICT | PRED | P** *predict-options* **;**
> **MODEL** *model-options* **;**

The PREDICT and MODEL statements are hierarchical; the PREDICT statement is followed by one or more MODEL statements. All the MODEL statements following a PREDICT statement use the variable and neighborhood specifications in that PREDICT statement.

You must specify at least one PREDICT statement and one MODEL statement. You must supply a single COORDINATES statement to identify the x and y coordinate variables in the input data set. You must also specify a single GRID statement to include the grid information.

The following table outlines the options available in PROC KRIGE2D classified by function.

Table 37.1. Options Available in the KRIGE2D Procedure

Task	Statement	Option
Data Set Options		
specify input data set	PROC KRIGE2D	DATA=
specify grid data set	GRID	GDATA=
specify model data set	MODEL	MDATA=
write kriging estimates and standard errors	PROC KRIGE2D	OUTEST=
write neighborhood information for each grid point	PROC KRIGE2D	OUTNBHD=
Declaring the Role of Variables		
specify the variables to be estimated (kriged)	PREDICT	VAR=
specify the x and y coordinate variables in the DATA= data set	COORDINATES	XC= YC=
specify the x and y coordinate variables in the GDATA= data set	GRID	XC= YC=
Controlling Kriging Neighborhoods		
specify the radius of a neighborhood for all grid points	PREDICT	RADIUS=
specify the number of neighbors for all grid points	PREDICT	NUMPOINTS=
specify the maximum of neighbors for all grid points	PREDICT	MAXPOINTS=
specify the minimum of neighbors for all grid points	PREDICT	MINPOINTS=
specify action when maximum not met	PREDICT	NODECREMENT
specify action when minimum not met	PREDICT	NOINCREMENT
Controlling the Semivariogram Model		
specify a nugget effect	MODEL	NUGGET=
specify a functional form	MODEL	FORM=
specify a range parameter	MODEL	RANGE=
specify a scale parameter	MODEL	SCALE=
specify an angle for an anisotropic model	MODEL	ANGLE=
specify a minor-major axis ratio for an anisotropic model	MODEL	RATIO=

PROC KRIGE2D Statement

PROC KRIGE2D *options* **;**

You can specify the following options in the PROC KRIGE2D statement.

DATA=*SAS-data-set*

specifies a SAS data set containing the x and y coordinate variables and the VAR= variables in the PREDICT statement.

OUTEST=*SAS-data-set*
OUTE=*SAS-data-set*

specifies a SAS data set in which to store the kriging estimates, standard errors and grid location. For details, see the section "OUTEST=*SAS-data-set* " on page 2060.

OUTNBHD=*SAS-data-set*
OUTN=*SAS-data-set*

specifies a SAS data set in which to store the neighborhood information for each grid point. Information is written to this data set only if one or more PREDICT statements have options specifying local kriging. For details, see the section "OUTNBHD=*SAS-data-set* " on page 2060.

SINGULARMSG=*number*
SMSG=*number*

controls the number of warning messages displayed for a singular matrix. When local kriging is performed, a separate kriging system is solved for each grid point. Anytime a singular matrix is encountered, a warning message is displayed up to a total of SINGULARMSG=*n* times. The default is SINGULARMSG=10.

COORDINATES Statement

COORDINATES | COORD *coordinate-variables* **;**

The following two options specify the names of the variables in the DATA= data set containing the values of the x and y coordinates of the data.

Only one COORDINATES statement is allowed, and it is applied to all PREDICT statements. In other words, it is assumed that all the VAR= variables in all PREDICT statements have the same x and y coordinates.

This is not a limitation. Since each VAR= variable is processed separately, observations for which the current VAR= variable is missing are excluded. With the next VAR= variable, the entire data are read again, this time excluding missing values in this next variable. Hence, a single run of PROC KRIGE2D can be used for variables measured at different locations without overlap.

XCOORD= *(variable-name)*
XC= *(variable-name)*

specifies the name of the variable containing the x coordinate of the data locations in the DATA= data set.

YCOORD= *(variable-name)*
YC= *(variable-name)*

specifies the name of the variable containing the y coordinate of the data locations in the DATA= data set.

GRID Statement

GRID *grid-options* **;**

You can use the following options to specify the grid of spatial locations for the kriging estimates. The grid specification is applied to all PREDICT and MODEL statements.

There are two basic methods for specifying the grid. You can specify the x and y coordinates explicitly, or they can be read from a SAS data set. The options for the explicit specification of grid locations are as follows.

X=*number*
X=$x_1, \ldots, x_m$
X=x_1 *to* x_m
X=x_1 *to* x_m *by* δx

specifies the x coordinate of the grid locations.

Y=*number*
Y=$y_1, \ldots, y_m$
Y=y_1 *to* y_m
Y=y_1 *to* y_m *by* δy

specifies the y coordinate of the grid locations.

For example, the following two GRID statements are equivalent.

```
grid x=1,2,3,4,5  y=0,2,4,6,8,10;
grid x=1 to 5 y=0 to 10 by 2;
```

To specify grid locations from a SAS data set, you must give the name of the data set and the variables containing the values of the x and y coordinates.

GRIDDATA=*SAS-data-set*
GDATA=*SAS-data-set*

specifies a SAS data set containing the x and y grid coordinates.

XCOORD= *(variable-name)*
XC= *(variable-name)*

specifies the name of the variable containing the x coordinate of the grid locations in the GRIDDATA= data set.

YCOORD= *(variable-name)*
YC= *(variable-name)*

specifies the name of the variable containing the y coordinate of the grid locations in the GRIDDATA= data set.

PREDICT Statement

PREDICT | PRED | P *predict-options* **;**

You can specify the following options in a PREDICT statement.

MAXPOINTS=*number*
MAXPOINTS=*number*
MAXP=*number*

specifies the maximum number of data points in a neighborhood. You specify this option in conjunction with the RADIUS= option. When the number of data points in the neighborhood formed at a given grid point by the RADIUS= option is greater than the MAXPOINTS= value, the RADIUS= value is decreased just enough to honor the MAXPOINTS= value unless you specify the NODECREMENT option.

MINPOINTS=*number*
MINP=*number*
MIN=*number*

specifies the minimum number of data points in a neighborhood. You specify this option in conjunction with the RADIUS= option. When the number of data points in the neighborhood formed at a given grid point by the RADIUS= option is less than the MINPOINTS= value, the RADIUS= value is increased just enough to honor the MINPOINTS= value unless you specify the NOINCREMENT option. The default is MINPOINTS=20. If enough data are available, this value should be increased to 30 to improve estimation.

NODECREMENT | NODECR

requests that the RADIUS= value not be decremented when the MAX= value is exceeded at a grid point. This option is relevant only when you specify both a RADIUS= value and a MAXPOINTS= value. In this case, when the number of points in the neighborhood constructed from the RADIUS= specification is greater than the MAXPOINTS= value, the RADIUS= value is decremented enough to honor the MAXPOINTS= value, and the kriging system is solved for this grid point. If you specify the NODECREMENT option, no decrementing is done, estimation is skipped at this grid point, and a message is written to the log.

NOINCREMENT | NOINCR

requests that the RADIUS= value not be incremented when the MIN= value is not met at a grid point. This option is relevant only when you specify both a RADIUS= value and a MINPOINTS= number. In this case, when the number of points in the neighborhood constructed from the RADIUS= specification is less than the MINPOINTS= value, the RADIUS= value is incremented enough to honor the MINPOINTS= value, and the kriging system is solved for this grid point. If you specify the NOINCREMENT option, no incrementing is done, estimation is skipped at this grid point, and a message is written to the log.

NUMPOINTS=*number*
NPOINTS=*number*
NPTS=*number*
NP=*number*

specifies the exact size of a neighborhood. This option is incompatible with all other PREDICT statement options controlling the neighborhood; it must appear by itself.

RADIUS=*number*
R=*number*

specifies the radius to use in a local kriging regression. When you specify this option, a separate kriging system is solved at each grid point by finding the neighborhood of this grid point consisting of all data points within the distance specified by the RADIUS= value. See the MAXPOINTS= and MINPOINTS= options for additional control on the neighborhood.

VAR= *variable-name*

specifies the single numeric variable used in the kriging system.

MODEL Statement

MODEL *model-options* **;**

You can use the following options to specify a semivariogram or covariance model. The specified model is used in the kriging system defined by the most previous PREDICT statement.

There are two ways to specify a semivariogram or covariance model. In the first method, you specify the required parameters SCALE, RANGE, and FORM, and possibly the optional parameters NUGGET, ANGLE, and RATIO, explicitly in the MODEL statement.

In the second method, you specify an MDATA= data set. This data set contains variables corresponding to the required SCALE, RANGE, and FORM parameters, and, optionally, variables for the NUGGET, ANGLE, and RATIO parameters.

The two methods are exclusive; either you specify all parameters explicitly, or they all are read from the MDATA= data set.

ANGLE=*angle*
ANGLE=*(angle*$_1$*,. . . ,angle*$_k$*)*

specifies the angle of the major axis for anisotropic models, measured in degrees clockwise from the N-S axis. In the case of a nested semivariogram model, you can specify an angle for each nesting. The default is ANGLE=0.

FORM=SPHERICAL | EXPONENTIAL | GAUSSIAN | POWER
FORM=SPH | EXP | GAUSS | PW

specifies the functional form of the semivariogram model. All the supported models are two-parameter models (SCALE= and RANGE=). A FORM= value is required; in the case of a nested semivariogram model, you must specify a form for each nesting.

See the section “Theoretical Semivariogram Models” beginning on page 2045 for details on how the FORM= forms are determined.

MDATA=*SAS-data-set*

specifies the input data set that contains parameter values for the covariance or semivariogram model. The MDATA= data set must contain variables named SCALE, RANGE, and FORM, and it can optionally contain variables NUGGET, ANGLE, and RATIO.

The FORM variable must be a character variable, assuming only the values allowed in the explicit FORM= syntax described previously. The RANGE and SCALE variables must be numeric. The optional variables ANGLE, RATIO, and NUGGET must also be numeric if present.

The number of observations present in the MDATA= data set corresponds to the level of nesting of the semivariogram model. For example, to specify a nonnested model using a spherical covariance, an MDATA= data set might look like

```
data md1;
   input scale range form $;
   datalines;
   25 10 SPH
run;
```

The PROC KRIGE2D statement to use the MDATA= specification is of the form

```
proc krige2d data=...;
   pred var=....;
   model mdata=md1;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc krige2d data=...;
   pred var=....;
   model scale=25 range=10 form=sph;
run;
```

The following MDATA= data set is an example of an anisotropic nested model:

```
data md1;
   input scale range form $ nugget angle ratio;
   datalines;
   20 8 S  5 35 0.7
   12 3 G  5 0  0.8
   4  1 G  5 45 0.5
   ;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc krige2d data=...;
   pred var=....;
   model scale=(20,12,4) range=(8,3,1) form=(S,G,G)
         angle=(35,0,45) ratio=(0.7,0.8,0.5) nugget=5;
run;
```

This example is somewhat artificial in that it is usually hard to detect different anisotropy directions and ratios for different nestings using an experimental semivariogram. Note that the NUGGET value is the same for all nestings. This is always the case; the nugget effect is a single additive term for all models. For further details, see the section "Theoretical and Computational Details of the Semivariogram" on page 4872 in the chapter on the VARIOGRAM procedure.

NUGGET=*number*

specifies the nugget effect for the model. The nugget effect is due to a discontinuity in the semivariogram as determined by plotting the sample semivariogram (see the chapter on the VARIOGRAM procedure for details). For models without any nugget effect, this option is left out; the default is NUGGET=0.

RANGE=*range*

RANGE=*(range$_1$,. . . ,range$_k$)*

specifies the range parameter in semivariogram models. In the case of a nested semivariogram model, you must specify a range for each nesting.

The range parameter is the divisor in the exponent in all supported models except the power model. It has the units of distance or distance squared for these models, and it is related to the correlation scale for the underlying spatial process. See the section "Theoretical Semivariogram Models" beginning on page 2045 for details on how the RANGE= values are determined.

RATIO=*ratio*

RATIO=*(ratio$_1$,. . . ,ratio$_k$)*

specifies the ratio of the length of the minor axis to the length of the major axis for anisotropic models. The value of the RATIO= option must be between 0 and 1. In the case of a nested semivariogram model, you can specify a ratio for each nesting. The default is RATIO=1.

SCALE=*scale*

SCALE=*(scale$_1$,. . . ,scale$_k$)*

specifies the scale parameter in semivariogram models. In the case of a nested semivariogram model, you must specify a scale for each nesting.

The scale parameter is the multiplicative factor in all supported models; it has the same units as the variance of the VAR= variable in the preceding PREDICT statement. See the section "Theoretical Semivariogram Models" beginning on page 2045 for details on how the SCALE= values are determined.

SINGULAR=*number*

gives the singularity criteria for solving kriging systems. The larger the value of the SINGULAR= option, the easier it is for a kriging system to be declared singular.

The default is SINGULAR=1E-7. See the section "Details of Ordinary Kriging" beginning on page 2056 for more detailed information.

Details

Theoretical Semivariogram Models

PROC VARIOGRAM computes the sample, or experimental semivariogram. Prediction of the spatial process at unsampled locations by techniques such as ordinary kriging requires a theoretical semivariogram or covariance.

When you use PROC VARIOGRAM and PROC KRIGE2D to perform spatial prediction, you must determine a suitable theoretical semivariogram based on the sample semivariogram. While there are various methods of fitting semivariogram models, such as least squares, maximum likelihood, and robust methods (Cressie 1993, section 2.6), these techniques are not appropriate for data sets resulting in a small number of variogram points. Instead, a visual fit of the variogram points to a few standard models is often satisfactory. Even when there are sufficient variogram points, a visual check against a fitted theoretical model is appropriate (Hohn 1988, p. 25ff).

In some cases, a plot of the experimental semivariogram suggests that a single theoretical model is inadequate. Nested models, anisotropic models, and the nugget effect increase the scope of theoretical models available. All of these concepts are discussed in this section. The specification of the final theoretical model is provided by the syntax of PROC KRIGE2D.

Note the general flow of investigation. After a suitable choice is made of the LAGDIST= and MAXLAG= options and, possibly, the NDIR= option (or a DIRECTIONS statement), the experimental semivariogram is computed. Potential theoretical models, possibly incorporating nesting, anisotropy, and the nugget effect, are computed by a DATA step, then they are plotted against the experimental semivariogram and evaluated. A suitable theoretical model is thus found visually, and the specification of the model is used in PROC KRIGE2D. This flow is illustrated in Figure 37.3; also see the "Getting Started" section on page 4852 in the chapter on the VARIOGRAM procedure for an illustration in a simple case.

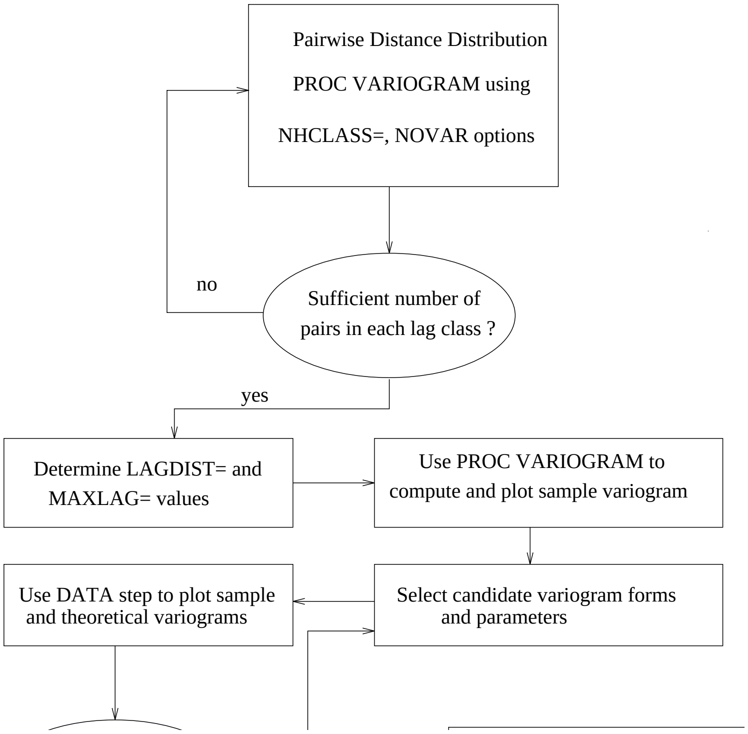

Pairwise Distance Distribution
PROC VARIOGRAM using
NHCLASS=, NOVAR options
no
Sufficient number of pairs in each lag class ?
yes
Determine LAGDIST= and MAXLAG= values
Use PROC VARIOGRAM to compute and plot sample variogram
Use DATA step to plot sample and theoretical variograms
Select candidate variogram forms and parameters

Figure 37.3. Flowchart for Variogram Selection

Four theoretical models are supported by PROC KRIGE2D: the spherical, Gaussian, exponential, and power models. For the first three types, the parameters a_0 and c_0, corresponding to the RANGE= and SCALE= options in the MODEL statement in PROC KRIGE2D, have the same dimensions and have similar affects on the shape of $\gamma_z(h)$, as illustrated in the following paragraph.

In particular, the dimension of c_0 is the same as the dimension of the variance of the spatial process $\{Z(r), r \in D \subset \mathcal{R}^2\}$. The dimension of a_0 is length with the same units as h.

These three model forms are now examined in more detail.

The Spherical Semivariogram Model

The form of the spherical model is

$$\gamma_z(h) = \begin{cases} c_0 \left[\frac{3}{2} \frac{h}{a_0} - \frac{1}{2} (\frac{h}{a_0})^3 \right], & \text{for } h \le a_0 \\ c_0, & \text{for } h > a_0 \end{cases}$$

The shape is displayed in Figure 37.4 using range $a_0 = 1$ and scale $c_0 = 4$.

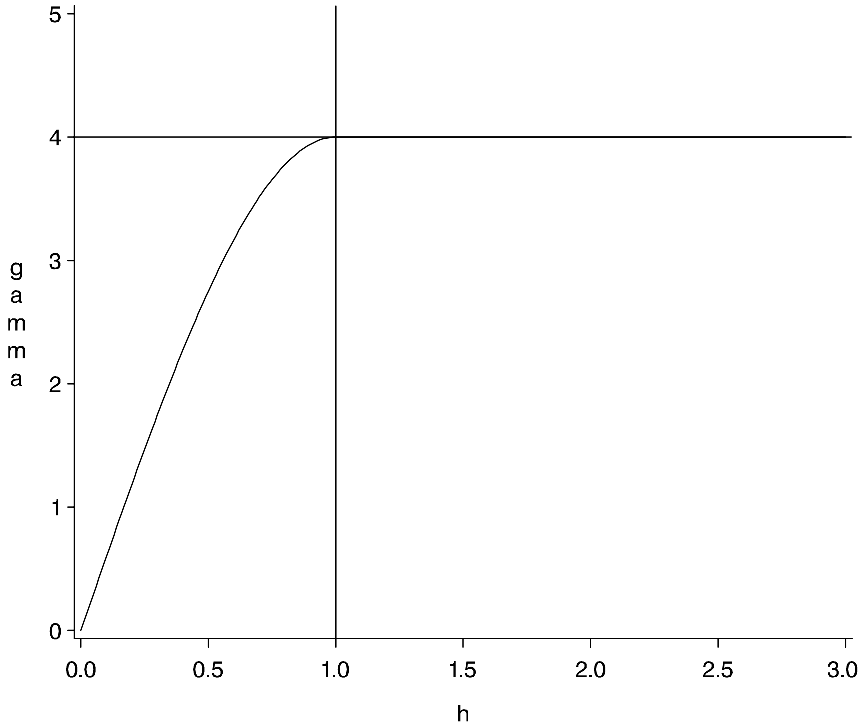


Figure 37.4. Spherical Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$

The vertical line at $h = 1$ is the "effective range" as defined by Duetsch and Journel (1992), or the "range ϵ" defined by Christakos (1992). This quantity, denoted r_ϵ, is the h-value where the covariance is approximately zero. For the spherical model, it is *exactly* zero; for the Gaussian and exponential models, the definition of r_ϵ is modified slightly.

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is called the "sill." In the case of the spherical model, $\gamma_z(h)$ actually reaches this value. For the other two model forms, the sill is a horizontal asymptote.

The Gaussian Semivariogram Model

The form of the Gaussian model is

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h^2}{a_0^2} \right) \right]$$

The shape is displayed in Figure 37.5 using range $a_0 = 1$ and scale $c_0 = 4$.

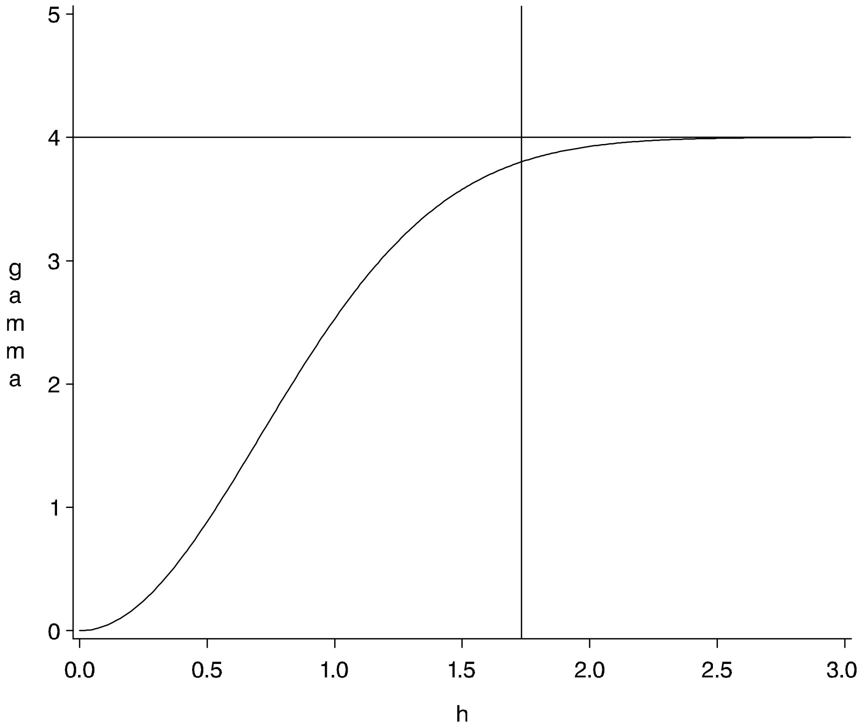


Figure 37.5. Gaussian Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$

The vertical line at $h = r_\epsilon = \sqrt{3}$ is the effective range, or the range ϵ (that is, the h-value where the covariance is approximately 5% of its value at zero).

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is the sill; $\gamma_z(h)$ approaches the sill asymptotically.

The Exponential Semivariogram Model

The form of the exponential model is

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h}{a_0}\right)\right]$$

The shape is displayed in Figure 37.6 using range $a_0 = 1$ and scale $c_0 = 4$.

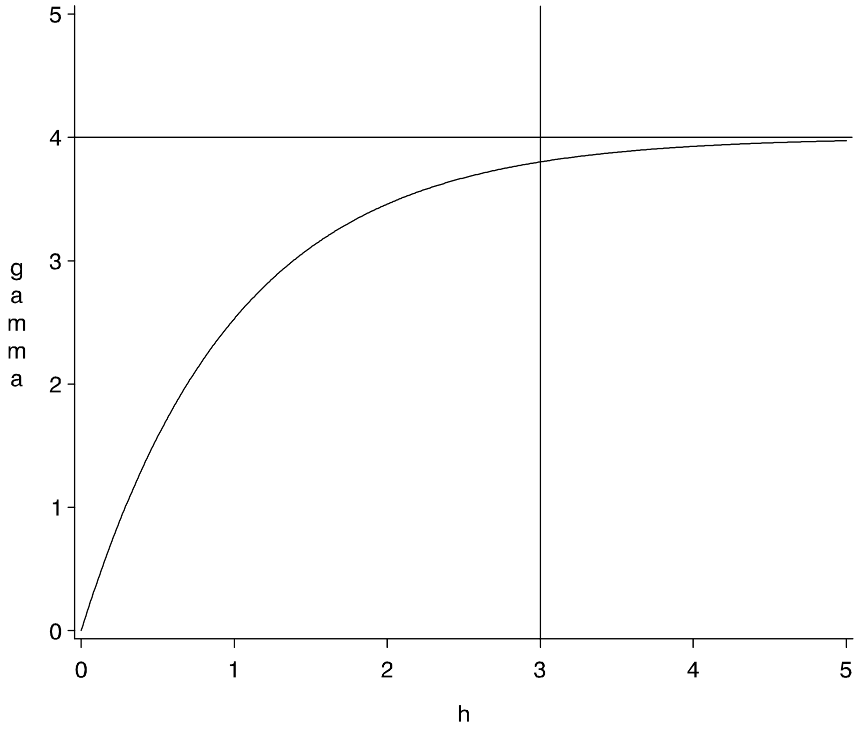


Figure 37.6. Exponential Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$

The vertical line at $h = r_\epsilon = 3$ is the effective range, or the range ϵ (that is, the h-value where the covariance is approximately 5% of its value at zero).

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is the sill, as in the other model forms.

It is noted from Figure 37.5 and Figure 37.6 that the major distinguishing feature of the Gaussian and exponential forms is the shape in the neighborhood of the origin $h = 0$. In general, small lags are important in determining an appropriate theoretical form based on a sample semivariogram.

The Power Semivariogram Model

The form of the power model is

$$\gamma_z(h) = c_0 h^{a_0}$$

For this model, the parameter a_0 is a dimensionless quantity, with typical values $0 < a_0 < 2$. Note that the value of $a_0 = 1$ yields a straight line. The parameter c_0 has dimensions of the variance, as in the other models. There is no sill for the power model. The shape of the power model with $a_0 = 0.4$ and $c_0 = 4$ is displayed in Figure 37.7.

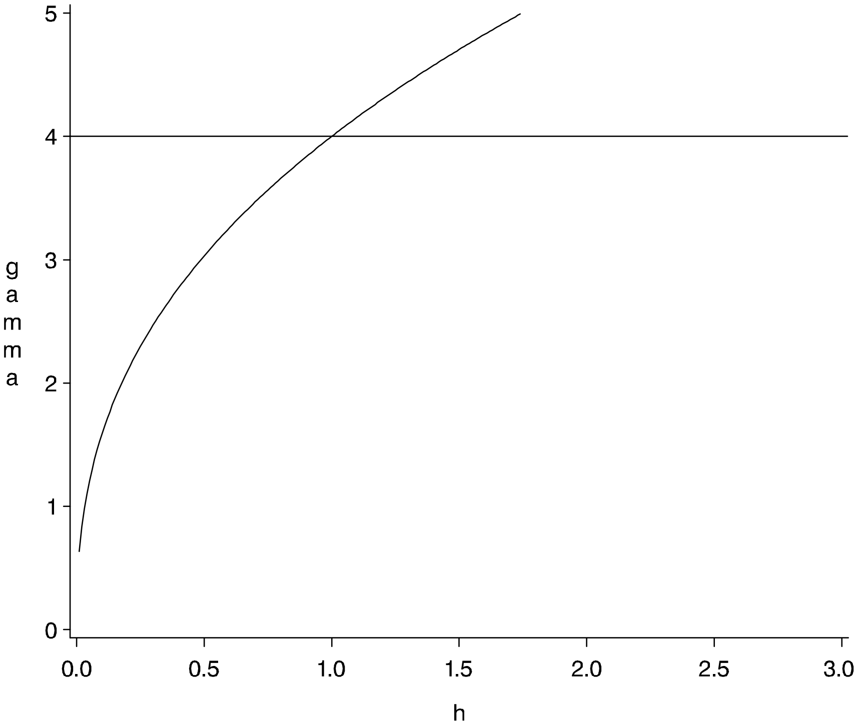


Figure 37.7. Power Semivariogram Model with Parameters $a_0 = 0.4$ and $c_0 = 4$

Nested Models

For a given set of spatial data, a plot of an experimental semivariogram may not seem to fit any one of the theoretical models. In such a case, the covariance structure of the spatial process may be a sum of two or more covariances. This is common in geologic applications where there are correlations at different length scales. At small lag distances h, the smaller scale correlations dominate, while the large scale correlations dominate at larger lag distances.

As an illustration, consider two semivariogram models, an exponential and a spherical.

$$\gamma_{z,1}(h) = c_{0,1} \exp(-\frac{h}{a_{0,1}})$$

and

$$\gamma_{z,2}(h) = \left\{ \begin{array}{ll} c_{0,2} \left[\frac{3}{2}\frac{h}{a_{0,2}} - \frac{1}{2}(\frac{h}{a_{0,2}})^3 \right], & \text{for } h \le a_{0,2} \\ c_{0,2}, & \text{for } h > a_{0,2} \end{array} \right\}$$

with $c_{0,1} = 1, a_{0,1} = 2.5, c_{0,2} = 2$, and $a_{0,2} = 1$. If both of these correlation structures are present in a spatial process $\{Z(r), r \in D\}$, then a plot of the experimental semivariogram would resemble the sum of these two semivariograms. This is illustrated in Figure 37.8.

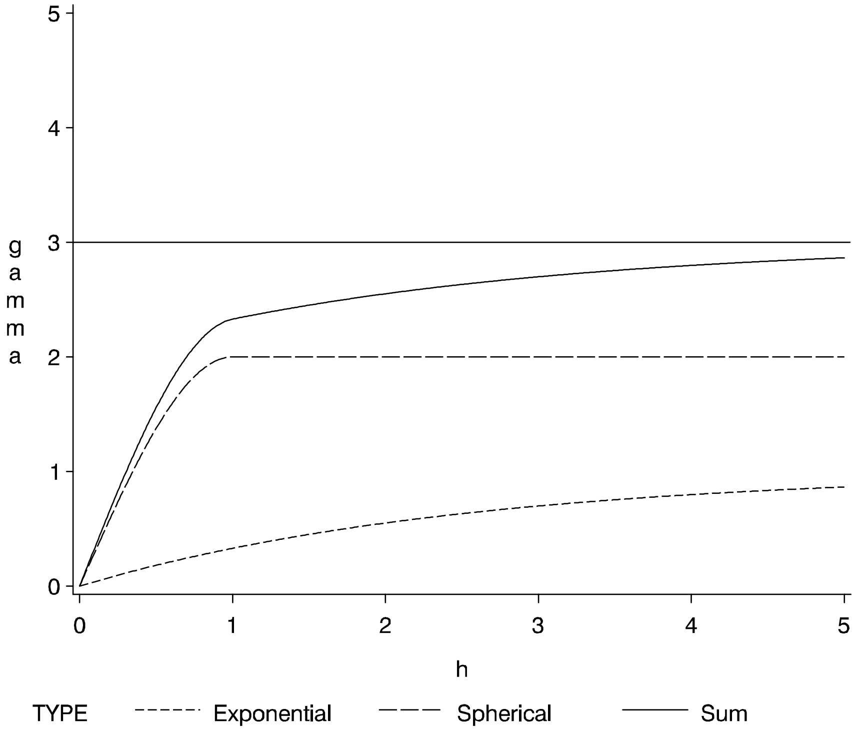


Figure 37.8. Sum of Exponential and Spherical Structures at Different Scales

This sum of $\gamma_1(h)$ and $\gamma_2(h)$ in Figure 37.8 does not resemble any *single* theoretical semivariogram; however, the shape at $h = 1$ is similar to a spherical. The asymptotic approach to a sill at three variance units, along with the shape around $h = 0$, indicates an exponential structure. Note that the sill value is the sum of the individual sills $c_{0,1} = 1$ and $c_{0,2} = 2$.

Refer to Hohn (1988, p. 38ff) for further examples of nested correlation structures.

The Nugget Effect

For all the variogram models considered previously, the following property holds:

$$\gamma_z(0) = \lim_{h \downarrow 0} \gamma_z(h) = 0$$

However, a plot of the experimental semivariogram may indicate a discontinuity at $h = 0$; that is, $\gamma_z(h) \to c_n > 0$ as $h \to 0$, while $\gamma_z(0) = 0$. The quantity c_n is called the “nugget effect”; this term is from mining geostatistics where nuggets literally exist, and it represents variations at a much smaller scale than any of the measured pairwise distances, that is, at distances $h \ll h_{min}$, where

$$h_{min} = \min_{i,j} h_{ij} = \min_{i,j} \mid r_i - r_j \mid$$

There are conceptual and theoretical difficulties associated with a nonzero nugget effect; refer to Cressie (1993, section 2.3.1) and Christakos (1992, section 7.4.3) for details. There is no *practical* difficulty however; you simply visually extrapolate the experimental semivariogram as $h \rightarrow 0$. The importance of availability of data at small lag distances is again illustrated.

As an example, an exponential semivariogram with a nugget effect c_n has the form

$$\gamma_z(h) = c_n + c_0 \left[1 - \exp\left(-\frac{h}{a_0}\right)\right], h > 0$$

and

$$\gamma_z(0) = 0$$

This is illustrated in Figure 37.9 for parameters $a_0 = 1$, $c_0 = 4$, and nugget effect $c_n = 1.5$.

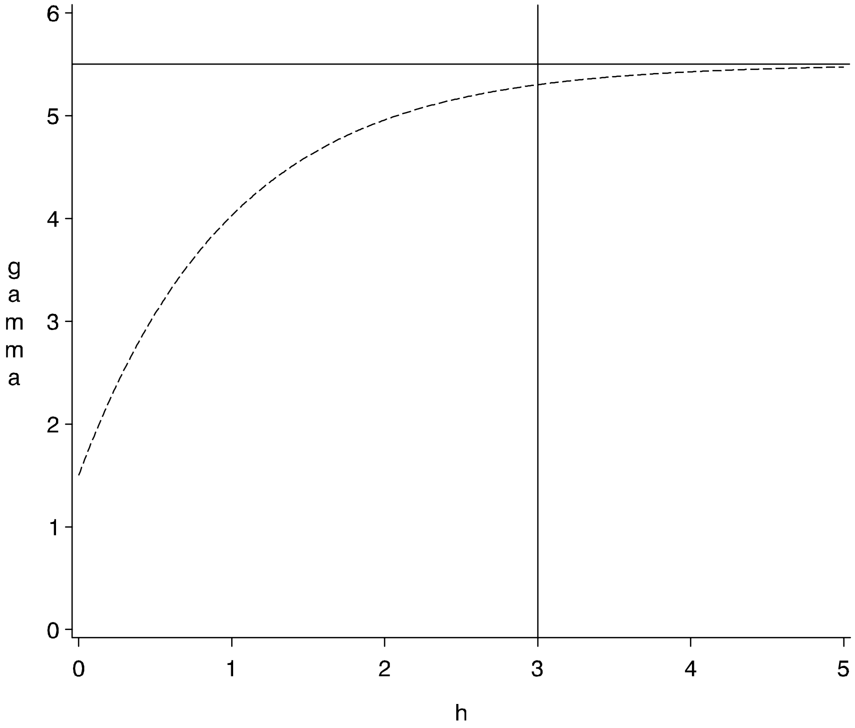


Figure 37.9. Exponential Semivariogram Model with a Nugget Effect $c_n = 1.5$

You can specify the nugget effect in PROC KRIGE2D with the NUGGET= option in the MODEL statement. It is a separate, additive term independent of direction; that is, it is isotropic. There is a way to approximate an anisotropic nugget effect; this is described in the following section.

Anisotropic Models

In all the theoretical models considered previously, the lag distance h entered as a scalar value. This implies that the correlation between the spatial process at two point pairs P_1, P_2 is dependent *only* on the separation distance $h = \mid P_1 P_2 \mid$, not on the orientation of the two points. A spatial process $\{Z(r), r \in D\}$ with this property is called isotropic, as is the associated covariance or semivariogram.

However, real spatial phenomena often show directional effects. Particularly in geologic applications, measurements along a particular direction may be highly correlated, while the perpendicular direction shows little or no correlation. Such processes are called anisotropic. Refer to Journel and Huijbregts (1978, section III.B.4) for more details.

There are two types of anisotropy. The simplest type occurs when the same covariance *form* and scale parameter c_0 is present in all directions but the range a_0 changes with direction. In this case, there is a single sill, but the semivariogram reaches the sill in a shorter lag distance along a certain direction.

This type of anisotropy is called "geometric" and is discussed in the following section.

Geometric Anisotropy

Geometric anisotropy is illustrated in Figure 37.10, where an anisotropic Gaussian semivariogram is plotted. The two curves displayed in this figure are generated using $a_0 = 1$ in the NE–SW direction and $a_0 = 3$ in the SE–NW direction.

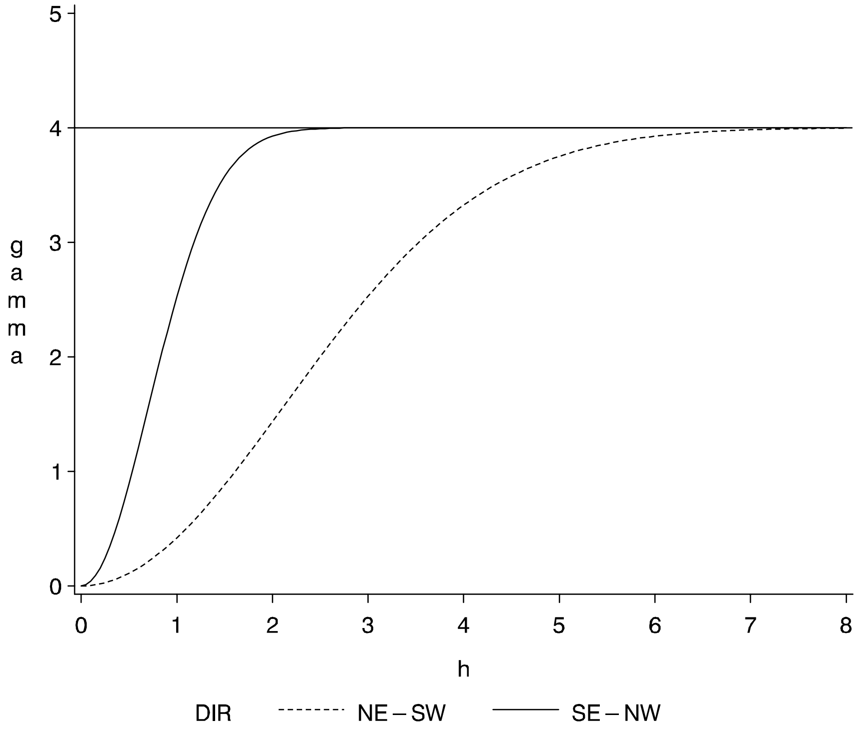


Figure 37.10. Geometric Anisotropy with Major Axis along E–W Direction

As you can see from the figure, the SE–NW curve gets “close” to the sill at approximately $h = 2$, while the NE–SW curve does so at $h = 6$. The ratio of the shorter to longer distance is $\frac{2}{6} = \frac{1}{3}$. This is the value to use in the RATIO= parameter in the MODEL statement in PROC KRIGE2D. Since the longer, or major, axis is in the NE–SW direction, the ANGLE= parameter in the MODEL statement in PROC KRIGE2D should be 45^o (all angles are measured clockwise from north).

The terminology associated with geometric anisotropy is that of ellipses. To see how this comes about, consider the following hypothetical set of calculations.

Let $\{Z(r), r \in D\}$ be a geometrically anisotropic process, and assume that there are sufficient data points to calculate an experimental semivariogram at a large number of angle classes $\theta \in \{0, \delta\theta, 2\delta\theta, \cdots, 180^o\}$. At each of these angles θ, the experimental semivariogram is plotted and the effective range r_ϵ is recorded. A diagram, in polar coordinates, of (r_ϵ, θ) yields an ellipse, with the major axis in the direction of the largest r_ϵ and the minor axis perpendicular. Denote the largest r_ϵ by r_ϵ^{max}, the smallest by r_ϵ^{min}, and their ratio by

$$R = \frac{r_\epsilon^{min}}{r_\epsilon^{max}}$$

By a rotation, a new set of axes are aligned along the major and minor axis. Then, a rescaling elongates the minor axis so its length equals that of the major axis of the ellipse.

First, the angle θ of the major axis of the ellipse (measured clockwise from north) is transformed to standard Cartesian orientation or counter-clockwise from the x-axis (east). Let $\varphi = 90^o - \theta$ denote the transformed angle. The matrix to transform the distance h is in terms of φ and the ratio R and it is given by

$$\mathbf{H} = \left(\begin{array}{cc} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi)/R & \cos(\varphi)/R \end{array} \right)$$

For a given point pair P_1P_2, with coordinates $(x_1, y_1), (x_2, y_2)$, the transformed interpair distance is computed by first transforming the components $\delta x = x_1 - x_2$ and $\delta y = y_1 - y_2$ by

$$\left(\begin{array}{c} \delta x' \\ \delta y' \end{array} \right) = \mathbf{H} \left(\begin{array}{c} \delta x \\ \delta y \end{array} \right)$$

The transformed interpair distance is then

$$h' = \sqrt{(\delta x')^2 + (\delta y')^2}$$

The original semivariogram, a function of *both* h and θ, is then transformed to a function only of h':

$$\hat{\gamma}(h') = \gamma(h, \theta)$$

This single semivariogram is then used for kriging purposes.

The interpretation of the major/minor axis in the case of geometric anisotropy is that the direction of the major axis is the direction in which the spatial process $\{Z(r), r \in D\}$ is most highly correlated; the process is least correlated in the perpendicular direction.

In some cases, these directions are known a priori. This can occur in mining applications where the geology of a region is known in advance. In most cases, however, nothing is known about possible anisotropy. Depending on the amount of data available, using four to six directions is usually sufficient to determine the presence of anisotropy and find the approximate major/minor axis directions.

The most convenient way of performing this is to use the NDIR= option in the COMPUTE statement in PROC VARIOGRAM to obtain a separate experimental semivariogram for each direction. After determining the direction of the major axis, use a DIRECTIONS statement on a subsequent run of PROC VARIOGRAM with this direction and its perpendicular direction. For example, if the initial run of PROC VARIOGRAM with NDIR=6 in the COMPUTE statement indicates that $\theta = 45^o$ is the major axis (has the largest r_ϵ), then rerun PROC VARIOGRAM with

```
DIRECTIONS 45,135;
```

Then, determine the ratio of r_ϵ for the minor and major axis for the RATIO= parameter in the COMPUTE statement of PROC KRIGE2D. This ratio is ≤ 1 for modeling geometric anisotropy. In the other type of anisotropy, *zonal* anisotropy, the RATIO= parameter is set to a large number for reasons explained in the following section.

Zonal Anisotropy

In zonal anisotropy, either the *form* covariance structure or the parameter c_0 (or both) is different in different directions. In particular, the sill is different for different directions. In geologic applications, this is the more common type of anisotropy. It is not possible to transform such a structure into an isotropic semivariogram.

Instead, nesting and geometric anisotropy are used together to approximate zonal anisotropy. For example, suppose the spatial process has a correlation structure in the N–S direction described by $\gamma_{z,1}$, a spherical model with sill at $c_0 = 6$ and range $a_0 = 2$, while in the E–W direction the correlation structure, described by $\gamma_{z,2}$, is again a spherical model but with sill at $c_0 = 3$ and range $a_0 = 1$.

You can approximate this structure in PROC KRIGE2D by specifying two nested models with large RATIO= values. In particular, the appropriate MODEL statement is

```
MODEL FORM=(S,S) ANGLE=(0,90) SCALE=(6,3)
      RANGE=(2,1) RATIO=(1E8,1E8);
```

The large values of the RATIO= parameter for each nested structure have the effect of an “infinite” range parameter in the direction of the minor axis. Hence, there is no variation in $\gamma_{z,1}$ in the E–W direction and no variation in $\gamma_{z,2}$ in the N–S direction.

Anisotropic Nugget Effect

Note that an isotropic nugget effect can be approximated by using nested models, with one of the nested models having a small range. Applying a geometric anisotropy specification to this nested structure results in an anisotropic nugget effect.

Details of Ordinary Kriging

Introduction

There are three common characteristics often observed with spatial data (that is, data indexed by their spatial locations).

(i) slowly varying, large-scale variations in the measured values

(ii) irregular, small-scale variations

(iii) similarity of measurements at locations close together

As an illustration, consider a hypothetical example in which an organic solvent leaks from an industrial site and spreads over a large area. Assume the solvent is absorbed and immobilized into the subsoil above any ground-water level, so you can ignore any time dependence.

For you to find the areal extent and the concentration values of the solvent, measurements are required. Although the problem is inherently three-dimensional, if you measure total concentration in a column of soil or take a depth-averaged concentration, it can be handled reasonably well with two-dimensional techniques.

You usually assume that measured concentrations are higher closer to the source and decrease at larger distances from the source. On top of this smooth variation, there are small-scale variations in the measured concentrations, due perhaps to the inherent variability of soil properties.

You also tend to suspect that measurements made close together yield similar concentration values, while measurements made far apart can have very different values.

These physically reasonable qualitative statements have no explicit probabilistic content, and there are a number of numerical smoothing techniques, such as inverse distance weighting and splines, that make use of large-scale variations and “close distance-close value” characteristics of spatial data to interpolate the measured concentrations for contouring purposes.

While characteristics (i) and (iii) are handled by such smoothing methods, characteristic (ii), the small-scale residual variation in the concentration field, is not accounted for.

There may be situations, due to the use of the prediction map or due to the relative magnitude of the irregular fluctuations, where you cannot ignore these small-scale irregular fluctuations. In other words, the smoothed or estimated values of the concentration field alone are not a sufficient characterization; you also need the possible spread around these contoured values.

Spatial Random Fields

One method of incorporating characteristic (ii) into the construction of a contour map is to model the concentration field as a spatial random field (SRF). The mathematical details of SRF models are given in a number of texts, for example, Cressie (1993) and Christakos (1992). The mathematics of SRFs are formidable. However, under certain simplifying assumptions, they produce classical linear estimators with very simple properties, allowing easy implementation for prediction purposes. These estimators, primarily ordinary kriging (OK), give both a prediction and a standard error of prediction at unsampled locations. This allows the construction of a map of both predicted values and level of uncertainty about the predicted values.

The key assumption in applying the SRF formalism is that the measurements come from a single realization of the SRF. However, in most geostatistical applications, the focus is on a single, unique realization. This is unlike most other situations in stochastic modeling in which there will be future experiments or observational activities (at least conceptually) under similar circumstances. This renders many traditional ideas of statistical inference ambiguous and somewhat counterintuitive.

There are additional logical and methodological problems in applying a stochastic model to a unique but partly unknown natural process; refer to the introduction in Matheron (1971) and Cressie (1993, section 2.3). These difficulties have resulted in attempts to frame the estimation problem in a completely deterministic way (Isaaks and Srivastava 1988; Journel 1985).

Additional problems with kriging, and with spatial estimation methods in general, are related to the necessary assumption of ergodicity of the spatial process. This assumption is required to estimate the covariance or semivariogram from sample data. Details are provided in Cressie (1993, pp. 52–58).

Despite these difficulties, ordinary kriging remains a popular and widely used tool in modeling spatial data, especially in generating surface plots and contour maps. An abbreviated derivation of the OK estimator for point estimation and the associated standard error is discussed in the following section. Full details are given in Journel and Huijbregts (1978), Christakos (1992), and Cressie (1993).

Ordinary Kriging

Denote the SRF by $Z(r), r \in D \subset \mathcal{R}^2$. Following the notation in Cressie (1993), the following model for $Z(r)$ is assumed:

$$Z(r) = \mu + \varepsilon(r)$$

Here, μ is the fixed, unknown mean of the process, and $\varepsilon(r)$ is a zero mean SRF representing the variation around the mean.

In most practical applications, an additional assumption is required in order to estimate the covariance C_z of the $Z(r)$ process. This assumption is second-order stationarity:

$$C_z(r_1, r_2) = E[\varepsilon(r_1)\varepsilon(r_2)] = C_z(r_1 - r_2)$$

This requirement can be relaxed slightly when you are using the semivariogram instead of the covariance. In this case, second-order stationarity is required of the differences $\varepsilon(r_1) - \varepsilon(r_2)$ rather than $\varepsilon(r)$:

$$\gamma_Z(r_1, r_2) = \frac{1}{2}E[\varepsilon(r_1) - \varepsilon(r_2)]^2 = \gamma_Z(r_1 - r_2)$$

By performing local kriging, the spatial processes represented by the previous equation for Z(*r*) are more general than they appear. In local kriging, at an unsampled location r_0, a separate model is fit using only data in a neighborhood of r_0. This has the effect of fitting a separate mean μ at each point, and it is similar to the "kriging with trend" (KT) method discussed in Journel and Rossi (1989).

Given the N measurements $Z(r_1), \ldots, Z(r_N)$ at known locations $r_1, \ldots, r_N$, you want to obtain an estimate $\hat{Z}$ of Z at an unsampled location r_0. When the following three requirements are imposed on the estimator $\hat{Z}$, the OK estimator is obtained.

(i) $\hat{Z}$ is linear in $Z(r_1), \cdots, Z(r_N)$.

(ii) $\hat{Z}$ is unbiased.

(ii) $\hat{Z}$ minimizes the mean-square prediction error $E(Z(r_0) - \hat{Z}(r_0))^2$.

Linearity requires the following form for $\hat{Z}(r_0)$:

$$\hat{Z}(r_0) = \sum_{i=1}^{N} \lambda_i Z(r_i)$$

Applying the unbiasedness condition to the preceding equation yields

$$E\hat{Z}(r_0) = \mu \Rightarrow \mu = \sum_{i=1}^{N} \lambda_i EZ(r_i) \Rightarrow$$

$$\sum_{i=1}^{N} \lambda_i \mu = \mu \Rightarrow \sum_{i=1}^{N} \lambda_i = 1$$

Finally, the third condition requires a constrained linear optimization involving $\lambda_1, \cdots, \lambda_N$ and a Lagrange parameter $2m$. This constrained linear optimization can be expressed in terms of the function $L(\lambda_1, \cdots, \lambda_N, m)$ given by

$$L = E\left(Z(r_0) - \sum_{i=1}^{N} \lambda_i Z(r_i)\right)^2 - 2m\left(\sum_{i=1}^{N} \lambda_i - 1\right)$$

Define the $N \times 1$ column vector $\boldsymbol{\lambda}$ by

$$\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_N)^T$$

and the $(N+1) \times 1$ column vector $\boldsymbol{\lambda_0}$ by

$$\boldsymbol{\lambda_0} = (\lambda_1, \cdots, \lambda_N, m)^T = \begin{pmatrix} \boldsymbol{\lambda} \\ m \end{pmatrix}$$

The optimization is performed by solving

$$\frac{\partial L}{\partial \boldsymbol{\lambda_0}} = \mathbf{0}$$

in terms of $\lambda_1, \cdots, \lambda_N$ and m.

The resulting matrix equation can be expressed in terms of either the covariance $C_z(r)$ or semivariogram $\gamma_z(r)$. In terms of the covariance, the preceding equation results in the following matrix equation:

$$\mathbf{C}\boldsymbol{\lambda_0} = \mathbf{C_0}$$

where

$$\mathbf{C} = \begin{pmatrix} C_z(0) & C_z(r_1 - r_2) & \cdots & C_z(r_1 - r_N) & 1 \\ C_z(r_2 - r_1) & C_z(0) & \cdots & C_z(r_2 - r_N) & 1 \\ & & \ddots & & \\ C_z(r_N - r_1) & C_z(r_N - r_2) & \cdots & C_z(0) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}$$

and

$$\mathbf{C_0} = \begin{pmatrix} C_z(r_0 - r_1) \\ C_z(r_0 - r_2) \\ \vdots \\ C_z(r_0 - r_N) \\ 1 \end{pmatrix}$$

The solution to the previous matrix equation is

$$\hat{\boldsymbol{\lambda}}_{\mathbf{0}} = \mathbf{C^{-1}C_0}$$

Using this solution for $\boldsymbol{\lambda}$ and m, the ordinary kriging estimate at r_0 is

$$\hat{Z}(r_0) = \lambda_1 Z(r_1) + \cdots + \lambda_N Z(r_N)$$

with associated prediction error

$$\sigma_z(r_0) = C_z(0) - \boldsymbol{\lambda}'\mathbf{c_0} + m$$

where $\mathbf{c_0}$ is $\mathbf{C_0}$ with the 1 in the last row removed, making it an $N \times 1$ vector.

These formulas are used in the best linear unbiased prediction (BLUP) of random variables (Robinson 1991). Further details are provided in Cressie (1993, pp. 119–123).

Because of possible numeric problems when solving the previous matrix equation, Duetsch and Journel (1992) suggest replacing the last row and column of 1s in the preceding matrix $\mathbf{C}$ by $C_z(0)$, keeping the 0 in the $(N+1, N+1)$ position and similarly replacing the last element in the preceding right-hand vector $\mathbf{C_0}$ with $C_z(0)$. This results in an equivalent system but avoids numeric problems when $C_z(0)$ is large or small relative to 1.

Output Data Sets

The KRIGE2D procedure produces two data sets: the OUTEST=*SAS-data-set* and the OUTNBHD=*SAS-data-set*. These data sets are described as follows.

OUTEST=*SAS-data-set*

The OUTEST= data set contains the kriging estimates and the associated standard errors. The OUTEST= data set contains the following variables:

- ESTIMATE, which is the kriging estimate for the current variable
- GXC, which is the x-coordinate of the grid point at which the kriging estimate is made
- GYC, which is the y-coordinate of the grid point at which the kriging estimate is made
- LABEL, which is the label for the current PREDICT/MODEL combination producing the kriging estimate. If you do not specify a label, default labels of the form Pred*j*.Model*k* are used.
- NPOINTS, which is the number of points used in the estimation. This number varies for each grid point if local kriging is performed.
- STDERR, which is the standard error of the kriging estimate
- VARNAME, which is the variable name

OUTNBHD=*SAS-data-set*

When you specify the RADIUS= option or the NUMPOINTS= option in the PREDICT statement, local kriging is performed. Local kriging is simply ordinary kriging at a given grid location using only those data points in a neighborhood defined by the RADIUS= value or the NUMPOINTS= value.

The OUTNBHD= data set contains one observation for each data point in each neighborhood. Hence, this data set can be large. For example, if the grid specification results in 1,000 grid points and each grid point has a neighborhood of 100 points, the resulting OUTNBHD= data set contains 100,000 points.

The OUTNBHD= data set contains the following variables:

- GXC, which is the x-coordinate of the grid point
- GYC, which is the y-coordinate of the grid point
- LABEL, which is the label for the current PREDICT/MODEL combination. If you do not specify a label, default labels of the form Pred*j*.Model*k* are used.
- NPOINTS, which is the number of points used in the estimation
- RADIUS, which is the radius used for each neighborhood
- VALUE, which is the value of the variable at the current data point
- VARNAME, which is the variable name of the current variable
- XC, which is the x-coordinate of the current data point
- YC, which is the y-coordinate of the current data point

Computational Resources

To generate a predicted value at a single grid point using N data points, PROC KRIGE2D must solve the following kriging system:

$$\mathbf{C}\boldsymbol{\lambda}_\mathbf{0} = \mathbf{C_0}$$

where $\mathbf{C}$ is $(N+1) \times (N+1)$, and the right-hand side vector $\mathbf{C_0}$ is $(N+1) \times 1$.

Holding the matrix and vector associated with this system in core requires approximately $\frac{N^2}{2}$ doubles (with typically eight bytes per double). The CPU time used in solving the system is proportional to N^3. For large N, this time dominates the time to compute the $\frac{(N+1)(N+2)}{2}$ elements of the covariance matrix $\mathbf{C}$ from the specified covariance or variogram model. This latter computation is proportional to N^2.

For local kriging, the kriging system is set up and solved for each grid point. Part of the set up process involves determining the neighborhood of each grid point. A fast K-D tree algorithm is used to determine neighborhoods. For G grid points, the dominant CPU time factor is setting up and solving the G kriging systems. The N in the preceding algorithm is the number of data points in a given neighborhood, and it can differ for each grid point.

In global kriging, the entire input data set and all grid points are used to set up and solve the single system

$$\mathbf{C}\boldsymbol{\lambda}_\mathbf{0} = \mathbf{C_0}$$

Again $\mathbf{C}$ is $(N+1) \times (N+1)$, but $\boldsymbol{\lambda}_\mathbf{0}$ is now $(N+1) \times G$, where G is the number of grid points, and N is the number of nonmissing observations in the input data set. The right-hand side matrix $\mathbf{C_0}$ is $(N+1) \times G$. Memory requirements are approximately $\frac{N^2}{2} + GN$ doubles. The CPU time used in solving the system is still dominated by the N^3 factorization of the left-hand side.

Example

Example 37.1. Investigating the Effect of Model Specification on Prediction

In the "Getting Started" section of the chapter on the VARIOGRAM procedure, a particular variogram is chosen for the coal seam thickness data. The chosen variogram is Gaussian with a scale (sill) of $c_0 = 7.5$, and a range of $a_0 = 30$. This choice of the variogram is based on a visual fit—a comparison of the plots of the regular and robust sample variograms and the Gaussian variogram for various scale (sill) and range values.

Another possible choice of model is the spherical variogram with the same scale (sill) of $c_0 = 7.5$ but with a range of $a_0 = 60$. This choice of range is again based on a visual fit; while not as good as the Gaussian model, the fit is reasonable.

It is generally held that spatial prediction is robust against model specification, while the standard error computation is not so robust.

This example investigates the effect of using these different models on the prediction and associated standard errors.

```
data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;

/*- Run KRIGE2D on original Gaussian model  ------------*/
proc krige2d data=thick outest=est1;
   pred var=thick r=60;
   model scale=7.5 range=30 form=gauss;
   coord xc=east yc=north;
   grid x=0 to 100 by 10 y=0 to 100 by 10;
run;

/*- Run KRIGE2D using Spherical Model, modified range  -*/
proc krige2d data=thick outest=est2;
   pred var=thick r=60;
   model scale=7.5 range=60 form=spherical;
   coord xc=east yc=north;
   grid x=0 to 100 by 10 y=0 to 100 by 10;
run;

data compare ;
   merge est1(rename=(estimate=g_est stderr=g_std))
         est2(rename=(estimate=s_est stderr=s_std));
   est_dif=g_est-s_est;
   std_dif=g_std-s_std;
run;

proc print data=compare;
   title ’Comparison of Gaussian and Spherical Models’;
   title2 ’Differences of Estimates and Standard Errors’;
   var gxc gyc npoints g_est s_est est_dif  g_std s_std
                       std_dif;
run;
```

Output 37.1.1. Comparison of Gaussian and Spherical Models

```
                    Comparison of Gaussian and Spherical Models
            Differences of Estimates and Standard Errors: First 50 Observations

Obs   GXC   GYC   NPOINTS    g_est     s_est     est_dif    g_std     s_std     std_dif

  1     0     0      23     43.9408  42.6700   1.27087  0.68260  2.05947  -1.37687
  2     0    10      28     41.6828  41.6780   0.00483  0.55909  2.03464  -1.47554
  3     0    20      31     38.9601  39.7285  -0.76843  0.30185  1.93478  -1.63293
  4     0    30      32     36.1701  37.3275  -1.15739  0.12705  1.54844  -1.42139
  5     0    40      39     33.8376  35.4320  -1.59440  0.04872  1.37821  -1.32949
  6     0    50      38     32.8375  34.3930  -1.55550  0.02983  1.22584  -1.19602
  7     0    60      35     33.9576  34.3155  -0.35785  0.00195  0.54122  -0.53927
  8     0    70      30     36.9502  37.6669  -0.71673  0.04006  1.20451  -1.16444
  9     0    80      31     41.1097  41.1016   0.00812  0.04705  0.99544  -0.94839
 10     0    90      28     43.6671  42.5216   1.14546  0.10236  1.57357  -1.47121
 11     0   100      23     41.9443  42.6511  -0.70681  0.53646  2.20792  -1.67146
 12    10     0      25     44.6795  44.1959   0.48355  0.07833  1.09743  -1.01910
 13    10    10      31     42.8397  42.7496   0.09008  0.10982  1.46686  -1.35703
 14    10    20      34     40.3120  40.3634  -0.05140  0.05315  1.54889  -1.49574
 15    10    30      39     37.7593  37.7648  -0.00544  0.00889  0.94136  -0.93247
 16    10    40      44     35.6365  35.5471   0.08940  0.00595  0.75920  -0.75325
 17    10    50      44     35.0603  34.7042   0.35612  0.01564  1.05033  -1.03469
 18    10    60      41     36.0716  35.4737   0.59794  0.01321  1.18277  -1.16957
 19    10    70      36     38.1196  38.1040   0.01565  0.00315  0.89157  -0.88842
 20    10    80      33     41.2799  41.0734   0.20644  0.02446  1.22772  -1.20326
 21    10    90      30     43.2193  42.8904   0.32890  0.05988  1.49438  -1.43450
 22    10   100      26     41.0358  43.1350  -2.09918  0.19050  1.93434  -1.74384
 23    20     0      29     44.4890  44.4359   0.05317  0.06179  1.23618  -1.17439
 24    20    10      35     43.3391  43.2938   0.04531  0.00526  0.95512  -0.94986
 25    20    20      39     41.1293  40.9885   0.14079  0.00675  1.18544  -1.17870
 26    20    30      43     38.6060  38.5300   0.07598  0.00898  1.08973  -1.08075
 27    20    40      49     36.5013  36.5275  -0.02623  0.03037  1.33620  -1.30583
 28    20    50      49     36.1158  35.7959   0.31990  0.02535  1.31986  -1.29451
 29    20    60      49     36.8115  36.5397   0.27182  0.00835  1.11490  -1.10656
 30    20    70      39     38.4308  38.5182  -0.08746  0.00257  0.89419  -0.89162
 31    20    80      36     41.0601  41.0449   0.01511  0.00766  1.18548  -1.17781
 32    20    90      33     43.1788  43.1073   0.07144  0.00613  0.94924  -0.94311
 33    20   100      27     42.7757  43.4689  -0.69313  0.06770  1.52094  -1.45324
 34    30     0      35     43.3601  43.9579  -0.59779  0.04662  1.32306  -1.27644
 35    30    10      39     43.1539  43.1448   0.00912  0.00245  0.72413  -0.72167
 36    30    20      44     41.2400  41.2166   0.02336  0.00528  1.10234  -1.09706
 37    30    30      52     38.9296  39.0178  -0.08816  0.00489  1.04501  -1.04012
 38    30    40      57     37.2813  37.3412  -0.05992  0.00804  0.89242  -0.88438
 39    30    50      57     36.7198  36.7558  -0.03597  0.00652  0.83517  -0.82865
 40    30    60      55     37.2047  37.3407  -0.13597  0.00682  1.00330  -0.99648
 41    30    70      48     38.8856  38.8919  -0.00628  0.00285  1.01430  -1.01145
 42    30    80      43     41.0627  41.0663  -0.00359  0.00260  0.97336  -0.97077
 43    30    90      36     43.0969  43.0465   0.05038  0.00194  0.51312  -0.51118
 44    30   100      29     44.5840  43.3474   1.23663  0.13593  1.57267  -1.43674
 45    40     0      36     42.8186  43.5157  -0.69706  0.01976  1.25689  -1.23713
 46    40    10      40     42.8970  42.9168  -0.01984  0.00301  0.95163  -0.94862
 47    40    20      52     41.1025  41.1824  -0.07989  0.00193  0.96204  -0.96012
 48    40    30      60     39.3288  39.2992   0.02960  0.00451  1.05561  -1.05111
 49    40    40      67     38.2096  37.9680   0.24161  0.01791  1.29139  -1.27349
 50    40    50      68     37.3139  37.5055  -0.19150  0.04039  1.51095  -1.47056
```

The predicted values at each of the grid locations do not differ greatly for the two variogram models. However, the standard error of prediction for the spherical model is substantially larger than the Gaussian model.

References

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Cressie, N.A.C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons, Inc.

Duetsch, C.V. and Journel, A.G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Hohn, M.E. (1988), *Geostatistics and Petroleum Geology*, New York: Van Nostrand Reinhold.

Isaaks, E.H. and Srivastava, R.M. (1988), "Spatial Continuity Measures for Probabilistic and Deterministic Geostatistics," *Mathematical Geology*, 20 (4), 313–341.

Journel, A.G. (1985), "The Deterministic Side of Geostatistics," *Mathematical Geology*, 17 (1), 1–15.

Journel, A.G. and Huijbregts, Ch.J. (1978), *Mining Geostatistics*, New York: Academic Press.

Journel, A.G. and Rossi, M. (1989), "When Do We Need a Trend Model in Kriging?" *Mathematical Geology*, 21 (7), 715–739.

Matheron, G. (1971), *The Theory of Regionalized Variables and Its Applications*, Les Cahiers du Centre de Morphologie Mathematique de Fontainebleau.

Robinson, G.K. (1991), "That BLUP is a Good Thing: The Estimation of Random Effects," *Statistical Science*, 6, 15–51.

Chapter 38

The LATTICE Procedure

Chapter Contents

Chapter 38
The LATTICE Procedure

Overview

The LATTICE procedure computes the analysis of variance and analysis of simple covariance for data from an experiment with a lattice design. PROC LATTICE analyzes balanced square lattices, partially balanced square lattices, and some rectangular lattices.

In balanced square lattices, the number of treatments is equal to the square of the number of units per block. Incomplete blocks are grouped to form mutually orthogonal replications. The number of replicates in the basic plan is always 1 plus the number of units per block.

The partially balanced square lattices are similar to balanced lattices, although the number of replicates may vary. Partially balanced designs are constructed of the replicates in the basic plan, but not all replicates are included the same number of times, and some may not be included at all.

In rectangular lattices, there are k units per block and $k(k+1)$ treatments. As in square lattices, blocks are grouped to form mutually orthogonal replicates in the basic plan. PROC LATTICE can analyze simple rectangular lattices (two orthogonal replications) and triple rectangular lattices (three orthogonal replications). The experiment can include several repetitions of the basic plan.

The LATTICE procedure determines from the data set which type of design has been used. It also checks to see if the design is valid and displays an appropriate message if it is not.

Getting Started

An example of a balanced square design is an experiment to investigate the effects of nine diets on the growth rate of pigs.

In some breeds of pigs, past experience has shown that a large part of the total variation in growth rates between animals can be attributed to the litter. Therefore, this experiment is planned so that litter differences do not contribute to the intrablock error.

First, the pigs are separated into sets of three litter-mates. Each block is assigned two sets of the three litter-mates. In a given block, one pig from each set receives a diet. Therefore, the experimental unit is a pair of pigs feeding in a particular pen on one of the nine diets. The response variable, growth rate, is the sum of the growth rates for the two pigs in a particular pen. To get the adjusted diet mean per pig, the adjusted treatment mean for the pen must be divided by 2.

The special numeric SAS variables named Group, Block, Treatmnt, and Rep must be used to define the design. In this example, the Treatmnt variable ranges from 1 to 9 and indicates the particular diet. The Block variable is 1, 2, or 3 and indicates the pen containing the two pigs. The Group variable ranges from 1 to 4 and specifies which replication within the basic plan includes the experimental unit. In this example, you would not use the Rep variable since the entire basic plan is not replicated.

You can use the following DATA step and PROC LATTICE statement to analyze this experiment. The response variable is Weight.

```
title 'Examining the Growth Rate of Pigs';
data Pigs;
   input Group Block Treatmnt Weight @@;
   datalines;
1 1 1 2.20  1 1 2 1.84  1 1 3 2.18
1 2 4 2.05  1 2 5 0.85  1 2 6 1.86
1 3 7 0.73  1 3 8 1.60  1 3 9 1.76
2 1 1 1.19  2 1 4 1.20  2 1 7 1.15
2 2 2 2.26  2 2 5 1.07  2 2 8 1.45
2 3 3 2.12  2 3 6 2.03  2 3 9 1.63
3 1 1 1.81  3 1 5 1.16  3 1 9 1.11
3 2 2 1.76  3 2 6 2.16  3 2 7 1.80
3 3 3 1.71  3 3 4 1.57  3 3 8 1.13
4 1 1 1.77  4 1 6 1.57  4 1 8 1.43
4 2 2 1.50  4 2 4 1.60  4 2 9 1.42
4 3 3 2.04  4 3 5 0.93  4 3 7 1.78
;

proc lattice data=Pigs;
   var Weight;
run;
```

The SAS code produces the output shown in Figure 38.1.

```
                        Examining the Growth Rate of Pigs

                              The Lattice Procedure

                         Analysis of Variance for Weight

                                                           Sum of         Mean
      Source                                      DF      Squares       Square

      Replications                                 3      0.07739      0.02580
      Blocks within Replications (Adj.)            8       1.4206       0.1776
         Component B                               8       1.4206       0.1776
      Treatments (Unadj.)                          8       3.2261       0.4033
      Intra Block Error                           16       1.2368      0.07730
      Randomized Complete Block Error             24       2.6574       0.1107
      Total                                       35       5.9609       0.1703


                        Additional Statistics for Weight

                Description                                Value

                Variance of Means in Same Block          0.04593
                LSD at .01 Level                          0.6259
                LSD at .05 Level                          0.4543
                Efficiency Relative to RCBD               120.55
```

```
                        Examining the Growth Rate of Pigs

                              The Lattice Procedure

                                Adjusted Treatment
                                 Means for Weight

                              Treatment         Mean

                                      1       1.8035
                                      2       1.7544
                                      3       1.9643
                                      4       1.7267
                                      5       0.9393
                                      6       1.8448
                                      7       1.3870
                                      8       1.4347
                                      9       1.5004
```

Figure 38.1. Output from Example LATTICE Procedure

Diet 3 yields the highest mean growth rate at 1.9643 pounds for the two pigs (0.9822 per pig), while diet 5 has the lowest rate at 0.9393 (0.4696 per pig). The efficiency of the experiment relative to a randomized complete block design is 120.55 percent, so using the lattice design increased precision, producing more accurate estimates of the treatment effects. The different elements of the LATTICE procedure's output are discussed in the "Displayed Output" section on page 2074.

Syntax

The following statements are available in PROC LATTICE.

```
PROC LATTICE < options >;
    BY variables ;
    VAR variables ;
```

Three specific numeric SAS variables, Group, Block, and Treatmnt, *must* be present in the data set to which PROC LATTICE is applied. A fourth numeric variable named Rep must be present when the design involves repetition of the entire basic plan. (See the "Input Data Set" section on page 2073 for more information.)

Every numeric variable other than Group, Block, Treatmnt, or Rep in the input SAS data set may be considered a response variable. A VAR statement tells PROC LATTICE that only the variables listed in the VAR statement are to be considered response variables. If the VAR statement is omitted, then all numeric variables, excluding Group, Block, Treatmnt, and Rep, are considered response variables. PROC LATTICE performs an analysis for each response variable.

PROC LATTICE Statement

PROC LATTICE < *options* >**;**

You can specify the following options in the PROC LATTICE statement.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC LATTICE. If you omit the DATA= option, the most recently created SAS data set is used.

COVARIANCE
COV

calculates sums of products for every possible pair of response variables. A sum of products is given for each source of variation in the analysis of variance table. For each pair of response variables, the one appearing later in the data set (or in the VAR statement) is the covariable.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LATTICE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LATTICE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the response variables. If you do not include a VAR statement, all numeric variables in the data set are considered response variables (except Group, Block, Treatmnt, and Rep).

Details

Input Data Set

Four numeric SAS variables, Group, Block, Treatmnt, and Rep, are used in the input data set to define the lattice design. The Group, Block, and Treatmnt variables are required in the data set to which PROC LATTICE is applied. The Rep variable must be present when the design involves repetition of the entire basic plan.

Group specifies which orthogonal replication in the basic plan includes the experimental unit. Values of Group must be 1, 2, . . . , *n*, where *n* is the number of replicates in the basic plan.

Block specifies the block in which the experimental unit is present. Values of Block must be 1, 2, . . . , *m*, where *m* is the number of blocks in a replication.

Treatmnt specifies which treatment was applied to the experimental unit. Values of Treatmnt must be 1, 2, . . . , *i*, where *i* is the number of treatments in a replication.

Rep specifies which repetition of the basic plan includes the experimental unit. Values of Rep must be 1, 2, . . . , *p*, where *p* is the number of replications of the entire basic plan. Thus, the experiment has a total of *np* replicates.

Missing Values

If a value of Group, Block, Treatmnt, or Rep is missing, the analysis is not performed and an appropriate error message is displayed.

If a value of a response variable is missing, this entire variable is dropped from the analysis. If other response variables exist that do not have missing values, they are analyzed.

Displayed Output

For each response variable, PROC LATTICE displays

- an "Analysis of Variance" table and related statistics, including the following as separate sources of variations:

  - Replications
  - Blocks within Replications (adjusted for treatments)
  - Treatments (unadjusted)
  - Intra Block Error
  - Randomized Complete Block Error

  The Blocks within Replications sum of squares is further broken down into "Component A" and "Component B." If there is no repetition of the basic plan, the Component B sum of squares is the same as the Blocks within Replications sum of squares. If there is repetition of the basic plan, the Component A sum of squares reflects the variation among blocks that contain the same treatments.

  The source of variation called Randomized Complete Block Error is the sum of the Blocks within Replications sum of squares and the Intra Block Error sum of squares. It is the appropriate error term if the experimental design is a randomized complete block design, with the replications filling the roles of complete blocks.

- two values for the Variance of Means. For some lattice designs, these are only approximations. The first value is applicable when the two treatments appear in the same block; the other (when it appears) applies when the two treatments never appear in the same block (a possibility in partially balanced and rectangular designs).
- an Average of Variance. Except with small designs, it is sufficient to use this average variance of means for tests between treatments (whether the two treatments appear in the same block or not); refer to Cochran and Cox (1957).

- the Least Significant Differences (LSDs) at the .01 and .05 levels of significance, based on the Average of Variance
- Efficiency Relative to RCBD, the efficiency of the lattice design relative to a randomized complete block design. The efficiency is the ratio of the randomized complete block mean squared error to the effective error variance; refer to Cochran and Cox (1957).
- the Adjusted Treatment Means. These are adjusted for blocks if the relative precision is greater than 105 percent.

When you specify the COVARIANCE option, PROC LATTICE produces Sums of Products and the Mean Product for each source of variation in the analysis of variance table.

ODS Table Names

PROC LATTICE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 38.1. ODS Tables Produced in PROC LATTICE

ODS Table Name	**Description**	**PROC LATTICE Option**
ANOVA	Analysis of variance	default
AdjTreatmentMeans	Adjusted treatment means	default
Statistics	Additional statistics	default

Example

Example 38.1. Analysis of Variance through PROC LATTICE

In the following example, from Cochran and Cox (1957, p. 406), the data are yields (Yield) in bushels per acre of 25 varieties (Treatmnt) of soybeans. The data are collected in two replications (Group) of 25 varieties in five blocks (Block) containing five varieties each. This is an example of a partially balanced square lattice design.

```
data Soy;
  do Group = 1 to 2;
    do Block = 1 to 5;
      do Plot = 1 to 5;
         input Treatmnt Yield @@;
         output;
      end;
    end;
  end;
  drop Plot;
  datalines;
1   6   2   7   3   5   4   8   5   6
6   16  7   12  8  12   9   13  10  8
11  17  12  7   13  7   14  9   15  14
16 18   17  16  18  13  19  13  20  14
21  14  22  15  23  11  24  14  25  14
1   24  6   13  11  24  16  11  21  8
2   21  7   11  12  14  17  11  22  23
3   16  8   4   13  12  18  12  23  12
4   17  9   10  14  30  19  9   24  23
5   15  10  15  15  22  20  16  25  19
;

proc print data=Soy;
   id Treatmnt;
run;

proc lattice data=Soy;
run;
```

The results from these statements are shown in Output 38.1.1 and Output 38.1.2.

Output 38.1.1. Displayed Output from PROC PRINT

```
                       Treatmnt    Group    Block    Yield

                           1         1        1        6
                           2         1        1        7
                           3         1        1        5
                           4         1        1        8
                           5         1        1        6
                           6         1        2       16
                           7         1        2       12
                           8         1        2       12
                           9         1        2       13
                          10         1        2        8
                          11         1        3       17
                          12         1        3        7
                          13         1        3        7
                          14         1        3        9
                          15         1        3       14
                          16         1        4       18
                          17         1        4       16
                          18         1        4       13
                          19         1        4       13
                          20         1        4       14
                          21         1        5       14
                          22         1        5       15
                          23         1        5       11
                          24         1        5       14
                          25         1        5       14
                           1         2        1       24
                           6         2        1       13
                          11         2        1       24
                          16         2        1       11
                          21         2        1        8
                           2         2        2       21
                           7         2        2       11
                          12         2        2       14
                          17         2        2       11
                          22         2        2       23
                           3         2        3       16
                           8         2        3        4
                          13         2        3       12
                          18         2        3       12
                          23         2        3       12
                           4         2        4       17
                           9         2        4       10
                          14         2        4       30
                          19         2        4        9
                          24         2        4       23
                           5         2        5       15
                          10         2        5       15
                          15         2        5       22
                          20         2        5       16
                          25         2        5       19
```

Output 38.1.2. Displayed Output from PROC LATTICE

```
                           The Lattice Procedure

                        Analysis of Variance for Yield

                                                      Sum of        Mean
      Source                                  DF     Squares      Square

      Replications                             1      212.18      212.18
      Blocks within Replications (Adj.)        8      501.84     62.7300
         Component B                           8      501.84     62.7300
      Treatments (Unadj.)                     24      559.28     23.3033
      Intra Block Error                       16      218.48     13.6550
      Randomized Complete Block Error         24      720.32     30.0133
      Total                                   49     1491.78     30.4445


                        Additional Statistics for Yield

                Description                                Value

                Variance of Means in Same Block          15.7915
                Variance of Means in Different Bloc      17.9280
                Average of Variance                      17.2159
                LSD at .01 Level                         12.1189
                LSD at .05 Level                          8.7959
                Efficiency Relative to RCBD               174.34
```

```
                           The Lattice Procedure

                             Adjusted Treatment
                              Means for Yield

                           Treatment         Mean

                                   1      19.0681
                                   2      16.9728
                                   3      14.6463
                                   4      14.7687
                                   5      12.8470
                                   6      13.1701
                                   7       9.0748
                                   8       6.7483
                                   9       8.3707
                                  10       8.4489
                                  11      23.5511
                                  12      12.4558
                                  13      12.6293
                                  14      20.7517
                                  15      19.3299
                                  16      12.6224
                                  17      10.5272
                                  18      10.7007
                                  19       7.3231
                                  20      11.4013
                                  21      11.6259
                                  22      18.5306
                                  23      12.2041
                                  24      17.3265
                                  25      15.4048
```

The efficiency of the experiment relative to a randomized complete block design is 174.34 percent. Precision is gained using the lattice design via the recovery of intrablock error information, enabling more accurate estimates of the treatment effects. Variety 8 of soybean had the lowest adjusted treatment mean (6.7483 bushels per acre), while variety 11 of soybean had the highest adjusted treatment mean (23.5511 bushels per acre).

References

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons, Inc.

Comstock, R.E., Peterson, W.J., and Stewart, H.A. (1948), “An Application of the Balanced Lattice Design in a Feeding Trial with Swine,” *Journal of Animal Science*, 7, 320–331.

Cornelius, P.L. (1983), “Lattice Designs,” *Encyclopedia of Statistical Sciences*, 4, 510–518.

Robinson, H.F. and Watson, G.S. (1949), “Analysis of Simple and Triple Rectangular Designs,” *North Carolina Agricultural Experiment Station Technical Bulletin*, 88.

Chapter 39
The LIFEREG Procedure

Chapter Contents

Chapter 39
The LIFEREG Procedure

Overview

The LIFEREG procedure fits parametric models to failure time data that can be right-, left-, or interval-censored. The models for the response variable consist of a linear effect composed of the covariates and a random disturbance term. The distribution of the random disturbance can be taken from a class of distributions that includes the extreme value, normal, logistic, and, by using a log transformation, the exponential, Weibull, lognormal, loglogistic, and three-parameter gamma distributions.

The model assumed for the response $\mathbf{y}$ is

$$\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon$$

where $\mathbf{y}$ is a vector of response values, often the log of the failure times, $\mathbf{X}$ is a matrix of covariates or independent variables (usually including an intercept term), β is a vector of unknown regression parameters, σ is an unknown scale parameter, and ϵ is a vector of errors assumed to come from a known distribution (such as the standard normal distribution). The distribution may depend on additional shape parameters. These models are equivalent to accelerated failure time models when the log of the response is the quantity being modeled. The effect of the covariates in an accelerated failure time model is to change the scale, and not the location, of a baseline distribution of failure times.

The LIFEREG procedure estimates the parameters by maximum likelihood using a Newton-Raphson algorithm. PROC LIFEREG estimates the standard errors of the parameter estimates from the inverse of the observed information matrix.

The accelerated failure time model assumes that the effect of independent variables on an event time distribution is multiplicative on the event time. Usually, the scale function is $\exp(\mathbf{x_c}'\beta_\mathbf{c})$, where $\mathbf{x_c}$ is the vector of covariate values (not including the intercept term) and $\beta_\mathbf{c}$ is a vector of unknown parameters. Thus, if T_0 is an event time sampled from the baseline distribution corresponding to values of zero for the covariates, then the accelerated failure time model specifies that, if the vector of covariates is $\mathbf{x_c}$, the event time is $T = \exp(\mathbf{x_c}'\beta_\mathbf{c})T_0$. If $y = \log(T)$ and $y_0 = \log(T_0)$, then

$$y = \mathbf{x_c}'\beta_\mathbf{c} + y_0$$

This is a linear model with y_0 as the error term.

In terms of survival or exceedance probabilities, this model is

$$\Pr(T > t \mid \mathbf{x_c}) = \Pr(T_0 > \exp(-\mathbf{x_c}'\beta_\mathbf{c})t)$$

The probability on the left-hand side of the equal sign is evaluated given the value $\mathbf{x_c}$ for the covariates, and the right-hand side is computed using the baseline probability distribution but at a scaled value of the argument. The right-hand side of the equation represents the value of the baseline Survival Function evaluated at $\exp(-\mathbf{x_c}'\beta_c)t$.

Usually, an intercept parameter and a scale parameter are allowed in the model. In terms of the original untransformed event times, the effects of the intercept term and the scale term are to scale the event time and power the event time, respectively. That is, if

$$\log(T_0) = \mu + \sigma \log(T_\epsilon)$$

then

$$T_0 = \exp(\mu) T_\epsilon^\sigma$$

Although it is possible to fit these models to the original response variable using the NOLOG option, it is more common to model the log of the response variable. Because of this log transformation, zero values for the observed failure times are not allowed unless the NOLOG option is specified. Similarly, small values for the observed failure times lead to large negative values for the transformed response. The NOLOG option should only be used if you want to fit a distribution appropriate for the untransformed response, the extreme value instead of the Weibull, for example.

The parameter estimates for the normal distribution are sensitive to large negative values, and care must be taken that the fitted model is not unduly influenced by them. Likewise, values that are extremely large even after the log transformation have a strong influence in fitting the extreme value (Weibull) and normal distributions. You should examine the residuals and check the effects of removing observations with large residuals or extreme values of covariates on the model parameters. The logistic distribution gives robust parameter estimates in the sense that the estimates have a bounded influence function.

The standard errors of the parameter estimates are computed from large sample normal approximations using the observed information matrix. In small samples, these approximations may be poor. Refer to Lawless (1982) for additional discussion and references. You can sometimes construct better confidence intervals by transforming the parameters. For example, large sample theory is often more accurate for $\log(\sigma)$ than σ. Therefore, it may be more accurate to construct confidence intervals for $\log(\sigma)$ and transform these into confidence intervals for σ. The parameter estimates and their estimated covariance matrix are available in an output SAS data set and can be used to construct additional tests or confidence intervals for the parameters. Alternatively, tests of parameters can be based on log-likelihood ratios. Refer to Cox and Oakes (1984) for a discussion of the merits of some possible test methods including score, Wald, and likelihood ratio tests. It is believed that likelihood ratio tests are generally more reliable in small samples than tests based on the information matrix.

The log-likelihood function is computed using the log of the failure time as a response. This log likelihood differs from the log likelihood obtained using the failure time as the response by an additive term of $\sum \log(t_i)$, where the sum is over the noncensored failure times. This term does not depend on the unknown parameters and does not affect parameter or standard error estimates. However, many published values of log likelihoods use the failure time as the basic response variable and, hence, differ by the additive term from the value computed by the LIFEREG procedure.

The classic Tobit model (Tobin 1958) also fits into this class of models but with data usually censored on the left. The data considered by Tobin in his original paper came from a survey of consumers where the response variable is the ratio of expenditures on durable goods to the total disposable income. The two explanatory variables are the age of the head of household and the ratio of liquid assets to total disposable income. Because many observations in this data set have a value of zero for the response variable, the model fit by Tobin is

$$\mathbf{y} = \max(\mathbf{x}'\beta + \epsilon, 0)$$

which is a regression model with left censoring. $\mathbf{x}' = (1, \mathbf{x_c}')$.

Getting Started

The following examples demonstrate how you can use the LIFEREG procedure to fit a parametric model to failure time data.

Suppose you have a response variable y that represents failure time, censor is a binary variable with censor=0 indicating censored values, and x1 and x2 are two linearly independent variables. The following statements perform a typical accelerated failure time model analysis. Higher-order effects such as interactions and nested effects are allowed in the independent variables list, but are not shown in this example.

```
proc lifereg;
   model y*censor(0) = x1 x2;
run;
```

PROC LIFEREG can operate on interval-censored data. The model syntax for specifying the censored interval is

```
proc lifereg;
   model (begin, end) = x1 x2;
run;
```

You can also model binomial data using the *events/trials* syntax for the response, as illustrated in the following statements:

```
proc lifereg;
   model r/n=x1 x2;
run;
```

The variable n represents the number of trials and the variable r represents the number of events.

Modeling Right-Censored Failure Time Data

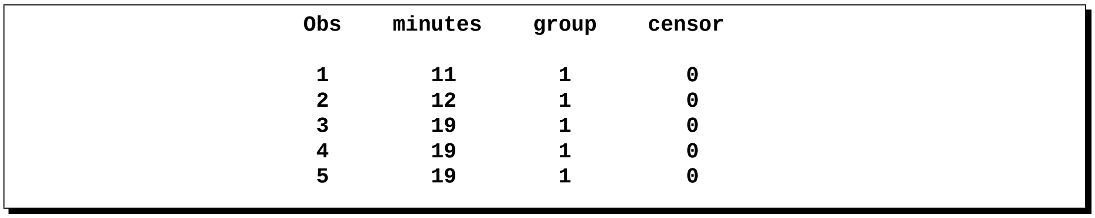
The following example demonstrates how you can use the LIFEREG procedure to fit a model to right-censored failure time data.

Suppose you conduct a study of two headache pain relievers. You divide patients into two groups, with each group receiving a different type of pain reliever. You record the time taken (in minutes) for each patient to report headache relief. Because some of the patients never report relief for the entire study, some of the observations are censored.

The following DATA step creates the SAS data set headache:

```
data headache;
   input minutes group censor @@;
   datalines;
11  1  0   12  1  0   19  1  0   19  1  0
19  1  0   19  1  0   21  1  0   20  1  0
21  1  0   21  1  0   20  1  0   21  1  0
20  1  0   21  1  0   25  1  0   27  1  0
30  1  0   21  1  1   24  1  1   14  2  0
16  2  0   16  2  0   21  2  0   21  2  0
23  2  0   23  2  0   23  2  0   23  2  0
25  2  1   23  2  0   24  2  0   24  2  0
26  2  1   32  2  1   30  2  1   30  2  0
32  2  1   20  2  1
;
```

The data set headache contains the variable minutes, which represents the reported time to headache relief, the variable group, the group to which the patient is assigned, and the variable censor, a binary variable indicating whether the observation is censored. Valid values of the variable censor are 0 (no) and 1 (yes). The following figure shows the first five records of the data set headache.



```
                    Obs    minutes    group    censor

                     1        11        1         0
                     2        12        1         0
                     3        19        1         0
                     4        19        1         0
                     5        19        1         0
```

Figure 39.1. Headache Data

The following statements invoke the LIFEREG procedure:

```
proc lifereg;
   class group;
   model minutes*censor(1)=group;
   output out=new cdf=prob;
run;
```

The CLASS statement specifies the variable group as the classification variable. The MODEL statement syntax indicates that the response variable minutes is right-censored when the variable censor takes the value 1. The MODEL statement specifies the variable group as the single explanatory variable. Because the MODEL statement does not specify the DISTRIBUTION= option, the LIFEREG procedure fits the default type 1 extreme value distribution using log(minutes) as the response. This is equivalent to fitting the Weibull distribution.

The OUTPUT statement creates the output data set new. In addition to the variables in the original data set headache, the SAS data set new also contains the variable prob. This new variable is created by the CDF= option to contain the estimates of the cumulative distribution function evaluated at the observed response.

The results of this analysis are displayed in the following figures.

```
                      The LIFEREG Procedure

                        Model Information

            Data Set                      WORK.HEADACHE
            Dependent Variable             Log(minutes)
            Censoring Variable                   censor
            Censoring Value(s)                        1
            Number of Observations                   38
            Noncensored Values                       30
            Right Censored Values                     8
            Left Censored Values                      0
            Interval Censored Values                  0
            Name of Distribution                Weibull
            Log Likelihood                  -9.37930239


                     Class Level Information

                   Name       Levels    Values

                   group           2    1 2
```

Figure 39.2. Model Fitting Information from the LIFEREG Procedure

Figure 39.2 displays the class level information and model fitting information. There are 30 noncensored observations and 8 right-censored observations. The log likelihood for the Weibull distribution is -9.3793. The log-likelihood value can be used to compare the goodness of fit for different models.

```
                              The LIFEREG Procedure

                           Analysis of Parameter Estimates

                                 Standard   95% Confidence     Chi-
Parameter       DF Estimate     Error       Limits        Square Pr > ChiSq

Intercept        1   3.3091    0.0589   3.1938   3.4245 3161.70     <.0001
group         1  1  -0.1933    0.0786  -0.3473  -0.0393    6.05     0.0139
group         2  0   0.0000    0.0000   0.0000   0.0000     .         .
Scale            1   0.2122    0.0304   0.1603   0.2809
Weibull Shape    1   4.7128    0.6742   3.5604   6.2381
```

Figure 39.3. Model Parameter Estimates from the LIFEREG Procedure

The table of parameter estimates is displayed in Figure 39.3. Both the intercept and the slope parameter for the variable group are significantly different from 0 at the 0.05 level. Because the variable group has only one degree of freedom, parameter estimates are given for only one level of the variable group (group=1). However, the estimate for the intercept parameter provides a baseline for group=2. The resulting model is

$$\log(\text{minutes}) = \begin{cases} 3.30911843 - 0.1933025 & \text{for group=1} \\ 3.30911843 & \text{for group=2} \end{cases}$$

Note that the Weibull shape parameter for this model is the reciprocal of the extreme value scale parameter estimate shown in Figure 39.3 $(1/0.21219 = 4.7128)$.

The following statements produce a graph of the cumulative distribution values versus the variable minutes. The LEGEND1 statement defines the appearance of the legend that displays on the plot. The two AXIS statements define the appearance of the plot axes. The SYMBOL statements control the plotting symbol, color, and method of smoothing.

```
legend1 frame cframe=ligr cborder=black
   position=center value=(justify=center);

axis1 label=(angle=90 rotate=0 'Estimated CDF') minor=none;
axis2 minor=none;

symbol1 c=white i=spline;
symbol2 c=yellow i=spline;

proc sort data=new;
   by prob;

proc gplot data=new;
   plot prob*minutes=group/ frame cframe=ligr
        legend=legend1 vaxis=axis1 haxis=axis2;
run;
```

The SORT procedure sorts the data set new by the variable prob. Then the GPLOT procedure plots the variable prob versus the variable minutes using the grouping variable as the identification variable. The LEGEND=, VAXIS=, and HAXIS= options specify the previously defined legend and axis statements.

Figure 39.4 displays the estimated cumulative distribution function for each group.

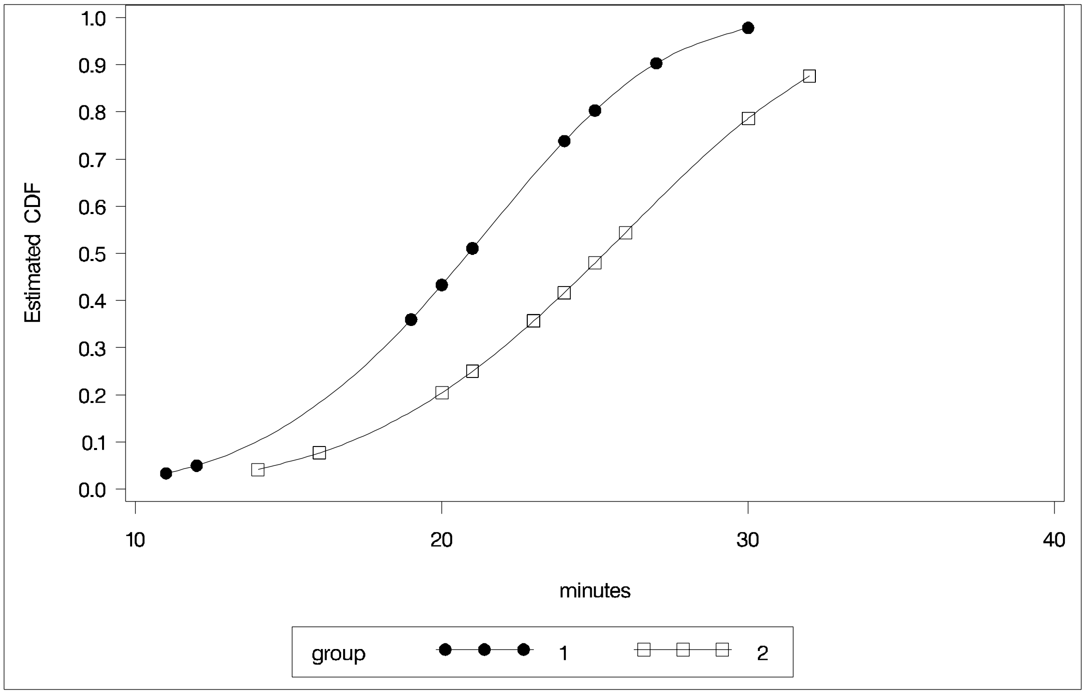


Figure 39.4. Plot of the Estimated Cumulative Distribution Function

Syntax

The following statements are available in PROC LIFEREG.

PROC LIFEREG < *options* > **;**
  BY *variables* **;**
  CLASS *variables* **;**
  INSET <*keyword-list*> < **/** *options* > **;**
  MODEL *response*=<*effects*> < **/** *options* > **;**
  OUTPUT < **OUT=***SAS-data-set* >
    *keyword***=***name* < . . . *keyword***=***name* >
    < *options* > **;**
  PROBPLOT < **/** *options* > **;**
  WEIGHT *variable* **;**

The PROC LIFEREG statement invokes the procedure. The MODEL statement is required and specifies the variables used in the regression part of the model as well as the distribution used for the error, or random, component of the model. Only a single MODEL statement can be used with one invocation of the LIFEREG procedure. If multiple MODEL statements are present, only the last is used. Main effects and interaction terms can be specified in the MODEL statement, similar to the GLM procedure. Initial values can be specified in the MODEL statement or in an INEST= data set. If no initial values are specified, the starting estimates are obtained by ordinary least squares. The CLASS statement determines which explanatory variables are treated as categorical. The WEIGHT statement identifies a variable with values that are used to weight the observations. Observations with zero or negative weights are not used to fit the model, although predicted values can be computed for them. The OUTPUT statement creates an output data set containing predicted values and residuals.

PROC LIFEREG Statement

PROC LIFEREG $<$ *options* $>$ **;**

The PROC LIFEREG statement invokes the procedure. You can specify the following options in the PROC LIFEREG statement.

COVOUT

writes the estimated covariance matrix to the OUTEST= data set if convergence is attained.

DATA=*SAS-data-set*

specifies the input SAS data set used by PROC LIFEREG. By default, the most recently created SAS data set is used.

GOUT=*graphics-catalog*

specifies a graphics catalog in which to save graphics output.

INEST= *SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= Data Set" on page 2121 for a detailed description of the contents of the INEST= data set.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOPRINT

suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. The following table illustrates how PROC LIFEREG interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, refer to the chapter titled "The SORT Procedure" in the *SAS Procedures Guide*.

OUTEST=*SAS-data-set*

specifies an output SAS data set containing the parameter estimates, the maximized log likelihood, and, if the COVOUT option is specified, the estimated covariance matrix. See the section "OUTEST= Data Set" on page 2121 for a detailed description of the contents of the OUTEST= data set.

XDATA= *SAS-data-set*

specifies an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement for probability plotting. If there are covariates specified in a MODEL statement and a probability plot is requested with a PROBPLOT statement, you specify fixed values for the effects in the MODEL statement with the XDATA= data set. See the section "XDATA= Data Set" on page 2122 for a detailed description of the contents of the XDATA= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LIFEREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LIFEREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

Variables that are classification variables rather than quantitative numeric variables must be listed in the CLASS statement. For each explanatory variable listed in the CLASS statement, indicator variables are generated for the levels assumed by the CLASS variable. If the CLASS statement is used, it must appear before the MODEL statement.

INSET Statement

INSET < *keyword-list* > < / *options*>**;**

The box or table of summary information produced on plots made with the PROBPLOT statement is called an *inset*. You can use the INSET statement to customize the information that is displayed in the inset box as well as to customize the appearance of the inset box. To supply the information that is displayed in the inset box, you specify *keywords* corresponding to the information that you want shown. For example, the following statements produce a probability plot with the number of observations, the number of right-censored observations, the name of the distribution, and the estimated Weibull shape parameter in the inset.

```
proc lifereg data=epidemic;
   model life = dose / dist = Weibull;
   probplot ;
   inset nobs right dist shape;
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset containing the number of observations and the name of the distribution, labeled “Sample Size” and “Distribution” in the inset.

```
inset nobs='Sample Size' dist='Distribution';
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

If you specify more than one INSET statement, only the first one is used.

The following table lists keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 39.1. INSET Statement Keywords

CONFIDENCE	confidence coefficient for all confidence intervals
DIST	name of the distribution
INTERVAL	number of interval-censored observations
LEFT	number of left-censored observations
NOBS	number of observations
NMISS	number of observations with missing values
RIGHT	number of right-censored observations
SCALE	value of the scale parameter
SHAPE	value of the shape parameter
UNCENSORED	number of uncensored observations

The following *options* control the appearance of the box. All *options* are specified after the slash (/) in the INSET statement.

CFILL=*color*
specifies the color for the filling box.

CFILLH=*color*
specifies the color for the filling box header.

CFRAME=*color*
specifies the color for the frame.

CHEADER=*color*
specifies the color for text in the header.

CTEXT=*color*
specifies the color for the text.

FONT=*font*
specifies the software font for the text.

HEIGHT=*value*
specifies the height of the text.

HEADER=*'quoted string'*
specifies text for the header or box title.

NOFRAME
omits the frame around the box.

POS= *value* <**DATA | PERCENT**>
determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in screen percent units or axis data units. The default is screen percent units.

REFPOINT= *name*

specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. The REFPOINT= option specifies which corner of the inset frame you have specified with coordinates (x, y), and it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

MODEL Statement

<label:> **MODEL** response<*censor(list)>=effects < / options > **;**

<label:> **MODEL** *(lower,upper)=effects < / options >* **;**

<label:> **MODEL** *events/trials=effects < / options >* **;**

Only a single MODEL statement can be used with one invocation of the LIFEREG procedure. If multiple MODEL statements are present, only the last is used. The optional *label* is used to label the model estimates in the output SAS data set and OUTEST= data set.

The first MODEL syntax is appropriate for right censoring. The variable *response* is possibly right-censored. If the *response* variable can be right-censored, then a second variable, denoted *censor*, must appear after the *response* variable with a list of parenthesized values, separated by commas or blanks, to indicate censoring. That is, if the *censor* variable takes on a value given in the list, the *response* is a right-censored value; otherwise, it is an observed value.

The second MODEL syntax specifies two variables, *lower* and *upper*, that contain values of the endpoints of the censoring interval. If the two values are the same (and not missing), it is assumed that there is no censoring and the actual response value is observed. If the lower value is missing, then the upper value is used as a left-censored value. If the upper value is missing, then the lower value is taken as a right-censored value. If both values are present and the lower value is less than the upper value, it is assumed that the values specify a censoring interval. If the lower value is greater than the upper value or both values are missing, then the observation is not used in the analysis although predicted values can still be obtained if none of the covariates are missing. The following table summarizes the ways of specifying censoring.

lower	*upper*	**Comparison**	**Interpretation**
not missing	not missing	equal	no censoring
not missing	not missing	lower < upper	censoring interval
missing	not missing		upper used as left-censoring value
not missing	missing		lower used as right-censoring value
not missing	not missing	lower > upper	observation not used
missing	missing		observation not used

The third MODEL syntax specifies two variables that contain count data for a binary response. The value of the first variable, *events*, is the number of successes. The value of the second variable, *trials*, is the number of tries. The values of both *events* and (*trials*-*events*) must be nonnegative, and *trials* must be positive for the response to be valid. The values of the two variables do not need to be integers and are not modified to be integers.

The *effects* following the equal sign are the covariates in the model. Higher-order effects, such as interactions and nested terms, are allowed in the list, similar to the GLM procedure. Variable names and combinations of variable names representing higher-order terms are allowed to appear in this list. Class variables can be used as effects, and indicator variables are generated for the class levels. If you do not specify any covariates following the equal sign, an intercept-only model is fit.

Examples of three valid MODEL statements are

```
a: model time*flag(1,3)=temp;

b: model (start, finish)=;

c: model r/n=dose;
```

Model statement a indicates that the response is contained in a variable named time and that, if the variable flag takes on the values 1 or 3, the observation is right-censored. The explanatory variable is temp, which could be a class variable. Model statement b indicates that the response is known to be in the interval between the values of the variables start and finish and that there are no covariates except for a default intercept term. Model statement c indicates a binary response, with the variable r containing the number of responses and the variable n containing the number of trials.

The following options can appear in the MODEL statement.

Task	Option
Model specification	
set the significance level	ALPHA=
specify distribution type for failure time	DISTRIBUTION=
request no log transformation of response	NOLOG
initial estimate for intercept term	INTERCEPT=
hold intercept term fixed	NOINT
initial estimates for regression parameters	INITIAL=
initialize scale parameter	SCALE=
hold scale parameter fixed	NOSCALE
initialize first shape parameter	SHAPE1=
hold first shape parameter fixed	NOSHAPE1
Model fitting	
set convergence criterion	CONVERGE=
set maximum iterations	MAXITER=
set tolerance for testing singularity	SINGULAR=
Output	
display estimated correlation matrix	CORRB
display estimated covariance matrix	COVB
display iteration history, final gradient, and second derivative matrix	ITPRINT

ALPHA=*value*

sets the significance level for the confidence intervals for regression parameters and estimated survival probabilities. The value must be between 0 and 1. By default, ALPHA = 0.05.

CONVERGE=*value*

sets the convergence criterion. Convergence is declared when the maximum change in the parameter estimates between Newton-Raphson steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E−8.

CONVG=*value*

sets the relative Hessian convergence criterion. *value* must be between 0 and 1. After convergence is determined with the change in parameter criterion specified with the CONVERGE= option, the quantity $tc = \frac{\mathbf{g}'\mathbf{H}^{-1}\mathbf{g}}{|f|}$ is computed and compared to *value*, where **g** is the gradient vector, **H** is the Hessian matrix for the model parameters, and f is the log-likelihood function. If tc is greater than *value*, a warning that the relative Hessian convergence criterion has been exceeded is displayed. This criterion detects the occasional case where the change in parameter convergence criterion is satisfied, but a maximum in the log-likelihood function has not been attained. By default, CONVG=1E−4.

CORRB

produces the estimated correlation matrix of the parameter estimates.

COVB

produces the estimated covariance matrix of the parameter estimates.

DISTRIBUTION=*distribution-type*
DIST=*distribution-type*
D=*distribution-type*

specifies the distribution type assumed for the failure time. By default, PROC LIFEREG fits a type 1 extreme value distribution to the log of the response. This is equivalent to fitting the Weibull distribution, since the scale parameter for the extreme value distribution is related to a Weibull shape parameter and the intercept is related to the Weibull scale parameter in this case. When the NOLOG option is specified, PROC LIFEREG models the untransformed response with a type 1 extreme value distribution as the default. See the section "Supported Distributions" on page 2111 for descriptions of the distributions. The following are valid values for *distribution-type*:

EXPONENTIAL	the exponential distribution, which is treated as a restricted Weibull distribution
GAMMA	a generalized gamma distribution (Lawless, 1982, p. 240). The standard two-parameter gamma distribution is not available in PROC LIFEREG.
LLOGISTIC	a loglogistic distribution
LNORMAL	a lognormal distribution
LOGISTIC	a logistic distribution (equivalent to LLOGISTIC when the NOLOG option is specified)
NORMAL	a normal distribution (equivalent to LNORMAL when the NOLOG option is specified)
WEIBULL	a Weibull distribution. If NOLOG is specified, it fits a type 1 extreme value distribution to the raw, untransformed data.

By default, PROC LIFEREG transforms the response with the natural logarithm before fitting the specified model when you specify the GAMMA, LLOGISTIC, LNORMAL, or WEIBULL option. You can suppress the log transformation with the NOLOG option. The following table summarizes the resulting distributions when the preceding distribution options are used in combination with the NOLOG option.

DISTRIBUTION=	NOLOG specified?	Resulting distribution
EXPONENTIAL	No	Exponential
EXPONENTIAL	Yes	One-parameter extreme value
GAMMA	No	Generalized gamma
GAMMA	Yes	Generalized gamma with untransformed responses
LOGISTIC	No	Logistic
LOGISTIC	Yes	Logistic (NOLOG has no effect)
LLOGISTIC	No	Log-logistic
LLOGISTIC	Yes	Logistic
LNORMAL	No	Lognormal
LNORMAL	Yes	Normal
NORMAL	No	Normal
NORMAL	Yes	Normal (NOLOG has no effect)
WEIBULL	No	Weibull
WEIBULL	Yes	Extreme value

INITIAL=*values*

sets initial values for the regression parameters. This option can be helpful in the case of convergence difficulty. Specified values are used to initialize the regression coefficients for the covariates specified in the MODEL statement. The intercept parameter is initialized with the INTERCEPT= option and is not included here. The values are assigned to the variables in the MODEL statement in the same order in which they are listed in the MODEL statement. Note that a class variable requires $k-1$ values when the class variable takes on k different levels. The order of the class levels is determined by the ORDER= option. If there is no intercept term, the first class variable requires k initial values. If a BY statement is used, all class variables must take on the same number of levels in each BY group or no meaningful initial values can be specified. The INITIAL= option can be specified as follows.

Type of List	**Specification**
list separated by blanks	`initial=3 4 5`
list separated by commas	`initial=3,4,5`
x to y	`initial=3 to 5`
x to y by z	`initial=3 to 5 by 1`
combination of methods	`initial=1,3 to 5,9`

By default, PROC LIFEREG computes initial estimates with ordinary least squares. See the section "Computational Method" on page 2108 for details.

Note: The INITIAL= option is overwritten by the INEST= option. See the section "INEST= Data Set" on page 2121 for details.

INTERCEPT=*value*

initializes the intercept term to *value*. By default, the intercept is initialized by an ordinary least squares estimate.

ITPRINT

displays the iteration history, the final evaluation of the gradient, and the final evaluation of the negative of the second derivative matrix, that is, the negative of the Hessian.

MAXITER=*n*

sets the maximum allowable number of iterations during the model estimation. By default, MAXITER=50.

NOINT

holds the intercept term fixed. Because of the usual log transformation of the response, the intercept parameter is usually a scale parameter for the untransformed response, or a location parameter for a transformed response.

NOLOG

requests that no log transformation of the response variable be performed. By default, PROC LIFEREG models the log of the response variable for the GAMMA, LLOGISTIC, LOGNORMAL, and WEIBULL distribution options.

NOSCALE

holds the scale parameter fixed. Note that if the log transformation has been applied to the response, the effect of the scale parameter is a power transformation of the original response. If no SCALE= value is specified, the scale parameter is fixed at the value 1.

NOSHAPE1

holds the first shape parameter, SHAPE1, fixed. If no SHAPE1= value is specified, SHAPE1 is fixed at a value that depends on the DISTRIBUTION type.

SCALE=*value*

initializes the scale parameter to *value*. If the Weibull distribution is specified, this scale parameter is the scale parameter of the type 1 extreme value distribution, not the Weibull scale parameter. Note that, with a log transformation, the exponential model is the same as a Weibull model with the scale parameter fixed at the value 1.

SHAPE1=*value*

initializes the first shape parameter to *value*. If the specified distribution does not depend on this parameter, then this option has no effect. The only distribution that depends on this shape parameter is the generalized gamma distribution. See the "Supported Distributions" section on page 2111 for descriptions of the parameterizations of the distributions.

SINGULAR=*value*

sets the tolerance for testing singularity of the information matrix and the crossproducts matrix for the initial least-squares estimates. Roughly, the test requires that a pivot be at least this value times the original diagonal value. By default, SINGULAR=1E−12.

OUTPUT Statement

OUTPUT <**OUT=***SAS-data-set*> *keyword***=***name* <*...keyword***=***name*> **;**

The OUTPUT statement creates a new SAS data set containing statistics calculated after fitting the model. At least one specification of the form *keyword*=*name* is required.

All variables in the original data set are included in the new data set, along with the variables created as options to the OUTPUT statement. These new variables contain fitted values and estimated quantiles. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets). Each OUTPUT statement applies to the preceding MODEL statement. See Example 39.1 for illustrations of the OUTPUT statement.

The following specifications can appear in the OUTPUT statement:

OUT=*SAS-data-set* specifies the new data set. By default, the procedure uses the DATAn convention to name the new data set.

keyword=*name* specifies the statistics to include in the output data set and gives names to the new variables. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

CENSORED specifies an indicator variable to signal censoring. The variable takes on the value 1 if the observation is censored; otherwise, it is 0.

CDF specifies a variable to contain the estimates of the cumulative distribution function evaluated at the observed response. See the "Predicted Values" section on page 2114 for more information.

CONTROL specifies a variable in the input data set to control the estimation of quantiles. See Example 39.1 for an illustration. If the specified variable has the value of 1, estimates for all the values listed in the QUANTILE= list are computed for that observation in the input data set; otherwise, no estimates are computed. If no CONTROL= variable is specified, all quantiles are estimated for all observations. If the response variable in the MODEL statement is binomial, then this option has no effect.

CRESIDUAL | CRES specifies a variable to contain the Cox-Snell residuals

$$-\log(S(u_i))$$

where S is the standard survival function and

$$u_i = \frac{y_i - \mathbf{x}_i'\mathbf{b}}{\sigma}$$

If the response variable in the corresponding model statement is binomial, then the residuals are not computed, and this variable contains missing values.

SRESIDUAL | SRES specifies a variable to contain the standardized residuals

$$\frac{y_i - \mathbf{x}_i'\mathbf{b}}{\sigma}$$

If the response variable in the corresponding model statement is binomial, then the residuals are not computed, and this variable contains missing values.

PREDICTED | P specifies a variable to contain the quantile estimates. If the response variable in the corresponding model statement is binomial, then this variable contains the estimated probabilities, $1 - F(-\mathbf{x}'\mathbf{b})$.

QUANTILES | QUANTILE | Q gives a list of values for which quantiles are calculated. The values must be between 0 and 1, noninclusive. For each value, a corresponding quantile is estimated. This option is not used if the response variable in the corresponding MODEL statement is binomial. The QUANTILES option can be specified as follows.

Type of List	**Specification**
list separated by blanks	`.2 .4 .6 .8`
list separated by commas	`.2,.4,.6,.8`
x to y	`.2 to .8`
x to y by z	`.2 to .8 by .1`
combination of methods	`.1,.2 to .8 by .2`

By default, QUANTILES=0.5. When the response is not binomial, a numeric variable, _PROB_, is added to the OUTPUT data set whenever the QUANTILES= option is specified. The variable _PROB_ gives the probability value for the quantile estimates. These are the values taken from the QUANTILES= list and are given as values between 0 and 1, not as values between 0 and 100.

STD_ERR | STD specifies a variable to contain the estimates of the standard errors of the estimated quantiles or $\mathbf{x}'\mathbf{b}$. If the response used in the MODEL statement is a binomial response, then these are the standard errors of $\mathbf{x}'\mathbf{b}$. Otherwise, they are the standard errors of the quantile estimates. These estimates can be used to compute confidence intervals for the quantiles. However, if the model is fit to the log of the event time, better confidence intervals can usually be computed by transforming the confidence intervals for the log response. See Example 39.1 for such a transformation.

XBETA specifies a variable to contain the computed value of $\mathbf{x}'\mathbf{b}$, where $\mathbf{x}$ is the covariate vector and **b** is the vector of parameter estimates.

PROBPLOT Statement

PROBPLOT | PPLOT < / *options* > **;**

You can use the PROBPLOT statement to create a probability plot from lifetime data. The data can be uncensored, right-censored, or arbitrarily censored. You can specify any number of PROBPLOT statements after a MODEL statement. The syntax used for the response in the MODEL statement determines the type of censoring assumed in creating the probability plot. The model fit with the MODEL statement is plotted along with the data. If there are covariates in the model, they are set to constant values specified in the XDATA= data set when creating the probability plot. If no XDATA= data set is specified, continuous variables are set to their overall mean values and categorical variables specified in the CLASS statement are set to their highest levels.

You can specify the following options to control the content, layout, and appearance of a probability plot.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an ANNOTATE data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the probability plot. The ANNOTATE= data set you specify in the PROBPLOT statement is used for all plots created by the statement.

CAXIS=*color*
CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CCENSOR=*color*

specifies the color for filling the censor plot area. The default is the first color in the device color list.

CENBIN

plots censored data as frequency counts (rounding-off for non-integer frequency) rather than as individual points.

CENCOLOR=*color*

specifies the color for the censor symbol. The default is the first color in the device color list.

CENSYMBOL=*symbol* **| (***symbol list***)**

specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, and circle) or a letter (A–Z). If you do not specify the CENSYMBOL= option, the symbol used for censored values is the same as for failures.

CFIT=*color*

specifies the color for the fitted probability line and confidence curves. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*
CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**
DES='*string***'**

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HCL

computes and draws confidence limits for the predicted probabilities in the horizontal direction.

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the lifetime axis scale. The HLOWER= option specifies *value* as the lower lifetime axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

HOFFSET=*value*

specifies the offset for the horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper lifetime axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1' . . . 'labeln'*
HREFLABEL=*'label1' . . . 'labeln'*
HREFLAB=*'label1' . . . 'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows the valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around probability plots.

INTERTILE=*value*

specifies the distance between tiles.

ITPRINTEM

displays the iteration history for the Turnbull algorithm.

JITTER=*value*

specifies the amount to jitter overlaying plot symbols, in units of symbol width.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*), and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*
LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

MAXITEM=*n1 <,n2> n1*

specifies the maximum number of iterations allowed for the Turnbull algorithm. Iteration history will be displayed in increments of *n2* if requested with the ITPRINTEM option. See the section "Arbitrarily Censored Data" on page 2119 for details.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'LIFEREG'.

NOCENPLOT

suppresses the plotting of censored data points.

NOCONF

suppresses the default percentile confidence bands on the probability plot.

NODATA

suppresses plotting of the estimated empirical probability plot.

NOFIT

suppresses the fitted probability (percentile) line and confidence bands.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOPOLISH

suppresses setting small interval probabilities to zero in the Turnbull algorithm.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

NPINTERVALS=*interval type*

specifies one of the two kinds of confidence limits for the estimated cumulative probabilities, pointwise (NPINTERVALS=POINT) or simultaneous (NPINTERVALS=SIMUL), requested by the PPOUT option to be displayed in the tabular output.

PCTLIST=*value-list*

specifies the list of percentages for which to compute percentile estimates. *value-list* must be a list of values separated by blanks or commas. Each value in the list must be between 0 and 100.

PLOWER=*value*

specifies the lower limit on the probability axis scale. The PLOWER= option specifies *value* as the lower probability axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

PRINTPROBS

displays intervals and associated probabilities for the Turnbull algorithm.

PUPPER=*value*

specifies the upper limit on the probability axis scale. The PUPPER= option specifies *value* as the upper probability axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

PPOS=*character-list*

specifies the plotting position type. See the section "Probability Plotting" on page 2116 for details.

PPOS	**Method**
EXPRANK	expected ranks
MEDRANK	median ranks
MEDRANK1	median ranks (exact formula)
KM	Kaplan-Meier
MKM	modified Kaplan-Meier (default)

PPOUT

specifies that a table of the cumulative probabilities plotted on the probability plot be displayed. Kaplan-Meier estimates of the cumulative probabilities are also displayed, along with standard errors and confidence limits. The confidence limits can be pointwise or simultaneous, as specified by the NPINTERVALS= option.

PROBLIST=*value-list*

specifies the list of initial values for the Turnbull algorithm.

ROTATE

requests probability plots with probability scale on the horizontal axis.

SQUARE

makes the layout of the probability plots square.

TOLLIKE=*value*

specifies the criterion for convergence in the Turnbull algorithm.

TOLPROB=*value*

specifies the criterion for setting the interval probability to zero in the Turnbull algorithm.

VAXISLABEL=‘*string***’**

specifies a label for the vertical axis.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the entries for the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*’label1’ . . . ’labeln’*
VREFLABEL=*’label1’ . . . ’labeln’*
VREFLAB=*’label1’ . . . ’labeln’*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=n

specifies the horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table.

n	**label placement**
1	left
2	right

WAXIS=n

specifies line thickness for axes and frame. The default value is 1.

WFIT=n

specifies line thickness for fitted curves. The default value is 1.

WGRID=n
specifies line thickness for grids. The default value is 1.

WREFL=n
specifies line thickness for reference lines. The default value is 1.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with nonpositive or missing values for the weight variable do not contribute to the fit of the model. The WEIGHT variable multiplies the contribution to the log likelihood for each observation.

Details

Missing Values

Any observation with missing values for the dependent variable is not used in the model estimation unless it is one and only one of the values in an interval specification. Also, if one of the explanatory variables or the censoring variable is missing, the observation is not used. For any observation to be used in the estimation of a model, only the variables needed in that model have to be nonmissing. Predicted values are computed for all observations with no missing explanatory variable values. If the censoring variable is missing, the CENSORED= variable in the OUT= SAS data set is also missing.

Model Specification

Main effects as well as interaction terms are allowed in the model specification, similar to the GLM procedure. For numeric variables, a main effect is a linear term equal to the value of the variable unless the variable appears in the CLASS statement. For variables listed in the CLASS statement, PROC LIFEREG creates indicator variables (variables taking the values zero or one) for every level of the variable except the last level. If there is no intercept term, the first class variable has indicator variables created for all levels including the last level. The levels are ordered according to the ORDER= option. Estimates of a main effect depend upon other effects in the model and, therefore, are adjusted for the presence of other effects in the model.

Computational Method

By default, the LIFEREG Procedure computes initial values for the parameters using ordinary least squares (OLS) ignoring censoring. This might not be the best set of starting values for a given set of data. For example, if there are extreme values in your data the OLS fit may be excessively influenced by the extreme observations, causing an overflow or convergence problems. See Example 39.3 for one way to deal with convergence problems.

You can specify the INITIAL= option in the MODEL statement to override these starting values. You can also specify the INTERCEPT=, SCALE=, and SHAPE= options to set initial values of the intercept, scale, and shape parameters. For models with multilevel interaction effects, it is a little difficult to use the INITIAL= option to provide starting values for all parameters. In this case, you can use the INEST= data set. See the section "INEST= Data Set" on page 2121 for detail. The INEST= data set overrides all previous specifications for starting values of parameters.

The rank of the design matrix $\mathbf{X}$ is estimated before the model is fit. Columns of $\mathbf{X}$ that are judged linearly dependent on other columns have the corresponding parameters set to zero. The test for linear dependence is controlled by the SINGULAR= option in the MODEL statement. Variables are included in the model in the order in which they are listed in the MODEL statement with the nonclass variables included in the model before any class variables.

The log-likelihood function is maximized by means of a ridge-stabilized Newton-Raphson algorithm. The maximized value of the log likelihood can take positive or negative values, depending on the specified model and the values of the maximum likelihood estimates of the model parameters.

If convergence of the maximum likelihood estimates is attained, a Type III chi-square test statistic is computed for each effect, testing whether there is any contribution from any of the levels of the effect. This statistic is computed as a quadratic form in the appropriate parameter estimates using the corresponding submatrix of the asymptotic covariance matrix estimate. Refer to Chapter 32, "The GLM Procedure," and Chapter 11, "The Four Types of Estimable Functions," for more information about Type III estimable functions.

The asymptotic covariance matrix is computed as the inverse of the observed information matrix. Note that if the NOINT option is specified and class variables are used, the first class variable contains a contribution from an intercept term. The results are displayed in an ODS table named **Type3Analysis**.

Chi-square tests for individual parameters are Wald tests based on the observed information matrix and the parameter estimates. If an effect has a single degree of freedom in the parameter estimates table, the chi-square test for this parameter is equivalent to the Type III test for this effect.

In releases previous to Version 8.2, a multiple degree of freedom statistic was computed for each effect to test for contribution from any level of the effect. In general, the Type III test statistic in a main effect only model (no interaction terms) will be equal to the previously computed effect statistic, unless there are collinearities among the effects. If there are collinearities, the Type III statistic will adjust for them, and the value of the Type III statistic and the number of degrees of freedom might not be equal to those of the previous effect statistic.

Suppose there are n observations from the model $\mathbf{y} = \mathbf{X}\beta + \sigma\epsilon$, where $\mathbf{X}$ is an $n \times k$ matrix of covariate values (including the intercept), $\mathbf{y}$ is a vector of responses, and

ϵ is a vector of errors with survival function S, cumulative distribution function F, and probability density function f. That is, $S(t) = \Pr(\epsilon_i > t)$, $F(t) = \Pr(\epsilon_i \leq t)$, and $f(t) = dF(t)/dt$, where ϵ_i is a component of the error vector. Then, if all the responses are observed, the log likelihood, L, can be written as

$$L = \sum \log \left(\frac{f(u_i)}{\sigma} \right)$$

where $u_i = \frac{1}{\sigma}(y_i - \mathbf{x}_i'\beta)$.

If some of the responses are left, right, or interval censored, the log likelihood can be written as

$$L = \sum \log \left(\frac{f(u_i)}{\sigma} \right) + \sum \log \left(S(u_i) \right) + \sum \log \left(F(u_i) \right) + \sum \log \left(F(u_i) - F(v_i) \right)$$

with the first sum over uncensored observations, the second sum over right-censored observations, the third sum over left-censored observations, the last sum over interval-censored observations, and

$$v_i = \frac{1}{\sigma}(z_i - \mathbf{x}_i'\beta)$$

where z_i is the lower end of a censoring interval.

If the response is specified in the binomial format, *events/trials*, then the log-likelihood function is

$$L = \sum r_i \log(P_i) + (n_i - r_i) \log(1 - P_i)$$

where r_i is the number of events and n_i is the number of trials for the ith observation. In this case, $P_i = 1 - F(-\mathbf{x}_i'\beta)$. For the symmetric distributions, logistic and normal, this is the same as $F(\mathbf{x}_i'\beta)$. Additional information on censored and limited dependent variable models can be found in Kalbfleisch and Prentice (1980) and Maddala (1983).

The estimated covariance matrix of the parameter estimates is computed as the negative inverse of **I**, which is the information matrix of second derivatives of L with respect to the parameters evaluated at the final parameter estimates. If **I** is not positive definite, a positive definite submatrix of **I** is inverted, and the remaining rows and columns of the inverse are set to zero. If some of the parameters, such as the scale and intercept, are restricted, the corresponding elements of the estimated covariance matrix are set to zero. The standard error estimates for the parameter estimates are taken as the square roots of the corresponding diagonal elements.

For restrictions placed on the intercept, scale, and shape parameters, one-degree-of-freedom Lagrange multiplier test statistics are computed. These statistics are computed as

$$\chi^2 = \frac{g^2}{V}$$

where g is the derivative of the log likelihood with respect to the restricted parameter at the restricted maximum and

$$V = \mathbf{I}_{11} - \mathbf{I}_{12}\mathbf{I}_{22}^{-1}\mathbf{I}_{21}$$

where the 1 subscripts refer to the restricted parameter and the 2 subscripts refer to the unrestricted parameters. The information matrix is evaluated at the restricted maximum. These statistics are asymptotically distributed as chi-squares with one degree of freedom under the null hypothesis that the restrictions are valid, provided that some regularity conditions are satisfied. See Rao (1973, p. 418) for a more complete discussion. It is possible for these statistics to be missing if the observed information matrix is not positive definite. Higher degree-of-freedom tests for multiple restrictions are not currently computed.

A Lagrange multiplier test statistic is computed to test this constraint. Notice that this test statistic is comparable to the Wald test statistic for testing that the scale is one. The Wald statistic is the result of squaring the difference of the estimate of the scale parameter from one and dividing this by the square of its estimated standard error.

Supported Distributions

For each distribution, the baseline survival function (S) and the probability density function(f) are listed for the additive random disturbance (y_0 or $\log(T_0)$) with location parameter μ and scale parameter σ. See the section "Overview" on page 2083. These distributions apply when the log of the response is modeled (this is the default analysis). The corresponding survival function (G) and its density function (g) are given for the untransformed baseline distribution (T_0).

For example, for the WEIBULL distribution, $S(w)$ and $f(w)$ are the survival function and the probability density function for the extreme value distribution (distribution of the log of the response) while $G(t)$ and $g(t)$ are the survival function and the probability density function of a Weibull distribution (using the untransformed response).

The chosen baseline functions define the meaning of the intercept, scale, and shape parameters. Only the gamma distribution has a free shape parameter in the following parameterizations. Notice that some of the distributions do not have mean zero and that σ is not, in general, the standard deviation of the baseline distribution.

Additionally, it is worth mentioning that, for the Weibull distribution, the accelerated failure time model is also a proportional-hazards model. However, the parameterization for the covariates differs by a multiple of the scale parameter from the parameterization commonly used for the proportional hazards model.

The distributions supported in the LIFEREG procedure follow. μ = Intercept and σ = Scale in the output.

Exponential

$$
\begin{aligned}
S(w) &= \exp(-\exp(w-\mu)) \\
f(w) &= \exp(w-\mu)\exp(-\exp(w-\mu)) \\
G(t) &= \exp(-\alpha t) \\
g(t) &= \alpha\exp(-\alpha t)
\end{aligned}
$$

where $\exp(-\mu) = \alpha$.

Generalized Gamma

$S(w) = S'(u)$, $f(w) = \sigma^{-1}f'(u)$, $G(t) = G'(v)$, $g(t) = \frac{v}{t\sigma}g'(v)$. $u = \frac{w-\mu}{\sigma}$, $v = \exp(\frac{\log(t)-\mu}{\sigma})$, and

$$
\begin{aligned}
S'(u) &= \begin{cases} 1 - \frac{\Gamma\left(\delta^{-2}, \delta^{-2}\exp(\delta u)\right)}{\Gamma(\delta^{-2})} & \text{if } \delta > 0 \\ \frac{\Gamma\left(\delta^{-2}, \delta^{-2}\exp(\delta u)\right)}{\Gamma(\delta^{-2})} & \text{if } \delta < 0 \end{cases} \\
f'(u) &= \frac{|\delta|}{\Gamma\left(\delta^{-2}\right)}\left(\delta^{-2}\exp(\delta u)\right)^{\delta^{-2}}\exp\left(-\exp(\delta u)\delta^{-2}\right) \\
G'(v) &= \begin{cases} 1 - \frac{\Gamma\left(\delta^{-2}, \delta^{-2}v^{\delta}\right)}{\Gamma(\delta^{-2})} & \text{if } \delta > 0 \\ \frac{\Gamma\left(\delta^{-2}, \delta^{-2}v^{\delta}\right)}{\Gamma(\delta^{-2})} & \text{if } \delta < 0 \end{cases} \\
g'(v) &= \frac{|\delta|}{v\Gamma\left(\delta^{-2}\right)}\left(\delta^{-2}v^{\delta}\right)^{\delta^{-2}}\exp\left(-v^{\delta}\delta^{-2}\right)
\end{aligned}
$$

where $\Gamma(a)$ denotes the complete gamma function, $\Gamma(a, z)$ denotes the incomplete gamma function, and δ is a free shape parameter. The δ parameter is referred to as Shape by PROC LIFEREG. Refer to Lawless, 1982, p.240 and Klein and Moeschberger, 1997, p.386 for a description of the generalized gamma distribution.

Loglogistic

$$
\begin{aligned}
S(w) &= \left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^{-1} \\
f(w) &= \frac{\exp\left(\frac{w-\mu}{\sigma}\right)}{\sigma\left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^{2}} \\
G(t) &= \frac{1}{1 + \alpha t^{\gamma}}
\end{aligned}
$$

$$g(t) = \frac{\alpha\gamma t^{\gamma-1}}{(1+\alpha t^{\gamma})^2}$$

where $\gamma = 1/\sigma$ and $\alpha = \exp(-\mu/\sigma)$.

Lognormal

$$\begin{aligned} S(w) &= 1 - \Phi\left(\frac{w-\mu}{\sigma}\right) \\ f(w) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left(\frac{w-\mu}{\sigma}\right)^2\right) \\ G(t) &= 1 - \Phi\left(\frac{\log(t)-\mu}{\sigma}\right) \\ g(t) &= \frac{1}{\sqrt{2\pi}\sigma t}\exp\left(-\frac{1}{2}\left(\frac{\log(t)-\mu}{\sigma}\right)^2\right) \end{aligned}$$

where Φ is the cumulative distribution function for the normal distribution.

Weibull

$$\begin{aligned} S(w) &= \exp\left(-exp\left(\frac{w-\mu}{\sigma}\right)\right) \\ f(w) &= \frac{1}{\sigma}\exp\left(\frac{w-\mu}{\sigma}\right)\exp\left(-\exp\left(\frac{w-\mu}{\sigma}\right)\right) \\ G(t) &= \exp\left(-\alpha t^{\gamma}\right) \\ g(t) &= \gamma\alpha t^{\gamma-1}\exp\left(-\alpha t^{\gamma}\right) \end{aligned}$$

where $\sigma = 1/\gamma$ and $\alpha = \exp(-\mu/\sigma)$.

If your parameterization is different from the ones shown here, you can still use the procedure to fit your model. For example, a common parameterization for the Weibull distribution is

$$\begin{aligned} g(t;\lambda,\beta) &= \left(\frac{\beta}{\lambda}\right)^{\beta}\left(\frac{t}{\alpha}\right)^{\beta-1}\exp\left(-\left(\frac{t}{\lambda}\right)^{\beta}\right) \\ G(t;\lambda,\beta) &= \exp\left(-\left(\frac{t}{\lambda}\right)^{\beta}\right) \end{aligned}$$

so that $\lambda = \exp(\mu)$ and $\beta = 1/\sigma$.

Again note that the expected value of the baseline log response is, in general, not zero and that the distributions are not symmetric in all cases. Thus, for a given set of covariates, $\mathbf{x}$, the expected value of the log response is not always $\mathbf{x}'\beta$.

Some relations among the distributions are as follows:

- The gamma with Shape=1 is a Weibull distribution.
- The gamma with Shape=0 is a lognormal distribution.
- The Weibull with Scale=1 is an exponential distribution.

Predicted Values

For a given set of covariates, $\mathbf{x}$ (including the intercept term), the pth quantile of the log response, y_p, is given by

$$y_p = \mathbf{x}'\beta + \sigma u_p$$

where u_p is the pth quantile of the baseline distribution. The estimated quantile is computed by replacing the unknown parameters with their estimates, including any shape parameters on which the baseline distribution might depend. The estimated quantile of the original response is obtained by taking the exponential of the estimated log quantile unless the NOLOG option is specified in the preceding MODEL statement.

The standard errors of the quantile estimates are computed using the estimated covariance matrix of the parameter estimates and a Taylor series expansion of the quantile estimate. The standard error is computed as

$$\text{STD} = \sqrt{\mathbf{z}'\mathbf{V}\mathbf{z}}$$

where $\mathbf{V}$ is the estimated covariance matrix of the parameter vector $(\beta', \sigma, \delta)'$, and $\mathbf{z}$ is the vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \hat{u}_p \\ \hat{\sigma}\frac{\partial u_p}{\partial \delta} \end{bmatrix}$$

where δ is the vector of the shape parameters. Unless the NOLOG option is specified, this standard error estimate is converted into a standard error estimate for $\exp(y_p)$ as $\exp(\hat{y}_p)$STD. It may be more desirable to compute confidence limits for the log response and convert them back to the original response variable than to use the standard error estimates for $\exp(y_p)$ directly. See Example 39.1 for a 90% confidence interval of the response constructed by exponentiating a confidence interval for the log response.

The variable, CDF, is computed as

$$\mathrm{CDF}_i = F(u_i)$$

where the residual

$$u_i = \left(\frac{y_i - \mathbf{x}'_i\mathbf{b}}{\hat{\sigma}} \right)$$

and F is the baseline cumulative distribution function.

Confidence Intervals

Confidence intervals are computed for all model parameters and are reported in the "Analysis of Parameter Estimates" table. The confidence coefficient can be specified with the ALPHA=α MODEL statement option, resulting in a $(1-\alpha) \times 100\%$ two-sided confidence coefficient. The default confidence coefficient is 95$\%$, corresponding to $\alpha = .05$.

Regression Parameters

A two-sided $(1-\alpha) \times 100\%$ confidence interval $[\beta_{iL}, \quad \beta_{iU}]$ for the regression parameter β_i is based on the asymptotic normality of the maximum likelihood estimator $\hat{\beta}_i$ and is computed by

$$\beta_{iL} = \hat{\beta}_i - z_{1-\alpha/2}(\mathrm{SE}_{\hat{\beta}_i})$$

$$\beta_{iU} = \hat{\beta}_i + z_{1-\alpha/2}(\mathrm{SE}_{\hat{\beta}_i})$$

where $\mathrm{SE}_{\hat{\beta}_i}$ is the estimated standard error of $\hat{\beta}_i$, and z_p is the $p \times 100\%$ percentile of the standard normal distribution.

Scale Parameter

A two-sided $(1-\alpha) \times 100\%$ confidence interval $[\sigma_L, \quad \sigma_U]$ for the scale parameter σ in the location-scale model is based on the asymptotic normality of the logarithm of the maximum likelihood estimator $\log(\hat{\sigma})$, and is computed by

$$\sigma_L = \hat{\sigma} / \exp[z_{1-\alpha/2}(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$$

$$\sigma_U = \hat{\sigma} \exp[z_{1-\alpha/2}(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$$

Refer to Meeker and Escobar (1998) for more information.

Weibull Scale and Shape Parameters

The Weibull distribution scale parameter η and shape parameter β are obtained by transforming the extreme value location parameter μ and scale parameter σ:

$$\eta = \exp(\mu)$$

$$\beta = 1/\sigma$$

Consequently, two-sided $(1-\alpha) \times 100\%$ confidence intervals for the Weibull scale and shape parameters are computed as

$$[\eta_L,\ \ \eta_U] = [\exp(\mu_L),\ \ \exp(\mu_U)]$$

$$[\beta_L,\ \ \beta_U] = [1/\sigma_U,\ \ 1/\sigma_L]$$

Gamma Shape Parameter

A two-sided $(1-\alpha) \times 100\%$ confidence interval for the 3-parameter gamma shape parameter δ is computed by

$$[\delta_L,\ \ \delta_U] = [\hat{\delta} - z_{1-\alpha/2}(\mathrm{SE}_{\hat{\delta}}),\ \ \hat{\delta} + z_{1-\alpha/2}(\mathrm{SE}_{\hat{\delta}})]$$

Probability Plotting

Probability plots are useful tools for the display and analysis of lifetime data. Probability plots use an inverse distribution scale so that a cumulative distribution function (CDF) plots as a straight line. A nonparametric estimate of the CDF of the lifetime data will plot approximately as a straight line, thus providing a visual assessment of goodness-of-fit.

You can use the PROBPLOT statement in LIFEREG to create probability plots of data that are complete, right-censored, interval-censored, or a combination of censoring types (arbitrarily censored). A line representing the maximum likelihood fit from the MODEL statement and pointwise parametric confidence bands for the cumulative probabilities are also included on the plot.

A random variable Y belongs to a *location-scale* family of distributions if its CDF F is of the form

$$Pr\{Y \leq y\} = F(y) = G\left(\frac{y-\mu}{\sigma}\right)$$

where μ is the location parameter and σ is the scale parameter. Here, G is a CDF that cannot depend on any unknown parameters, and G is the CDF of Y if $\mu = 0$

and $\sigma = 1$. For example, if Y is a normal random variable with mean μ and standard deviation σ,

$$G(u) = \Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2})\, du$$

and

$$F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right)$$

The normal, extreme value, and logistic distributions are location-scale models. The 3-parameter gamma distribution is a location-scale model if the shape parameter δ is fixed. If T has a lognormal, Weibull, or log-logistic distribution, then $\log(T)$ has a distribution that is a location-scale model. Probability plots are constructed for lognormal, Weibull, and log-logistic distributions by using $\log(T)$ instead of T in the plots.

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample with distribution function $F(y)$. A probability plot is a plot of the points $y_{(i)}$ against $m_i = G^{-1}(a_i)$, where $a_i = \hat{F}(y_i)$ is an estimate of the CDF $F(y_{(i)}) = G\left(\frac{y_{(i)}-\mu}{\sigma}\right)$. The nonparametric CDF estimates a_i are sometimes called *plotting positions*. The axis on which the points m_i are plotted is usually labeled with a probability scale (the scale of a_i).

If F is one of the location-scale distributions, then y is the lifetime; otherwise, the log of the lifetime is used to transform the distribution to a location-scale model.

If the data actually have the stated distribution, then $\hat{F} \approx F$,

$$m_i = G^{-1}(\hat{F}(y_i)) \approx G^{-1}(G\left(\frac{y_{(i)}-\mu}{\sigma}\right)) = \frac{y_{(i)}-\mu}{\sigma}$$

and points $(y_{(i)}, m_i)$ should fall approximately on a straight line.

There are several ways to compute the nonparametric CDF estimates used in probability plots from lifetime data. These are discussed in the next two sections.

Complete and Right-Censored Data

The censoring times must be taken into account when you compute plotting positions for right-censored data. The modified Kaplan-Meier method described in the following section is the default method for computing nonparametric CDF estimates for display on probability plots. Refer to Abernethy (1996), Meeker and Escobar (1998), and Nelson (1982) for discussions of the methods described in the following sections.

Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. Order the data in increasing order. Label all the data with reverse ranks r_i, with $r_1 = n, \ldots, r_n = 1$. For the lifetime (not censoring time) corresponding to reverse rank r_i, compute the survival function estimate

$$S_i = \left[\frac{r_i}{r_i + 1}\right] S_{i-1}$$

with $S_0 = 1$. The expected rank plotting position is computed as $a_i = 1 - S_i$. The option PPOS=EXPRANK specifies the expected rank plotting position.

For the Kaplan-Meier method,

$$S_i = \left[\frac{r_i - 1}{r_i}\right] S_{i-1}$$

The Kaplan-Meier plotting position is then computed as $a'_i = 1 - S_i$. The option PPOS=KM specifies the Kaplan-Meier plotting position.

For the modified Kaplan-Meier method, use

$$S'_i = \frac{S_i + S_{i-1}}{2}$$

where S_i is computed from the Kaplan-Meier formula with $S_0 = 1$. The plotting position is then computed as $a''_i = 1 - S'_i$. The option PPOS=MKM specifies the modified Kaplan-Meier plotting position. If the PPOS option is not specified, the modified Kaplan-Meier plotting position is used as the default method.

For complete samples, $a_i = i/(n+1)$ for the expected rank method, $a'_i = i/n$ for the Kaplan-Meier method, and $a''_i = (i - .5)/n$ for the modified Kaplan-Meier method. If the largest observation is a failure for the Kaplan-Meier estimator, then $F_n = 1$ and the point is not plotted.

Median Ranks

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. A failure order number j_i is assigned to the *i*th failure: $j_i = j_{i-1} + \Delta$, where $j_0 = 0$. The increment Δ is initially 1 and is modified when a censoring time is encountered in the ordered sample. The new increment is computed as

$$\Delta = \frac{(n+1) - \text{ previous failure order number}}{1 + \text{ number of items beyond previous censored item}}$$

The plotting position is computed for the ith failure time as

$$a_i = \frac{j_i - .3}{n + .4}$$

For complete samples, the failure order number j_i is equal to i, the order of the failure in the sample. In this case, the preceding equation for a_i is an approximation to the median plotting position computed as the median of the *i*th-order statistic from the uniform distribution on (0, 1). In the censored case, j_i is not necessarily an integer, but the preceding equation still provides an approximation to the median plotting position. The PPOS=MEDRANK option specifies the median rank plotting position.

Arbitrarily Censored Data

The LIFEREG procedure can create probability plots for data that consists of combinations of exact, left-censored, right-censored, and interval-censored lifetimes, that is, arbitrarily censored data. The LIFEREG procedure uses an iterative algorithm developed by Turnbull (1976) to compute a nonparametric maximum likelihood estimate of the cumulative distribution function for the data. Since the technique is maximum likelihood, standard errors of the cumulative probability estimates are computed from the inverse of the associated Fisher information matrix. This algorithm is an example of the expectation-maximization (EM) algorithm. The default initial estimate assigns equal probabilities to each interval. You can specify different initial values with the PROBLIST= option. Convergence is determined if the change in the log likelihood between two successive iterations is less than delta, where the default value of delta is 10^{-8}. You can specify a different value for delta with the TOLLIKE= option. Iterations will be terminated if the algorithm does not converge after a fixed number of iterations. The default maximum number of iterations is 1000. Some data may require more iterations for convergence. You can specify the maximum allowed number of iterations with the MAXITEM= option in the PROBPLOT statement. The iteration history of the log likelihood is displayed if you specify the ITPRINTEM option. The iteration history of the estimated interval probabilities are also displayed if you specify both options ITPRINTEM and PRINTPROBS.

If an interval probability is smaller than a tolerance (10^{-6} by default) after convergence, the probability is set to zero, the interval probabilities are renormalized so that they add to one, and iterations are restarted. Usually the algorithm converges in just a few more iterations. You can change the default value of the tolerance with the TOLPROB= option. You can specify the NOPOLISH option to avoid setting small probabilities to zero and restarting the algorithm.

If you specify the ITPRINTEM option, a table summarizing the Turnbull estimate of the interval probabilities is displayed. The columns labeled “Reduced Gradient” and “Lagrange Multiplier” are used in checking final convergence of the maximum likelihood estimate. The Lagrange multipliers must all be greater than or equal to zero, or the solution is not maximum likelihood. Refer to Gentleman and Geyer (1994) for more details of the convergence checking. Also refer to Meeker and Escobar (1998, chap. 3) for more information.

See Example 39.6 on page 2142 for an illustration.

Nonparametric Confidence Intervals

You can use the PPOUT option in the PROBPLOT statement to create a table containing the nonparametric CDF estimates computed by the selected method, Kaplan-Meier CDF estimates, standard errors of the Kaplan-Meier estimator, and nonparametric confidence limits for the CDF. The confidence limits are either pointwise or simultaneous, depending on the value of the NPINTERVALS= option in the PROBPLOT statement. The method used in the LIFEREG procedure for computation of approximate pointwise and simultaneous confidence intervals for cumulative failure probabilities relies on the Kaplan-Meier estimator of the cumulative distribution function of failure time and approximate standard deviation of the Kaplan-Meier estimator. For the case of arbitrarily censored data, the Turnbull algorithm, discussed previously, provides an extension of the Kaplan-Meier estimator. Both the Kaplan-Meier and the Turnbull estimators provide an estimate of the standard error of the CDF estimator, $\mathrm{se}_{\hat{F}}$, that is used in computing confidence intervals.

Pointwise Confidence Intervals

Approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, section 3.6) as

$$[F_L, \quad F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right]$$

where

$$w = \exp \left[\frac{z_{1-\alpha/2}\mathrm{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right]$$

where z_p is the pth quantile of the standard normal distribution.

Simultaneous Confidence Intervals

Approximate $(1-\alpha)100\%$ simultaneous confidence bands valid over the lifetime interval (t_a, t_b) are computed as the “Equal Precision” case of Nair (1984) and Meeker and Escobar (1998, section 3.8) as

$$[F_L, \quad F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right]$$

where

$$w = \exp \left[\frac{e_{a,b,1-\alpha/2}\mathrm{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right]$$

where the factor $x = e_{a,b,1-\alpha/2}$ is the solution of

$$x \exp(-x^2/2) \log \left[\frac{(1-a)b}{(1-b)a} \right] / \sqrt{8\pi} = \alpha/2$$

The time interval (t_a, t_b) over which the bands are valid depends in a complicated way on the constants a and b defined in Nair (1984), $0 < a < b < 1$. a and b are chosen by default so that the confidence bands are valid between the lowest and highest times corresponding to failures in the case of multiply censored data, or, to the lowest and highest intervals for which probabilities are computed for arbitrarily censored data. You can optionally specify a and b directly with the NPINTERVALS=SIMULTANEOUS(a, b) option in the PROBPLOT statement.

INEST= Data Set

If specified, the INEST= data set specifies initial estimates for all the parameters in the model. The INEST= data set must contain the intercept variable (named Intercept) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in one BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values. Combining the INEST= data set and the MAXITER= option in the MODEL statement, partial scoring can be done, such as predicting on a validation data set by using the model built from a training data set.

You can specify starting values for the iterative algorithm in the INEST= data set. This data set overwrites the INITIAL= option in the MODEL statement, which is a little difficult to use for models including multilevel interaction effects. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization. See Example 39.3 on page 2133 for an illustration.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates and the log likelihood for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different modeling using the LIFEREG procedure. If the COVOUT option is specified, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates. Note that, if the LIFEREG procedure does not converge, the parameter estimates are set to missing in the OUTEST data set.

The OUTEST= data set contains all variables specified in the MODEL statement and the BY statement. One observation consists of parameter values for the model with the dependent variable having the value -1. If the COVOUT option is specified, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL a character variable containing the label of the MODEL statement, if present. Otherwise, the variable's value is blank.

NAME a character variable containing the name of the dependent variable for the parameter estimates observations or the name of the row for the covariance matrix estimates

TYPE a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates

DIST a character variable containing the name of the distribution modeled

LNLIKE a numeric variable containing the last computed value of the log likelihood

INTERCEPT a numeric variable containing the intercept parameter estimates and covariances

SCALE a numeric variable containing the scale parameter estimates and covariances

SHAPE1 a numeric variable containing the first shape parameter estimates and covariances if the specified distribution has additional shape parameters

Any BY variables specified are also added to the OUTEST= data set.

XDATA= Data Set

The XDATA= data set is used for plotting the predicted probability when there are covariates specified in a MODEL statement and a probability plot is specified with a PROBPLOT statement. See Example 39.4 on page 2136 for an illustration.

The XDATA= data set is an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement.The XDATA= data set has the same structure as the DATA= data set but is not required to have all the variables or observations that appear in the DATA= data set.

The XDATA= data set must contain all the independent variables in the MODEL statement and variables in the CLASS statement. Even though variables in the CLASS statement may not be used, valid values are required for these variables in the XDATA= data set. Missing values are not allowed. Missing values are not allowed in the XDATA= data set for any of the independent variables either. Missing values are allowed for the dependent variables and other variables if they are included in the XDATA= data set.

If BY processing is used, the XDATA= data set should also include the BY variables, and there must be at least one valid observation for each BY group. If there is more than one valid observation in a BY group, the last one read is used for that BY group.

If there is no XDATA= data set in the PROC LIFEREG statement, by default, the LIFEREG procedure will use the overall mean for effects containing a continuous variable (or variables) and the highest level of a single classification variable as reference level. The rules are summarized as follows:

- If the effect contains a continuous variable (or variables), the overall mean of this effect (not the variables) is used.
- If the effect is a single classification variable, the highest level of the variable is used.

Computational Resources

Let p be the number of parameters estimated in the model. The minimum working space (in bytes) needed is

$$16p^2 + 100p$$

However, if sufficient space is available, the input data set is also kept in memory; otherwise, the input data set is reread for each evaluation of the likelihood function and its derivatives, with the resulting execution time of the procedure substantially increased.

Let n be the number of observations used in the model estimation. Each evaluation of the likelihood function and its first and second derivatives requires $O(np^2)$ multiplications and additions, n individual function evaluations for the log density or log distribution function, and n evaluations of the first and second derivatives of the function. The calculation of each updating step from the gradient and Hessian requires $O(p^3)$ multiplications and additions. The $O(v)$ notation means that, for large values of the argument, v, $O(v)$ is approximately a constant times v.

Displayed Output

For each model, PROC LIFEREG displays

- the name of the Data Set
- the name of the Dependent Variable
- the name of the Censoring Variable
- the Censoring Value(s) that indicate a censored observation
- the number of Noncensored and Censored Values
- the final estimate of the maximized log likelihood
- the iteration history and the Last Evaluation of the Gradient and Hessian if the ITPRINT option is specified (not shown)

For each explanatory variable in the model, the LIFEREG procedure displays

- the name of the Variable
- the degrees of freedom (DF) associated with the variable in the model
- the Estimate of the parameter
- the standard error (Std Err) estimate from the observed information matrix

- an approximate chi-square statistic for testing that the parameter is zero (the class variables also have an overall chi-square test statistic computed that precedes the individual level parameters)
- the probability of a larger chi-square value (Pr>Chi)
- the Label of the variable or, if the variable is a class level, the Value of the class variable

If there are constrained parameters in the model, such as the scale or intercept, then PROC LIFEREG displays a Lagrange multiplier test for the constraint.

ODS Table Names

PROC LIFEREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 39.2. ODS Tables Produced in PROC LIFEREG

ODS Table Name	**Description**	**Statement**	**Option**
ClassLevels	Class variable levels	CLASS	default*
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
EMIterHistory	Iteration history for Turnbull algorithm	PROBPLOT	ITPRINTEM
IterHistory	Iteration history	MODEL	ITPRINT
LagrangeStatistics	Lagrange statistics	MODEL	NOINT | NOSCALE
LastGrad	Last evaluation of the gradient	MODEL	ITPRINT
LastHess	Last evaluation of the Hessian	MODEL	ITPRINT
ModelInfo	Model information	MODEL	default
NObs	Observations Summary	PROC	default
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ProbEstimates	Nonparametric CDF estimates	PROBPLOT	PPOUT
Turnbull	Probability estimates from Turnbull algorithm	PROBPLOT	ITPRINTEM
Type3Analysis	Type 3 tests	MODEL	default*

* Depends on data.

Examples

Example 39.1. Motorette Failure

This example fits a Weibull model and a lognormal model to the example given in Kalbfleisch and Prentice (1980, p. 5). An output data set called models is specified to contain the parameter estimates. By default, the natural log of the variable time is used by the procedure as the response. After this log transformation, the Weibull model is fit using the extreme value baseline distribution, and the lognormal is fit using the normal baseline distribution.

Since the extreme value and normal distributions do not contain any shape parameters, the variable SHAPE1 is missing in the models data set. An additional output data set, out, is created that contains the predicted quantiles and their standard errors for values of the covariate corresponding to temp=130 and temp=150. This is done with the control variable, which is set to 1 for only two observations.

Using the standard error estimates obtained from the output data set, approximate 90% confidence limits for the predicted quantities are then created in a subsequent DATA step for the log response. The logs of the predicted values are obtained because the values of the P= variable in the OUT= data set are in the same units as the original response variable, time. The standard errors of the quantiles of the log(time) are approximated (using a Taylor series approximation) by the standard deviation of time divided by the mean value of time. These confidence limits are then converted back to the original scale by the exponential function. The following statements produce Output 39.1.1 through Output 39.1.5.

```
title 'Motorette Failures With Operating Temperature as a Covariate';
data motors;
   input time censor temp @@;
   if _N_=1 then
      do;
         temp=130;
         time=.;
         control=1;
         z=1000/(273.2+temp);
         output;
         temp=150;
         time=.;
         control=1;
         z=1000/(273.2+temp);
         output;
      end;
   if temp>150;
   control=0;
   z=1000/(273.2+temp);
   output;
   datalines;
8064 0 150 8064 0 150 8064 0 150 8064 0 150 8064 0 150
8064 0 150 8064 0 150 8064 0 150 8064 0 150 8064 0 150
1764 1 170 2772 1 170 3444 1 170 3542 1 170 3780 1 170
4860 1 170 5196 1 170 5448 0 170 5448 0 170 5448 0 170
 408 1 190  408 1 190 1344 1 190 1344 1 190 1440 1 190
1680 0 190 1680 0 190 1680 0 190 1680 0 190 1680 0 190
 408 1 220  408 1 220  504 1 220  504 1 220  504 1 220
 528 0 220  528 0 220  528 0 220  528 0 220  528 0 220
;

proc print data=motors;
run;

proc lifereg data=motors outest=modela covout;
```

```
   a: model time*censor(0)=z;
      output out=outa quantiles=.1 .5 .9 std=std p=predtime
         control=control;
run;

proc lifereg data=motors outest=modelb covout;
   b: model time*censor(0)=z / dist=lnormal;
      output out=outb quantiles=.1 .5 .9 std=std p=predtime
         control=control;
run;


data models;
   set modela modelb;
run;

proc print data=models;
   id _model_;
   title 'fitted models';
run;

data out;
   set outa outb;
run;


data out1;
   set out;
   ltime=log(predtime);
   stde=std/predtime;
   upper=exp(ltime+1.64*stde);
   lower=exp(ltime-1.64*stde);
run;

proc print;
   id temp;
   title 'quantile estimates and confidence limits';
run;
```

Output 39.1.1. Motorette Failure Data

```
              Motorette Failures With Operating Temperature as a Covariate

                  Obs    time    censor    temp    control       z

                    1       .       0       130       1       2.48016
                    2       .       0       150       1       2.36295
                    3    1764       1       170       0       2.25632
                    4    2772       1       170       0       2.25632
                    5    3444       1       170       0       2.25632
                    6    3542       1       170       0       2.25632
                    7    3780       1       170       0       2.25632
                    8    4860       1       170       0       2.25632
                    9    5196       1       170       0       2.25632
                   10    5448       0       170       0       2.25632
                   11    5448       0       170       0       2.25632
                   12    5448       0       170       0       2.25632
                   13     408       1       190       0       2.15889
                   14     408       1       190       0       2.15889
                   15    1344       1       190       0       2.15889
                   16    1344       1       190       0       2.15889
                   17    1440       1       190       0       2.15889
                   18    1680       0       190       0       2.15889
                   19    1680       0       190       0       2.15889
                   20    1680       0       190       0       2.15889
                   21    1680       0       190       0       2.15889
                   22    1680       0       190       0       2.15889
                   23     408       1       220       0       2.02758
                   24     408       1       220       0       2.02758
                   25     504       1       220       0       2.02758
                   26     504       1       220       0       2.02758
                   27     504       1       220       0       2.02758
                   28     528       0       220       0       2.02758
                   29     528       0       220       0       2.02758
                   30     528       0       220       0       2.02758
                   31     528       0       220       0       2.02758
                   32     528       0       220       0       2.02758
```

Output 39.1.2. Motorette Failure: Model A

```
                                  The LIFEREG Procedure

                                    Model Information

                        Data Set                        WORK.MOTORS
                        Dependent Variable                Log(time)
                        Censoring Variable                   censor
                        Censoring Value(s)                        0
                        Number of Observations                   30
                        Noncensored Values                       17
                        Right Censored Values                    13
                        Left Censored Values                      0
                        Interval Censored Values                  0
                        Missing Values                            2
                        Name of Distribution                Weibull
                        Log Likelihood                 -22.95148315


                          Type III Analysis of Effects

                                              Wald
                       Effect       DF    Chi-Square    Pr > ChiSq

                       z             1       99.5239        <.0001


                         Analysis of Parameter Estimates

                              Standard   95% Confidence     Chi-
         Parameter     DF Estimate    Error       Limits       Square Pr > ChiSq

         Intercept      1 -11.8912   1.9655 -15.7435  -8.0389   36.60     <.0001
         z              1   9.0383   0.9060   7.2626  10.8141   99.52     <.0001
         Scale          1   0.3613   0.0795   0.2347   0.5561
         Weibull Shape  1   2.7679   0.6091   1.7982   4.2605
```

Output 39.1.3. Motorette Failure: Model B

```
                                    The LIFEREG Procedure

                                      Model Information

                           Data Set                      WORK.MOTORS
                           Dependent Variable              Log(time)
                           Censoring Variable                 censor
                           Censoring Value(s)                      0
                           Number of Observations                 30
                           Noncensored Values                     17
                           Right Censored Values                  13
                           Left Censored Values                    0
                           Interval Censored Values                0
                           Missing Values                          2
                           Name of Distribution            Lognormal
                           Log Likelihood               -24.47381031


                                Type III Analysis of Effects

                                                 Wald
                          Effect       DF    Chi-Square    Pr > ChiSq

                          z             1       42.0001        <.0001


                               Analysis of Parameter Estimates

                                        Standard   95% Confidence     Chi-
                   Parameter     DF Estimate    Error       Limits       Square Pr > ChiSq

                   Intercept      1 -10.4706   2.7719 -15.9034  -5.0377   14.27     0.0002
                   z              1   8.3221   1.2841   5.8052  10.8389   42.00     <.0001
                   Scale          1   0.6040   0.1107   0.4217   0.8652
```

Output 39.1.4. Motorette Failure: Fitted Models

```
                                          fitted models

_MODEL_    _NAME_       _TYPE_     _DIST_       _STATUS_      _LNLIKE_        time    Intercept          z       _SCALE_

   A       time         PARMS     Weibull     0 Converged     -22.9515     -1.0000     -11.8912     9.03834      0.36128
   A       Intercept    COV       Weibull     0 Converged     -22.9515    -11.8912       3.8632    -1.77878      0.03448
   A       z            COV       Weibull     0 Converged     -22.9515      9.0383      -1.7788     0.82082     -0.01488
   A       Scale        COV       Weibull     0 Converged     -22.9515      0.3613       0.0345    -0.01488      0.00632
   B       time         PARMS     Lognormal   0 Converged     -24.4738     -1.0000     -10.4706     8.32208      0.60403
   B       Intercept    COV       Lognormal   0 Converged     -24.4738    -10.4706       7.6835    -3.55566      0.03267
   B       z            COV       Lognormal   0 Converged     -24.4738      8.3221      -3.5557     1.64897     -0.01285
   B       Scale        COV       Lognormal   0 Converged     -24.4738      0.6040       0.0327    -0.01285      0.01226
```

Output 39.1.5. Motorette Failure: Quantile Estimates and Confidence Limits

```
                                 quantile estimates and confidence limits

   temp    time    censor    control       z       _PROB_    predtime       std      ltime      stde       upper       lower

    130     .        0          1      2.48016      0.1     16519.27    5999.85     9.7123   0.36320    29969.51     9105.47
    130     .        0          1      2.48016      0.5     32626.65    9874.33    10.3929   0.30265    53595.71    19861.63
    130     .        0          1      2.48016      0.9     50343.22   15044.35    10.8266   0.29884    82183.49    30838.80
    150     .        0          1      2.36295      0.1      5726.74    1569.34     8.6529   0.27404     8976.12     3653.64
    150     .        0          1      2.36295      0.5     11310.68    2299.92     9.3335   0.20334    15787.62     8103.28
    150     .        0          1      2.36295      0.9     17452.49    3629.28     9.7672   0.20795    24545.37    12409.24
    130     .        0          1      2.48016      0.1     12033.19    5482.34     9.3954   0.45560    25402.68     5700.09
    130     .        0          1      2.48016      0.5     26095.68   11359.45    10.1695   0.43530    53285.36    12779.95
    130     .        0          1      2.48016      0.9     56592.19   26036.90    10.9436   0.46008   120349.65    26611.42
    150     .        0          1      2.36295      0.1      4536.88    1443.07     8.4200   0.31808     7643.71     2692.83
    150     .        0          1      2.36295      0.5      9838.86    2901.15     9.1941   0.29487    15957.38     6066.36
    150     .        0          1      2.36295      0.9     21336.97    7172.34     9.9682   0.33615    37029.72    12294.62
```

Example 39.2. Computing Predicted Values for a Tobit Model

The LIFEREG Procedure can be used to perform a Tobit analysis. The Tobit model, described by Tobin (1958), is a regression model for left-censored data assuming a normally distributed error term. The model parameters are estimated by maximum likelihood. PROC LIFEREG provides estimates of the parameters of the distribution of the **uncensored** data. Refer to Greene (1993) and Maddala (1983) for a more complete discussion of censored normal data and related distributions. This example shows how you can use PROC LIFEREG and the DATA step to compute two of the three types of predicted values discussed there.

Consider a continuous random variable Y, and a constant C. If you were to sample from the distribution of Y but discard values less than (greater than) C, the distribution of the remaining observations would be **truncated** on the left (right). If you were to sample from the distribution of Y and report values less than (greater than) C as C, the distribution of the sample would be left (right) **censored**.

The probability density function of the truncated random variable Y' is given by

$$f_{\mathrm{Y}'}(y) = \frac{f_{\mathrm{Y}}(y)}{\Pr(\mathrm{Y} > \mathrm{C})} \quad \text{for} \quad y > \mathrm{C}$$

where $f_{\mathrm{Y}}(y)$ is the probability density function of Y. PROC LIFEREG cannot compute the proper likelihood function to estimate parameters or predicted values for a truncated distribution.

Suppose the model being fit is specified as follows:

$$\mathrm{Y}_i^* = \mathbf{x}_i'\beta + \epsilon_i$$

where ϵ_i is a normal error term with zero mean and standard deviation σ.

Define the censored random variable Y_i as

$$\begin{aligned} \mathrm{Y}_i &= 0 \quad \text{if} \quad \mathrm{Y}_i^* \leq 0 \\ \mathrm{Y}_i &= \mathrm{Y}_i^* \quad \text{if} \quad \mathrm{Y}_i^* > 0 \end{aligned}$$

This is the Tobit model for left-censored normal data. Y_i^* is sometimes called the *latent variable*. PROC LIFEREG estimates parameters of the distribution of Y_i^* by maximum likelihood.

You can use the LIFEREG procedure to compute predicted values based on the mean functions of the latent and observed variables. The mean of the latent variable Y_i^* is $\mathbf{x}_i'\beta$ and you can compute values of the mean for different settings of $\mathbf{x}_i$ by specifying XBETA=*variable-name* in an OUTPUT statement. Estimates of $\mathbf{x}_i'\beta$ for each observation will be written to the OUT= data set. Predicted values of the observed variable Y_i can be computed based on the mean

$$E(\mathrm{Y}_i) = \Phi\left(\frac{\mathbf{x}_i'\beta}{\sigma}\right)(\mathbf{x}_i'\beta + \sigma\lambda_i)$$

where

$$\lambda_i = \frac{\phi(\mathbf{x}_i'\beta/\sigma)}{\Phi(\mathbf{x}_i'\beta/\sigma)}$$

ϕ and Φ represent the normal probability density and cumulative distribution functions.

Although the distribution of ϵ_i in the Tobit model is often assumed normal, you can use other distributions for the Tobit model in the LIFEREG procedure by specifying a distribution with the DISTRIBUTION= option in the MODEL statement. One distribution should be mentioned is the logistic distribution. For this distribution, the MLE has bounded influence function with respect to the response variable, but not the design variables. If you believe your data has outliers in the response direction, you might try this distribution for some robust estimation of the Tobit model.

With the logistic distribution the predicted values of the observed variable Y_i can be computed based on the mean of Y_i^*

$$E(\mathrm{Y}_i) = \sigma \ln(1 + \exp(\mathbf{x}_i'\beta/\sigma))$$

The following table shows a subset of the Mroz (1987) data set. In this data, Hours is the number of hours the wife worked outside the household in a given year, Yrs_Ed is the years of education, and Yrs_Exp is the years of work experience. A Tobit model will be fit to the hours worked with years of education and experience as covariates.

Hours	Yrs_Ed	Yrs_Exp
0	8	9
0	8	12
0	9	10
0	10	15
0	11	4
0	11	6
1000	12	1
1960	12	29
0	13	3
2100	13	36
3686	14	11
1920	14	38
0	15	14
1728	16	3
1568	16	19
1316	17	7
0	17	15

If the wife was not employed (worked 0 hours), her hours worked will be left-censored at zero. In order to accommodate left censoring in PROC LIFEREG, you need two variables to indicate censoring status of observations. You can think of these variables as lower and upper endpoints of interval censoring. If there is no censoring, set both variables to the observed value of Hours. To indicate left censoring, set the lower endpoint to missing and the upper endpoint to the censored value, zero in this case.

The following statements create a SAS data set with the variables Hours, Yrs_Ed, and Yrs_Exp from the preceding data. A new variable, Lower is created such that Lower=. if Hours=0 and Lower=Hours if Hours>0.

```
data subset;
   input Hours Yrs_Ed Yrs_Exp @@;
   if Hours eq 0
      then Lower=.;
      else Lower=Hours;
datalines;
0 8 9 0 8 12 0 9 10 0 10 15 0 11 4 0 11 6
1000 12 1 1960 12 29 0 13 3 2100 13 36
3686 14 11 1920 14 38 0 15 14 1728 16 3
1568 16 19 1316 17 7 0 17 15
;
```

The following statements fit a normal regression model to the left-censored Hours data using Yrs_Ed and Yrs_Exp as covariates. You will need the estimated standard deviation of the normal distribution to compute the predicted values of the censored distribution from the preceding formulas. The data set OUTEST contains the standard deviation estimate in a variable named _SCALE_. You also need estimates of $\mathbf{x}_i'\beta$. These are contained in the data set OUT as the variable Xbeta

```
proc lifereg data=subset outest=OUTEST(keep=_scale_);
   model (lower, hours) = yrs_ed yrs_exp / d=normal;
   output out=OUT xbeta=Xbeta;
run;
```

Output 39.2.1 shows the results of the model fit. These tables show parameter estimates for the uncensored, or latent variable, distribution.

Output 39.2.1. Parameter Estimates from PROC LIFEREG

```
                              The LIFEREG Procedure

                                Model Information

                    Data Set                       WORK.SUBSET
                    Dependent Variable                   Lower
                    Dependent Variable                   Hours
                    Number of Observations                  17
                    Noncensored Values                       8
                    Right Censored Values                    0
                    Left Censored Values                     9
                    Interval Censored Values                 0
                    Name of Distribution                Normal
                    Log Likelihood                 -74.9369977


                        Analysis of Parameter Estimates

                        Standard   95% Confidence     Chi-
      Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

      Intercept  1 -5598.64 2850.248 -11185.0 -12.2553    3.86     0.0495
      Yrs_Ed     1 373.1477 191.8872  -2.9442 749.2397    3.78     0.0518
      Yrs_Exp    1  63.3371  38.3632 -11.8533 138.5276    2.73     0.0987
      Scale      1 1582.870 442.6732 914.9433 2738.397
```

The following statements combine the two data sets created by PROC LIFEREG to compute predicted values for the censored distribution. The OUTEST= data set contains the estimate of the standard deviation from the uncensored distribution, and the OUT= data set contains estimates of $\mathbf{x}_i'\beta$.

```
data predict;
   drop lambda _scale_ _prob_;
   set out;
   if _n_ eq 1 then set outest;
   lambda = pdf('NORMAL',Xbeta/_scale_)
            / cdf('NORMAL',Xbeta/_scale_);
   Predict = cdf('NORMAL', Xbeta/_scale_)
             * (Xbeta + _scale_*lambda);
   label Xbeta='MEAN OF UNCENSORED VARIABLE'
         Predict = 'MEAN OF CENSORED VARIABLE';
run;

proc print data=predict noobs label;
   var hours lower yrs: xbeta predict;
run;
```

Output 39.2.2 shows the original variables, the predicted means of the uncensored distribution, and the predicted means of the censored distribution.

Output 39.2.2. Predicted Means from PROC LIFEREG

```
                                                  MEAN OF      MEAN OF
                                                UNCENSORED     CENSORED
      Hours      Lower     Yrs_Ed     Yrs_Exp    VARIABLE      VARIABLE

          0         .          8          9      -2043.42        73.46
          0         .          8         12      -1853.41        94.23
          0         .          9         10      -1606.94       128.10
          0         .         10         15       -917.10       276.04
          0         .         11          4      -1240.67       195.76
          0         .         11          6      -1113.99       224.72
       1000      1000         12          1      -1057.53       238.63
       1960      1960         12         29        715.91      1052.94
          0         .         13          3       -557.71       391.42
       2100      2100         13         36       1532.42      1672.50
       3686      3686         14         11        322.14       805.58
       1920      1920         14         38       2032.24      2106.81
          0         .         15         14        885.30      1170.39
       1728      1728         16          3        561.74       951.69
       1568      1568         16         19       1575.13      1708.24
       1316      1316         17          7       1188.23      1395.61
          0         .         17         15       1694.93      1809.97
```

Example 39.3. Overcoming Convergence Problems by Specifying Initial Values

This example illustrates the use of parameter initial value specification to help overcome convergence difficulties.

The following statements create a data set and request a Weibull regression model be fit to the data.

```
data raw;
   input censor x c1 @@;
   datalines;
0 16 0.00   0 17 0.00   0 18 0.00
0 17 0.04   0 18 0.04   0 18 0.04
0 23 0.40   0 22 0.40   0 22 0.40
0 33 4.00   0 34 4.00   0 35 4.00
1 54 40.00  1 54 40.00  1 54 40.00
1 54 400.00 1 54 400.00 1 54 400.00
;
run;

proc print;
run;

title ’OLS (default) initial values’;
proc lifereg data=raw;
   model x*censor(1) = c1 / distribution = weibull itprint;
run;
```

Output 39.3.1 shows the data set contents.

Output 39.3.1. Contents of the Data Set

```
                  Obs    censor     x        c1

                    1       0      16      0.00
                    2       0      17      0.00
                    3       0      18      0.00
                    4       0      17      0.04
                    5       0      18      0.04
                    6       0      18      0.04
                    7       0      23      0.40
                    8       0      22      0.40
                    9       0      22      0.40
                   10       0      33      4.00
                   11       0      34      4.00
                   12       0      35      4.00
                   13       1      54     40.00
                   14       1      54     40.00
                   15       1      54     40.00
                   16       1      54    400.00
                   17       1      54    400.00
                   18       1      54    400.00
```

Convergence was not attained in 50 iterations for this model, as the messages to the log indicate:

```
WARNING: Convergence was not attained in 50 iterations. You may want to
         increase the maximum number of iterations (MAXITER= option) or
         change the convergence criteria (CONVERGE = value) in the MODEL
         statement.
WARNING: The procedure is continuing in spite of the above warning. Results
         shown are based on the last maximum likelihood iteration. Validity
         of the model fit is questionable.
```

The first line (iter=0) of the iteration history table, in Output 39.3.2, shows the default initial ordinary least squares (OLS) estimates of the parameters.

Output 39.3.2. Initial Least Squares

```
                            OLS (default) initial values

    Iter     Ridge        Loglike        Intercept              c1           Scale

       0         0     -22.891088     3.2324769714    0.0020664542    0.3995754195
```

The log logistic distribution is more robust to large values of the response than the Weibull, so one approach to improving the convergence performance is to fit a log logistic distribution, and if this converges, use the resulting parameter estimates as initial values in a subsequent fit of a model with the Weibull distribution.

The following statements fit a log logistic distribution to the data.

```
proc lifereg data=raw;
   model x*censor(1) = c1 / distribution = llogistic;
run;
```

The algorithm converges, and the maximum likelihood estimates for the log logistic distribution are shown in Output 39.3.3

Output 39.3.3. Estimates from the Log Logistic Distribution

```
                              The LIFEREG Procedure

                                Model Information

                      Data Set                        WORK.RAW
                      Dependent Variable                Log(x)
                      Censoring Variable                censor
                      Censoring Value(s)                     1
                      Number of Observations                18
                      Noncensored Values                    12
                      Right Censored Values                  6
                      Left Censored Values                   0
                      Interval Censored Values               0
                      Name of Distribution           LLogistic
                      Log Likelihood              12.093136846


                          Analysis of Parameter Estimates

                               Standard   95% Confidence     Chi-
      Parameter      DF Estimate    Error       Limits       Square Pr > ChiSq

      Intercept       1   2.8983   0.0318   2.8360   2.9606 8309.43     <.0001
      c1              1   0.1592   0.0133   0.1332   0.1852  143.85     <.0001
      Scale           1   0.0498   0.0122   0.0308   0.0804
```

The following statements re-fit the Weibull model using the maximum likelihood estimates from the log logistic fit as initial values.

```
proc lifereg data=raw outest=outest;
   model x*censor(1) = c1 / itprint distribution = weibull
                            intercept=2.898 initial=0.16 scale=0.05;
   output out=out xbeta=xbeta;
run;
```

Examination of the resulting output in Output 39.3.4 shows that the convergence problem has been solved by specifying different initial values.

As an example, the following invocation of PROC LIFEREG, using the INEST= data set providing starting values for the three parameters, is equivalent to the previous invocation.

```
data in;
   input  intercept c1 scale;
   datalines;
2.898 0.16 0.05
;
proc lifereg data=raw inest=in outest=outest;
   model x*censor(1) = c1 / itprint distribution = weibull;
   output out=out xbeta=xbeta;
run;
```

Output 39.3.4. Final Estimates from the Weibull Distribution

```
                                The LIFEREG Procedure

                                  Model Information

                         Data Set                           WORK.RAW
                         Dependent Variable                   Log(x)
                         Censoring Variable                   censor
                         Censoring Value(s)                        1
                         Number of Observations                   18
                         Noncensored Values                       12
                         Right Censored Values                     6
                         Left Censored Values                      0
                         Interval Censored Values                  0
                         Name of Distribution                Weibull
                         Log Likelihood                 11.232023272


            Algorithm converged.


                            Analysis of Parameter Estimates

                                    Standard   95% Confidence     Chi-
             Parameter     DF Estimate    Error       Limits       Square Pr > ChiSq

             Intercept      1   2.9699   0.0326   2.9059   3.0338 8278.86     <.0001
             c1             1   0.1435   0.0165   0.1111   0.1758   75.43     <.0001
             Scale          1   0.0844   0.0189   0.0544   0.1308
             Weibull Shape  1  11.8526   2.6514   7.6455  18.3749
```

Example 39.4. Analysis of Arbitrarily Censored Data with Interaction Effects

The following artificial data are for a study of the natural recovery time of mice after injection of a certain toxin. 20 mice were grouped by sex (sex: 1 = Male, 2 = Female) with equal sizes. Their ages (in days) were recorded at the injection. Their recovery times (in minutes) were also recorded. Toxin density in blood was used to decide whether a mouse recovered. Mice were checked at two times for recovery. If a mouse had recovered at the first time, the observation is left-censored, and no further measurement is made. The variable time1 is set to missing and time2 is set to the measurement time to indicate left-censoring. If a mouse had not recovered at the first time, it was checked later at a second time. If it had recovered by the second measurement time, the observation is interval-censored and the variable time1 is set to the first measurement time and time2 is set to the second measurement time. If there was no recovery at the second measurement, the observation is right-censored, and time1 is set to the second measurement time and time2 is set to missing to indicate right-censoring.

The following statements create a SAS data set containing the data from the experiment and fit a Weibull model with age, sex, and age and sex interaction as covariates.

```
title 'Natural Recovery Time';
   data mice;
      input sex age time1 time2 ;
      datalines;
   1  57  631   631
   1  45  .     170
   1  54  227   227
   1  43  143   143
```

```
1  64  916   .
1  67  691   705
1  44  100   100
1  59  730   .
1  47  365   365
1  74  1916  1916
2  79  1326  .
2  75  837   837
2  84  1200  1235
2  54  .     365
2  74  1255  1255
2  71  1823  .
2  65  537   637
2  33  583   683
2  77  955   .
2  46  577   577
;

data xrow1;
   input sex age time1 time2 ;
   datalines;
1  50  .  .
;

data xrow2;
   input sex age time1 time2 ;
   datalines;
 2  60.6  .  .
 ;

proc lifereg data=mice xdata=xrow1;
   class sex  ;
   model (time1, time2) = age sex age*sex / dist=Weibull;

   probplot  / nodata
       font = swiss
       plower=.5
       vref(intersect) = 75
       vreflab = '75 Percent'
       vreflabpos = 2
       cfit=blue
       cframe=ligr
       ;
   inset  / cfill = white
       ctext = blue;
run;
```

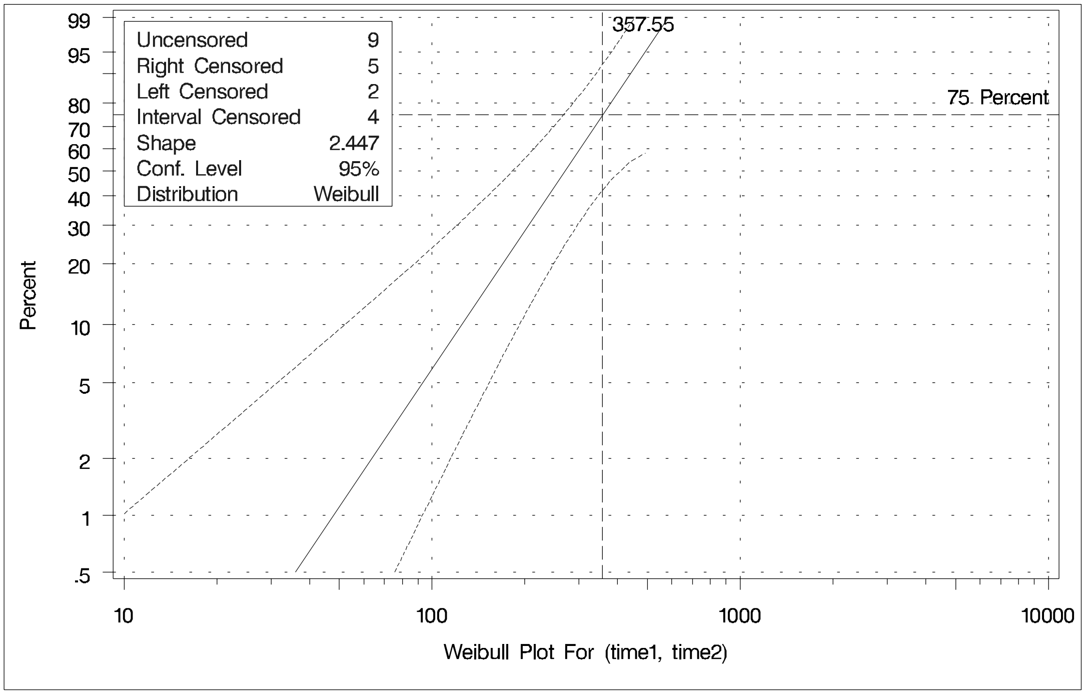
Standard output is shown in Output 39.4.1. Tables containing general model information, Type III tests for the main effects and interaction terms, and parameter estimates are created.

Output 39.4.1. Parameter Estimates for the Interaction Model

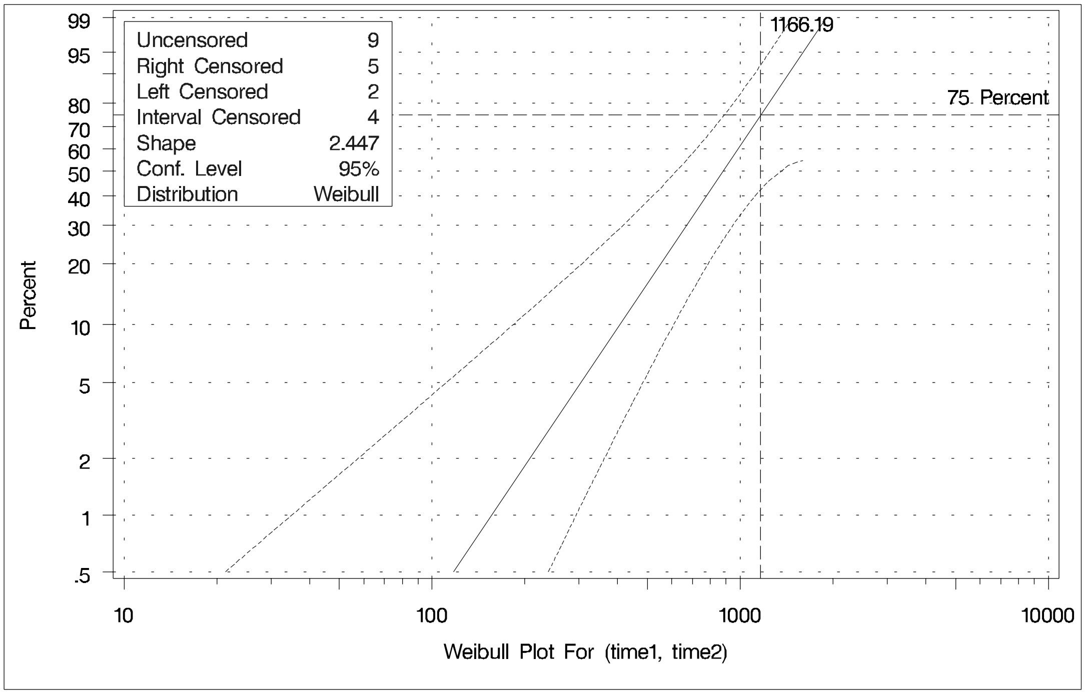
```
                              Natural Recovery Time

                              The LIFEREG Procedure

                                Model Information

                         Data Set                       WORK.MICE
                         Dependent Variable            Log(time1)
                         Dependent Variable            Log(time2)
                         Number of Observations                20
                         Noncensored Values                     9
                         Right Censored Values                  5
                         Left Censored Values                   2
                         Interval Censored Values               4
                         Name of Distribution             Weibull
                         Log Likelihood              -25.91033295


                           Type III Analysis of Effects

                                               Wald
                       Effect       DF    Chi-Square    Pr > ChiSq

                       age           1       33.8496        <.0001
                       sex           1       14.0245        0.0002
                       age*sex       1       10.7196        0.0011


                          Analysis of Parameter Estimates

                                      Standard   95% Confidence     Chi-
        Parameter       DF Estimate    Error        Limits       Square Pr > ChiSq

        Intercept        1   5.4110   0.5549   4.3234   6.4986   95.08     <.0001
        age              1   0.0250   0.0086   0.0081   0.0419    8.42     0.0037
        sex           1  1  -3.9808   1.0630  -6.0643  -1.8974   14.02     0.0002
        sex           2  0   0.0000   0.0000   0.0000   0.0000     .         .
        age*sex       1  1   0.0613   0.0187   0.0246   0.0980   10.72     0.0011
        age*sex       2  0   0.0000   0.0000   0.0000   0.0000     .         .
        Scale            1   0.4087   0.0900   0.2654   0.6294
        Weibull Shape    1   2.4468   0.5391   1.5887   3.7682
```

The following two plots display the predicted probability against the recovery time for two different populations. Output 39.4.2 is created with the PROBPLOT statement with the option XDATA= xrow1, which specifies the population with sex = 1, age = 50. Although the SAS statements are not shown , Output 39.4.3 is created with the PROBPLOT statement with the option XDATA= xrow2, which specifies the population with sex = 2, age = 60.6. These are the default values that the LIFEREG procedure would use for the probability plot if the XDATA= option had not been specified. Reference lines are used to display specified predicted probability points and their relative locations on the plot.

Output 39.4.2. Probability Plot for Recovery Time with sex = 1, age = 50



Output 39.4.3. Probability Plot for Recovery Time with sex = 2, age = 60.6



Example 39.5. Probability Plotting–Right Censoring

The following statements create a SAS data set containing observed and right-censored lifetimes of 70 diesel engine fans (Nelson 1982, p. 318).

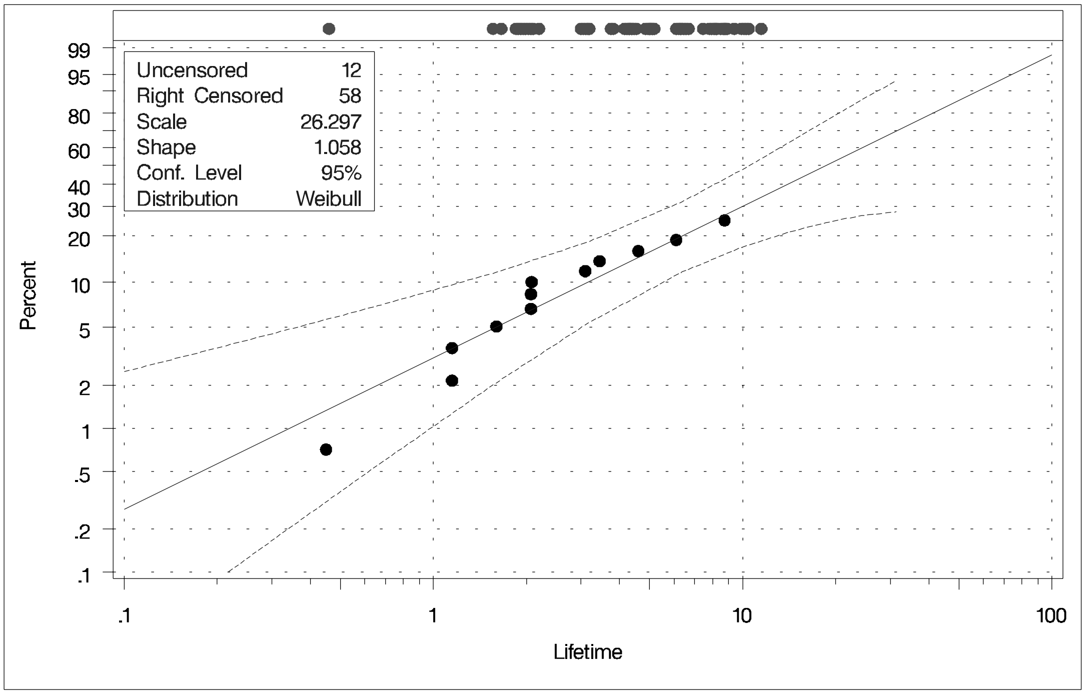
```
title ’Engine Fan Lifetime Study’;
   data fan;
      input lifetime censor@@;
      lifetime = lifetime / 1000;
      label lifetime = Lifetime;
      datalines;
       450 0    460 1   1150 0   1150 0   1560 1
      1600 0   1660 1   1850 1   1850 1   1850 1
      1850 1   1850 1   2030 1   2030 1   2030 1
      2070 0   2070 0   2080 0   2200 1   3000 1
      3000 1   3000 1   3000 1   3100 0   3200 1
      3450 0   3750 1   3750 1   4150 1   4150 1
      4150 1   4150 1   4300 1   4300 1   4300 1
      4300 1   4600 0   4850 1   4850 1   4850 1
      4850 1   5000 1   5000 1   5000 1   6100 1
      6100 0   6100 1   6100 1   6300 1   6450 1
      6450 1   6700 1   7450 1   7800 1   7800 1
      8100 1   8100 1   8200 1   8500 1   8500 1
      8500 1   8750 1   8750 0   8750 1   9400 1
      9900 1  10100 1  10100 1  10100 1  11500 1
      ;
   run;
```

Some of the fans had not failed at the time the data were collected, and the unfailed units have right-censored lifetimes. The variable LIFETIME represents either a failure time or a censoring time in thousands of hours. The variable CENSOR is equal to 0 if the value of LIFETIME is a failure time, and it is equal to 1 if the value is a censoring time. The following statements use the LIFEREG procedure to produce the probability plot with an inset for the engine lifetimes.

```
symbol v=dot c=white;
   proc lifereg;
      model lifetime*censor( 1 ) = / d = weibull;
      probplot
          cencolor = red
          cframe   = ligr
          cfit     = blue
          ppout
          npintervals=simul
          ;
      inset /
          cfill = white
          ctext = blue;
   run;
```

The resulting graphical output is shown in Output 39.5.1. The estimated CDF, a line representing the maximum likelihood fit, and pointwise parametric confidence bands are plotted in the body of Output 39.5.1. The values of right-censored observations are plotted along the top of the graph. The “Cumulative Probability Estimates” table is also created in Output 39.5.2.

Output 39.5.1. Probability Plot for the Fan Data



Output 39.5.2. CDF Estimates

```
                              The LIFEREG Procedure

                         Cumulative Probability Estimates
                                    Simultaneous                     Kaplan-
                                   95% Confidence       Kaplan-        Meier
                    Cumulative         Limits             Meier     Standard
      Lifetime     Probability     Lower      Upper    Estimate        Error

          0.45          0.0071    0.0007     0.2114      0.0143       0.0142
          1.15          0.0215    0.0033     0.2114      0.0288       0.0201
          1.15          0.0360    0.0073     0.2168      0.0433       0.0244
           1.6          0.0506    0.0125     0.2304      0.0580       0.0282
          2.07          0.0666    0.0190     0.2539      0.0751       0.0324
          2.07          0.0837    0.0264     0.2760      0.0923       0.0361
          2.08          0.1008    0.0344     0.2972      0.1094       0.0392
           3.1          0.1189    0.0436     0.3223      0.1283       0.0427
          3.45          0.1380    0.0535     0.3471      0.1477       0.0460
           4.6          0.1602    0.0653     0.3844      0.1728       0.0510
           6.1          0.1887    0.0791     0.4349      0.2046       0.0581
          8.75          0.2488    0.0884     0.6391      0.2930       0.0980
```

Example 39.6. Probability Plotting–Arbitrarily Censoring

Table 39.3 contains microprocessor failure data from Nelson (1990). Units were inspected at predetermined time intervals. The data consist of inspection interval endpoints (in hours) and the number of units failing in each interval. A missing (.) lower endpoint indicates left censoring, and a missing upper endpoint indicates right censoring. These can be thought of as semi-infinite intervals with a lower (upper) endpoint of negative (positive) infinity for left (right) censoring.

Table 39.3. Interval-Censored Data

Lower Endpoint	Upper Endpoint	Number Failed
.	6	6
6	12	2
24	48	2
24	.	1
48	168	1
48	.	839
168	500	1
168	.	150
500	1000	2
500	.	149
1000	2000	1
1000	.	147
2000	.	122

The following SAS program will compute the Turnbull estimate and create a lognormal probability plot.

```
data micro;
   input t1 t2 f ;
   datalines;
   . 6 6
   6 12 2
   12 24 0
   24 48 2
   24 .  1
   48 168 1
   48 .   839
   168 500 1
   168 .   150
   500 1000 2
   500 .    149
   1000 2000 1
   1000 . 147
   2000 . 122
   ;

symbol v=dot c=white;
proc lifereg data=micro;
   model ( t1 t2 ) = / d=lognormal intercept=25 scale=5;
   weight f;
```

```
      probplot
         cframe = ligr
         cfit   = blue
         pupper = 10
         itprintem
         printprobs
         maxitem = (1000,25)
         ppout;
      inset / cfill = white;
   run;
```

The two initial values INTERCEPT= 25 and SCALE= 5 in the MODEL statement are used to aid convergence in the model-fitting algorithm.

The following tables are created by the PROBPLOT statement in addition to the standard tabular output from the MODEL statement. Output 39.6.1 shows the iteration history for the Turnbull estimate of the CDF for the microprocessor data. With both options ITPRINTEM and PRINTPROBS specified in the PROBPLOT statement, this table contains the log likelihoods and interval probabilities for every 25th iteration and the last iteration. It would only contain the log likelihoods if the option PRINTPROBS were not specified.

Output 39.6.1. Iteration History for the Turnbull Estimate

```
                                 The LIFEREG Procedure

            Iteration History for the Turnbull Estimate of the CDF
Iteration   Loglikelihood       (., 6)      (6, 12)     (24, 48)    (48, 168)
                             (168, 500)   (500, 1000)   (1000, 2000)    (2000, .)

        0      -1133.4051        0.125        0.125        0.125        0.125
                                 0.125          0.125          0.125        0.125
       25      -104.16622   0.00421644   0.00140548   0.00140648   0.00173338
                            0.00237846    0.00846094     0.04565407   0.93474475
       50      -101.15151   0.00421644   0.00140548   0.00140648   0.00173293
                            0.00234891    0.00727679     0.01174486   0.96986811
       75      -101.06641   0.00421644   0.00140548   0.00140648   0.00173293
                            0.00234891    0.00727127     0.00835638    0.9732621
      100      -101.06534   0.00421644   0.00140548   0.00140648   0.00173293
                            0.00234891    0.00727125     0.00801814   0.97360037
      125      -101.06533   0.00421644   0.00140548   0.00140648   0.00173293
                            0.00234891    0.00727125     0.00798438   0.97363413
      130      -101.06533   0.00421644   0.00140548   0.00140648   0.00173293
                            0.00234891    0.00727125       0.007983   0.97363551
```

Output 39.6.2. Summary for the Turnbull Algorithm

```
                           The LIFEREG Procedure

      Lower        Upper                          Reduced         Lagrange
   Lifetime     Lifetime    Probability          Gradient       Multiplier

          .            6         0.0042                 0                0
          6           12         0.0014                 0                0
         24           48         0.0014                 0                0
         48          168         0.0017                 0                0
        168          500         0.0023                 0                0
        500         1000         0.0073      -7.219342E-9                0
       1000         2000         0.0080      -0.037063236                0
       2000            .         0.9736      0.0003038877                0
```

The table in Output 39.6.2 summarizes the Turnbull estimates of the interval probabilities, the reduced gradients, and Lagrange multipliers as described in the section "Arbitrarily Censored Data" on page 2119.

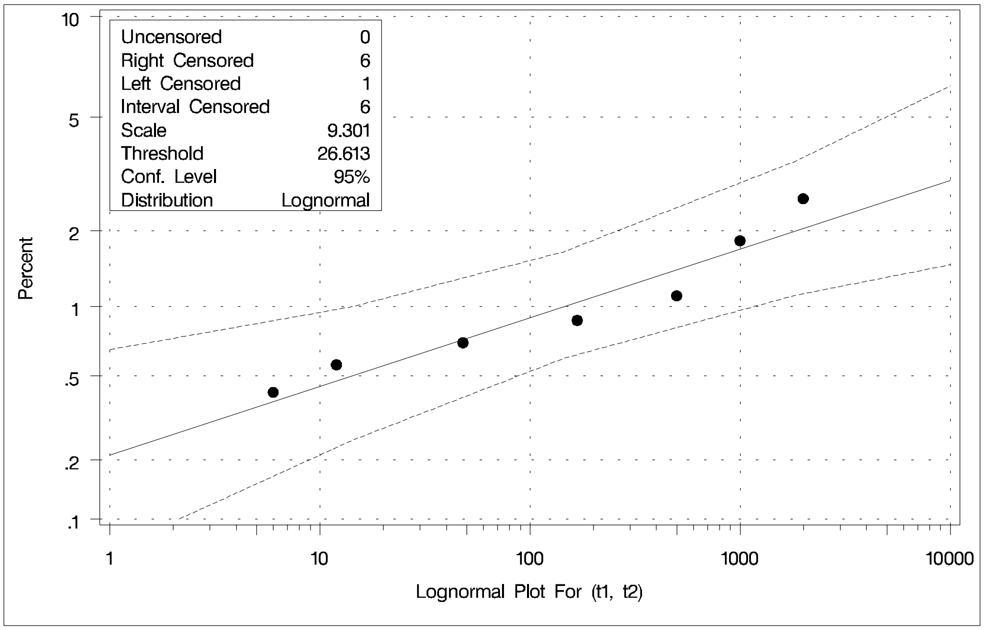
Output 39.6.3. Final CDF Estimates for Turnbull Algorithm

```
                           The LIFEREG Procedure

                      Cumulative Probability Estimates
                                              Pointwise 95%
                                               Confidence
      Lower        Upper     Cumulative          Limits          Standard
   Lifetime     Lifetime    Probability      Lower     Upper        Error

          6            6         0.0042     0.0019    0.0094       0.0017
         12           24         0.0056     0.0028    0.0112       0.0020
         48           48         0.0070     0.0038    0.0130       0.0022
        168          168         0.0088     0.0047    0.0164       0.0028
        500          500         0.0111     0.0058    0.0211       0.0037
       1000         1000         0.0184     0.0094    0.0357       0.0063
       2000         2000         0.0264     0.0124    0.0553       0.0101
```

Output 39.6.3 shows the final estimate of the CDF, along with standard errors and nonparametric confidence limits. Two kinds of nonparametric confidence limits, pointwise or simultaneous, are available. The default is the pointwise nonparametric confidence limits. You can specify the simultaneous nonparametric confidence limits by the NPINTERVALS= SIMUL option.

Output 39.6.4 shows the CDF estimates, the maximum likelihood fit, and the pointwise parametric confidence limits plotted on a lognormal probability plot.

Output 39.6.4. Lognormal Probability Plot for the Microprocessor Data



References

Abernethy, R.B. (1996), *The New Weibull Handbook*, Second Edition, North Palm Beach, FL: Robert B. Abernethy.

Allison, P.D. (1995), *Survival Analysis Using the SAS System: A Practical Guide*, Cary, NC: SAS Institute.

Cox, D.R. (1972), “Regression Models and Life Tables (with discussion),” *Journal of the Royal Statistical Society, Series B*, 34, 187–220.

Cox, D.R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman and Hall.

Elandt-Johnson, R.C. and Johnson, N.L. (1980), *Survival Models and Data Analysis*, New York: John Wiley & Sons, Inc.

Gentleman, R. and Geyer, C.J. (1994), “Maximum Likelihood for Interval Censored Data: Consistency and Computation,” *Biometrika*, 81 (3), 618–623.

Green, W.H. (1993), *Econometric Analysis*, Second Edition, New York: Cambridge University Press.

Gross, A.J. and Clark, V.A. (1975), *Survival Distributions: Reliability Applications in the Biomedical Sciences*, New York: John Wiley & Sons, Inc.

Kalbfleisch, J.D. and Prentice, R.L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons, Inc.

Klein, J.P. and Moeschberger, M.L. (1997), *Survival Analysis: Techniques for Censored and Truncated Data*, Berlin: Springer–Verlag.

Lawless, J.E. (1982), *Statistical Models and Methods for Lifetime Data*, New York: John Wiley & Sons, Inc.

Lee, E.T. (1980), *Statistical Methods for Survival Data Analysis*, Belmont, CA: Lifetime Learning Publications.

Maddala, G.S. (1983), *Limited-Dependent and Qualitative Variables in Econometrics*, New York: Cambridge University Press.

Meeker, W.Q. and Escobar, L.A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Mroz, T.A. (1987), “The Sensitivity of an Empirical Model of Married Women’s Work to Economic and Statistical Assumptions,” *Econometrica* 55, 765–799.

Nair, V.N. (1984), “Confidence Bands for Survival Functions With Censored Data: A Comparative Study,” *Technometrics*, 26 (3), 265–275.

Nelson, W. (1982), *Applied Life Data Analysis*, New York: John Wiley & Sons.

Nelson, W. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, New York: John Wiley & Sons.

Rao, C.R. (1973), *Linear Statistical Inference and Its Applications*, New York: John Wiley & Sons, Inc.

Tobin, J. (1958), “Estimation of Relationships for Limited Dependent Variables,” *Econometrica*, 26, 24–36.

Turnbull, B.W. (1976), “The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data,” *Journal of the Royal Statistical Society, Series B,* 38, 290–295.

Chapter 40
The LIFETEST Procedure

Chapter Contents

Chapter 40
The LIFETEST Procedure

Overview

A common feature of lifetime or survival data is the presence of right-censored observations due either to withdrawal of experimental units or to termination of the experiment. For such observations, you know only that the lifetime exceeded a given value; the exact lifetime remains unknown. Such data cannot be analyzed by ignoring the censored observations because, among other considerations, the longer-lived units are generally more likely to be censored. The analysis methodology must correctly use the censored observations as well as the uncensored observations.

Several texts that discuss the survival analysis methodology are Collett (1994), Cox and Oakes (1984), Kalbfleisch and Prentice (1980), Klein and Moeschberger (1997), Lawless (1982), and Lee (1992). Users interested in the theory should consult Fleming and Harrington (1991) and Andersen et al. (1992).

Usually, a first step in the analysis of survival data is the estimation of the distribution of the survival times. Survival times are often called *failure* times, and *event* times are uncensored survival times. The survival distribution function (SDF), also known as the survivor function, is used to describe the lifetimes of the population of interest. The SDF evaluated at t is the probability that an experimental unit from the population will have a lifetime exceeding t, that is

$$S(t) = \Pr(T > t)$$

where $S(t)$ denotes the survivor function and T is the lifetime of a randomly selected experimental unit. The LIFETEST procedure can be used to compute nonparametric estimates of the survivor function either by the product-limit method (also called the Kaplan-Meier method) or by the life-table method (also called the acturial method).

Some functions closely related to the SDF are the cumulative distribution function (CDF), the probability density function (PDF), and the hazard function. The CDF, denoted $F(t)$, is defined as $1 - S(t)$ and is the probability that a lifetime does not exceed t. The PDF, denoted $f(t)$, is defined as the derivative of $F(t)$, and the hazard function, denoted $h(t)$, is defined as $f(t)/S(t)$. If the life-table method is chosen, the estimates of the probability density function and the hazard function can also be computed. Plots of these estimates can be produced by a graphical or line printer device, or by launching the experimental graphics based on the output delivery system (ODS). For specific information about the experimental graphics that is available in PROC LIFETEST, see the section "ODS Graphics" on page 2190.

An important task in the analysis of survival data is the comparison of survival curves. It is of interest to determine whether the underlying populations of k ($k \geq 2$) samples have identical survivor functions. PROC LIFETEST provides nonparametic

k-sample tests based on weighted comparisons of the estimated hazard rate of the individual population under the null and alternative hypotheses. Corresponding to various weight functions, a variety of tests can be specified, which include the log-rank test, Wilcoxon test, Tarone-Ware test, Peto-Peto test, modified Peto-Peto test, and the Fleming-Harrington G_ρ family of tests. PROC LIFETEST also provides corresponding trend tests to detect ordered alternatives. Stratified tests can be specified to adjust for prognostic factors that affect the events rates in the various populations. A likelihood ratio test, based on an underlying exponential model, is also included to compare the survival curves of the samples.

There are other prognostic variables called covariates that are thought to be related to the failure time. These covariates can also be used to construct statistics to test for association between the covariates and the lifetime variable. PROC LIFETEST can compute two such test statistics: censored data linear rank statistics based on the exponential scores and the Wilcoxon scores. The corresponding tests are known as the log-rank test and the Wilcoxon test, respectively. These tests are computed by pooling over any defined strata, thus adjusting for the stratum variables.

Getting Started

You can use the LIFETEST procedure to compute nonparametric estimates of the survivor functions, to compare survival curves, and to compute rank tests for association of the failure time variable with covariates.

For simple analyses, only the PROC LIFETEST and TIME statements are required. Consider a sample of survival data. Suppose that the time variable is T and the censoring variable is C with value 1 indicating censored observations. The following statements compute the product-limit estimate for the sample:

```
proc lifetest;
   time t*c(1);
run;
```

You can use the STRATA statement to divide the data into various strata. A separate survivor function is then estimated for each stratum, and tests of the homogeneity of strata are performed. However, if the GROUP= option is also specified in the STRATA statement, the GROUP= variable is used to identify the samples whose survivor functions are to be compared and the STRATA variables are used to define the strata for the stratified tests. You can specify covariates in the TEST statement. PROC LIFETEST computes linear rank statistics to test the effects of these covariates on survival.

For example, consider the results of a small randomized trial on rats. Suppose you randomize 40 rats that have been exposed to a carcinogen into two treatment groups (Drug X and Placebo). The event of interest is death from cancer induced by the carcinogen. The response is the time from randomization to death. Four rats died of other causes; their survival times are regarded as censored observations. Interest lies in whether the survival distributions differ between the two treatments.

The data set Exposed contains four variables: Days (survival time in days from treatment to death), Status (censoring indicator variable: 0 if censored and 1 if not censored), Treatment (treatment indicator), and Sex (gender: F if female and M if male).

```
proc format;
   value Rx 1='Drug X' 0='Placebo';
data exposed;
  input Days  Status Treatment Sex $ @@;
  format Treatment Rx.;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
255   1   1   M   171   1   1   F
224   0   1   F   325   1   1   M
225   1   1   F   325   1   1   M
287   1   1   M   217   1   1   F
319   1   1   M   255   1   1   F
264   1   1   M   256   1   1   F
237   0   0   F   291   1   0   M
156   1   0   F   323   1   0   M
270   1   0   M   253   1   0   M
257   1   0   M   206   1   0   F
242   1   0   M   206   1   0   F
157   1   0   F   237   1   0   M
249   1   0   M   211   1   0   F
180   1   0   F   229   1   0   F
226   1   0   F   234   1   0   F
268   0   0   M   209   1   0   F
;
```

PROC LIFETEST is invoked to compute the product-limit estimate of the survivor function for each treatment and to compare the survivor functions between the two treatments.

```
ods html;
ods graphics on;

proc lifetest data=Exposed;
   time Days*Status(0);
   strata Treatment;
run;

ods graphics off;
ods html close;
```

In the TIME statement, the survival time variable, Days, is crossed with the censoring variable, Status, with the value 0 indicating censoring. That is, the values of Days are considered censored if the corresponding values of Status are 0; otherwise, they are considered as event times. In the STRATA statement, the variable Treatment is specified, which indicates that the data are to be divided into strata based on the values of Treatment. PROC LIFETEST computes the product-limit estimate for each stratum and tests whether the survivor functions are identical across strata. The experimental ODS GRAPHICS statement is specified to display the estimated survivor functions.

The results of the analysis are displayed in the following figures.

Figure 40.1 displays the product-limit survival estimate for the Drug X group (Treatment=1). The figure lists, for each observed time, the survival estimate, failure rate, standard error of the estimate, number of failures, and number of subjects remaining in the study.

```
                              The LIFETEST Procedure

                          Stratum 1: Treatment = Drug X

                         Product-Limit Survival Estimates

                                               Survival
                                               Standard      Number       Number
         Days       Survival      Failure        Error       Failed        Left

        0.000         1.0000            0            0          0            20
      171.000         0.9500       0.0500       0.0487          1            19
      179.000         0.9000       0.1000       0.0671          2            18
      217.000         0.8500       0.1500       0.0798          3            17
      224.000*             .            .            .          3            16
      225.000         0.7969       0.2031       0.0908          4            15
      255.000              .            .            .          5            14
      255.000         0.6906       0.3094       0.1053          6            13
      256.000              .            .            .          7            12
      256.000              .            .            .          8            11
      256.000              .            .            .          9            10
      256.000         0.4781       0.5219       0.1146         10             9
      262.000         0.4250       0.5750       0.1135         11             8
      264.000         0.3719       0.6281       0.1111         12             7
      287.000         0.3188       0.6813       0.1071         13             6
      319.000              .            .            .         14             5
      319.000         0.2125       0.7875       0.0942         15             4
      325.000              .            .            .         16             3
      325.000         0.1063       0.8938       0.0710         17             2
      355.000         0.0531       0.9469       0.0517         18             1
      378.000*        0.0531            .            .         18             0

             NOTE: The marked survival times are censored observations.
```

Figure 40.1. Survivor Function Estimate for the Drug X-Treated Rats

Figure 40.2 displays summary statistics of survival times for the Drug X group. It contains estimates of the 25th, 50th, and 75th percentiles and the corresponding 95% confidence limits.

The median survival time for rats in this treatment is 256 days. The mean and standard error are also displayed; however, it is noted that these values are underestimated because the largest observed time is censored and the estimation is restricted to the largest event time.

```
                              Quartile Estimates

                              Point     95% Confidence Interval
                 Percent    Estimate      [Lower      Upper)

                      75     319.000     262.000     325.000
                      50     256.000     255.000     319.000
                      25     255.000     217.000     256.000


                              Mean    Standard Error

                           271.131           11.877

NOTE: The mean survival time and its standard error were underestimated because
 the largest observation was censored and the estimation was restricted to the
                             largest event time.
```

Figure 40.2. Summary Statistics of Survival Times for Drug X-Treated Rats

```
                       Stratum 2: Treatment = Placebo

                       Product-Limit Survival Estimates

                                              Survival
                                              Standard     Number       Number
         Days      Survival     Failure        Error      Failed        Left

        0.000        1.0000           0            0           0          20
      156.000        0.9500      0.0500       0.0487           1          19
      157.000        0.9000      0.1000       0.0671           2          18
      180.000        0.8500      0.1500       0.0798           3          17
      206.000             .           .            .           4          16
      206.000        0.7500      0.2500       0.0968           5          15
      209.000        0.7000      0.3000       0.1025           6          14
      211.000        0.6500      0.3500       0.1067           7          13
      226.000        0.6000      0.4000       0.1095           8          12
      229.000        0.5500      0.4500       0.1112           9          11
      234.000        0.5000      0.5000       0.1118          10          10
      237.000        0.4500      0.5500       0.1112          11           9
      237.000*            .           .            .          11           8
      242.000        0.3937      0.6063       0.1106          12           7
      249.000        0.3375      0.6625       0.1082          13           6
      253.000        0.2812      0.7188       0.1038          14           5
      257.000        0.2250      0.7750       0.0971          15           4
      268.000*            .           .            .          15           3
      270.000        0.1500      0.8500       0.0891          16           2
      291.000        0.0750      0.9250       0.0693          17           1
      323.000             0      1.0000            0          18           0

           NOTE: The marked survival times are censored observations.
```

Figure 40.3. Survivor Function Estimate for Placebo-Treated Rats

Figure 40.3 and Figure 40.4 display the survival estimates and the summary statistics of the survival times for Placebo (Treatment=0). The median survival time for rats in this treatment is 235 days.

```
                         Quartile Estimates

                          Point     95% Confidence Interval
             Percent    Estimate      [Lower      Upper)

                  75     257.000     237.000     291.000
                  50     235.500     209.000     253.000
                  25     207.500     180.000     234.000


                          Mean    Standard Error

                        235.156           10.211
```

Figure 40.4. Summary Statistics of Survival Times for Placebo-Treated Rats

A summary of the number of censored and event observations is shown in Figure 40.5. The figure lists, for each stratum, the number of event and censored observations, and the percentage of censored observations.

```
            Summary of the Number of Censored and Uncensored Values

                                                                 Percent
     Stratum    Treatment       Total  Failed    Censored    Censored

           1    Drug X             20      18           2       10.00
           2    Placebo            20      18           2       10.00
     ----------------------------------------------------------------
       Total                       40      36           4       10.00
```

Figure 40.5. Number of Event and Censored Observations

Results of the comparison of survival curves between the two treatments are shown in Figure 40.6. The rank tests for homogeneity indicate a significant difference between the treatments (p=0.0175 for the log-rank test and p=0.0249 for the Wilcoxon test). Rats treated with Drug X live significantly longer than those treated with Placebo. The log-rank test, which places more weight on larger survival times, is more significant than the Wilcoxon test, which places more weight on early survival times. As noted earlier, the exponential model is not appropriate for the given survival data; consequently, the result of the likelihood ratio test should be ignored.

```
                     Test of Equality over Strata

                                                Pr >
          Test        Chi-Square      DF    Chi-Square

          Log-Rank        5.6485       1        0.0175
          Wilcoxon        5.0312       1        0.0249
          -2Log(LR)       0.1983       1        0.6561
```

Figure 40.6. Results of the 2-sample Tests

Figure 40.7 displays the graph of the product-limit survivor function estimates versus survival time. The two treatments differ primarily at larger survival times.

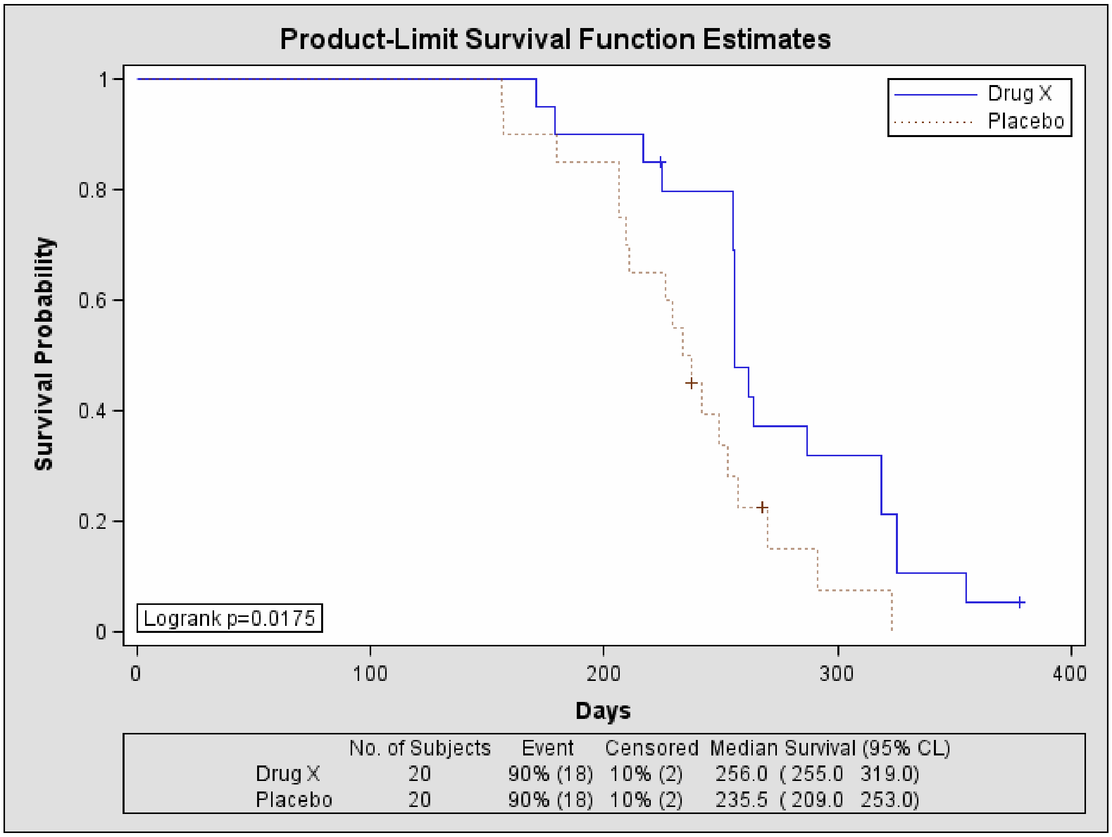


Figure 40.7. Plot of Estimated Survivor Functions (Experimental)

This graphical display is requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the LIFETEST procedure, see the section “ODS Graphics” on page 2190.

Next, suppose male rats and female rats are thought to have different survival rates, and you want to assess the treatment effect while adjusting for the gender differences. By specifying the variable Sex in the STRATA statement as a stratifying variable and by specifying the variable Treatment in the GROUP= option, you can carry out a stratified test to test Treatment while adjusting for Sex. The test statistics are computed by pooling over the strata defined by the values of Sex, thus controlling for the effect of Sex. The NOTABLE option is added to the PROC LIFETEST statement to avoid estimating a survival curve for each gender.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   strata Sex / group=Treatment;
run;
```

Results of the stratified tests are shown in Figure 40.8. The treatment effect is statistically significant for both the log-rank test (*p*=0.0071) and the Wilcoxon test (*p*=0.0150). As compared to the results of the unstratified tests in Figure 40.6, the significance of the treatment effect has been sharpened by controlling for the effect of the gender of the subjects.

```
                          The LIFETEST Procedure

                  Stratified Test of Equality over Group

                                                        Pr >
                Test        Chi-Square       DF     Chi-Square

                Log-Rank        7.2466        1         0.0071
                Wilcoxon        5.9179        1         0.0150
```

Figure 40.8. Results of the Stratified 2-sample Tests

Since Treatment is a binary variable, another way to study the effect of Treatment is to carry out a censored linear rank test with Treatment as an independent variable. Although this test is less popular than the 2-sample test, nevertheless, in situations where the independent variables are continuous and are difficult to discretize, it may be infeasible to perform the k-sample test. To compute the censored linear rank statistics to test the Treatment effect, Treatment is specified in the TEST statement.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   test Treatment;
run;
```

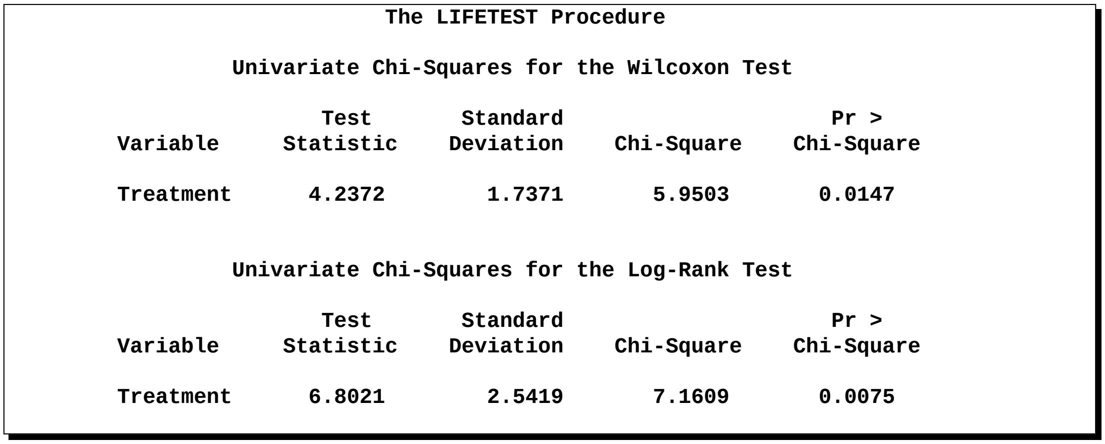
```
                          The LIFETEST Procedure

            Univariate Chi-Squares for the Wilcoxon Test

                        Test        Standard                      Pr >
     Variable       Statistic      Deviation    Chi-Square    Chi-Square

     Treatment         3.9525         1.7524        5.0875        0.0241


            Univariate Chi-Squares for the Log-Rank Test

                        Test        Standard                      Pr >
     Variable       Statistic      Deviation    Chi-Square    Chi-Square

     Treatment         6.2708         2.6793        5.4779        0.0193
```

Figure 40.9. Results of Linear Rank Tests of Treatment

Results of the linear rank tests are shown Figure 40.9. The *p*-values are very similar to those of the 2-sample tests in Figure 40.6.

With Sex as a prognostic factor that you want to control, you can compute a stratified linear rank statistic to test the effect of Treatment by specifying Sex in the STRATA statement and Treatment in the TEST statement. The NOTEST option is specified in the STRATA statement to suppress the k-sample tests for Sex.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   strata Sex / notest;
   test Treatment;
run;
```



```
                              The LIFETEST Procedure

                 Univariate Chi-Squares for the Wilcoxon Test

                         Test       Standard                      Pr >
      Variable      Statistic      Deviation    Chi-Square    Chi-Square

      Treatment        4.2372         1.7371        5.9503        0.0147


                 Univariate Chi-Squares for the Log-Rank Test

                         Test       Standard                      Pr >
      Variable      Statistic      Deviation    Chi-Square    Chi-Square

      Treatment        6.8021         2.5419        7.1609        0.0075
```

Figure 40.10. Result of Stratified Linear Rank Tests of Treatment

Results of the stratified linear rank tests are shown in Figure 40.10. The p-values are very similar to those of the stratified 2-sample tests in Figure 40.8.

Syntax

The following statements are available in PROC LIFETEST:

PROC LIFETEST < *options* > **;**
  TIME *variable* < ***censor(list)*** > **;**
  BY *variables* **;**
  FREQ *variable* **;**
  ID *variables* **;**
  STRATA *variable* < *(list)* > < . . . *variable* < *(list)* > > **;**
  SURVIVAL *options* **;**
  TEST *variables* **;**

The simplest use of PROC LIFETEST is to request the nonparametric estimates of the survivor function for a sample of survival times. In such a case, only the PROC LIFETEST statement and the TIME statement are required. You can use the STRATA statement to divide the data into various strata. A separate survivor function is then estimated for each stratum, and tests of the homogeneity of strata are performed. However, if the GROUP= option is also specify in the STRATA statement, stratified tests are carried out to test the k samples defined by the GROUP= variable while controlling for the effect of the STRATA variables. You can use the SURVIVAL statement to output the estimates of the survivor function into a SAS data set. You can specify covariates in the TEST statement. PROC LIFETEST computes linear rank statistics to test the effects of these covariates on survival.

The PROC LIFETEST statement invokes the procedure. All statements except the TIME statement are optional, and there is no required order for the statements following the PROC LIFETEST statement. The TIME statement is used to specify the variables that define the survival time and censoring indicator. The STRATA statement specifies a variable or set of variables defining the strata for the analysis. The SURVIVAL statement enables you to specify a transformation to be used in the computation of the confidence intervals; it also enables you to output simultaneous confidence intervals. The TEST statement specifies a list of numeric covariates to be tested for their association with the response survival time. Each variable is tested individually, and a joint test statistic is also computed. The ID statement provides a list of variables whose values are used to identify observations in the product-limit estimates of the survival function. When only the TIME statement appears, no strata are defined and no tests of homogeneity are performed.

PROC LIFETEST Statement

PROC LIFETEST < *options* > **;**

The PROC LIFETEST statement invokes the procedure. The following options can appear in the PROC LIFETEST statement and are described in alphabetic order. If no options are requested, PROC LIFETEST computes and displays product-limit estimates of the survival distribution within each stratum and tests the equality of the survival functions across strata.

Table 40.1. Summary of PROC LIFETEST Statement Options

Task	Options	Description
Specify Data Set	DATA=	specifies the input SAS data set
	OUTSURV=	names an output data set to contain survival estimates and confidence limits
	OUTTEST=	names an output data set to contain rank test statistics for association of survival time with covariates limits
Estimate Survival	METHOD=	specifies method to compute survivor function
	ALPHA=	sets confidence level for survival estimates
	INTERVALS=	specifies interval endpoints for life-table estimates
	NINTERVAL=	specifies number of intervals for life-table estimates
	WIDTH=	specifies width of intervals for life-table estimates

Table 40.1. (continued)

Task	Options	Description
Plot Survival	PLOTS=	specifies plots
	MAXTIME=	sets maximum value of time variable for plotting
Traditional High-Resolution Graphics	ANNOTATE=	specifies an annotate data set that adds features to plots
	CENSOREDSYMBOL=	defines symbol used for censored observations in plots
	DESCRIPTION=	specifies string that appears in the description field of the PROC GREPLAY master menu for the plots
	EVENTSYMBOL=	specifies symbol used for event observations in plots
	GOUT=	specifies graphics catalog name for saving graphics output
	LANNOTATE=	specifies an input data set that contains variables for local annotation
Line Printer Plots	LINEPRINTER	specifies that plots are produced by line printer
	FORMCHAR(1,2,7,9)=	defines characters used for line printer plot axes
	NOCENSPLOT	suppresses the plot of censored observations
Control Output	NOPRINT	suppresses display of printed output
	NOTABLE	suppresses display of survival function estimates
	INTERVALS=	displays only the product-limit estimate for the smallest time within each specified interval
	TIMELIST=	specifies a list of time points at which the Kaplan-Meier estimates are displayed
	REDUCEOUT	specifies that only INTERVAL= or TIMELIST= observations are listed in the OUTSURV= data set
Miscellaneous	ALPHAQT=	sets confidence level for survival time quartiles
	MISSING	allows missing values to be a stratum level
	SINGULAR=	sets tolerance for testing singularity of covariance matrix of rank statistics
	TIMELIM=	specifies the time limit used to estimate the mean survival time and its standard error

ALPHA=*value*

specifies a number between 0.0001 and 0.9999 that sets the confidence level for the confidence intervals for the survivor function. The confidence level for the interval is 1 - ALPHA. For example, the option ALPHA=0.05 requests a 95% confidence interval for the SDF at each time point. The default value is 0.05.

ALPHAQT=*value*

specifies a number between 0.0001 and 0.9999 that sets the level for the confidence intervals for the quartiles of the survival time. The confidence level for the interval is 1 - ALPHAQT. For example, the option ALPHAQT=0.05 requests a 95% confidence interval for the quantiles of the survival time. The default value is 0.05.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an input data set that contains appropriate variables for annotation of the traditional high-resolution graphics. The ANNOTATE= option enables you to add features (for example, labels explaining extreme observations) to plots produced on graphics devices. The ANNOTATE= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS statement is specified. The data set specified must be an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*.

The data set specified with the ANNOTATE= option in the PROC LIFETEST statement is “global” in the sense that the information in this data set is displayed on every plot produced by a single invocation of PROC LIFETEST.

CENSOREDSYMBOL=*name* **|** *’string’*
CS=*name* **|** *’string’*

specifies the symbol value for the censored observations in the traditional high-resolution graphics. The value, *name* or *’string’*, is the symbol value specification allowed in SAS/GRAPH software. The default is CS=CIRCLE. If you want to omit plotting the censored observations, specify CS=NONE. The CENSOREDSYMBOL= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS statement is specified.

DATA=*SAS-data-set*

names the SAS data set used by PROC LIFETEST. By default, the most recently created SAS data set is used.

DESCRIPTION=’*string* **’**
DES=’*string* **’**

specifies a descriptive string of up to 40 characters that appears in the “Description” field of the traditional high-resolution graphics catalog. The description does not appear on the plots. By default, PROC LIFETEST assigns a description of the form PLOT OF *vname* vs *hname*, where *vname* and *hname* are the names of the *y* variable and the *x* variable, respectively. The DESCRIPTION= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS is specified.

EVENTSYMBOL=*name* **|** *'string'*
ES=*name* **|** *'string'*

specifies the symbol value for the event observations in the traditional high-resolution graphics. The value, *name* or *'string'*, is the symbol value specification allowed in SAS/GRAPH software. The default is ES=NONE. The EVENTSYMBOL= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS statement is specified.

FORMCHAR(1,2,7,9)=*'string'*

defines the characters used for constructing the vertical and horizontal axes of the line printer plots. The string should be four characters. The first and second characters define the vertical and horizontal bars, respectively, which are also used in drawing the *steps* of the product-limit survival function. The third character defines the tick mark for the axes, and the fourth character defines the lower left corner of the plot. If the FORMCHAR option in PROC LIFETEST is not specified, the value supplied, if any, with the system option FORMCHAR= is used. The default is FORMCHAR(1,2,7,9)='|-+-'. Any character or hexadecimal string can be used to customize the plot appearance. To send the plot output to a printer with the IBM graphics character set (1 or 2) or display it directly on your PC screen, you can use the following hexadecimal representation

```
formchar(1,2,7,9)='B3C4C5C0'x
```

or system option

```
formchar='B3C4DAC2BFC3C5B4C0C1D9'x
```

Refer to the chapter titled "The PLOT Procedure," in the *SAS Procedures Guide* or the section "System Options" in *SAS Language Reference: Dictionary* for further information.

GOUT=*graphics-catalog*

specifies the graphics catalog for saving traditional high-resolution graphics output from PROC LIFETEST. The default is WORK.GSEG. The GOUT= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS statement is specified. For more information, refer to the chapter titled "The GREPLAY Procedure" in *SAS/GRAPH Software: Reference*.

INTERVALS=*values*

specifies a list of interval endpoints for the life-table method. These endpoints must all be nonnegative numbers. The initial interval is assumed to start at zero whether or not zero is specified in the list. Each interval contains its lower endpoint but does not contain its upper endpoint. When this option is used with the product-limit method, it reduces the number of survival estimates displayed by displaying only the estimates for the smallest time within each specified interval. The INTERVALS= option can be specified in any of the following ways:

list separated by blanks	`intervals=1 3 5 7`
list separated by commas	`intervals=1,3,5,7`
x to y	`intervals=1 to 7`
x to y by z	`intervals=1 to 7 by 1`
combination of the above	`intervals=1,3 to 5,7`

For example, the specification

```
intervals=5,10 to 30 by 10
```

produces the set of intervals

$$\{[0,5),[5,10),[10,20),[20,30),[30,\infty)\}$$

LANNOTATE=*SAS-data-set*
LANN=*SAS-data-set*

specifies an input data set that contains variables for local annotation of traditional high-resolution graphics. You can use the LANNOTATE= option to specify a different annotation for each BY group, in which case the BY variables must be included in the LANNOTATE= data set. The LANNOTATE= option cannot be used if the LINEPRINTER option or the experimental ODS GRAPHICS statement is specified. The data set specified must be an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*.

If there is no BY-group processing, the ANNOTATE= and LANNOTATE= options have the same effects.

LINEPRINTER
LS

specifies that plots are produced by a line printer instead of by a graphical device. This option cannot be used if the experimental ODS GRAPHICS statement is specified.

MAXTIME=*value*

specifies the maximum value of the time variable allowed on the plots so that outlying points do not determine the scale of the time axis of the plots. This parameter only affects the displayed plots and has no effect on the calculations.

METHOD=*type*

specifies the method used to compute the survival function estimates. Valid values for *type* are as follows.

PL | KM	specifies that product-limit (PL) or Kaplan-Meier (KM) estimates are computed.
ACT | LIFE | LT	specifies that life-table (or actuarial) estimates are computed.

By default, METHOD=PL.

MISSING

allows missing values for numeric variables and blank values for character variables as valid stratum levels. See the section "Missing Values" on page 2171 for details.

By default, PROC LIFETEST does not use observations with missing values for any stratum variables.

NINTERVAL=*value*

specifies the number of intervals used to compute the life-table estimates of the survivor function. This parameter is overridden by the WIDTH= option or the INTERVALS= option. When you specify the NINTERVAL= option, PROC LIFETEST tries to find an interval that results in round numbers for the endpoints. Consequently, the number of intervals may be different from the number requested. Use the INTERVALS= option to control the interval endpoints. The default is NINTERVAL=10.

NOCENSPLOT
NOCENS

requests that the plot of censored observations be suppressed when the PLOTS= option is specified. This option is not needed when the life-table method is used to compute the survival estimates, since the plot of censored observations is not produced.

NOPRINT

suppresses the display of output. This option is useful when only an output data set is needed. Note that this option temporarily disables the Output Delivery System (ODS).

For more information, see Chapter 14, "Using the Output Delivery System."

NOTABLE

suppresses the display of survival function estimates. Only the number of censored and event times, plots, and test results are displayed.

OUTSURV=*SAS-data-set*
OUTS=*SAS-data-set*

creates an output SAS data set to contain the estimates of the survival function and corresponding confidence limits for all strata. See the section "Output Data Sets" on page 2183 for more information on the contents of the OUTSURV= SAS data set.

OUTTEST=*SAS-data-set*
OUTT=*SAS-data-set*

creates an output SAS data set to contain the overall chi-square test statistic for association with failure time for the variables in the TEST statement, the values of the univariate rank test statistics for each variable in the TEST statement, and the estimated covariance matrix of the univariate rank test statistics. See the section "Output Data Sets" on page 2183 for more information on the contents of the OUTTEST= SAS data set.

PLOTS= *(type <(NAME=name)> <, ..., type <(NAME=name)> >)*

creates plots of survival estimates or censored observations, where *type* is the type of plot and *name* is a catalog entry name of up to eight characters. Valid values of *type* are as follows:

CENSORED | C specifies a plot of censored observations by strata (product-limit method only).

SURVIVAL | S specifies a plot of the estimated SDF versus time.

LOGSURV | LS specifies a plot of the $-\log(\text{estimated SDF})$ versus time.

LOGLOGS | LLS specifies a plot of the $\log(-\log(\text{estimated SDF}))$ versus $\log(\text{time})$.

HAZARD | H specifies a plot of the estimated hazard function versus time (life-table method only).

PDF | P specifies a plot of the estimated probability density function versus time (life-table method only).

Parentheses are required in specifying the plots. For example,

```
plots = (s)
```

requests a plot of the estimated survivor function versus time, and

```
plots = (s(name=Surv2), h(name=Haz2))
```

requests a plot of the estimated survivor function versus time and a plot of the estimated hazard function versus time, with Surv2 and Haz2 as their catalog names, respectively.

REDUCEOUT

specifies that the OUTSURV= data set contains only those observations that are included in the INTERVALS= or TIMELIST= option. This option has no effect if the OUTSURV= option is not specified. It also has no effect if neither the INTERVALS= option nor the TIMELIST= option is specified.

SINGULAR=*value*

specifies the tolerance for testing singularity of the covariance matrix for the rank test statistics. The test requires that a pivot for sweeping a covariance matrix be at least this number times a norm of the matrix. The default value is 1E-12.

TIMELIM=*time-limit*

specifies the time limit used in the estimation of the mean survival time and its standard error. The mean survival time can be shown to be the area under the Kaplan-Meier survival curve. However, if the largest observed time in the data is censored, the area under the survival curve is not a closed area. In such a situation, you can choose a time limit L and estimate the mean survival curve limited to a time L (Lee 1992, pp. 72–76). This option is ignored if the largest observed time is an event time. Valid *time-limit* values are as follows:

EVENT | LET specifies that the time limit L is the largest event time in the data. TIMELIM=EVENT is the default.

OBSERVED | LOT specifies that the time limit L is the largest observed time in the data.

number specifies that the time limit L is the given *number*. The *number* must be positive and at least as large as the largest event time in the data.

TIMELIST=*number-list*

specifies a list of time points at which the Kaplan-Meier estimates are displayed. The time points are listed in the column labeled as Timelist. Since the Kaplan-Meier survival curve is a decreasing step function, each given time point falls in an interval that has a constant survival estimate. The event time that corresponds to the beginning of the time interval is displayed along with its survival estimate.

WIDTH=*value*

sets the width of the intervals used in the life-table calculation of the survival function. This parameter is overridden by the INTERVALS= option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LIFETEST to obtain separate analyses on observations in groups defined by the BY variables.

The BY statement is more efficient than the STRATA statement for defining strata in large data sets. However, if you use the BY statement to define strata, PROC LIFETEST does not pool over strata for testing the association of survival time with covariates nor does it test for homogeneity across the BY groups.

Interval size is computed separately for each BY group. When intervals are determined by default, they may be different for each BY group. To make intervals the same for each BY group, use the INTERVALS= option in the PROC LIFETEST statement.

When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LIFETEST procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable containing the frequency of occurrence of each observation. PROC LIFETEST treats each observation as if it appeared n times, where n is the value of the FREQ variable for the observation. The FREQ statement is useful for producing life tables when the data are already in the form of a summary data set. If not an integer, the frequency value is truncated to an integer. If the frequency value is less than one, the observation is not used.

ID Statement

ID *variables* **;**

The ID variable values are used to label the observations of the product-limit survival function estimates. SAS format statements can be used to format the values of the ID variables.

STRATA Statement

STRATA *variable* < *(list)* > < . . . *variable* < *(list)* > > < */options* >**;**

The STRATA statement indicates which variables determine strata levels for the computations. The strata are formed according to the nonmissing values of the designated strata variables. The MISSING option can be used to allow missing values as a valid stratum level. Other options enable you to specify various k-sample tests, trend tests and stratified tests.

In the preceding syntax, *variable* is a variable whose values determine the stratum levels and *list* is a list of endpoints for a numeric variable. The values for *variable* can be formatted or unformatted. If the variable is a character variable, or if the variable is numeric and no list appears, then the strata are defined by the unique values of the strata variable. More than one variable can be specified in the STRATA statement, and each numeric variable can be followed by a list. Each interval contains its lower endpoint but does not contain its upper endpoint. The corresponding strata are formed by the combination of levels. If a variable is numeric and is followed by a list, then the levels for that variable correspond to the intervals defined by the list. The initial interval is assumed to start at $-\infty$ and the final interval is assumed to end at ∞.

The specification of STRATA variables can have any of the following forms:

list separated by blanks	**strata age(5 10 20 30)**
list separated by commas	**strata age(5,10,20,30)**
x to y	**strata age(5 to 10)**
x to y by z	**strata age(5 to 30 by 10)**
combination of the above	**strata age(5,10 to 50 by 10)**

For example, the specification

```
strata age(5,20 to 50 by 10) sex;
```

indicates the following levels for the Age variable

$$\{(-\infty,5),[5,20),[20,30),[30,40),[40,50),[50,\infty)\}$$

This statement also specifies that the age strata is further subdivided by values of the variable Sex. In this example, there are 6 age groups by 2 sex groups, forming a total of 12 strata.

The specification of several variables (for example, A B C) is equivalent to the A*B*C... syntax of the TABLES statement in the FREQ procedure. The number of strata levels usually grows very rapidly with the number of STRATA variables, so you must be cautious when specifying the list of STRATA variables.

The following options can appear in the STRATA statement after a slash ("/"). Other than the MISSING option, these options are dedicated to the tests of the two or more samples of survival data.

GROUP=*variable*
specifies the variable whose formatted values identify the various samples whose underlying survival curves are to be compared. The tests are stratified on the levels of the STRATA variables. For instance, in a multicenter trial in which two forms of therapy are to be compared, you specify the variable identifying therapies as the GROUP= variable and the variable identifying centers as the STRATA variable, in order to perform a stratified 2-sample test to compare the therapies while controlling the effect of the centers.

MISSING
allows missing values to be a stratum level or a valid value of the GROUP= variable.

NODETAIL
suppresses the display of the rank statistics and the corresponding covariance matrices for various strata. If the TREND option is specified, the display of the scores for computing the trend tests is suppressed.

NOTEST
suppresses the k-sample tests, stratified tests, and trend tests

TREND

computes the trend tests for testing the null hypothesis that the k population hazards rate are the same versus an ordered alternatives. If there is only one STRATA variable and the variable is numeric, the unformatted values of the variable are used as the scores; otherwise, the scores are $1, 2, \ldots,$ in the given order of the strata.

TEST=(*list*)

enables you to select the weight functions for the k-sample tests, stratified tests, or trend tests. You can specify a *list* containing one or more of the following keywords.

LOGRANK specifies the log-rank test

WILCOXON specifies the Wilcoxon test. The test is also referred to as the Gehan test or the Breslow test.

TARONE specifies the Tarone-Ware test

PETO specifies the Peto-Peto test. The test is also referred to as the Peto-Peto-Prentice test.

MODPETO specifies the modified Peto-Peto test

FLEMING(ρ1, ρ2) specifies the family of tests in Harrington and Fleming (1982), where ρ1 and ρ2 are nonegative numbers. FLEMING(ρ1,ρ2) reduces to the Fleming-Harrington G^ρ family (Fleming and Harrington 1981) when ρ2=0, which you can specify FLEMING(ρ) with one argument. When ρ=0, the test becomes the log-rank test. When ρ=1, the test should be very close to the Peto-Peto test.

LR specifies the likelihood ratio test based on the exponential model.

ALL specifies all the nonparametric tests with ρ1=1 and ρ2=0 for the FLEMING(. , .) test.

By default, TEST=(LOGRANK WILCOXON LR) for the k-sample tests, and TEST=(LOGRANK WILCOXON) for stratified and trend tests.

SURVIVAL Statement

SURVIVAL *options* **;**

The SURVIVAL statement creates an output SAS data set containing the results of the estimation of the survivor function. Although you can use the OUTSURV= option in the PROC LIFETEST statement to produce the output data set, the SURVIVAL statement enables you to output confidence bands and to specify a transformation of survival time in the computation of the pointwise confidence intervals and the confidence bands. Options in the PROC LIFETEST statements (ALPHA=, INTERVALS=, REDUCEOUT, and TIMELIST=) that applies to the OUTSURV= data can also be specified in the SURVIVAL statements. You can plot these survival estimates using the experimental ODS graphics (see the section “ODS Graphics” on page 2190).

Table 40.2. Summary of SURVIVAL Statement Options

Task	**Options**	**Description**
Specify Data Set	OUT=	specifies the output SAS data set
Specify Transformation	CONFTYPE=	specifies the transformation for the computation of pointwise and simultaneous confidence intervals for the survivor function
Specify Confidence Bands	CONFBAND=	specifies the confidence bands to be output
	BANDMAX=	specifies the maximum time for the confidence bands
	BANDMIN=	specifies the minimum time for the confidence bands
Standard Errors	STDERR	outputs the standard errors

BANDMAXTIME=*value*
BANDMAX=*value*

specifies the maximum time for the confidence bands. The default is the largest observed event time. If the specified BANDMAX= time exceeds the largest observed event time, it is truncated to the largest observed event time.

BANDMINTIME=*value*
BANDMIN=*value*

specifies the minimum time for the confidence bands. The default is the smallest observed event time. For the equal precision band, if the BANDMIN= value is less than the smallest observed event time, it is defaulted to the smallest observed event time.

CONFBAND=*keyword*

specifies the confidence bands to output. Confidence bands are available only for the product-limit method. You can use the following *keywords*:

ALL	outputs both the Hall-Wellner and the equal precision confidence bands.
EP	outputs the equal precision confidence band.
HW	outputs the Hall and Wellner confidence band.

CONFTYPE=*keyword*

specifies the transformation applied to $S(t)$ to obtain the pointwise confidence intervals as well as the confidence bands. The following *keywords* can be used and the default is CONFTYPE=LOGLOG.

ASINSQRT the arcsine-square root transformation

$$g(x) = \sin^{-1}(\sqrt{x})$$

LOGLOG the log-log transformation

$$g(x) = \log(-\log(x))$$

This is also referred to as the log cumulative hazard transformation since it is applying the logarithmic function $\log(.)$ to the cumulative hazard function. Collett (1994) and Lachin (2000) refer it as the complementary log-log transformation.

LINEAR the identity transformation

$$g(x) = x$$

LOG the logarithmic transformation

$$g(x) = \log(x)$$

LOGIT the logit transformation

$$g(x) = \log\left(\frac{x}{1-x}\right)$$

OUT=*SAS-Data-Set*

names the SAS data set that contains the survival estimates. If the OUT= option is omitted, PROC LIFETEST creates an output SAS data set with the default name DATAn. If you do not want to create this output SAS data set, set OUT=_NULL_.

STDERR

specifies that the standard error of the survivor function (SDF_STDERR) be output. If the life-table method is used, the standard error of the density function (PDF_STDERR) and the standard error of the hazard function (HAZ_STDERR) are also output.

TEST Statement

TEST *variables* **;**

The TEST statement specifies a list of numeric (continuous) covariates that you want tested for association with the failure time.

Two sets of rank statistics are computed. These rank statistics and their variances are pooled over all strata. Univariate (marginal) test statistics are displayed for each of the covariates.

Additionally, a sequence of test statistics for joint effects of covariates is displayed. The first element of the sequence is the largest univariate test statistic. Other variables are then added on the basis of the largest increase in the joint test statistic. The process continues until all the variables have been added or until the remaining variables are linearly dependent on the previously added variables.

See the section "Rank Tests for the Association of Survival Time with Covariates" on page 2180 for more information.

TIME Statement

TIME *variable* < ***censor(list)*** > **;**

The TIME statement is required. It is used to indicate the failure time variable, where *variable* is the name of the failure time variable that can be optionally followed by an asterisk, the name of the censoring variable, and a parenthetical list of values that correspond to right censoring. The censoring values should be numeric, nonmissing values. For example, the statement

```
time T*Flag(1,2);
```

identifies the variable T as containing the values of the event or censored time. If the variable Flag has value 1 or 2, the corresponding value of T is a right-censored value.

Details

Missing Values

Observations with a missing value for either the failure time or the censoring variable are not used in the analysis. If a stratum variable value is missing, survival function estimates are computed for the strata labeled by the missing value, but these data are not used in any rank tests. However, the MISSING option can be used to request that missing values be treated as valid stratum values. If any variable specified in the TEST statement has a missing value, that observation is not used in the calculation of the rank statistics.

Computational Formulas

Product-Limit Method

Let $t_1 < t_2 < \cdots < t_k$ represent the distinct event times. For each $i = 1, \ldots, k$, let n_i be the number of surviving units, the size of the risk set, just prior to t_i. Let d_i be the number of units that fail at t_i, and let $s_i = n_i - d_i$.

The product-limit estimate of the SDF at t_i is the cumulative product

$$\hat{S}(t_i) = \prod_{j=1}^{i} \left(1 - \frac{d_j}{n_j}\right)$$

Notice that the estimator is defined to be right continuous; that is, the events at t_i are included in the estimate of $S(t_i)$. The corresponding estimate of the standard error is computed using Greenwood's formula (Kalbfleisch and Prentice 1980) as

$$\hat{\sigma}\left(\hat{S}(t_i)\right) = \hat{S}(t_i)\sqrt{\sum_{j=1}^{i} \frac{d_j}{n_j s_j}}$$

The first sample quartile of the survival time distribution is given by

$$q_{0.25} = \frac{1}{2}(\inf\left\{t : 1 - \hat{S}(t) \geq 0.25\right\} + \sup\left\{t : 1 - \hat{S}(t) \leq 0.25\right\})$$

Confidence intervals for the quartiles are based on the sign test (Brookmeyer and Crowley 1982). The $100(1-\alpha)\%$ confidence interval for the first quartile is given by

$$I_{0.25} = \left\{t : (1 - \hat{S}(t) - 0.25)^2 \leq c_\alpha \hat{\sigma}^2 \left(\hat{S}(t)\right)\right\}$$

where c_α is the upper α percentile of a central chi-squared distribution with 1 degree of freedom. The second and third sample quartiles and the corresponding confidence intervals are calculated by replacing the 0.25 in the last two equations by 0.50 and 0.75, respectively.

The estimated mean survival time is

$$\hat{\mu} = \sum_{i=1}^{k} \hat{S}(t_{i-1})(t_i - t_{i-1})$$

where t_0 is defined to be zero. When the largest observed time is censored, this sum underestimates the mean. The standard error of $\hat{\mu}$ is estimated as

$$\hat{\sigma}(\hat{\mu}) = \sqrt{\frac{m}{m-1} \sum_{i=1}^{k-1} \frac{A_i^2}{n_i s_i}}$$

where

$$\begin{aligned} A_i &= \sum_{j=i}^{k-1} \hat{S}(t_j)(t_{j+1} - t_j) \\ m &= \sum_{j=1}^{k} d_j \end{aligned}$$

If the largest observed time is not an event, you can use the TIMELIM= option to specify a time limit L and estimate the mean survival time limited to the time L and its standard error by replacing k by $k+1$ with $t_{k+1} = L$.

Life-Table Method

The life-table estimates are computed by counting the numbers of censored and uncensored observations that fall into each of the time intervals $[t_{i-1}, t_i)$, $i = 1, 2, \ldots, k+1$, where $t_0 = 0$ and $t_{k+1} = \infty$. Let n_i be the number of units entering the interval $[t_{i-1}, t_i)$, and let d_i be the number of events occurring in the interval. Let $b_i = t_i - t_{i-1}$, and let $n'_i = n_i - w_i/2$, where w_i is the number of units censored in the interval. The *effective sample size* of the interval $[t_{i-1}, t_i)$ is denoted by n'_i. Let t_{mi} denote the midpoint of $[t_{i-1}, t_i)$.

The conditional probability of an event in $[t_{i-1}, t_i)$ is estimated by

$$\hat{q}_i = \frac{d_i}{n'_i}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{q}_i\right) = \sqrt{\frac{\hat{q}_i \hat{p}_i}{n'_i}}$$

where $\hat{p}_i = 1 - \hat{q}_i$.

The estimate of the survival function at t_i is

$$\hat{S}(t_i) = \begin{cases} 1 & i = 0 \\ \hat{S}(t_{i-1}) p_{i-1} & i > 0 \end{cases}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{S}(t_i)\right) = \hat{S}(t_i) \sqrt{\sum_{j=1}^{i-1} \frac{\hat{q}_j}{n'_j \hat{p}_j}}$$

The density function at t_{mi} is estimated by

$$\hat{f}(t_{mi}) = \frac{\hat{S}(t_i) \hat{q}_i}{b_i}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{f}(t_{mi})\right) = \hat{f}(t_{mi}) \sqrt{\sum_{j=1}^{i-1} \frac{\hat{q}_j}{n'_j \hat{p}_j} + \frac{\hat{p}_i}{n'_i \hat{q}_i}}$$

The estimated hazard function at t_{mi} is

$$\hat{h}(t_{mi}) = \frac{2\hat{q}_i}{b_i(1 + \hat{p}_i)}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{h}(t_{mi})\right) = \hat{h}(t_{mi}) \sqrt{\frac{1 - (b_i \hat{h}(t_{mi})/2)^2}{n'_i \hat{q}_i}}$$

Let $[t_{j-1}, t_j)$ be the interval in which $\hat{S}(t_{j-1}) \geq \hat{S}(t_i)/2 > \hat{S}(t_j)$. The median residual lifetime at t_i is estimated by

$$\hat{M}_i = t_{j-1} - t_i + b_j \frac{\hat{S}(t_{j-1}) - \hat{S}(t_i)/2}{\hat{S}(t_{j-1}) - \hat{S}(t_j)}$$

and the corresponding standard error is estimated by

$$\hat{\sigma}(\hat{M}_i) = \frac{\hat{S}(t_i)}{2\hat{f}(t_{mj})\sqrt{n'_i}}$$

Interval Determination

If you want to determine the intervals exactly, use the INTERVALS= option in the PROC LIFETEST statement to specify the interval endpoints. Use the WIDTH= option to specify the width of the intervals, thus indirectly determining the number of intervals. If neither the INTERVALS= option nor the WIDTH= option is specified in the life-table estimation, the number of intervals is determined by the NINTERVAL= option. The width of the time intervals is 2, 5, or 10 times an integer (possibly a negative integer) power of 10. Let $c = \log_{10}$(maximum observed time/number of intervals), and let b be the largest integer not exceeding c. Let $d = 10^{c-b}$ and let

$$a = 2 \times I(d \leq 2) + 5 \times I(2 < d \leq 5) + 10 \times I(d > 5)$$

with I being the indicator function. The width is then given by

$$\text{width} = a \times 10^b$$

By default, NINTERVAL=10.

Pointwise Confidence Limits Added to the Output Data Set

Pointwise confidence limits are computed for the survivor function, and for the density function and hazard function when the life-table method is used. Let α be specified by the ALPHA= option. Let $z_{\alpha/2}$ be the critical value for the standard normal distribution. That is, $\Phi(-z_{\alpha/2}) = \alpha/2$, where Φ is the cumulative distribution function of the standard normal random variable.

Survival Function

When the computation of confidence limits for the survivor function $S(t)$ is based on the asymptotic normality of the survival estimator $\hat{S}(t)$, the approximate confidence interval may include impossible values outside the range [0,1] at extreme values of t. This problem can be avoided by applying the asymptotic normality to a transformation of $S(t)$ for which the range is unrestricted. In addition, certain transformed confidence intervals for $S(t)$ perform better than the usual linear confidence intervals (Borgan and Liestøl 1990). The CONFTYPE= option enables you to pick one of the following transformations: the log-log function (Kalbfleisch and Prentice 1980), the arcsine-square root function (Nair 1984), the logit function (Meeker and Escobar 1998), the log function, and the linear function.

Let g be the tranformation that is being applied to the survivor function $S(t)$. By the delta method, the standard error of $g(\hat{S}(t))$ is estimated by

$$\tau(t) = \hat{\sigma}(g(\hat{S}(t)) = g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))$$

where g' is the first derivative of the function g. The 100(1-α)% confidence interval for $S(t)$ is given by

$$g^{-1}\left(g(\hat{S}(t)) \pm z_{\frac{\alpha}{2}} g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))\right)$$

where g^{-1} is the inverse function of g.

Arcsine-Square Root Transformation

The estimated variance of $\sin^{-1}(\sqrt{\hat{S}(t)})$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2[\hat{S}(t)]}{4\hat{S}(t)[1-\hat{S}(t)]}$. The 100(1-$\alpha$)% confidence interval for $S(t)$ is given by

$$\sin^2\{\max[0, \sin^{-1}(\sqrt{\hat{S}(t)}) - z_{\frac{\alpha}{2}}\hat{\tau}(t)]\} \leq S(t) \leq \sin^2\{\min[\frac{\pi}{2}, \sin^{-1}(\sqrt{\hat{S}(t)}) + z_{\frac{\alpha}{2}}\hat{\tau}(t)]\}$$

Linear Transformation

This is the same as having no transformation in which g is the identity. The 100(1-α)% confidence interval for $S(t)$ is given by

$$\hat{S}(t) - z_{\frac{\alpha}{2}}\hat{\sigma}(S(t))) \leq S(t) \leq \hat{S}(t) + z_{\frac{\alpha}{2}}\hat{\sigma}(S(t)))$$

Log Transformation

The estimated variance of $\log(\hat{S}(t))$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2(\hat{S}(t))}{\hat{S}^2(t)}$. The 100(1-$\alpha$)% confidence interval for $S(t)$ is given by

$$\hat{S}(t)\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t)) \leq S(t) \leq \hat{S}(t)\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t))$$

Log-log Transformation

The estimated variance of $\log(-\log(\hat{S}(t))$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2[\hat{S}(t)]}{[\hat{S}(t)\log(\hat{S}(t))]^2}$. The 100(1-$\alpha$)% confidence interval for $S(t)$ is given by

$$[\hat{S}(t)]^{\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t))} \leq S(t) \leq [\hat{S}(t)]^{\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t))}$$

Logit Transformation

The estimated variance of $\log(\frac{\hat{S}(t)}{1-\hat{S}(t)})$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2(\hat{S}(t))}{\hat{S}^2(t)[1-\hat{S}(t)]^2}$. The 100(1-$\alpha$)% confidence limits for $S(t)$ are given by

$$\frac{\hat{S}(t)}{\hat{S}(t) + (1-\hat{S}(t))\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t))} \le S(t) \le \frac{\hat{S}(t)}{\hat{S}(t) + (1-\hat{S}(t))\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t))}$$

Density and Hazard Functions

For the life-table method, a 100(1-α)% confidence interval for hazard function or density function at time t is computed as

$$\hat{g}(t) \pm z_{\alpha/2}\hat{\sigma}[\hat{g}(t)]$$

where $\hat{g}(t)$ is the estimate of either the hazard function or the density function at time t, and $\hat{\sigma}[\hat{g}(t)]$ is the corresponding standard error estimate.

Simultaneous Confidence Intervals for Kaplan-Meier Curve

The pointwise confidence interval for the survivor function $S(t)$ is valid for a single fixed time at which the inference is to be made. In some applications, it is of interest to find the upper and lower confidence bands that guarantee, with a given confidence level, that the survivor function falls within the band for all t in some interval. Hall and Wellner (1980) and Nair (1984) provide two different approaches for deriving the confidence bands. An excellent review can be found in Klein and Moeschberger (1997). You can use CONFBAND= option in the SURVIVAL statement to select the confidence bands. The EP confidence band provides confidence bounds that are proportional to the pointwise confidence interval, while those of the HW band are not proportional to the pointwise confidence bounds. The maximum time, t_U, for the bands can be specified by the BANDMAX= option; the minimum time, t_L, can be specified by the BANDMIN= option. Transformations that are used to improve the pointwise confidence intervals can be applied to improve the confidence bands. It may turn out that the upper and lower bounds of the confidence bands are not decreasing in $t_L < t < t_U$, which is contrary to the nonincreasing characteristic of survivor function. Meeker and Escobar (1998) suggest making an adjustment so that the bounds do not increase: if the upper bound is increasing on the right, it is made flat from the minimum to t_U; if the lower bound is increasing from the right, it is made flat from t_L to the maximum. PROC LIFETEST does not make any adjustment for the nondecreasing behavior of the confidence bands in the OUT= data set. However, the adjustment was made in the display of the confidence bands using ODS graphics.

For Kaplan-Meier estimation, let $t_1 < t_2 < \ldots < t_D$ be the D distinct events times, and that at time t_i, there are d_i events. Let Y_i be the number of individuals who are at risk at time t_i. The variance of $\hat{S}(t)$, given by the Greenwood formula, is $\hat{\sigma}^2[\hat{S}(t)] = \sigma_S^2(t)\hat{S}^2(t)$ where

$$\sigma_S^2(t) = \sum_{t_i \le t} \frac{d_i}{Y_i(Y_i - d_i)}$$

Let $t_L < t_U$ be the time range for the confidence band so that t_U is less than or equal to the largest event time. For the Hall-Wellner band, t_L can be zero, but for the equal precison band, t_L is greater than or equal to the smallest event time. Let

$$a_L = \frac{n\sigma_S^2(t_L)}{1 + n\sigma_S^2(t_L)} \quad \text{and} \quad a_U = \frac{n\sigma_S^2(t_U)}{1 + n\sigma_S^2(t_U)}$$

Let $\{W^0(u), 0 \leq u \leq 1\}$ be a Brownian bridge.

Hall-Wellner Band

The 100(1-α)% HW band of Hall and Wellner (1980) is

$$\begin{aligned}
&\hat{S}(t) - h_\alpha(a_L, a_U)n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t) \\
&\leq S(t) \leq \\
&\hat{S}(t) + h_\alpha(a_L, a_U)n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t)
\end{aligned}$$

for all $t_L \leq t \leq t_U$, where $h_\alpha(a_L, a_U)$ is given by

$$\alpha = \Pr\{\sup_{a_L \leq u \leq a_U} |W^0(u)| > h_\alpha(a_L, a_U)\}$$

The critical values are computed from the results in Chung (1986).

Note that the given confidence band has a formula similar to that of the (linear) pointwise confidence interval where $h_\alpha(a_L, a_U)$ and $n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t)$ in the former correspond to $z_{\frac{\alpha}{2}}$ and $\hat{\sigma}(\hat{S}(t))$ in the latter, respectively. You can obtain the other transformations (arcsine-square root, log-log, log, and logit) for the confidence bands by replacing $z_{\frac{\alpha}{2}}$ and $\hat{\tau}(t)$ in the corresponding pointwise confidence interval formula by $h_\alpha(a_L, a_U)$ and the following $\hat{\tau}(t)$, respectively.

Arcsine-Square Root Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{2}\sqrt{\frac{S(t)}{n[1 - S(t)]}}$$

Log Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}}$$

Log-log Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}|\log[\hat{S}(t)]|}$$

Logit Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}[1 - \hat{S}(t)]}$$

Equal Precision Band

The 100(1-α)% EP band of Nair (1984) is

$$\hat{S}(t) - e_\alpha(a_L, a_U)\hat{S}(t)\sigma_S(t) \leq S(t) \leq \hat{S}(t) + e_\alpha(a_L, a_U)\hat{S}(t)\sigma_S(t)$$

for all $t_L \leq t \leq t_U$, where $e_\alpha(a_L, a_U)$ is given by

$$\alpha = \Pr\{ \sup_{a_L \leq u \leq a_U} \frac{|W^0(u)|}{[u(1-u)]^{\frac{1}{2}}} > e_\alpha(a_L, a_U)\}$$

PROC LIFETEST uses the approximation of Miller and Siegmund (1982, Equation 8) to approximate the tail probability in which $e_\alpha(a_L, a_U)$ is obtained by solving x in

$$\frac{4x\phi(x)}{x} + \phi(x)\left(x - \frac{1}{x}\right)\log\left[\frac{a_U(1-a_L)}{a_L(1-a_U)}\right] = \alpha$$

where $\phi()$ is the standard normal density. Note that the given confidence bounds are proportional to the pointwise confidence intervals. As a matter of fact, this confidence band and the (linear) pointwise confidence interval have the same formula except for the critical values ($z_{\frac{\alpha}{2}}$ for the pointwise confidence interval and $e_\alpha(a_L, a_U)$ for the band). You can obtain the other transformations (arcsine-square root, log-log, log, and logit) for the confidence bands by replacing $z_{\frac{\alpha}{2}}$ by $e_\alpha(a_L, a_U)$ in the formulae of the pointwise confidence intervals.

Comparison of Two or More Groups of Survival Data

Let k be the number of groups. Let $S_i(t)$ be the underlying surivor function ith group, $i = 1, \ldots, k$. The null and alternative hypotheses to be tested are

$$H_0 : S_1(t) = S_2(t) = \ldots = S_k(t) \text{ for all } t \leq \tau$$

versus

$$H_1 : \text{at least one of the } S_i(t)\text{'s is different for some } t \leq \tau$$

respectively, where τ is the largest observed time. Let $t_1 < t_2 < \ldots < t_D$ be the distinct event times in the pooled sample. At time t_i, let $W(t_i)$ be a positive weight function, and let n_{ij} and d_{ij} be the size of the risk set and the number of events in the jth sample, respectively. Let $n_i = \sum_{j=1}^k n_{ij}$, $d_i = \sum_{j=1}^k d_{ij}$ and $s_i = n_i - d_i$.

Nonparametric tests

The rank statistics (Klein and Moeschberger 1997, Section 7.3) for testing H_0 versus H_1 have the form of a k-vector $\mathbf{v} = (v_1, v_2, \ldots, v_k)'$ with

$$v_j = \sum_{i=1}^{D} W(t_i)\left\{d_{ij} - \frac{n_{ij}d_i}{n_i}\right\}$$

and the estimated covariance matrix, $\mathbf{V} = (V_{jl})$, is given by

$$V_{jl} = \sum_{i=1}^{D} W^2(t_i) \left\{ \frac{d_i s_i (n_i n_{il} \delta_{jl} - n_{ij} n_{il})}{n_i^2 (n_i - 1)} \right\}$$

where δ_{jl} is 1 if $j = l$ and 0 otherwise. The term v_j can be interpreted as a weighted sum of observed minus expected numbers of failure under the null hypothesis of identical survival curves. The overall test statistic for homogeneity is $\mathbf{v}'\mathbf{V}^-\mathbf{v}$, where $\mathbf{V}^-$ denotes a generalized inverse of $\mathbf{V}$. This statistic is treated as having a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{V}$ for the purposes of computing an approximate probability level. The choices of the weight function $W(t_i)$ are given in the following table:

Test	$W(t_i)$
log-rank	1.0
Wilcoxon	n_i
Tarone-Ware	$\sqrt{n_i}$
Peto-Peto	$\tilde{S}(t_i)$
modified Peto-Peto	$\tilde{S}(t_i)\frac{n_i}{n_i+1}$
Harrington-Fleming (p,q)	$[\hat{S}(t_i)]^p[1 - \hat{S}(t_i)]^q, p \geq 0, q \geq 0$

where $\hat{S}(t)$ is the product-limit estimate at t for the pooled sample, and $\tilde{S}(t)$ is a survivor function estimate close to $\hat{S}(t)$ given by

$$\tilde{S}(t) = \prod_{t_i \leq t} \left(1 - \frac{d_i}{n_i + 1}\right)$$

Likelihood Ratio Test

The likelihood ratio test statistic (Lawless 1982) for test H_0 versus H_1 assumes that the data in the various samples are exponentially distributed and tests that the scale parameters are equal. The test statistic is computed as

$$Z = 2N \log\left(\frac{T}{N}\right) - 2\sum_{j=1}^{k} N_j \log\left(\frac{T_j}{N_j}\right)$$

where N_j is the total number of events in the jth stratum, $N = \sum_{j=1}^{k} N_j$, T_j is the total time on test in the jth stratum, and $T = \sum_{j=1}^{k} T_j$. The approximate probability value is computed by treating Z as having a chi-square distribution with $c-1$ degrees of freedom.

Trend Tests

Trend tests (Klein and Moeschberger 1997, Section 7.4) have more power to detect ordered alternatives as

$$H_2 : S_1(t) \geq S_2(t) \geq \ldots \geq S_k(t), t \leq \tau, \text{ with at least one inequality}$$

Let $a_1 < a_2 < \ldots < a_k$ be a sequence of scores associated with the k samples. The test statistic and its standard error are given by

$$\sum_{j=1}^{k} a_j v_j \quad \text{and} \quad \sum_{j=1}^{k}\sum_{l=1}^{k} a_j a_l V_{jl}$$

respectively. Under H_0, the $z-$score

$$Z = \frac{\sum_{j=1}^{k} a_j v_j}{\sqrt{\{\sum_{j=1}^{k}\sum_{l=1}^{k} a_j a_l V_{jl}\}}}$$

has, asymptotically, a standard normal distribution.

Stratified Tests

Suppose the test is to be stratified on M levels of a set of STRATA variables. Based only on the data of the sth stratum ($s = 1 \ldots M$), let $\mathbf{v}_s$ be the test statistic (Klein and Moeschberger 1997, Section 7.5) for the sth stratum, and let $\mathbf{V}_s$ be its covariance matrix. A global test statistic is constructed as

$$\chi^2 = \left(\sum_{i=1}^{M} \mathbf{v}_s\right)' \left(\sum_{i=1}^{M} \mathbf{V}_s\right)^{-1} \left(\sum_{i=1}^{M} \mathbf{v}_s\right)$$

Under the null hypothesis, the test statistic has a χ^2 distribution with the same df as the individual test for each stratum.

Rank Tests for the Association of Survival Time with Covariates

The rank tests for the association of covariates (Kalbfleisch and Prentice 1980, Chapter 6) are more general cases of the rank tests for homogeneity. In this section, the index α is used to label all observations, $\alpha = 1, 2, \ldots, n$, and the indices i, j range only over the observations that correspond to events, $i, j = 1, 2, \ldots, k$. The ordered event times are denoted as $t_{(i)}$, the corresponding vectors of covariates are denoted as $\mathbf{z}_{(i)}$, and the ordered times, both censored and event times, are denoted as t_α.

The rank test statistics have the form

$$\mathbf{v} = \sum_{\alpha=1}^{n} c_{\alpha,\delta_\alpha} \mathbf{z}_\alpha$$

where n is the total number of observations, c_{α,δ_α} are rank scores, which can be either log-rank or Wilcoxon rank scores, δ_α is 1 if the observation is an event and 0 if the observation is censored, and $\mathbf{z}_\alpha$ is the vector of covariates in the TEST statement for the αth observation. Notice that the scores, c_{α,δ_α}, depend on the censoring pattern and that the terms are summed up over all observations.

The log-rank scores are

$$c_{\alpha,\delta_\alpha} = \sum_{(j:t_{(j)} \le t_\alpha)} \left(\frac{1}{n_j} - \delta_\alpha \right)$$

and the Wilcoxon scores are

$$c_{\alpha,\delta_\alpha} = 1 - (1 + \delta_\alpha) \prod_{(j:t_{(j)} \le t_\alpha)} \frac{n_j}{n_j + 1}$$

where n_j is the number at risk just prior to $t_{(j)}$.

The estimates used for the covariance matrix of the log-rank statistics are

$$\mathbf{V} = \sum_{i=1}^{k} \frac{\mathbf{V}_i}{n_i}$$

where $\mathbf{V}_i$ is the corrected sum of squares and crossproducts matrix for the risk set at time $t_{(i)}$; that is,

$$\mathbf{V}_i = \sum_{(\alpha:t_\alpha \ge t_{(i)})} (\mathbf{z}_\alpha - \bar{\mathbf{z}}_i)'(\mathbf{z}_\alpha - \bar{\mathbf{z}}_i)$$

where

$$\bar{\mathbf{z}}_i = \sum_{(\alpha:t_\alpha \ge t_{(i)})} \frac{\mathbf{z}_\alpha}{n_i}$$

The estimate used for the covariance matrix of the Wilcoxon statistics is

$$\mathbf{V} = \sum_{i=1}^{k} \left[a_i(1 - a_i^*)(2\mathbf{z}_{(i)}\mathbf{z}'_{(i)} + \mathbf{S}_i) - (a_i^* - a_i) \left(a_i \mathbf{x}_i \mathbf{x}'_i + \sum_{j=i+1}^{k} a_j (\mathbf{x}_i \mathbf{x}'_j + \mathbf{x}_j \mathbf{x}'_i) \right) \right]$$

where

$$\begin{aligned}
a_i &= \prod_{j=1}^{i} \frac{n_j}{n_j + 1} \\
a_i^* &= \prod_{j=1}^{i} \frac{n_j + 1}{n_j + 2} \\
\mathbf{S}_i &= \sum_{(\alpha:t_{(i+1)} > t_\alpha > t_{(i)})} \mathbf{z}_\alpha \mathbf{z}'_\alpha \\
\mathbf{x}_i &= 2\mathbf{z}_{(i)} + \sum_{(\alpha:t_{(i+1)} > t_\alpha > t_{(i)})} \mathbf{z}_\alpha
\end{aligned}$$

In the case of tied failure times, the statistics $\mathbf{v}$ are averaged over the possible orderings of the tied failure times. The covariance matrices are also averaged over the tied failure times. Averaging the covariance matrices over the tied orderings produces functions with appropriate symmetries for the tied observations; however, the actual variances of the $\mathbf{v}$ statistics would be smaller than the preceding estimates. Unless the proportion of ties is large, it is unlikely that this will be a problem.

The univariate tests for each covariate are formed from each component of $\mathbf{v}$ and the corresponding diagonal element of $\mathbf{V}$ as v_i^2/V_{ii}. These statistics are treated as coming from a chi-square distribution for calculation of probability values.

The statistic $\mathbf{v}'\mathbf{V}^{-}\mathbf{v}$ is computed by sweeping each pivot of the $\mathbf{V}$ matrix in the order of greatest increase to the statistic. The corresponding sequence of partial statistics is tabulated. Sequential increments for including a given covariate and the corresponding probabilities are also included in the same table. These probabilities are calculated as the tail probabilities of a chi-square distribution with one degree of freedom. Because of the selection process, these probabilities should not be interpreted as p-values.

If desired for data screening purposes, the output data set requested by the OUTTEST= option can be treated as a sum of squares and crossproducts matrix and processed by the REG procedure using the option METHOD=RSQUARE. Then the sets of variables of a given size can be found that give the largest test statistics. Example 40.1 illustrates this process.

Computer Resources

The data are first read and sorted into strata. If the data are originally sorted by failure time and censoring state, with smaller failure times coming first and event values preceding censored values in cases of ties, the data can be processed by strata without additional sorting. Otherwise, the data are read into memory by strata and sorted.

Memory Requirements

For a given BY group, define

N the total number of observations

V the number of STRATA variables

C the number of covariates listed on the TEST statement

L total length of the ID variables in bytes

S number of strata

n maximum number of observations within strata

b $12 + 8C + L$

$m1$ $(112 + 16V) \times S$

$m2$ $50 \times b \times S$

$m3$ $(50+n)\times(b+4)$

$m4$ $8(C+4)^2$

$m5$ $20N+8S\times(S+4)$

The memory, in bytes, required to process the BY-group is at least

$$m1+\max(m2,m3)+m4$$

The test of equality of survival functions across strata requires additional memory ($m5$ bytes). However, if this additional memory is not available, PROC LIFETEST skips the test for equality of survival functions and finishes the other computations. Additional memory is required for the PLOTS= option. Temporary storage of $16n$ bytes is required to store the product-limit estimates for plotting.

Output Data Sets

OUTSURV= Data Set

You can specify either the OUTSURV= option in the PROC LIFETEST statement to create an output data set containing the following columns:

- any specified BY variables
- any specified STRATA variables, their values coming from either their original values or the midpoints of the stratum intervals if endpoints are used to define strata (semi-infinite intervals are labeled by their finite endpoint)
- STRATUM, a numeric variable that numbers the strata
- the time variable as given in the TIME statement. In the case of the product-limit estimates, it contains the observed failure or censored times. For the life-table estimates, it contains the lower endpoints of the time intervals.
- SURVIVAL, a variable containing the survivor function estimates
- CONFTYPE, a variable containing the name of the transformation applied to the survival time in the computation of confidence intervals (if the OUT= option is specified in the SURVIVAL statement)
- SDF_LCL, a variable containing the lower limits of the pointwise confidence intervals for the survivor function
- SDF_UCL, a variable containing the upper limits of the pointwise confidence intervals for the survivor function

If the estimation uses the product-limit method, then the data set also contains

- _CENSOR_, an indicator variable that has a value 1 for a censored observation and a value 0 for an event observation

If the estimation uses the life-table method, then the data set also contains

- MIDPOINT, a variable containing the value of the midpoint of the time interval
- PDF, a variable containing the density function estimates
- PDF_LCL, a variable containing the lower endpoint of the PDF confidence interval
- PDF_UCL, a variable containing the upper endpoint of the PDF confidence interval
- HAZARD, a variable containing the hazard estimates
- HAZ_LCL, a variable containing the lower endpoint of the hazard confidence interval
- HAZ_UCL, a variable containing the upper endpoint of the hazard confidence interval

Each survival function contains an initial observation with the value 1 for the SDF and the value 0 for the time. The output data set contains an observation for each distinct failure time if the product-limit method is used or an observation for each time interval if the life-table method is used. The product-limit survival estimates are defined to be right continuous; that is, the estimates at a given time include the factor for the failure events that occur at that time.

Labels are assigned to all the variables in the output data set except the BY variable and the STRATA variable.

OUT= Data Set

The OUT= option in the SURVIVAL statement creates an output data set containing all the variables listed in the OUTSURV= data set specified in the PROC LIFETEST statement with the additional variable

- CONFTYPE, a variable containing the type of transform used in the computation of the confidence intervals and bands for the survivor function

If the product-limit method is used, the OUT= data set also contains

- SDF_STDERR, a variable containing the standard error of the survivor function estimator (if the STDERR option is specified in the SURVIVAL statement)
- HW_LCL, a variable containing the lower limits of the Hall-Wellner confidence bands (if the CONFBAND=HW or CONFBAND=ALL is specified in the SURVIVAL statement)
- HW_UCL, a variable containing the upper limits of the Hall-Wellner confidence bands (if the CONFBAND=HW or CONFBAND=ALL is specified in the SURVIVAL statement)
- EP_LCL, a variable containing the lower limits of the equal precision confidence bands (if the CONFBAND=EP or CONFBAND=ALL is specified in the SURVIVAL statement)

- EP_UCL, a variable containing the upper limits of the equal precision confidence bands (if the CONFBAND=EP or CONFBAND=ALL is specified in the SURVIVAL statement)

In this release, the OUTSURV= data set is not created if you specify both the OUTSURV= option in the PROC LIFETEST statement and the OUT= option in the PROC statement.

OUTTEST= Data Set

The OUTTEST= option in the LIFETEST statement creates an output data set containing the rank statistics for testing the association of failure time with covariates. It contains

- any specified BY variables
- _TYPE_, a character variable of length 8 that labels the type of rank test, either “LOG-RANK” or “WILCOXON”
- _NAME_, a character variable of length 8 that labels the rows of the covariance matrix and the test statistics
- the TIME variable, containing the overall test statistic in the observation that has _NAME_ equal to the name of the time variable and the univariate test statistics under their respective covariates.
- all variables listed in the TEST statement

The output is in the form of a symmetric matrix formed by the covariance matrix of the rank statistics bordered by the rank statistics and the overall chi-square statistic. If the value of _NAME_ is the name of a variable in the TEST statement, the observation contains a row of the covariance matrix and the value of the rank statistic in the time variable. If the value of _NAME_ is the name of the TIME variable, the observation contains the values of the rank statistics in the variables from the TEST list and the value of the overall chi-square test statistic in the TIME variable.

Two complete sets of statistics labeled by the _TYPE_ variable are produced, one for the log-rank test and one for the Wilcoxon test.

Displayed Output

If you use the NOPRINT option in the PROC LIFETEST statement, the procedure does not display any output.

For each stratum, the LIFETEST procedure displays the following unless the NOTABLE option is specified.

- if you specify the STRATA statement, the values of the stratum variables
- if you request the product-limit estimate, the “Product-Limit Survival Estimates” table, which displays
  - the observed (event or censored) time

  - the estimate of the survivor function
  - the estimate of the cumulative distribution function of the failure time
  - the standard error estimate of the estimated survivor function
  - the number of event times that have been observed
  - the number of event or censored times that remain to be observed
  - if you specify the FREQ statement, the frequency of the observed event or censored times
  - if you specify the ID statement, the values of the ID variables

- if you request the product-limit estimate, the "Quartile Estimates" table, which contains

  - point estimates of the quartiles of the failure times
  - lower and upper confidence limits for the quartiles

- if you requested the product-limit estimate, the estimated mean survival time and its estimated standard error
- if you requested the life-table estimate, the "Life Table Survival Estimates" table, which displays

  - time intervals into which the failure and censored times are distributed; each interval is from the lower limit, up to but not including the upper limit. If the upper limit is infinity, the missing value is printed.
  - the number of events that occur in the interval
  - the number of censored observations that fall into the interval
  - the effective sample size for the interval
  - the estimate of conditional probability of events (failures) in the interval
  - the standard error of the estimated conditional probability of events
  - the estimate of the survival function at the beginning of the interval
  - the estimate of the cumulative distribution function of the failure time at the beginning of the interval
  - the standard error estimate of the estimated survival function
  - the estimate of the median residual lifetime, which is the amount of time elapsed before reducing the number of at-risk units to one-half. This is also known as the *median future lifetime* in Elandt-Johnson and Johnson (1980).
  - the estimated standard error of the estimated median residual lifetime
  - the density function estimated at the midpoint of the interval
  - the standard error estimate of the estimated density
  - the hazard rate estimated at the midpoint of the interval
  - the standard error estimate of the estimated hazard

The following results, processed across all strata, are displayed:

- the "Summary of the Number of Censored and Uncensored Values" table, which contains

  - if the STRATA statement is specified, the stratum identification
  - the total number of observations
  - the number of event observations
  - the number of censored observations
  - the percentage of censored observations
- the “Rank Statistics” table that contains the test statistics of the nonparametric k-sample tests or stratified k-sample tests
- the covariance matrix for the LOGRANK k-sample statistics
- the covariance matrix for the WILCOXON k-sample statistics
- the covariance matrix for the TARONE k-sample statistics (if requested)
- the covariance matrix for the PETO k-sample statistics (if requested)
- the covariance matrix for the MODPETO k-sample statistics (if requested)
- the covariance matrix for the FLEMING k-sample statistics (if requested)
- the “Test of Equality over Strata” table, which displays the results of the nonparametric k-sample test and the likelihood ratio test that is based on the exponential distribution (if the GROUP= option is not specified in the STRATA statement)
- the “Stratified Test of Equality over Group” table, which displays the results of the stratified k-sample tests for the GROUP= variable (if the GROUP= option in the STRATA statement is specified)
- the “Scores for Trend Test” tables, which displays the set of scores used to contruct the trend tests (if the TREND option is specified in the STRATA statement)
- the “Trend Tests” tables, which displays the results of the trend tests (if the TREND option is specified in the STRATA statement)
- the line printer plot or traditional high resolution graph of the estimated survivor function against failure time (if requested)
- the line printer plot or traditional high resolution graph of the negative log of the estimated survivor function against failure time (if requested)
- the line printer plot or traditional high resolution graph of the log of the negative log of the estimated survivor function against the log of the failure time (if requested)
- the line printer plot or traditional high resolution graph of the estimated hazard function against failure time (if requested and only for life-table method)
- the line printer plot or traditional high resolution graph of the estimated density function against failure time (if requested and only for life-table method)

If you specify the TEST statement, the PROC LIFETEST also displays

- the “Univariate Chi-Squares for the Log-Rank Test” table, which contains the log-rank test results for individual variables in the TEST statement

- the "Covariance Matrix for the Log-Rank Statistics" table
- the "Forward Stepwise Sequence of Chi-Squares for the Log-Rank Test" table, which contains
  - the sequence of partial chi-square statistics for the log-rank test in the order of the greatest increase to the overall test statistic
  - the degrees of freedom of the partial chi-square statistics
  - the approximate probability values of the partial chi-square statistics
  - the chi-square increments for including the given variables
  - the probability values of the chi-square increments
- the "Univariate Chi-Squares for the Wilcoxon Test" table, which contains the log-rank test results for individual variables in the TEST statement
- the "Covariance Matrix for the Wilcoxon Statistics" table
- the "Forward Stepwise Sequence of Chi-Squares for the Wilcoxon Test" table, which contains
  - the sequence of partial chi-square statistics for the Wilcoxon test in the order of the greatest increase to the overall test statistic
  - the degrees of freedom of the partial chi-square statistics
  - the approximate probability values of the partial chi-square statistics
  - the chi-square increments for including the given variables
  - the probability values of the chi-square increments

ODS Table Names

PROC LIFETEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 40.3. ODS Tables Produced in PROC LIFETEST

ODS Table Name	Description	Statement	Option
CensorPlot	line printer plot of censored observations	PROC	PLOT=(C|S|LS|LLS), METHOD=PL, and LINEPRINTER
CensoredSummary	number of event and censored observations	PROC	METHOD=PL
DensityPlot	traditional high-resolution or line printer plot of the density function	PROC	PLOT=(D) and METHOD=LT
HazardPlot	traditional high-resolution graph or line printer plot of the hazard function	PROC	PLOT=(H) and METHOD=LT
FlemingHomCov	covariance matrix for k-sample FLEMING statistics	STRATA	TEST=(FLEMING)

Table 40.3. (continued)

ODS Table Name	Description	Statement	Option
HomStats	test statistics for k-sample tests	STRATA	
HomTests	results of k-sample tests	STRATA	
LifetableEstimates	life-table survival estimates	PROC	METHOD=LT
LogForStepSeq	forward stepwise sequence for the log-rank statistics for association	TEST	
LogHomCov	covariance matrix for k-sample LOGRANK statistics	STRATA	TEST=(LOGRANK)
LogLogSurvivalPlot	traditional high-resolution graph or line printer plot of the log of the negative log survivor function	PROC	PLOT=(LLS)
LogSurvivalPlot	GSET or line printer plot of the log survivor function	PROC	PLOT=(LS)
LogTestCov	covariance matrix for log-rank statistics for association	TEST	
LogUniChisq	univariate chi-squares for log-rank statistic for association	TEST	
Means	mean and Standard Error of survival times	PROC	METHOD=PL
ModPetoHomCov	covariance matrix for k-sample MODPETO statistics	STRATA	TEST=(MODPETO)
NObs	Number of observations		default
PetoHomCov	covariance matrix for k-sample PETO statistics	STRATA	TEST=(PETO)
ProductLimitEstimates	product-limit survival estimates	PROC	METHOD=PL
Quartiles	quartiles of the survival distribution	PROC	METHOD=PL
SurvivalPlot	traditional high-resolution graph or line printer plot of the survivor function	PROC	PLOT=(S)
TaroneHomCov	covariance matrix for k-sample TARONE statistics	STRATA	TEST=(TARONE)
TrendScores	scores used to construct trend tests	STRATA	TREND
TrendTests	results of Trend Tests	STRATA	TREND
WilForStepSeq	forward stepwise sequence for the log-rank statistics for association	TEST	
WilHomCov	covariance matrix for k-sample WILCOXON statistics	STRATA	TEST=(WILCOXON)
WilTestCov	covariance matrix for log-rank statistics for association	TEST	
WilUniChiSq	univariate chi-squares for Wilcoxon statistic for association	TEST	

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the LIFETEST procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the PLOTS= option. For general information about ODS GRAPHICS, see Chapter 15, “Statistical Graphics Using ODS.”

You can specify the PLOTS=option in the SURVIVAL statement to request graphical displays of survival distribution and related function estimates:

PLOTS=(*type <,type,...,type>***)**

requests one or more plots of the survival estimates, where *type* is the type of plot. The following table shows the valid values of *type*.

Type	**Plot Description**
CL | PCL	pointwise confidence limits for the survival function
DENSITY | PDF	estimated density function (life-table method only)
EPB	equal precision bands for survivorship (product-limit method only)
HAZARD | HAZ	estimated hazard function (life-table method only)
HWB	Hall-Wellner confidence bands for survivorship (product-limit method only)
LOGLOGS | LLS	log of negative log of estimated survivor function versus log of time
LOGSURV | LS	negative log of estimated survivor function
STRATUM	a panel of three plots and a table of summary statistics for each stratum. For the product-limit method, the panel contains a plot of the product-limit estimates, a plot of the Hall-Wellner band, and a plot of the equal precision band. Pointwise confidence limits and the product-limit estimates are also plotted along with the confidence bands. For the life-table method, the panel contains a plot of life-table estimates, a plot of the estimated hazard function, and a plot of the estimated density. Corresponding pointwise confidence limits are also plotted along with the survival estimates.
SURVIVAL | S	estimated survivor function. Censored times are plotted as a plus sign on the product-limit curve.

You can use the ALPHA=, BANDMIN=, BANDMAX=, and CONFTYPE= options in the SURVIVAL statement to modify the confidence limits and confidence bands. To restrict the display of the plots up to a certain limit on the time scale, specify the following option in the SURVIVAL statement.

MAXTIME=*value*
specifies the maximum value of the time variable allowed on the plots so that outlying points do not determine the scale of the time axis of the plots.

ODS Graph Names

PROC LIFETEST assigns a name to each graph it creates using the Output Delivery System (ODS). You can use these names to reference the graphs when using ODS. The names are listed in Table 40.4.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 40.4. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Table 40.4. ODS Graphics Produced by PROC LIFETEST

ODS Graph Name	Plot Description	Statement	PLOTS=(type)
Density	Density estimate based on life-table method	SURVIVAL	PDF
EqualPrecision	Equal precision band overlaid with the estimated survival curve	SURVIVAL	EPB
HallWellner	Hall and Wellner band overlaid with the estimated survival curve	SURVIVAL	HWB
Hazard	Hazard estimate based on the life-table method	SURVIVAL	HAZ
LogNegLogSurvival	Log of negative log of the estimated survival function	SURVIVAL	LLS
NegLogSurvival	Negative log of the estimated survival function	SURVIVAL	LS
Survival	Estimated survival curve	SURVIVAL	S
SurvivalCL	Pointwise confidence limits overlaid with estimated survival curve	SURVIVAL	CL

Examples

Example 40.1. Product-Limit Estimates and Tests of Association for the VA Lung Cancer Data

This example uses the data presented in Appendix I of Kalbfleisch and Prentice (1980). The response variable, SurvTime, is the survival time in days of a lung cancer patient. Negative values of SurvTime are censored values. The covariates are Cell (type of cancer cell), Therapy (type of therapy: standard or test), Prior (prior therapy: 0=no, 10=yes), Age (age in years), DiagTime (time in months from diagnosis to entry into the trial), and Kps (performance status). A censoring indicator variable Censor is created from the data, with value 1 indicating a censored time and value 0 an event time. Since there are only two types of therapy, an indicator variable, Treatment, is constructed for therapy type, with value 0 for standard therapy and value 1 for test therapy.

```
data VALung;
   drop check m;
   retain Therapy Cell;
   infile cards column=column;
   length Check $ 1;
   label SurvTime='failure or censoring time'
      Kps='karnofsky index'
      DiagTime='months till randomization'
      Age='age in years'
      Prior='prior treatment?'
      Cell='cell type'
      Therapy='type of treatment'
      Treatment='treatment indicator';
   M=Column;
   input Check $ @@;
   if M>Column then M=1;
   if Check='s'|Check='t' then input @M Therapy $ Cell $ ;
   else input @M SurvTime Kps DiagTime Age Prior @@;
   if SurvTime > .;
   censor=(SurvTime<0);
   SurvTime=abs(SurvTime);
   Treatment=(Therapy='test');
   cards;
standard squamous
 72 60  7 69  0   411 70  5 64 10   228 60  3 38  0   126 60  9 63 10
118 70 11 65 10    10 20  5 49  0    82 40 10 69 10   110 80 29 68  0
314 50 18 43  0  -100 70  6 70  0    42 60  4 81  0     8 40 58 63 10
144 30  4 63  0   -25 80  9 52 10    11 70 11 48 10
standard small
 30 60  3 61  0   384 60  9 42  0     4 40  2 35  0    54 80  4 63 10
 13 60  4 56  0  -123 40  3 55  0   -97 60  5 67  0   153 60 14 63 10
 59 30  2 65  0   117 80  3 46  0    16 30  4 53 10   151 50 12 69  0
 22 60  4 68  0    56 80 12 43 10    21 40  2 55 10    18 20 15 42  0
139 80  2 64  0    20 30  5 65  0    31 75  3 65  0    52 70  2 55  0
287 60 25 66 10    18 30  4 60  0    51 60  1 67  0   122 80 28 53  0
 27 60  8 62  0    54 70  1 67  0     7 50  7 72  0    63 50 11 48  0
392 40  4 68  0    10 40 23 67 10
standard adeno
  8 20 19 61 10    92 70 10 60  0    35 40  6 62  0   117 80  2 38  0
132 80  5 50  0    12 50  4 63 10   162 80  5 64  0     3 30  3 43  0
 95 80  4 34  0
standard large
177 50 16 66 10   162 80  5 62  0   216 50 15 52  0   553 70  2 47  0
278 60 12 63  0    12 40 12 68 10   260 80  5 45  0   200 80 12 41 10
156 70  2 66  0  -182 90  2 62  0   143 90  8 60  0   105 80 11 66  0
103 80  5 38  0   250 70  8 53 10   100 60 13 37 10
test squamous
999 90 12 54 10   112 80  6 60  0   -87 80  3 48  0  -231 50  8 52 10
242 50  1 70  0   991 70  7 50 10   111 70  3 62  0     1 20 21 65 10
587 60  3 58  0   389 90  2 62  0    33 30  6 64  0    25 20 36 63  0
357 70 13 58  0   467 90  2 64  0   201 80 28 52 10     1 50  7 35  0
 30 70 11 63  0    44 60 13 70 10   283 90  2 51  0    15 50 13 40 10
test small
```

```
    25 30  2 69  0  -103 70 22 36 10    21 20  4 71  0    13 30  2 62  0
    87 60  2 60  0     2 40 36 44 10    20 30  9 54 10     7 20 11 66  0
    24 60  8 49  0    99 70  3 72  0     8 80  2 68  0    99 85  4 62  0
    61 70  2 71  0    25 70  2 70  0    95 70  1 61  0    80 50 17 71  0
    51 30 87 59 10    29 40  8 67  0
   test adeno
    24 40  2 60  0    18 40  5 69 10   -83 99  3 57  0    31 80  3 39  0
    51 60  5 62  0    90 60 22 50 10    52 60  3 43  0    73 60  3 70  0
     8 50  5 66  0    36 70  8 61  0    48 10  4 81  0     7 40  4 58  0
   140 70  3 63  0   186 90  3 60  0    84 80  4 62 10    19 50 10 42  0
    45 40  3 69  0    80 40  4 63  0
   test large
    52 60  4 45  0   164 70 15 68 10    19 30  4 39 10    53 60 12 66  0
    15 30  5 63  0    43 60 11 49 10   340 80 10 64 10   133 75  1 65  0
   111 60  5 64  0   231 70 18 67 10   378 80  4 65  0    49 30  3 37  0
   ;
```

PROC LIFETEST is invoked to compute the product-limit estimate of the survivor function for each type of cancer cell and to analyze the effects of the variables Age, Prior, DiagTime, Kps, and Treatment on the survival of the patients. These prognostic factors are specified in the TEST statement, and the variable Cell is specified in the STRATA statement. Traditional high-resolution graphs of the product-limit estimates, the log estimates, and the negative log-log estimates are requested through the PLOTS= option in the PROC LIFETEST statement. Because of a few large survival times, a MAXTIME of 600 is used to set the scale of the time axis; that is, the time scale extends from 0 to a maximum of 600 days in the plots. The variable Therapy is specified in the ID statement to identify the type of therapy for each observation in the product-limit estimates. The OUTTEST option specifies the creation of an output data set named Test to contain the rank test matrices for the covariates.

```
   symbol1 c=blue; symbol2 c=orange; symbol3 c=green;
   symbol4 c=red; symbol5 c=cyan; symbol6 c=black;
   title 'VA Lung Cancer Data';
   proc lifetest data=VALung plots=(s,ls,lls) outtest=Test maxtime=600;
      time SurvTime*Censor(1);
      id Therapy;
      strata Cell;
      test Age Prior DiagTime Kps Treatment;
   run;
```

Output 40.1.1 through Output 40.1.4 display the product-limit estimates of the survivor functions for the four cell types. Summary statistics of the survival times are also shown. The median survival times are 51 days, 156 days, 51 days, and 118 days for patients with adeno cells, large cells, small cells, and squamous cells, respectively.

Output 40.1.1. Estimation Results for Cell=adeno

```
                              Stratum 1: Cell = adeno

                          Product-Limit Survival Estimates

                                              Survival
                                              Standard     Number     Number
   SurvTime     Survival     Failure           Error       Failed      Left      Therapy

      0.000       1.0000           0               0          0         27
      3.000       0.9630      0.0370          0.0363          1         26      standard
      7.000       0.9259      0.0741          0.0504          2         25      test
      8.000            .           .               .          3         24      standard
      8.000       0.8519      0.1481          0.0684          4         23      test
     12.000       0.8148      0.1852          0.0748          5         22      standard
     18.000       0.7778      0.2222          0.0800          6         21      test
     19.000       0.7407      0.2593          0.0843          7         20      test
     24.000       0.7037      0.2963          0.0879          8         19      test
     31.000       0.6667      0.3333          0.0907          9         18      test
     35.000       0.6296      0.3704          0.0929         10         17      standard
     36.000       0.5926      0.4074          0.0946         11         16      test
     45.000       0.5556      0.4444          0.0956         12         15      test
     48.000       0.5185      0.4815          0.0962         13         14      test
     51.000       0.4815      0.5185          0.0962         14         13      test
     52.000       0.4444      0.5556          0.0956         15         12      test
     73.000       0.4074      0.5926          0.0946         16         11      test
     80.000       0.3704      0.6296          0.0929         17         10      test
     83.000*           .           .               .         17          9      test
     84.000       0.3292      0.6708          0.0913         18          8      test
     90.000       0.2881      0.7119          0.0887         19          7      test
     92.000       0.2469      0.7531          0.0850         20          6      standard
     95.000       0.2058      0.7942          0.0802         21          5      standard
    117.000       0.1646      0.8354          0.0740         22          4      standard
    132.000       0.1235      0.8765          0.0659         23          3      standard
    140.000       0.0823      0.9177          0.0553         24          2      test
    162.000       0.0412      0.9588          0.0401         25          1      standard
    186.000            0      1.0000               0         26          0      test

            NOTE: The marked survival times are censored observations.

                                 Quartile Estimates

                                 Point      95% Confidence Interval
                   Percent     Estimate       [Lower       Upper)

                        75       92.000       73.000     140.000
                        50       51.000       31.000      90.000
                        25       19.000        8.000      45.000


                                 Mean     Standard Error

                               65.556             10.127
```

Output 40.1.2. Estimation Results for Cell=large

```
                         Stratum 2: Cell = large

                     Product-Limit Survival Estimates

                                       Survival
                                       Standard     Number     Number
  SurvTime     Survival    Failure      Error       Failed      Left     Therapy

     0.000      1.0000           0           0         0         27
    12.000      0.9630      0.0370      0.0363         1         26      standard
    15.000      0.9259      0.0741      0.0504         2         25      test
    19.000      0.8889      0.1111      0.0605         3         24      test
    43.000      0.8519      0.1481      0.0684         4         23      test
    49.000      0.8148      0.1852      0.0748         5         22      test
    52.000      0.7778      0.2222      0.0800         6         21      test
    53.000      0.7407      0.2593      0.0843         7         20      test
   100.000      0.7037      0.2963      0.0879         8         19      standard
   103.000      0.6667      0.3333      0.0907         9         18      standard
   105.000      0.6296      0.3704      0.0929        10         17      standard
   111.000      0.5926      0.4074      0.0946        11         16      test
   133.000      0.5556      0.4444      0.0956        12         15      test
   143.000      0.5185      0.4815      0.0962        13         14      standard
   156.000      0.4815      0.5185      0.0962        14         13      standard
   162.000      0.4444      0.5556      0.0956        15         12      standard
   164.000      0.4074      0.5926      0.0946        16         11      test
   177.000      0.3704      0.6296      0.0929        17         10      standard
   182.000*          .           .           .        17          9      standard
   200.000      0.3292      0.6708      0.0913        18          8      standard
   216.000      0.2881      0.7119      0.0887        19          7      standard
   231.000      0.2469      0.7531      0.0850        20          6      test
   250.000      0.2058      0.7942      0.0802        21          5      standard
   260.000      0.1646      0.8354      0.0740        22          4      standard
   278.000      0.1235      0.8765      0.0659        23          3      standard
   340.000      0.0823      0.9177      0.0553        24          2      test
   378.000      0.0412      0.9588      0.0401        25          1      test
   553.000           0      1.0000           0        26          0      standard

           NOTE: The marked survival times are censored observations.

                            Quartile Estimates

                            Point     95% Confidence Interval
                 Percent    Estimate      [Lower      Upper)

                      75     231.000     164.000     340.000
                      50     156.000     103.000     216.000
                      25      53.000      43.000     133.000


                            Mean    Standard Error

                         170.506            25.098
```

Output 40.1.3. Estimation Results for Cell=small

```
                          Stratum 3: Cell = small

                      Product-Limit Survival Estimates

                                  Survival
                                  Standard    Number      Number
SurvTime    Survival   Failure     Error      Failed       Left      Therapy

   0.000      1.0000          0          0       0          48
   2.000      0.9792     0.0208     0.0206       1          47      test
   4.000      0.9583     0.0417     0.0288       2          46      standard
   7.000           .          .          .       3          45      standard
   7.000      0.9167     0.0833     0.0399       4          44      test
   8.000      0.8958     0.1042     0.0441       5          43      test
  10.000      0.8750     0.1250     0.0477       6          42      standard
  13.000           .          .          .       7          41      standard
  13.000      0.8333     0.1667     0.0538       8          40      test
  16.000      0.8125     0.1875     0.0563       9          39      standard
  18.000           .          .          .      10          38      standard
  18.000      0.7708     0.2292     0.0607      11          37      standard
  20.000           .          .          .      12          36      standard
  20.000      0.7292     0.2708     0.0641      13          35      test
  21.000           .          .          .      14          34      standard
  21.000      0.6875     0.3125     0.0669      15          33      test
  22.000      0.6667     0.3333     0.0680      16          32      standard
  24.000      0.6458     0.3542     0.0690      17          31      test
  25.000           .          .          .      18          30      test
  25.000      0.6042     0.3958     0.0706      19          29      test
  27.000      0.5833     0.4167     0.0712      20          28      standard
  29.000      0.5625     0.4375     0.0716      21          27      test
  30.000      0.5417     0.4583     0.0719      22          26      standard
  31.000      0.5208     0.4792     0.0721      23          25      standard
  51.000           .          .          .      24          24      standard
  51.000      0.4792     0.5208     0.0721      25          23      test
  52.000      0.4583     0.5417     0.0719      26          22      standard
  54.000           .          .          .      27          21      standard
  54.000      0.4167     0.5833     0.0712      28          20      standard
  56.000      0.3958     0.6042     0.0706      29          19      standard
  59.000      0.3750     0.6250     0.0699      30          18      standard
  61.000      0.3542     0.6458     0.0690      31          17      test
  63.000      0.3333     0.6667     0.0680      32          16      standard
  80.000      0.3125     0.6875     0.0669      33          15      test
  87.000      0.2917     0.7083     0.0656      34          14      test
  95.000      0.2708     0.7292     0.0641      35          13      test
  97.000*          .          .          .      35          12      standard
  99.000           .          .          .      36          11      test
  99.000      0.2257     0.7743     0.0609      37          10      test
 103.000*          .          .          .      37           9      test
 117.000      0.2006     0.7994     0.0591      38           8      standard
 122.000      0.1755     0.8245     0.0567      39           7      standard
 123.000*          .          .          .      39           6      standard
 139.000      0.1463     0.8537     0.0543      40           5      standard
 151.000      0.1170     0.8830     0.0507      41           4      standard
 153.000      0.0878     0.9122     0.0457      42           3      standard
 287.000      0.0585     0.9415     0.0387      43           2      standard
 384.000      0.0293     0.9707     0.0283      44           1      standard
 392.000           0     1.0000          0      45           0      standard

            NOTE: The marked survival times are censored observations.

                             Quartile Estimates

                             Point     95% Confidence Interval
                Percent     Estimate     [Lower      Upper)

                     75      99.000      59.000     151.000
                     50      51.000      25.000      61.000
                     25      20.000      13.000      25.000
```

Output 40.1.3. (continued)

```
                             Mean    Standard Error

                           78.981            14.837
```

Output 40.1.4. Estimation Results for Cell=squamous

```
                         Stratum 4: Cell = squamous

                       Product-Limit Survival Estimates

                                          Survival
                                          Standard     Number      Number
 SurvTime     Survival    Failure          Error       Failed       Left      Therapy

    0.000       1.0000          0              0          0          35
    1.000        .            .              .          1          34      test
    1.000       0.9429     0.0571         0.0392          2          33      test
    8.000       0.9143     0.0857         0.0473          3          32      standard
   10.000       0.8857     0.1143         0.0538          4          31      standard
   11.000       0.8571     0.1429         0.0591          5          30      standard
   15.000       0.8286     0.1714         0.0637          6          29      test
   25.000       0.8000     0.2000         0.0676          7          28      test
   25.000*       .            .              .          7          27      standard
   30.000       0.7704     0.2296         0.0713          8          26      test
   33.000       0.7407     0.2593         0.0745          9          25      test
   42.000       0.7111     0.2889         0.0772         10          24      standard
   44.000       0.6815     0.3185         0.0794         11          23      test
   72.000       0.6519     0.3481         0.0813         12          22      standard
   82.000       0.6222     0.3778         0.0828         13          21      standard
   87.000*       .            .              .         13          20      test
  100.000*       .            .              .         13          19      standard
  110.000       0.5895     0.4105         0.0847         14          18      standard
  111.000       0.5567     0.4433         0.0861         15          17      test
  112.000       0.5240     0.4760         0.0870         16          16      test
  118.000       0.4912     0.5088         0.0875         17          15      standard
  126.000       0.4585     0.5415         0.0876         18          14      standard
  144.000       0.4257     0.5743         0.0873         19          13      standard
  201.000       0.3930     0.6070         0.0865         20          12      test
  228.000       0.3602     0.6398         0.0852         21          11      standard
  231.000*       .            .              .         21          10      test
  242.000       0.3242     0.6758         0.0840         22           9      test
  283.000       0.2882     0.7118         0.0820         23           8      test
  314.000       0.2522     0.7478         0.0793         24           7      standard
  357.000       0.2161     0.7839         0.0757         25           6      test
  389.000       0.1801     0.8199         0.0711         26           5      test
  411.000       0.1441     0.8559         0.0654         27           4      standard
  467.000       0.1081     0.8919         0.0581         28           3      test
  587.000       0.0720     0.9280         0.0487         29           2      test
  991.000       0.0360     0.9640         0.0352         30           1      test
  999.000            0     1.0000              0         31           0      test

           NOTE: The marked survival times are censored observations.

                             Quartile Estimates

                             Point     95% Confidence Interval
                 Percent    Estimate      [Lower       Upper)

                      75     357.000     201.000     467.000
                      50     118.000      72.000     242.000
                      25      33.000      11.000     111.000


                             Mean    Standard Error

                          230.225            48.475
```

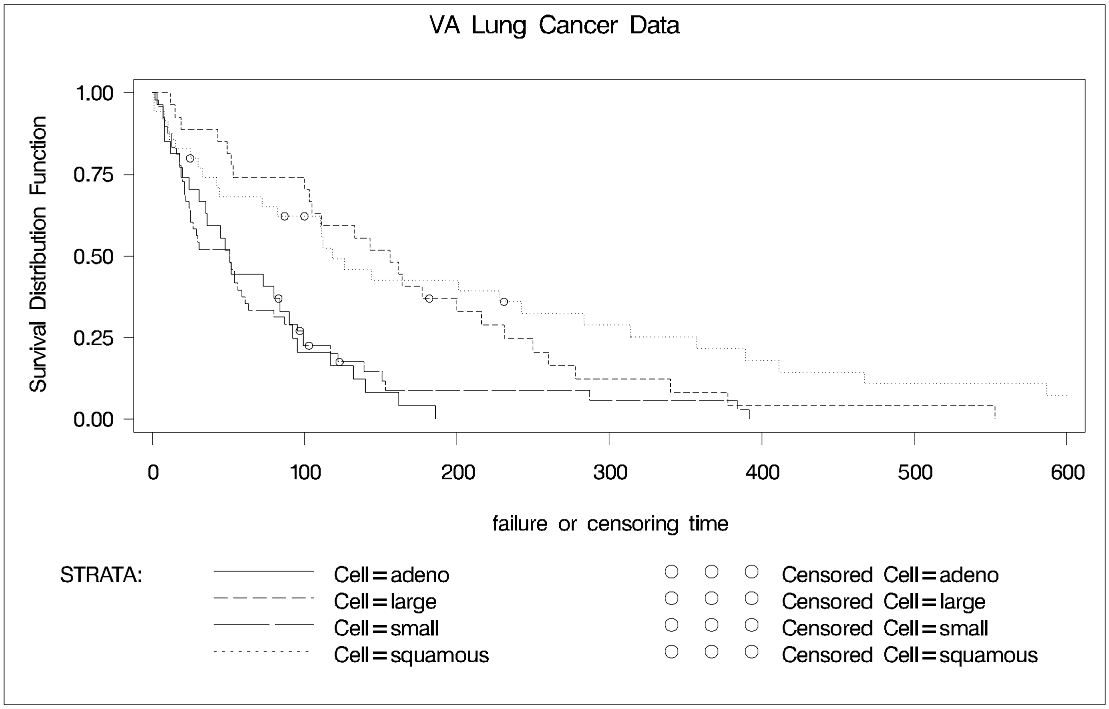
The distribution of event and censored observations among the four cell types is summarized in Output 40.1.5.

Output 40.1.5. Summary of Censored and Uncensored Values

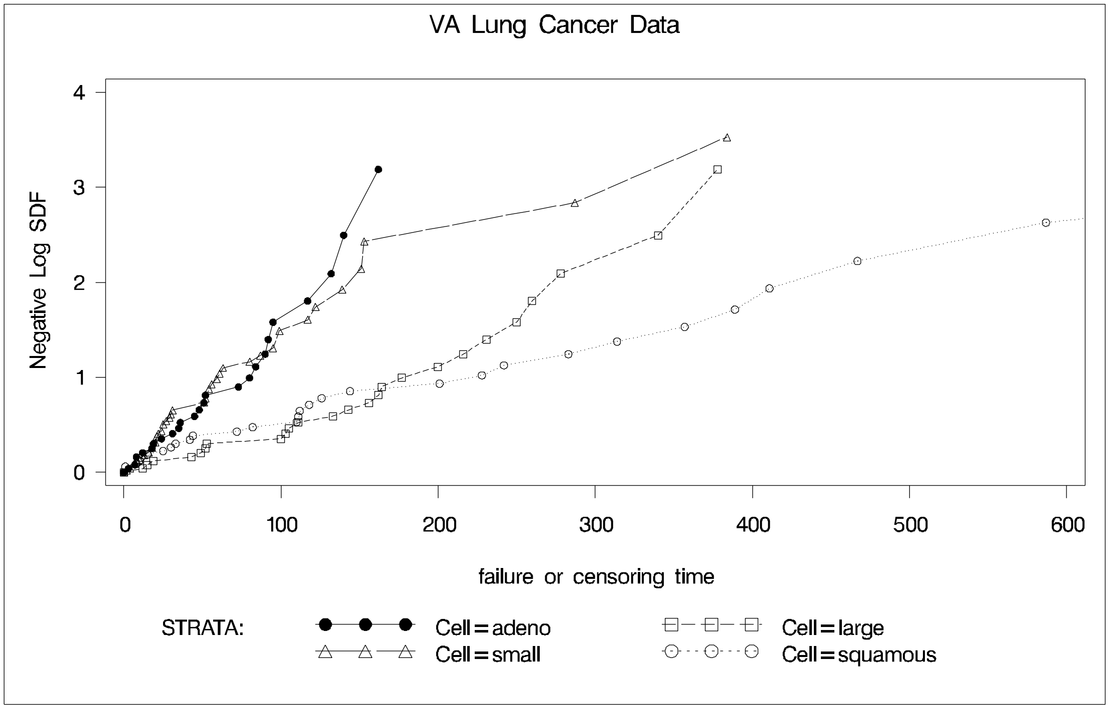
```
            Summary of the Number of Censored and Uncensored Values

                                                                 Percent
        Stratum     Cell            Total  Failed    Censored   Censored

              1     adeno              27      26           1       3.70
              2     large              27      26           1       3.70
              3     small              48      45           3       6.25
              4     squamous           35      31           4      11.43
        ----------------------------------------------------------------
          Total                       137     128           9       6.57
```

The graph of the estimated survivor functions is shown in Output 40.1.6. The adeno cell curve and the small cell curve are much closer to each other than to the large cell curve or the squamous cell curve. The survival rates of the adeno cell patients and the small cell patients decrease rapidly to approximately 29% in 90 days. Shapes of the large cell curve and the squamous cell curve are quite different, although both decrease less rapidly than those of the adeno and small cells. The squamous cell curve decreases more rapidly initially than the large cell curve, but the role is reversed in the later period.

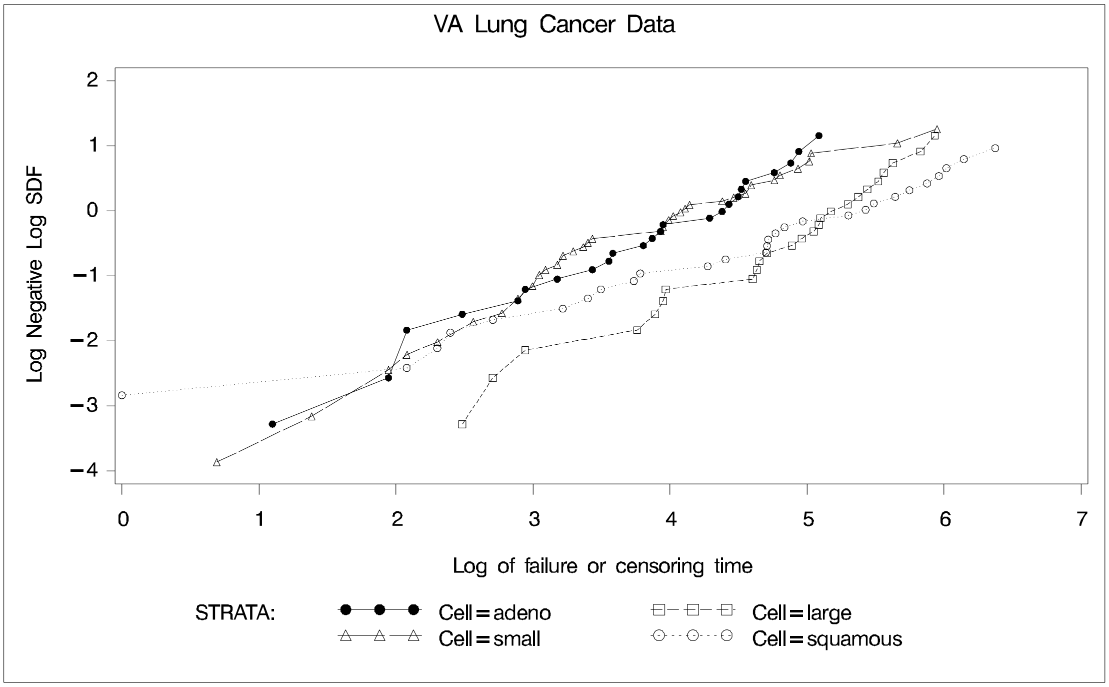
Output 40.1.6. Graph of the Estimated Survivor Functions



Output 40.1.7. Graph of Negative Log of the Estimated Survivor Functions



Output 40.1.8. Graph of Log of the Negative Log of the Estimated Survivor Functions



The graph of the negative log of the estimated survivor functions is displayed in Output 40.1.7. Output 40.1.8 displays the log of the negative log of the estimated survivor functions against the log of time.

Results of the homogeneity tests across cell types are given in Output 40.1.9. The log-rank and Wilcoxon statistics and their corresponding covariance matrices are displayed. Also given is a table that consists of the approximate chi-square statistics, degrees of freedom, and p-values for the log-rank, Wilcoxon, and likelihood ratio tests. All three tests indicate strong evidence of a significant difference among the survival curves for the four types of cancer cells ($p < 0.001$).

Output 40.1.9. Homogeneity Tests Across Strata

```
                                Rank Statistics

                        Cell          Log-Rank     Wilcoxon

                        adeno           10.306        697.0
                        large           -8.549      -1085.0
                        small           14.898       1278.0
                        squamous       -16.655       -890.0


                  Covariance Matrix for the Log-Rank Statistics

        Cell              adeno          large          small       squamous

        adeno           12.9662        -4.0701        -4.4087        -4.4873
        large           -4.0701        24.1990        -7.8117       -12.3172
        small           -4.4087        -7.8117        21.7543        -9.5339
        squamous        -4.4873       -12.3172        -9.5339        26.3384


                  Covariance Matrix for the Wilcoxon Statistics

        Cell              adeno          large          small       squamous

        adeno            121188         -34718         -46639         -39831
        large            -34718         151241         -59948         -56576
        small            -46639         -59948         175590         -69002
        squamous         -39831         -56576         -69002         165410


                           Test of Equality over Strata

                                                         Pr >
                   Test        Chi-Square       DF    Chi-Square

                   Log-Rank       25.4037        3        <.0001
                   Wilcoxon       19.4331        3        0.0002
                   -2Log(LR)      33.9343        3        <.0001
```

Results of the log-rank test of the prognostic variables are shown in Output 40.1.10. The univariate test results correspond to testing each prognostic factor marginally. The joint covariance matrix of these univariate test statistics is also displayed. In computing the overall chi-square statistic, the partial chi-square statistics following a forward stepwise entry approach are tabulated.

Consider the log-rank test in Output 40.1.10. Since the univariate test for Kps has the largest chi-square (43.4747) among all the covariates, Kps is entered first. At this stage, the partial chi-square and the chi-square increment for Kps are the same as the univariate chi-square. Among all the covariates not in the model (Age, Prior, DiagTime, Treatment), Treatment has the largest approximate chi-square increment (1.7261) and is entered next. The approximate chi-square for the model containing Kps and Treatment is 43.4747+1.7261=45.2008 with 2 degrees of freedom. The third covariate entered is Age. The fourth is Prior, and the fifth is DiagTime. The overall chi-square statistic on the last line of output is the partial chi-square for including all the covariates. It has a value of 46.4200 with 5 degrees of freedom, which is highly significant ($p < 0.0001$).

Output 40.1.10. Log-Rank Test of the Prognostic Factors

```
                  Univariate Chi-Squares for the Log-Rank Test

               Test    Standard               Pr >
 Variable  Statistic Deviation Chi-Square Chi-Square Label

 Age       -40.7383      105.7    0.1485    0.7000   age in years
 Prior     -19.9435    46.9836    0.1802    0.6712   prior treatment?
 DiagTime    -115.9    97.8708    1.4013    0.2365   months till randomization
 Kps         1123.1      170.3   43.4747    <.0001   karnofsky index
 Treatment  -4.2076     5.0407    0.6967    0.4039   treatment indicator


                  Covariance Matrix for the Log-Rank Statistics

Variable            Age          Prior       DiagTime            Kps      Treatment

Age             11175.4         -301.2         -892.2        -2948.4          119.3
Prior            -301.2         2207.5         2010.9           78.6           13.9
DiagTime         -892.2         2010.9         9578.7        -2295.3           21.9
Kps             -2948.4           78.6        -2295.3        29015.6           61.9
Treatment         119.3           13.9           21.9           61.9           25.4


          Forward Stepwise Sequence of Chi-Squares for the Log-Rank Test

                                            Pr >       Chi-Square       Pr >
    Variable       DF     Chi-Square    Chi-Square      Increment    Increment

    Kps             1       43.4747       <.0001         43.4747      <.0001
    Treatment       2       45.2008       <.0001          1.7261      0.1889
    Age             3       46.3012       <.0001          1.1004      0.2942
    Prior           4       46.4134       <.0001          0.1122      0.7377
    DiagTime        5       46.4200       <.0001         0.00665      0.9350

                     Variable     Label

                     Kps          karnofsky index
                     Treatment    treatment indicator
                     Age          age in years
                     Prior        prior treatment?
                     DiagTime     months till randomization
```

You can establish this forward stepwise entry of prognostic factors by passing the matrix corresponding to the log-rank test to the RSQUARE method in the REG procedure. PROC REG finds the sets of variables that yield the largest chi-square statistics.

```
data RSq;
   set Test;
   if _type_='LOG RANK';
      _type_='cov';

 proc print data=RSq;

 proc reg data=RSq(type=COV);
    model SurvTime=Age Prior DiagTime Kps Treatment
          / selection=rsquare;
    title 'All Possible Subsets of Covariates for the
           log-rank Test';
run;
```

Output 40.1.11 displays the univariate statistics and their covariance matrix for the log-rank test.

Output 40.1.11. Log-Rank Statistics and Covariance Matrix

```
Obs _TYPE_ _NAME_    SurvTime       Age    Prior DiagTime       Kps Treatment

 1   cov   SurvTime     46.42    -40.74   -19.94  -115.86   1123.14    -4.208
 2   cov   Age         -40.74  11175.44  -301.23  -892.24  -2948.45   119.297
 3   cov   Prior       -19.94   -301.23  2207.46  2010.85     78.64    13.875
 4   cov   DiagTime   -115.86   -892.24  2010.85  9578.69  -2295.32    21.859
 5   cov   Kps        1123.14  -2948.45    78.64 -2295.32  29015.62    61.945
 6   cov   Treatment    -4.21    119.30    13.87    21.86     61.95    25.409
```

Results of the best subset regression are shown in Output 40.1.12. The variable Kps generates the largest univariate test statistic among all the covariates, the pair Kps and Age generate the largest test statistic among any other pairs of covariates, and so on. The entry order of covariates is identical to that of PROC LIFETEST.

Output 40.1.12. Best Subset Regression from the REG Procedure

```
   All Possible Subsets of Covariates for the              log-rank Test

                          The REG Procedure
                           Model: MODEL1
                    Dependent Variable: SurvTime

                      R-Square Selection Method

         Number in
           Model      R-Square    Variables in Model

                1       0.9366    Kps
                1       0.0302    DiagTime
                1       0.0150    Treatment
                1       0.0039    Prior
                1       0.0032    Age
         ----------------------------------------------------------
                2       0.9737    Kps Treatment
                2       0.9472    Age Kps
                2       0.9417    Prior Kps
                2       0.9382    DiagTime Kps
                2       0.0434    DiagTime Treatment
                2       0.0353    Age DiagTime
                2       0.0304    Prior DiagTime
                2       0.0181    Prior Treatment
                2       0.0159    Age Treatment
                2       0.0075    Age Prior
         ----------------------------------------------------------
                3       0.9974    Age Kps Treatment
                3       0.9774    Prior Kps Treatment
                3       0.9747    DiagTime Kps Treatment
                3       0.9515    Age Prior Kps
                3       0.9481    Age DiagTime Kps
                3       0.9418    Prior DiagTime Kps
                3       0.0456    Age DiagTime Treatment
                3       0.0438    Prior DiagTime Treatment
                3       0.0355    Age Prior DiagTime
                3       0.0192    Age Prior Treatment
         ----------------------------------------------------------
                4       0.9999    Age Prior Kps Treatment
                4       0.9976    Age DiagTime Kps Treatment
                4       0.9774    Prior DiagTime Kps Treatment
                4       0.9515    Age Prior DiagTime Kps
                4       0.0459    Age Prior DiagTime Treatment
         ----------------------------------------------------------
                5       1.0000    Age Prior DiagTime Kps Treatment
```

Example 40.2. Confidence Bands for Survival of Bone Marrow Transplant Patients (Experimental)

This example uses the data of 137 bone marrow transplant patients extracted from Klein and Moeschberger (1997). At the time of transplant, each patient is classified into one of three risk categories: ALL (Acute Lymphoblastic Leukemia), low-risk AML (Acute Myeloctic Leukemia), and high-risk AML. The endpoint of interest is the disease-free survival, which is the time to death or relapse or the end of the study in days. The data are saved in the SAS data set BMT. In this data set, the variable Group represents the Patient’s risk category, the variable T represents the disease-free survival time, and the variable Status is the censoring indicator with value 1 indicating an event time and value 0 a censored time.

```
proc format;
   value risk 1=’ALL’ 2=’low-risk AML’ 3=’high-risk AML’;

data BMT;
   input Group T Status @@;
   format Group risk.;
   label T=’Time to Relapse’;
   datalines;
1 2081 0 1 1602 0 1 1496 0 1 1462 0 1 1433 0
1 1377 0 1 1330 0 1  996 0 1  226 0 1 1199 0
1 1111 0 1  530 0 1 1182 0 1 1167 0 1  418 1
1  383 1 1  276 1 1  104 1 1  609 1 1  172 1
1  487 1 1  662 1 1  194 1 1  230 1 1  526 1
1  122 1 1  129 1 1   74 1 1  122 1 1   86 1
1  466 1 1  192 1 1  109 1 1   55 1 1    1 1
1  107 1 1  110 1 1  332 1 2 2569 0 2 2506 0
2 2409 0 2 2218 0 2 1857 0 2 1829 0 2 1562 0
2 1470 0 2 1363 0 2 1030 0 2  860 0 2 1258 0
2 2246 0 2 1870 0 2 1799 0 2 1709 0 2 1674 0
2 1568 0 2 1527 0 2 1324 0 2  957 0 2  932 0
2  847 0 2  848 0 2 1850 0 2 1843 0 2 1535 0
2 1447 0 2 1384 0 2  414 1 2 2204 1 2 1063 1
2  481 1 2  105 1 2  641 1 2  390 1 2  288 1
2  421 1 2   79 1 2  748 1 2  486 1 2   48 1
2  272 1 2 1074 1 2  381 1 2   10 1 2   53 1
2   80 1 2   35 1 2  248 1 2  704 1 2  211 1
2  219 1 2  606 1 3 2640 0 3 2430 0 3 2252 0
3 2140 0 3 2133 0 3 1238 0 3 1631 0 3 2024 0
3 1345 0 3 1136 0 3  845 0 3  422 1 3  162 1
3   84 1 3  100 1 3    2 1 3   47 1 3  242 1
3  456 1 3  268 1 3  318 1 3   32 1 3  467 1
3   47 1 3  390 1 3  183 1 3  105 1 3  115 1
3  164 1 3   93 1 3  120 1 3   80 1 3  677 1
3   64 1 3  168 1 3   74 1 3   16 1 3  157 1
3  625 1 3   48 1 3  273 1 3   63 1 3   76 1
3  113 1 3  363 1
;
```

Klein and Moeschberger (1997, Section 4.4) describe in detail how to compute the Hall and Wellner (HW) and equal precision (EP) confidence bands. Now you can use the SURVIVAL statement in PROC LIFETEST to obtain these confidence bands. In the following code, PROC LIFETEST is invoked to compute the product-limit estimates of the disease-free survival. The SURVIVAL statement is included to create an output SAS data set (named Out1) that contains the survival function estimates and to plot them with the experimental graphics using the ODS. To obtain both the HW and EP confidence bands in the OUT= data set, you specify the CONFBAND=ALL option. The BANDMIN=100 and BANDMAX=600 options restrict the confidence bands for the survivor function $S(t)$ over the range $100 \leq t \leq 600$. The CONFTYPE=ASINSQRT option is specified to apply the arcsine-square root transform to the survivor function in computing the pointwise confidence intervals and the confidence bands. The experimental ODS graphics statement is specified to display the graphics using ODS. The specific plots to be displayed are specified by the PLOTS=(STRATUM, SURVIVAL, HWB) option, which includes a panel of plots for each stratum, a plot of the survivor functions estimates for all strata, and a plot of the Hall-Wellner bands for all strata. Since most of the events occur within 800 days, MAXTIME=800 is specified to restrict the display to such time.

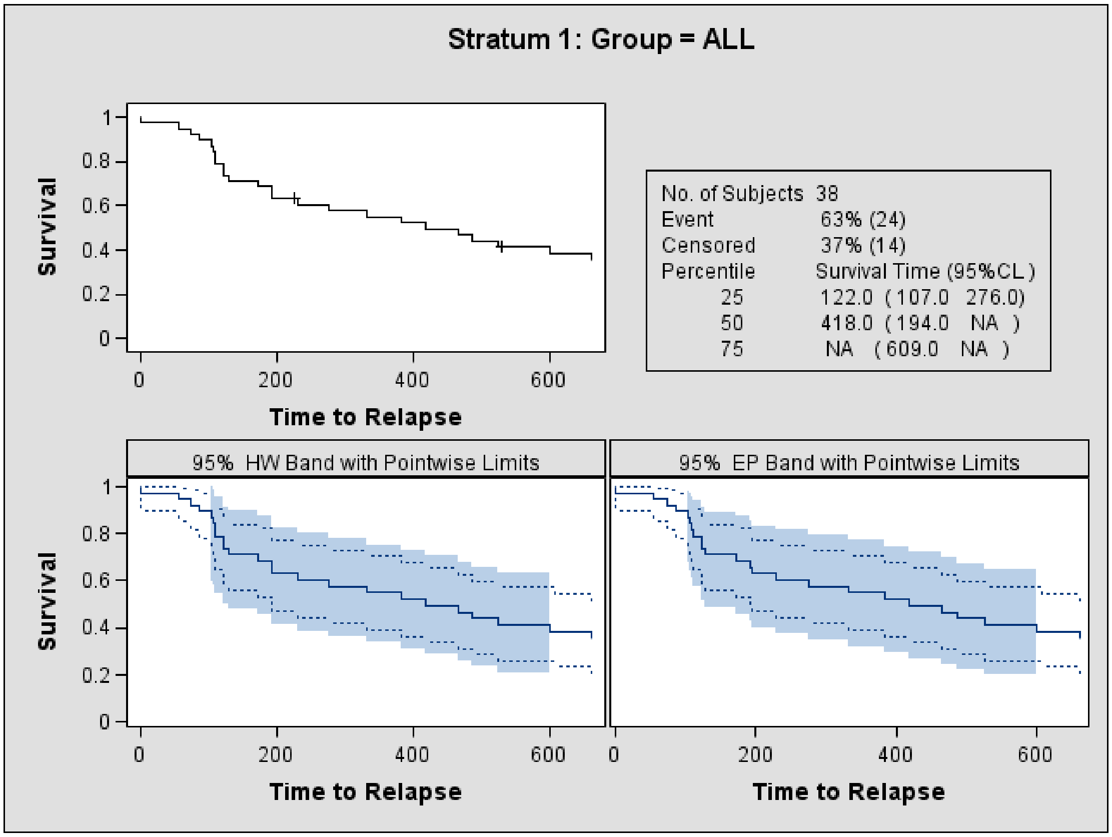
```
ods html;
ods graphics on;

proc lifetest data=BMT noprint;
   time T * Status(0);
   survival out=Out1
             confband=all bandmin=100 bandmax=600 maxtime=800
             conftype=asinsqrt
             plots=(stratum, survival, hwb);
  strata Group;
  run;

ods graphics off;
ods html close;

proc contents data=Out1;
   run;
```

The HW confidence bands for disease-free survival are represented by the variables HW_LCL and HW_UCL in the Out1 data set, and the EP confidence bands are represented by the variables EP_LCL and EP_UCL. Other variables in the Out1 data set are shown in the printed output of PROC CONTENTS in Output 40.2.1.

Output 40.2.1. Variables in the Out1 Data Set

```
                              The CONTENTS Procedure

                    Alphabetic List of Variables and Attributes

 #    Variable    Type    Len    Format    Label

 5    CONFTYPE    Char      8              Transform for Survival Confidence Interval
10    EP_LCL      Num       8              Equal Precision Band Lower 95.00% Limit
11    EP_UCL      Num       8              Equal Precision Band Upper 95.00% Limit
 1    Group       Num       8    RISK.
 8    HW_LCL      Num       8              Hall-Wellner Band Lower 95.00% Limit
 9    HW_UCL      Num       8              Hall-Wellner Band Upper 95.00% Limit
 6    SDF_LCL     Num       8              SDF Lower 95.00% Confidence Limit
 7    SDF_UCL     Num       8              SDF Upper 95.00% Confidence Limit
12    STRATUM     Num       8              Stratum Number
 4    SURVIVAL    Num       8              Survival Distribution Function Estimate
 2    T           Num       8              Time to Relapse
 3    _CENSOR_    Num       8              Censoring Flag: 0=Failed 1=Censored
```

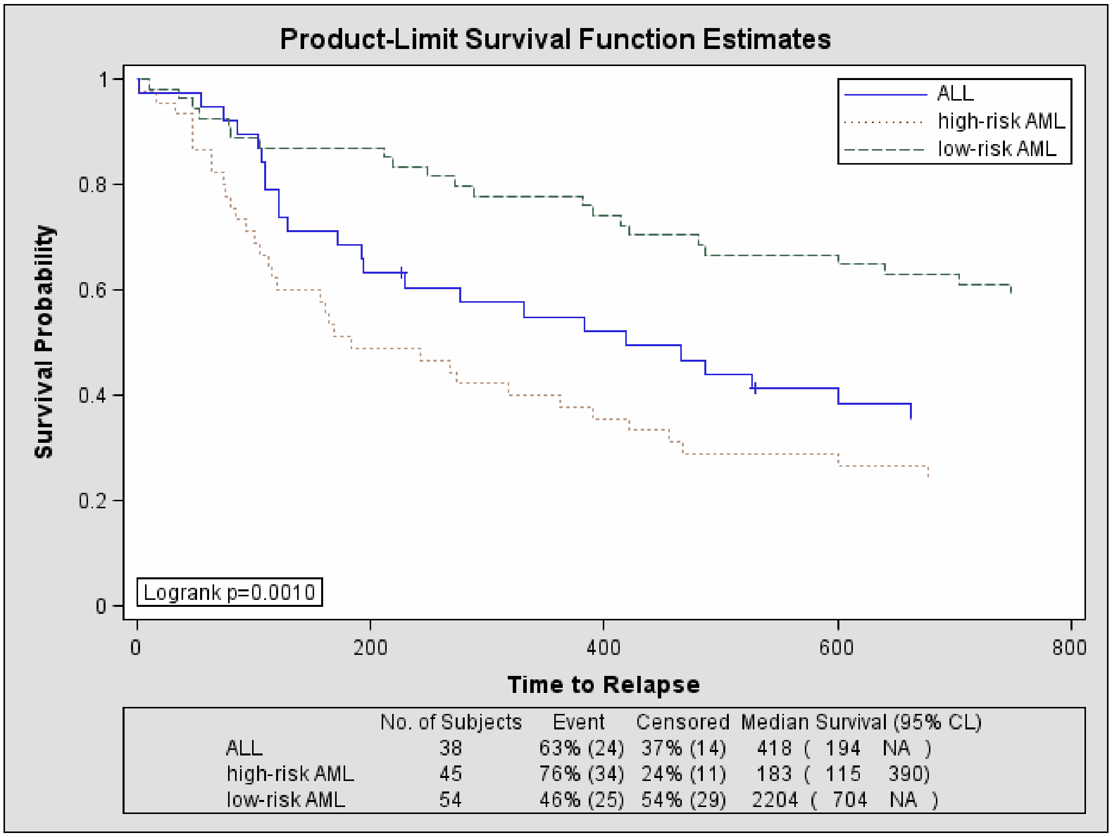
Output 40.2.2. Panel Plot for ALL Patients (Experimental)



```
No. of Subjects  38
Event             63% (24)
Censored          37% (14)
Percentile       Survival Time (95%CL)
      25          122.0  (107.0   276.0)
      50          418.0  (194.0   NA  )
      75           NA    (609.0   NA  )
```

The panel plot for the ALL patients is shown in Output 40.2.2. The upper left cell in this 2×2 matrix plot displays the estimated survival curve, the upper right cell shows a table of summary statistics, the lower left cell displays the Hall-Wellner band, and the lower right cell displays the equal precision band. Pointwise confidence limits are plotted along with the confidence bands. As expected, the confidence bands are much wider than the set of the pointwise confidence intervals. The EP confidence band appears to be slightly wider than the HW confidence band for survival time over 200 days. The panel plots for the low-risk and high-risk AML patients are not shown here.

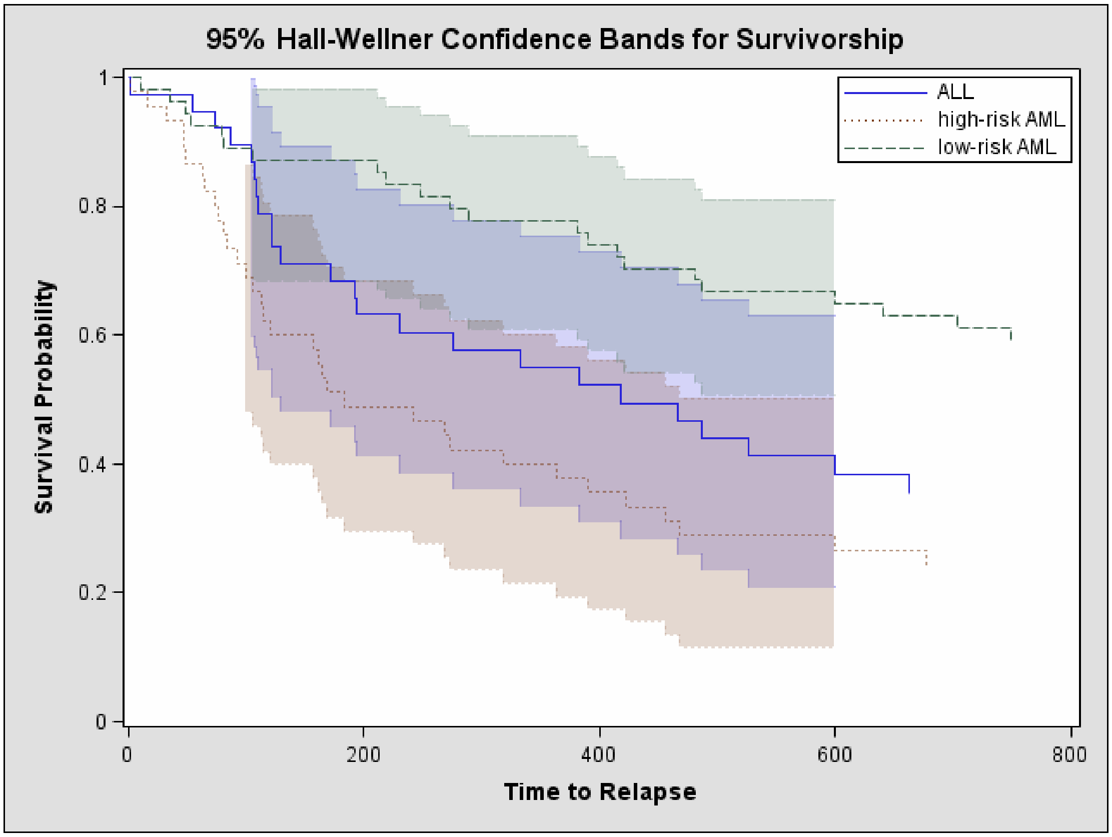
Output 40.2.3. Product-Limit Estimates of Survival of Bone Marrow Transplant Patients (Experimental)



```
                 No. of Subjects   Event     Censored  Median Survival (95% CL)
ALL                    38        63% (24)  37% (14)     418 (  194   NA  )
high-risk AML          45        76% (34)  24% (11)     183 (  115   390)
low-risk AML           54        46% (25)  54% (29)    2204 (  704   NA  )
```

Output 40.2.3 shows a plot of the estimated survival curves for the three leukemia groups. Censored observations are plotted as a plus sign. It appears that the low-risk AML patients have the best prognosis while the high-risk AML patients have the worse prognosis, with the ALL patients in between. Output 40.2.4 shows a plot of the Hall-Wellner bands for the three leukemia groups. The band for the ALL patients overlaps with those of the low-risk and high-risk AML patients, but there is very little overlapping between the band for the low-risk AML patients and the band for the high-risk patients. One would expect the low-risk AML patients to live much longer than the high-risk AML patients.

Output 40.2.4. Hall-Wellner Bands for the Survival of Bone Marrow Transplant Patients (Experimental)



The graphical display in Output 40.2.2 as well as those shown in Output 40.2.3 and Output 40.2.4 are requested by specifying the experimental ODS GRAPHICS statement and the experimental PLOTS= option in the SURVIVAL statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the LIFETEST procedure, see the section "ODS Graphics" on page 2190.

Example 40.3. Life-Table Estimates for Males with Angina Pectoris

The data in this example come from Lee (1992, p. 91) and represent the survival rate of males with angina pectoris. Survival time is measured as years from the time of diagnosis. The data are read as number of events and number of withdrawals in each one-year time interval for 16 intervals. Three variables are constructed from the data: Years (an artificial time variable with values that are the midpoints of the time intervals), Censored (a censoring indicator variable with value 1 indicating censored observations and value 0 indicating event observations), and Freq (the frequency variable). Two observations are created for each interval, one representing the event observations and the other representing the censored observations.

```
title 'Survival of Males with Angina Pectoris';
data males;
   keep Freq Years Censored;
   retain Years -.5;
   input fail withdraw @@;
   Years + 1;
   Censored=0;
   Freq=fail;
   output;
   Censored=1;
   Freq=withdraw;
   output;
   datalines;
456   0 226  39 152  22 171  23 135 24 125 107
 83 133  74 102  51  68  42  64  43 45  34  53
 18  33   9  27   6  23   0  30
;
```

PROC LIFETEST is invoked to compute the various life-table survival estimates, the median residual time, and their standard errors. The life-table method of computing estimates is requested by specifying METHOD=LT. The intervals are specified by the INTERVAL= option. Traditional high-resolution graphs of the life-table estimate, negative log of the estimate, negative log-log of the estimate, estimated density function, and estimated hazard function are requested by the PLOTS= option. No tests for homogeneity are carried out because the data are not stratified.

```
symbol1 c=blue;
proc lifetest data=males  method=lt intervals=(0 to 15 by 1)
              plots=(s,ls,lls,h,p);
   time Years*Censored(1);
   freq Freq;
run;
```

Results of the life-table estimation are shown in Output 40.3.1. The five-year survival rate is 0.5193 with a standard error of 0.0103. The estimated median residual lifetime, which is 5.33 years initially, has reached a maximum of 6.34 years at the beginning of the second year and decreases gradually to a value lower than the initial 5.33 years at the beginning of the seventh year.

Output 40.3.1. Life-Table Survivor Function Estimate

```
                                         Survival of Males with Angina Pectoris

                                             Life Table Survival Estimates

                                                                Conditional
                                            Effective  Conditional  Probability                         Survival   Median
         Interval          Number   Number     Sample  Probability     Standard                         Standard  Residual
     [Lower,     Upper)    Failed  Censored      Size   of Failure        Error    Survival    Failure     Error  Lifetime

           0          1       456        0     2418.0       0.1886      0.00796      1.0000          0         0    5.3313
           1          2       226       39     1942.5       0.1163      0.00728      0.8114     0.1886   0.00796    6.2499
           2          3       152       22     1686.0       0.0902      0.00698      0.7170     0.2830   0.00918    6.3432
           3          4       171       23     1511.5       0.1131      0.00815      0.6524     0.3476   0.00973    6.2262
           4          5       135       24     1317.0       0.1025      0.00836      0.5786     0.4214    0.0101    6.2185
           5          6       125      107     1116.5       0.1120      0.00944      0.5193     0.4807    0.0103    5.9077
           6          7        83      133      871.5       0.0952      0.00994      0.4611     0.5389    0.0104    5.5962
           7          8        74      102      671.0       0.1103       0.0121      0.4172     0.5828    0.0105    5.1671
           8          9        51       68      512.0       0.0996       0.0132      0.3712     0.6288    0.0106    4.9421
           9         10        42       64      395.0       0.1063       0.0155      0.3342     0.6658    0.0107    4.8258
          10         11        43       45      298.5       0.1441       0.0203      0.2987     0.7013    0.0109    4.6888
          11         12        34       53      206.5       0.1646       0.0258      0.2557     0.7443    0.0111         .
          12         13        18       33      129.5       0.1390       0.0304      0.2136     0.7864    0.0114         .
          13         14         9       27       81.5       0.1104       0.0347      0.1839     0.8161    0.0118         .
          14         15         6       23       47.5       0.1263       0.0482      0.1636     0.8364    0.0123         .
          15          .         0       30       15.0            0            0      0.1429     0.8571    0.0133         .

                                                       Evaluated at the Midpoint of the Interval

                                                 Median                  PDF                 Hazard
                         Interval              Standard                Standard              Standard
                    [Lower,     Upper)            Error        PDF       Error     Hazard       Error

                          0          1           0.1749     0.1886     0.00796   0.208219    0.009698
                          1          2           0.2001     0.0944     0.00598   0.123531    0.008201
                          2          3           0.2361     0.0646     0.00507    0.09441    0.007649
                          3          4           0.2361     0.0738     0.00543   0.119916    0.009154
                          4          5           0.1853     0.0593     0.00495   0.108043    0.009285
                          5          6           0.1806     0.0581     0.00503   0.118596    0.010589
                          6          7           0.1855     0.0439     0.00469        0.1    0.010963
                          7          8           0.2713     0.0460     0.00518   0.116719    0.013545
                          8          9           0.2763     0.0370     0.00502    0.10483    0.014659
                          9         10           0.4141     0.0355     0.00531   0.112299    0.017301
                         10         11           0.4183     0.0430     0.00627   0.155235    0.023602
                         11         12                .     0.0421     0.00685    0.17942    0.030646
                         12         13                .     0.0297     0.00668   0.149378     0.03511
                         13         14                .     0.0203     0.00651   0.116883    0.038894
                         14         15                .     0.0207     0.00804   0.134831    0.054919
                         15          .                .          .           .          .           .
```

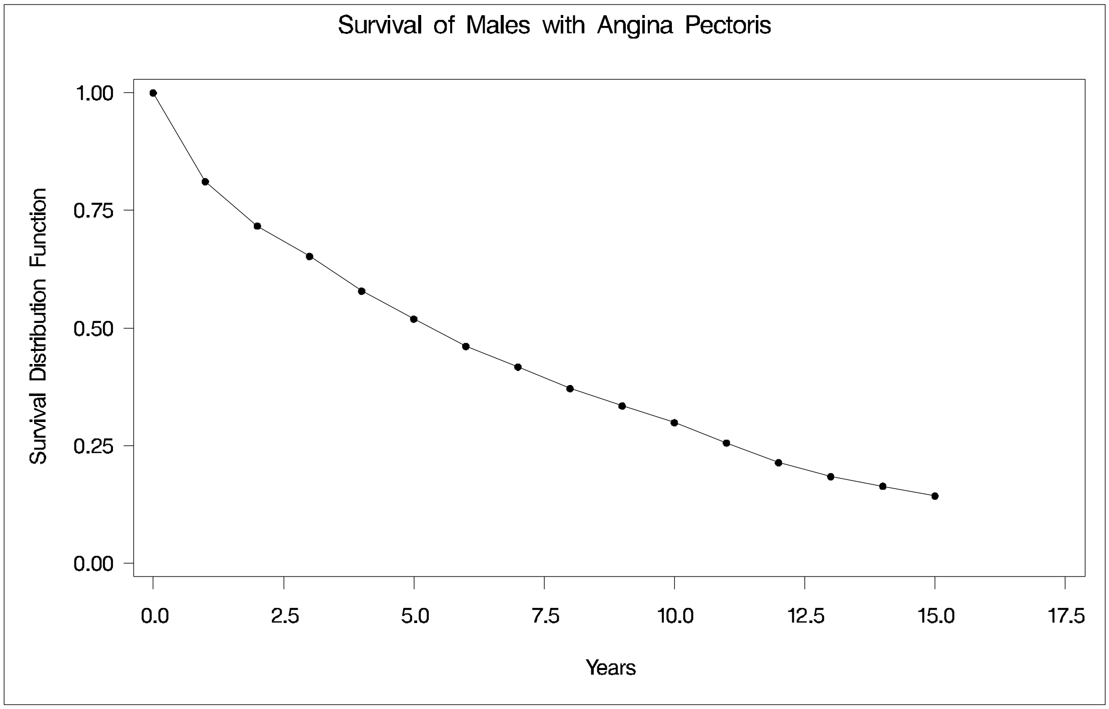
The breakdown of event and censored observation in the data is shown in Output 40.3.2. Note that 32.8% of the patients have withdrawn from the study.

Output 40.3.2. Summary of Censored and Event Observations

```
                      Survival of Males with Angina Pectoris

                Summary of the Number of Censored and Uncensored Values

                                                       Percent
                          Total  Failed    Censored    Censored

                           2418    1625         793       32.80

NOTE: There were 2 observations with missing values, negative time values or frequency values less than 1.
```

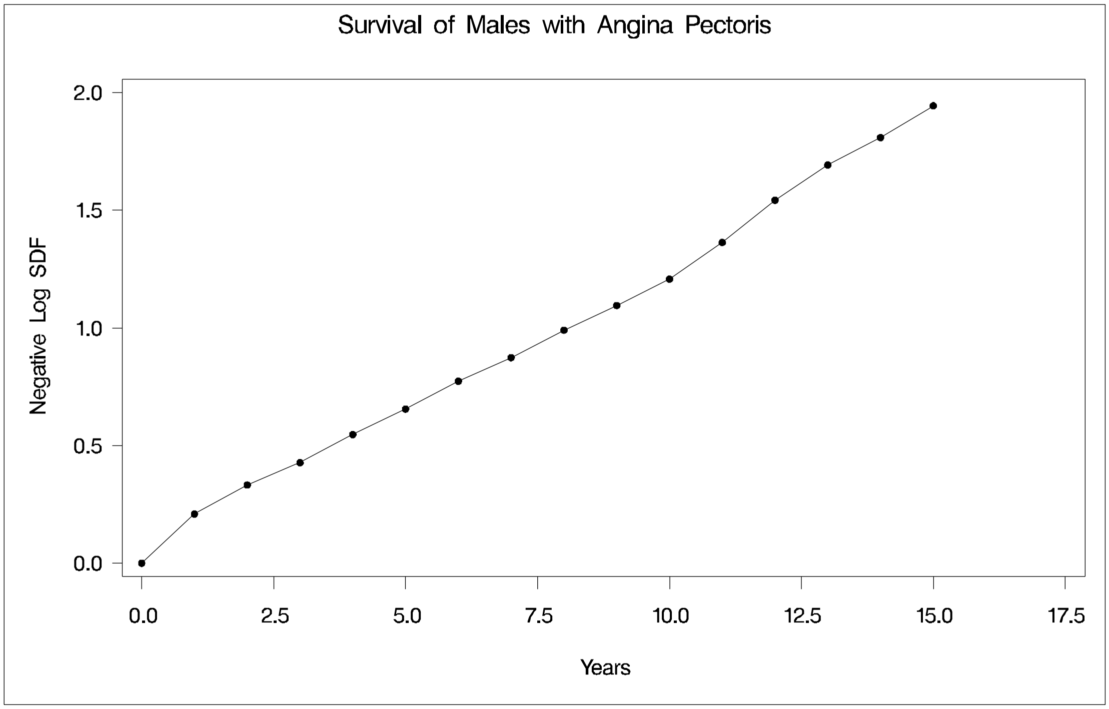
Output 40.3.3 displays the graph of the life-table survivor function estimate. The median survival time, read from the survivor function curve, is 5.33 years, and the 25th and 75th percentiles are 1.04 and 11.13 years, respectively.

Output 40.3.3. Life-Table Survivor Function Estimate



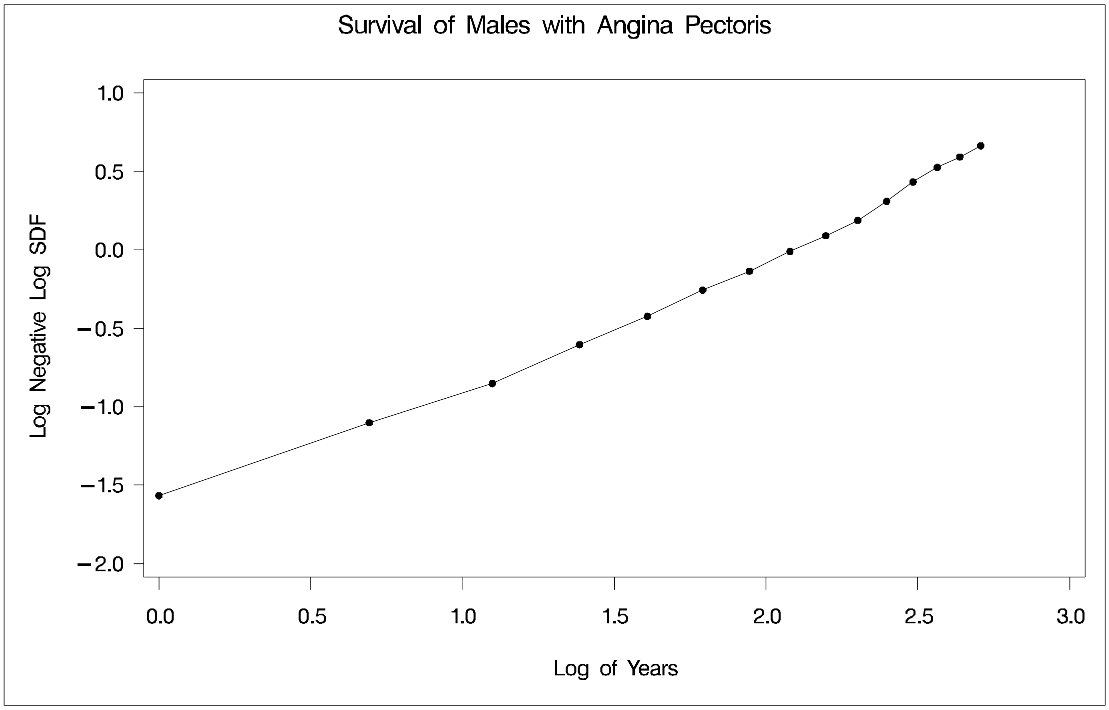
An exponential model may be appropriate for the survival of these male patients with angina pectoris since the curve of the negative log of the survivor function estimate versus the survival time (Output 40.3.4) approximates a straight line through the origin. Note that the graph of the log of the negative log of the survivor function estimate versus the log of time (Output 40.3.5) is practically a straight line.

Output 40.3.4. Negative Log of Survivor Function Estimate

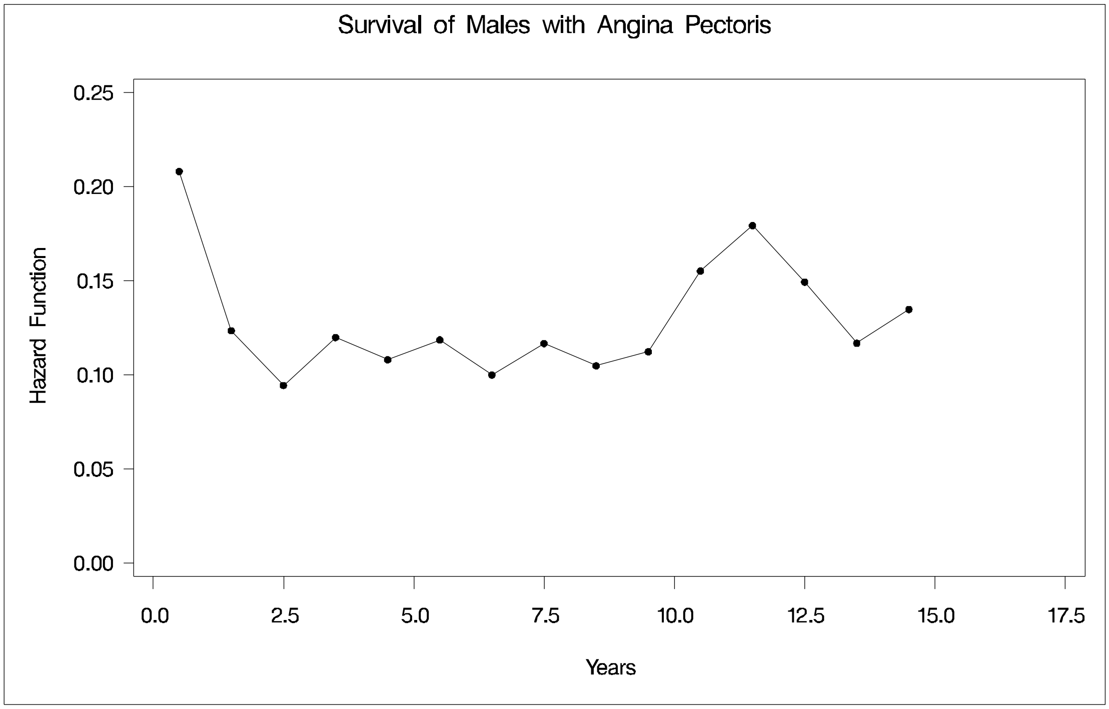


As discussed in Lee (1992), the graph of the estimated hazard function (Output 40.3.6) shows that the death rate is highest in the first year of diagnosis. From the end of the first year to the end of the tenth year, the death rate remains relatively constant, fluctuating between 0.09 and 0.12. The death rate is generally higher after the tenth year. This could indicate that a patient who has survived the first year has a better chance than a patient who has just been diagnosed. The profile of the median residual lifetimes also supports this interpretation.

Output 40.3.5. Log of Negative Log of Survivor Function Estimate

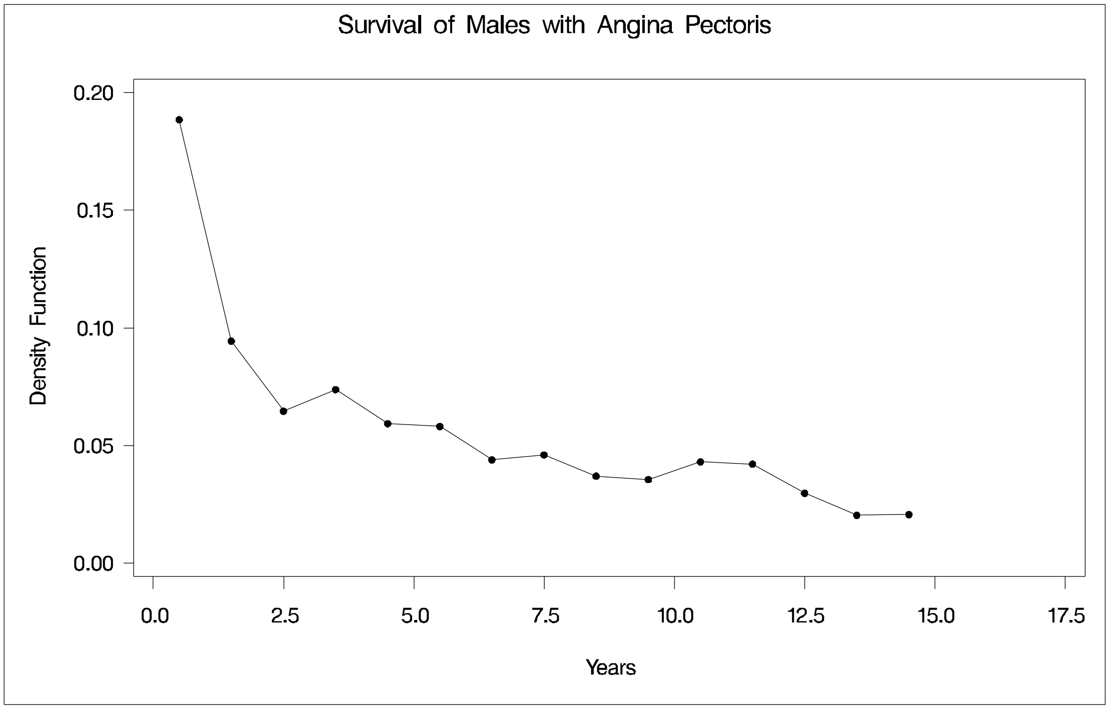


Output 40.3.6. Hazard Function Estimate



The density estimate is shown in (Output 40.3.7). Visually, it resembles that of an exponential distribution.

Output 40.3.7. Density Function Estimate



References

Andersen, P. K., Borgan, Ø., Gill, R. D., and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, New York: Springer-Verlag.

Borgan, Ø. and Liestøl, K. (1990), "A Note on Confidence Interval and Bands for the Survival Curves Based on Transformations," *Scandinavian Journal of Statistics*, 18, 35–41.

Brookmeyer, R. and Crowley, J. (1982), "A Confidence Interval for the Median Survival Time," *Biometrics*, 38, 29–41.

Chung, C. F. (1986), "Formulae for Probabilities Associated with Wiener and Brownian Bridge Processes," Technical Report 79, Laboratory for Research in Statistics and Probability, Ottawa, Canada: Carleton University.

Collett, D. (1994), *Modeling Survival Data in Medical Research*, London: Chapman and Hall.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman and Hall.

Elandt-Johnson, R. C. and Johnson, N. L. (1980), *Survival Models and Data Analysis*, New York: John Wiley & Sons, Inc.

Fleming, T. and Harrington, D. (1981), "A Class of Hypothesis Tests for One and Two Samples of Censored Survival Data," *Communications in Statistics*, 10, 763–794.

Fleming, T. R. and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: John Wiley & Sons, Inc.

Hall, W. J. and Wellner, J. A. (1980), "Confidence Bands for a Survival Curve for Censored Data," *Biometrika 69*.

Harrington, D. and Fleming, T. (1982), "A Class of Rank Test Procedures for Censored Survival Data," *Biometrika*, 69, 133–143.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons, Inc.

Klein, J. P. and Moeschberger, M. L. (1997), *Survival Analysis: Techniques for Censored and Truncated Data*, New York: Springer-Verlag.

Lachin, J. (2000), *Biostatistical Methods: The Assessment of Relative Risks*, New York: John Wiley & Sons, Inc.

Lawless, J. F. (1982), *Statistical Methods and Methods for Lifetime Data*, New York: John Wiley & Sons, Inc.

Lee, E. T. (1992), *Statistical Methods for Survival Data Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Meeker, W. Q. and Escobar, L. A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Miller, R. G. and Siegmund, D. (1982), "Maximally Selected Chi-Square Statistics," *Biometrics*, 1011–1016.

Nair, V. N. (1984), "Confidence Bands for Survival Functions with Censored Data: A Comparative Study," *Technometrics*, 26, 265–275.

Chapter 41
The LOESS Procedure

Chapter Contents

Chapter 41
The LOESS Procedure

Overview

The LOESS procedure implements a nonparametric method for estimating regression surfaces pioneered by Cleveland, Devlin, and Grosse (1988), Cleveland and Grosse (1991), and Cleveland, Grosse, and Shyu (1992). the LOESS procedure allows great flexibility because no assumptions about the parametric form of the regression surface are needed.

The SAS System provides many regression procedures such as the GLM, REG, and NLIN procedures for situations in which you can specify a reasonable parametric model for the regression surface. You can use the LOESS procedure for situations in which you do not know a suitable parametric form of the regression surface. Furthermore, the LOESS procedure is suitable when there are outliers in the data and a robust fitting method is necessary.

The main features of the LOESS procedure are as follows:

- fits nonparametric models
- supports the use of multidimensional data
- supports multiple dependent variables
- supports both direct and interpolated fitting using kd trees
- performs statistical inference
- performs automatic smoothing parameter selection
- performs iterative reweighting to provide robust fitting when there are outliers in the data
- supports multiple SCORE statements

Experimental graphics are now available with the LOESS procedure. For more information, refer to the "ODS Graphics" section on page 2248.

Local Regression and the Loess Method

Assume that for $i = 1$ to n, the ith measurement y_i of the response y and the corresponding measurement x_i of the vector x of p predictors are related by

$$y_i = g(x_i) + \epsilon_i$$

where g is the regression function and ϵ_i is a random error. The idea of local regression is that at a predictor x, the regression function $g(x)$ can be locally approximated by the value of a function in some specified parametric class. Such a local approximation is obtained by fitting a regression surface to the data points within a chosen neighborhood of the point x.

In the loess method, weighted least squares is used to fit linear or quadratic functions of the predictors at the centers of neighborhoods. The radius of each neighborhood is chosen so that the neighborhood contains a specified percentage of the data points. The fraction of the data, called the *smoothing parameter,* in each local neighborhood controls the smoothness of the estimated surface. Data points in a given local neighborhood are weighted by a smooth decreasing function of their distance from the center of the neighborhood.

In a direct implementation, such fitting is done at each point at which the regression surface is to be estimated. A much faster computational procedure is to perform such local fitting at a selected sample of points in predictor space and then to blend these local polynomials to obtain a regression surface.

You can use the LOESS procedure to perform statistical inference provided the error distribution satisfies some basic assumptions. In particular, such analysis is appropriate when the ϵ_i are i.i.d. normal random variables with mean 0. By using the iterative reweighting, the LOESS procedure can also provide statistical inference when the error distribution is symmetric but not necessarily normal. Furthermore, by doing iterative reweighting, you can use the LOESS procedure to perform robust fitting in the presence of outliers in the data.

While all output of the LOESS procedure can be optionally displayed, most often the LOESS procedure is used to produce output data sets that will be viewed and manipulated by other SAS procedures. PROC LOESS uses the Output Delivery System (ODS) to place results in output data sets. This is a departure from older SAS procedures that provide OUTPUT statements to create SAS data sets from analysis results.

Getting Started

Scatter Plot Smoothing

The following data from the Connecticut Tumor Registry presents age-adjusted numbers of melanoma incidences per 100,000 people for 37 years from 1936 to 1972 (Houghton, Flannery, and Viola 1980).

```
data Melanoma;
   input  Year Incidences @@;
   format Year d4.0;
   format DepVar d4.1;
datalines;
1936    0.9   1937   0.8  1938   0.8  1939   1.3
1940    1.4   1941   1.2  1942   1.7  1943   1.8
1944    1.6   1945   1.5  1946   1.5  1947   2.0
1948    2.5   1949   2.7  1950   2.9  1951   2.5
1952    3.1   1953   2.4  1954   2.2  1955   2.9
1956    2.5   1957   2.6  1958   3.2  1959   3.8
```

```
1960    4.2   1961   3.9  1962   3.7  1963   3.3
1964    3.7   1965   3.9  1966   4.1  1967   3.8
1968    4.7   1969   4.4  1970   4.8  1971   4.8
1972    4.8
;
```

The following PROC GPLOT statements produce the simple scatter plot of these data displayed in Figure 41.1.

```
symbol1 color=black value=dot ;
proc gplot data=Melanoma;
   title1 'Scatter Plot of Melanoma Data';
   plot Incidences*Year;
run;
```

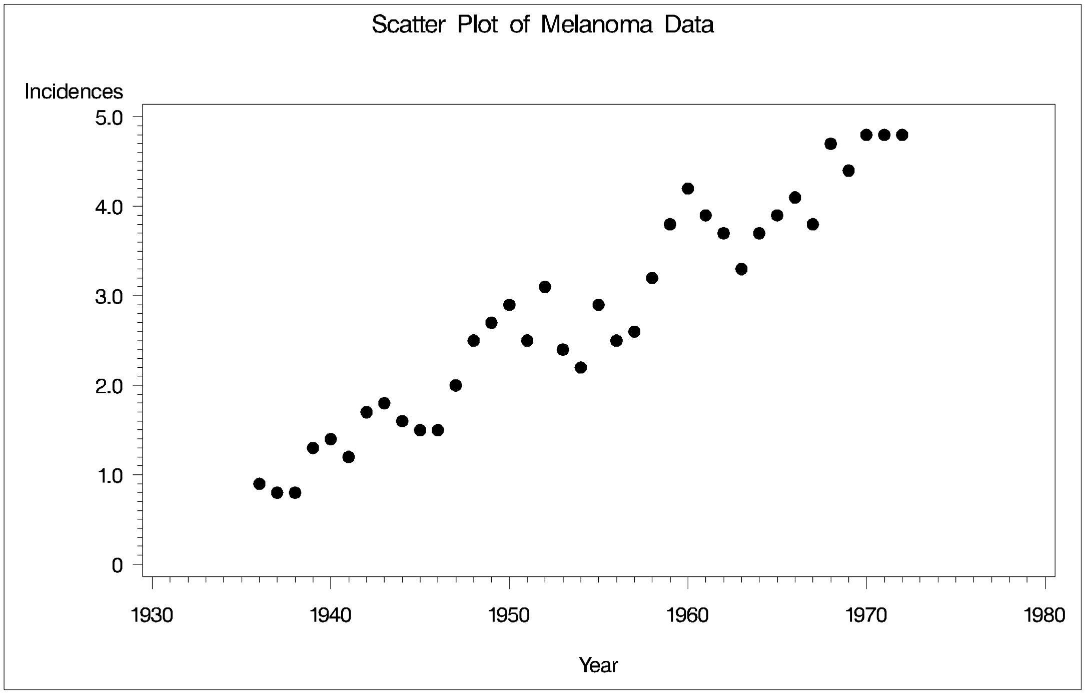


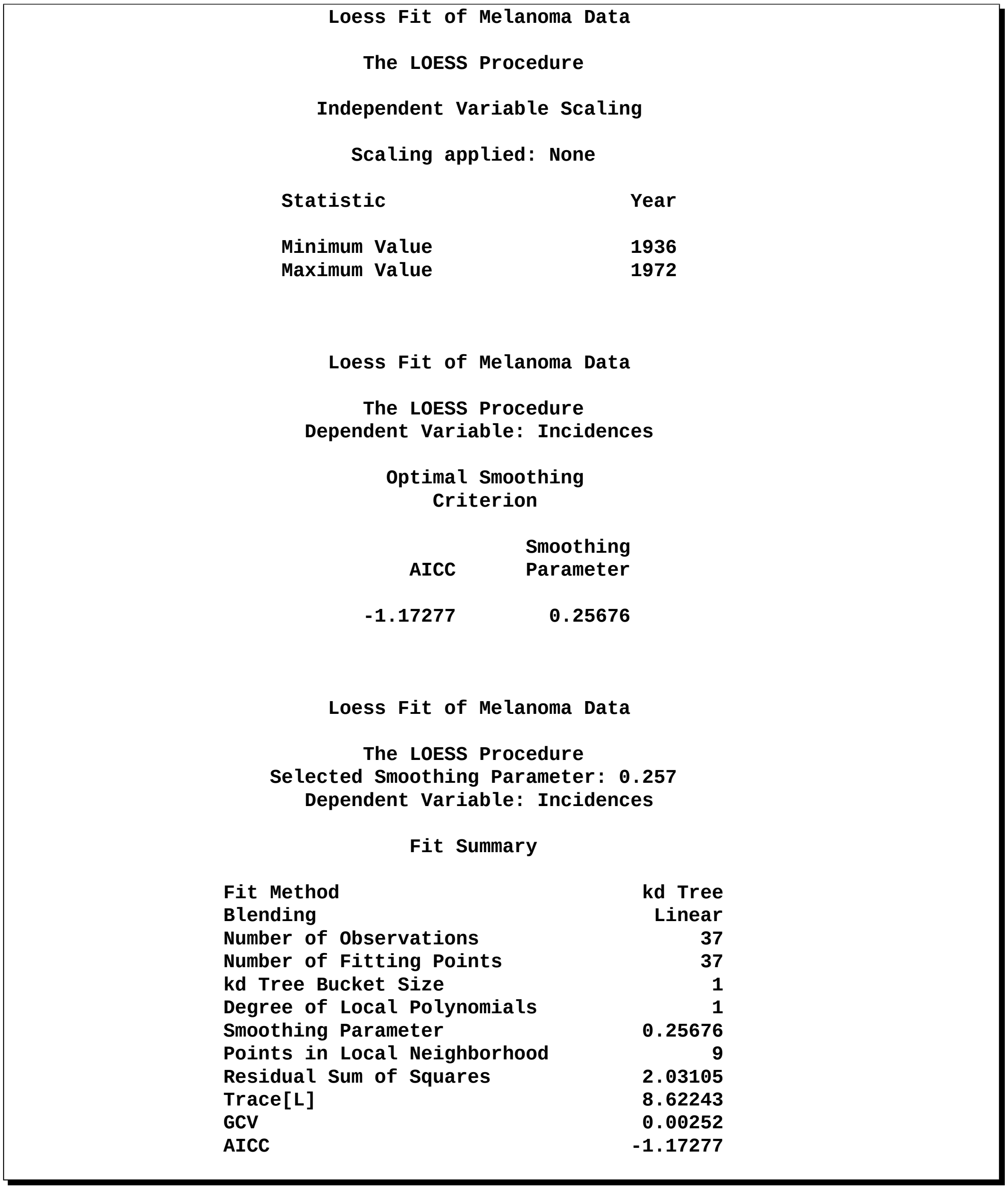
Figure 41.1. Scatter Plot of Incidences versus Year for the Melanoma Data

Suppose that you want to smooth the response variable Incidences as a function of the variable Year. The following PROC LOESS statements request this analysis:

```
proc loess data=Melanoma;
   model Incidences=Year/details(OutputStatistics);
run;
```

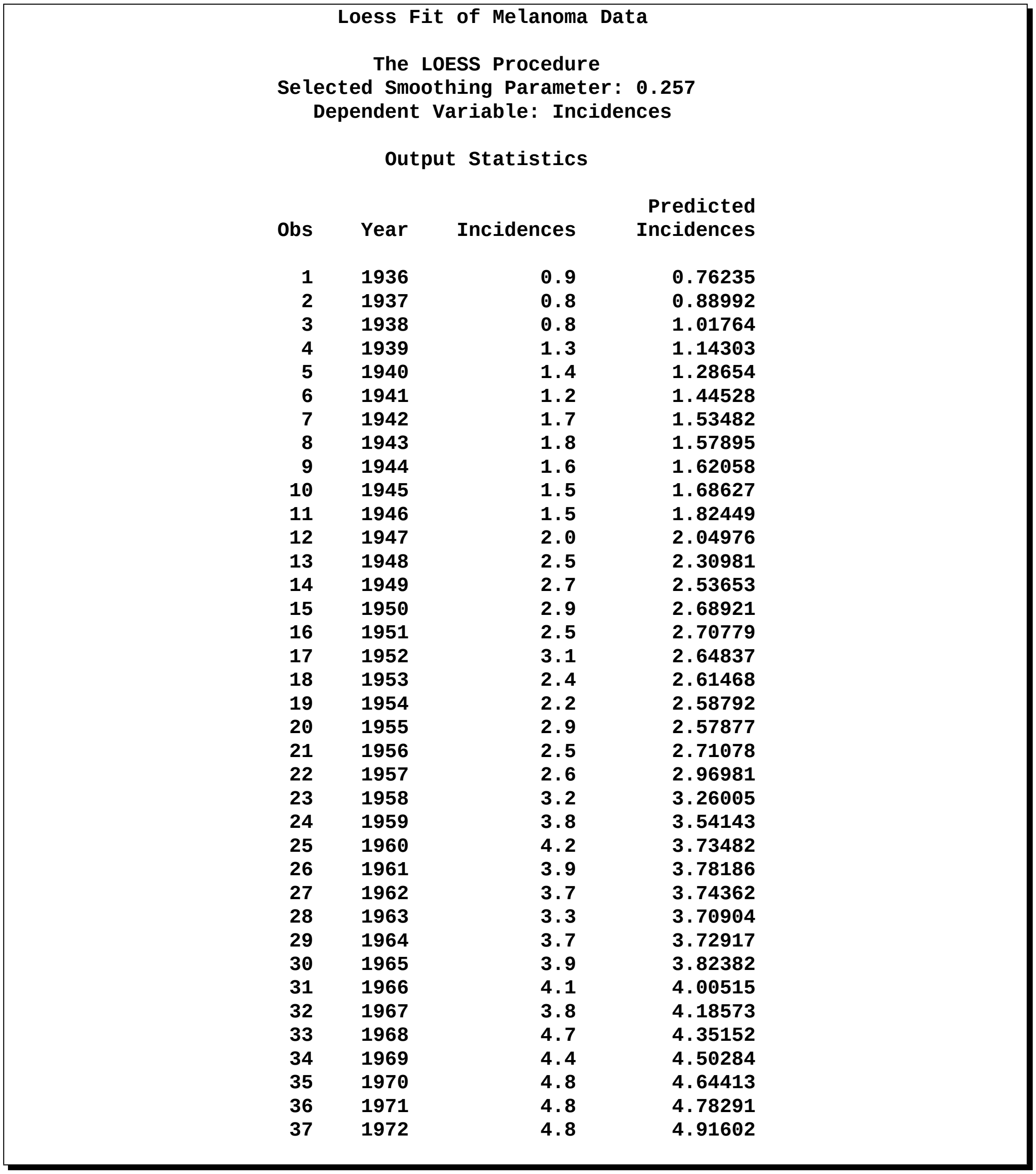
You use the PROC LOESS statement to invoke the procedure and specify the data set. The MODEL statement names the dependent and independent variables. You use options in the MODEL statement to specify fitting parameters and control the displayed output. For example, the MODEL statement option DETAILS(OutputStatistics) requests that the “Output Statistics” table be included in the displayed output. By default, this table is not displayed.

The results are displayed in Figure 41.2 and Figure 41.3.



```
                    Loess Fit of Melanoma Data

                       The LOESS Procedure

                  Independent Variable Scaling

                     Scaling applied: None

              Statistic                      Year

              Minimum Value                  1936
              Maximum Value                  1972



                    Loess Fit of Melanoma Data

                       The LOESS Procedure
                  Dependent Variable: Incidences

                         Optimal Smoothing
                             Criterion

                                    Smoothing
                          AICC      Parameter

                      -1.17277        0.25676



                    Loess Fit of Melanoma Data

                       The LOESS Procedure
             Selected Smoothing Parameter: 0.257
                  Dependent Variable: Incidences

                          Fit Summary

         Fit Method                                kd Tree
         Blending                                   Linear
         Number of Observations                         37
         Number of Fitting Points                       37
         kd Tree Bucket Size                             1
         Degree of Local Polynomials                     1
         Smoothing Parameter                       0.25676
         Points in Local Neighborhood                    9
         Residual Sum of Squares                   2.03105
         Trace[L]                                  8.62243
         GCV                                       0.00252
         AICC                                     -1.17277
```

Figure 41.2. Output from PROC LOESS



```
                   Loess Fit of Melanoma Data

                      The LOESS Procedure
               Selected Smoothing Parameter: 0.257
                  Dependent Variable: Incidences

                       Output Statistics

                                            Predicted
               Obs    Year    Incidences    Incidences

                 1    1936           0.9       0.76235
                 2    1937           0.8       0.88992
                 3    1938           0.8       1.01764
                 4    1939           1.3       1.14303
                 5    1940           1.4       1.28654
                 6    1941           1.2       1.44528
                 7    1942           1.7       1.53482
                 8    1943           1.8       1.57895
                 9    1944           1.6       1.62058
                10    1945           1.5       1.68627
                11    1946           1.5       1.82449
                12    1947           2.0       2.04976
                13    1948           2.5       2.30981
                14    1949           2.7       2.53653
                15    1950           2.9       2.68921
                16    1951           2.5       2.70779
                17    1952           3.1       2.64837
                18    1953           2.4       2.61468
                19    1954           2.2       2.58792
                20    1955           2.9       2.57877
                21    1956           2.5       2.71078
                22    1957           2.6       2.96981
                23    1958           3.2       3.26005
                24    1959           3.8       3.54143
                25    1960           4.2       3.73482
                26    1961           3.9       3.78186
                27    1962           3.7       3.74362
                28    1963           3.3       3.70904
                29    1964           3.7       3.72917
                30    1965           3.9       3.82382
                31    1966           4.1       4.00515
                32    1967           3.8       4.18573
                33    1968           4.7       4.35152
                34    1969           4.4       4.50284
                35    1970           4.8       4.64413
                36    1971           4.8       4.78291
                37    1972           4.8       4.91602
```

Figure 41.3. Output from PROC LOESS continued

Usually, such displayed results are of limited use. Most frequently the results are needed in an output data set so that they can be displayed graphically and analyzed further. For example, to place the "Output Statistics" table shown in Figure 41.3 in an output data set, you use the ODS OUTPUT statement as follows:

```
proc loess data=Melanoma;
   model Incidences=Year;
   ods output OutputStatistics=Results;
run;
```

The statement

```
    ods output OutputStatistics=Results;
```

requests that the “Output Statistics” table that appears in Figure 41.2 be placed in a SAS data set named Results. Note also that the DETAILS(OutputStatistics) option that caused this table to be included in the displayed output need not be specified.

The PRINT procedure displays the first five observations of this data set:

```
title1 ’First 5 Observations of the Results Data Set’;
proc print data=Results(obs=5);
  id obs;
run;
```

```
                  First 5 Observations of the Results Data Set

                                      Dep
                         Obs    Year   Var           Pred

                           1    1936   0.9        0.76235
                           2    1937   0.8        0.88992
                           3    1938   0.8        1.01764
                           4    1939   1.3        1.14303
                           5    1940   1.4        1.28654
```

Figure 41.4. PROC PRINT Output of the Results Data Set

You can now produce a scatter plot including the fitted loess curve as follows:

```
symbol1 color=black value=dot;
symbol2 color=black interpol=join value=none;

/* macro used in subsequent examples */
%let opts=vaxis=axis1 hm=3 vm=3 overlay;

axis1 label=(angle=90 rotate=0);

proc gplot data=Results;
  title1 ’Melanoma Data with Default LOESS Fit’;
  plot  DepVar*Year Pred*Year/ &opts;
run;
```

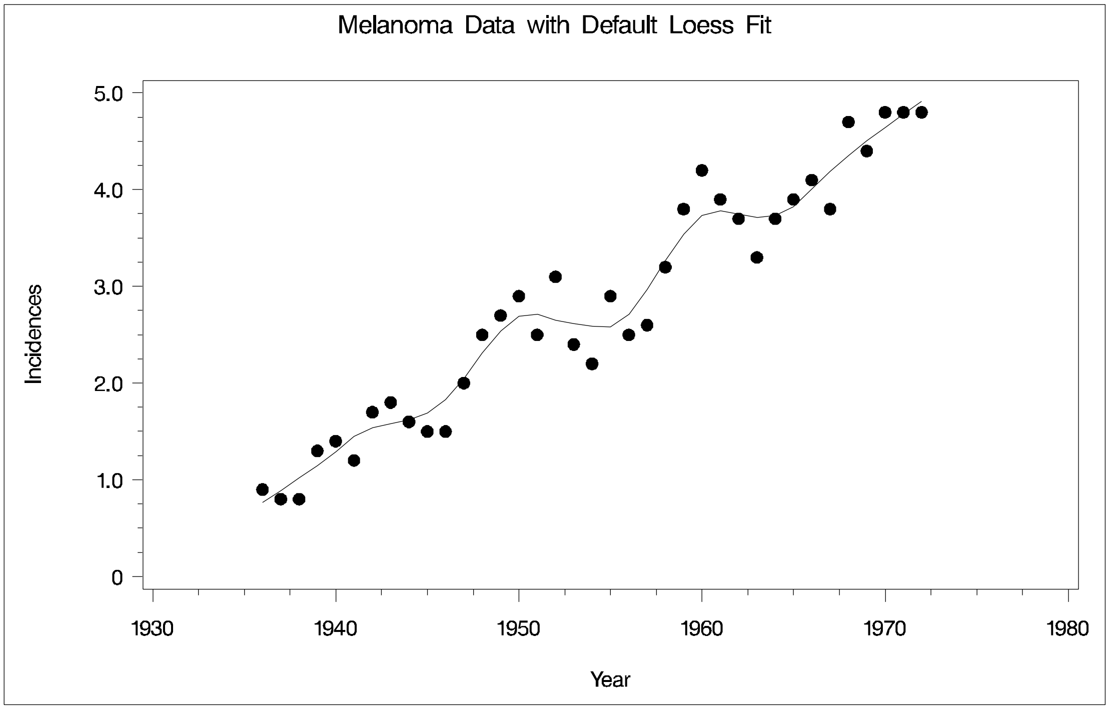


Figure 41.5. Default Loess Fit for Melanoma Data

The smoothing parameter value used in the loess fit shown in Figure 41.5 was obtained with a default method of selecting this value. This method minimizes a bias corrected AIC criterion (Hurvich, Simonoff, and Tsai 1998), which balances the residual sum of squares against the smoothness of the fit.

You can find the selected smoothing parameter value in the "Smoothing Criterion" table shown in Figure 41.2. Note that with this smoothing parameter value, the loess fit captures the increasing trend in the data as well the periodic pattern in the data, which is related to an 11-year sunspot activity cycle.

You can obtain a summary of all the models that PROC LOESS evaluated in choosing this smoothing parameter value in the "Model Summary" table. You request this optionally displayed table by adding the choice, ModelSummary, in the DETAILS option in the model statement as follows:

```
proc loess data=Melanoma;
   model Incidences=Year/details(OutputStatistics ModelSummary);
   ods output OutputStatistics=Results;
run;
```

Note that this example shows that you can request more than one optional table using the DETAILS option. The requested "Model Summary" table is shown in Figure 41.6

```
                          Loess Fit of Melanoma Data

                             The LOESS Procedure
                        Dependent Variable: Incidences

                                Model Summary

    Smoothing       Local
    Parameter      Points    Residual SS           GCV            AICC

      0.41892          15        3.42229       0.00339        -0.96252
      0.68919          25        4.05838       0.00359        -0.93459
      0.31081          11        2.51054       0.00279        -1.12034
      0.20270           7        1.58513       0.00239        -1.12221
      0.17568           6        1.56896       0.00241        -1.09706
      0.28378          10        2.50487       0.00282        -1.10402
      0.20270           7        1.58513       0.00239        -1.12221
      0.25676           9        2.03105       0.00252        -1.17277
      0.22973           8        2.02965       0.00256        -1.15145
      0.25676           9        2.03105       0.00252        -1.17277
```

Figure 41.6. Model Summary Table

Rather than use an automatic method for selecting the smoothing parameter value, you may want to examine loess fits for a range of values. You do this by using the SMOOTH= option in the MODEL statement as follows:

```
proc loess data=Melanoma;
   model Incidences=Year/smooth=0.1 0.2 0.3 0.4 residual;
   ods output OutputStatistics=Results;
run;
```

The RESIDUAL option causes the residuals to be added to the "Output Statistics" table. PROC PRINT displays the first five observations of this data set:

```
proc print data=Results(obs=5);
  id obs;
run;
```

```
                   First 5 Observations of the Results Data Set

                   Smoothing             Dep
           Obs     Parameter     Year    Var           Pred       Residual

             1        0.1        1936    0.9        0.90000              0
             2        0.1        1937    0.8        0.80000              0
             3        0.1        1938    0.8        0.80000              0
             4        0.1        1939    1.3        1.30000              0
             5        0.1        1940    1.4        1.40000              0
```

Figure 41.7. PROC PRINT Output of the Results Data Set

Note that the fits for all the smoothing parameters are placed in single data set and that ODS has added a SmoothingParameter variable to this data set that you can use to distinguish each fit.

The following statements display the loess fits obtained in a 2 by 2 plot grid:

```
goptions  nodisplay;
proc gplot data=Results;
   by SmoothingParameter;
   plot DepVar*Year=1 Pred*Year/ &opts name='fit';
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
    igout gseg;
    treplay 1:fit 2:fit2 3:fit1 4:fit3;
run; quit;
```

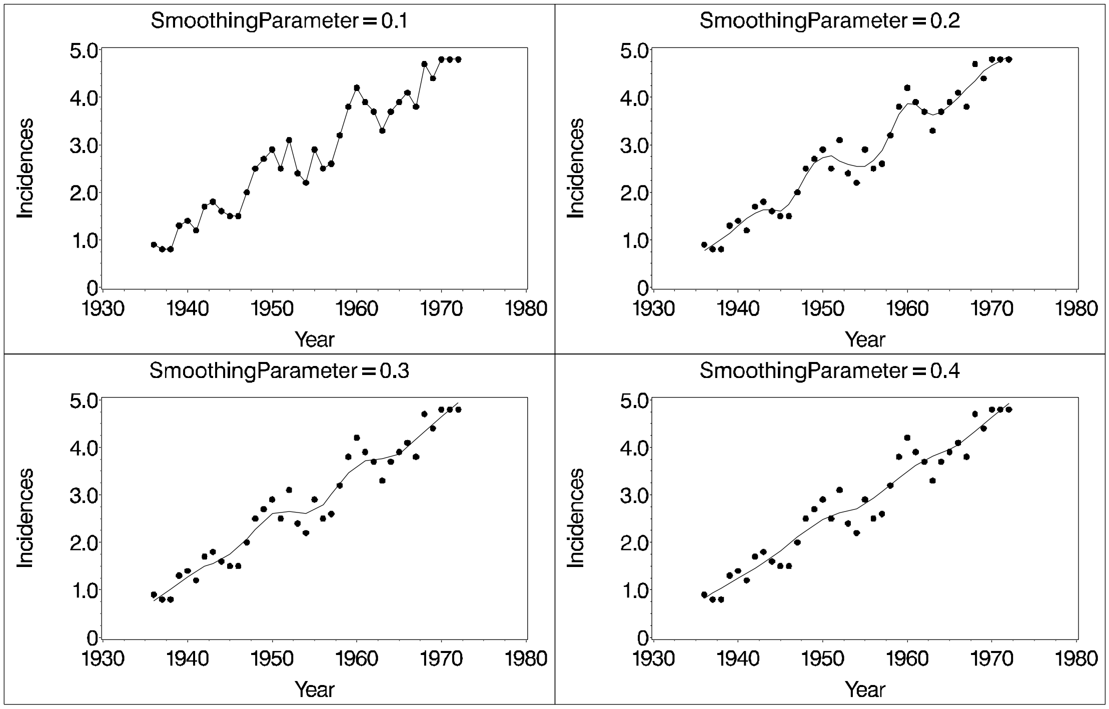


Figure 41.8. Loess Fits with a Range of Smoothing Parameters

If you examine the plots in Figure 41.8, you see that a good fit is obtained with smoothing parameter value 0.2. You can gain further insight in how to choose the smoothing parameter value by examining scatter plots of the fit residuals versus the year. To aid the interpretation of these scatter plots, you can again use PROC LOESS to smooth the response Residual as a function of Year.

```
proc loess data=Results;
   by SmoothingParameter;
   ods output OutputStatistics=residout;
   model Residual=Year / smooth=0.3;
run;

axis1 label = (angle=90 rotate=0)
      order = (-0.8 to 0.8 by 0.4);
goptions nodisplay;
proc gplot data=residout;
   by SmoothingParameter;
   format DepVar 3.1;
   plot  DepVar*Year Pred*Year / &opts vref=0 lv=2 vm=1
                                name='resids';
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
    igout gseg;
    treplay 1:resids 2:resids2 3:resids1 4:resids3;
run; quit;
```

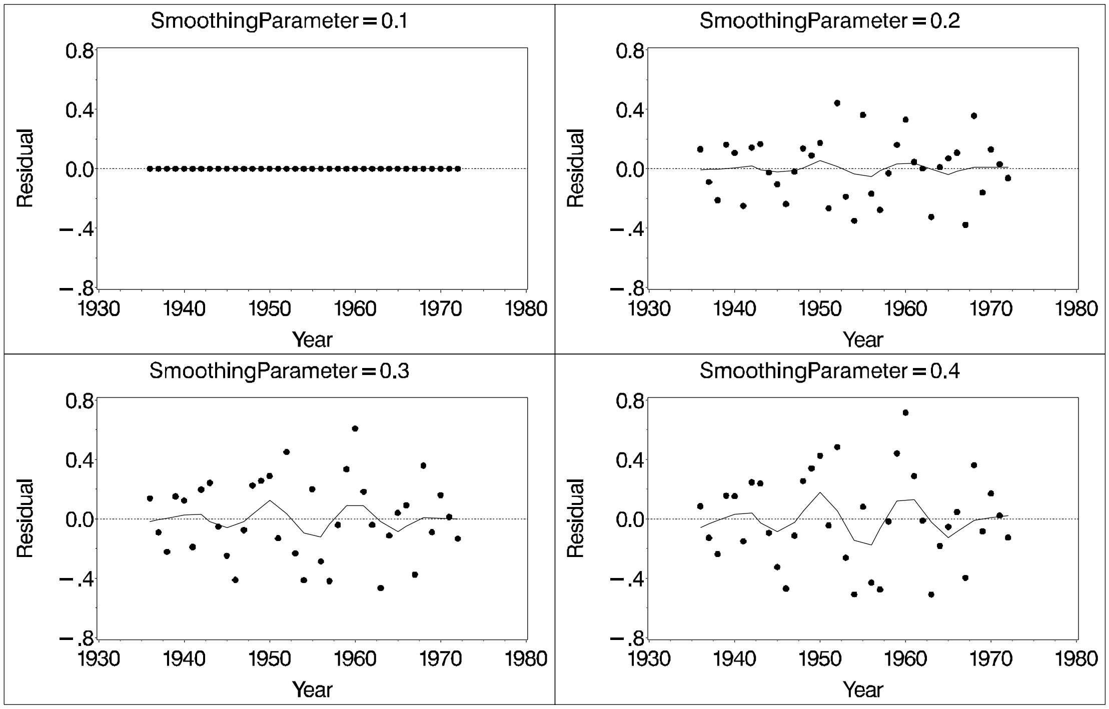


Figure 41.9. Scatter Plots of Residuals versus Year

Looking at the scatter plots in Figure 41.9 confirms that the choice 0.2 is reasonable. With smoothing parameter value 0.1, there is gross overfitting in the sense that the original data are exactly interpolated. The loess fits on the Residual versus Year scatter plots for smoothing parameter values 0.3 and 0.4 reveal that there is a periodic trend in the residuals that is much weaker when the value 0.2. This suggests that when the smoothing parameter value is above 0.3, an overly smooth fit is obtained that misses essential features in the original data.

Having now decided on a loess fit, you may want to obtain confidence limits for your model predictions. This is done by adding the CLM option in the MODEL statement. By default 95% limits are produced, but this can be changed by using the ALPHA= option in the MODEL statement. The following statements add 90% confidence limits to the Results data set and display the results graphically:

```
proc loess data=Melanoma;
   model Incidences=Year/smooth=0.2 residual clm
                         alpha=0.1;
   ods output OutputStatistics=Results;
run;

symbol3 color=green interpol=join value=none;
symbol4 color=green interpol=join value=none;
axis1 label = (angle=90 rotate=0)
      order = (0 to 6);
title1 'Age-adjusted Melanoma Incidences for 37 Years';

proc gplot data=Results;
  plot  DepVar*Year Pred*Year LowerCl*Year UpperCL*Year
                     /  &opts;
run;
```

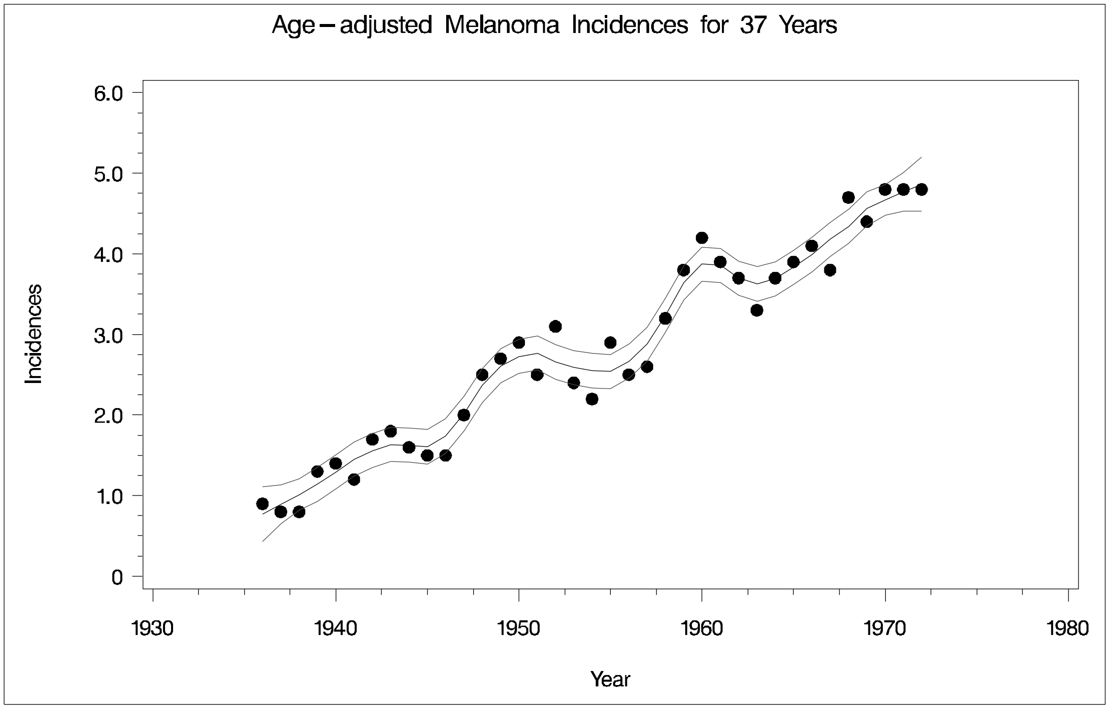


Figure 41.10. Loess fit of Melanoma Data with 90% Confidence Bands

Syntax

The following statements are available in PROC LOESS:

PROC LOESS <*DATA=SAS-data-set*> **;**
  MODEL *dependents=regressors* < */ options* > **;**
  ID *variables* **;**
  BY *variables* **;**
  WEIGHT *variable* **;**
  SCORE *DATA=SAS-data-set* < *ID=(variable list)* > < */ options* > **;**

The PROC LOESS and MODEL statements are required. The BY, WEIGHT, and ID statements are optional. The SCORE statement is optional, and more than one SCORE statement can be used.

The statements used with the LOESS procedure, in addition to the PROC LOESS statement, are as follows.

BY specifies variables to define subgroups for the analysis.

ID names variables to identify observations in the displayed output.

MODEL specifies the dependent and independent variables in the loess model, details and parameters for the computational algorithm, and the required output.

SCORE specifies a data set containing observations to be scored.

WEIGHT declares a variable to weight observations.

PROC LOESS Statement

PROC LOESS <*DATA=SAS-data-set*> **;**

The PROC LOESS statement is required. The only option in this statement is the DATA= option, which names a data set to use for the loess model.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LOESS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in Base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement is optional, and more than one ID statement can be used. The variables listed in any of the ID statements are displayed in the "Output Statistics" table beside each observation. Any variables specified as a regressor or dependent variable in the MODEL statement already appear in the "Output Statistics" table and are not treated as ID variables, even if they appear in the variable list of an ID statement.

MODEL Statement

MODEL *dependents=independent variables* < */ options* > **;**

The MODEL statement names the dependent variables and the independent variables. Variables specified in the MODEL statement must be numeric variables in the data set being analyzed.

Table 41.1 lists the options available in the MODEL statement.

Table 41.1. Model Statement Options

Option	Description
Fitting Parameters	
DIRECT	specifies direct fitting at every data point
SMOOTH=	specifies the list of smoothing values
DEGREE=	specifies the degree of local polynomials (1 or 2)
DROPSQUARE=	specifies the variables whose squares are to be dropped from local quadratic polynomials
BUCKET=	specifies the number of points in kd tree buckets
ITERATIONS=	specifies the number of reweighting iterations
DFMETHOD=	specifies the method of computing lookup degrees of freedom
SELECT=	specifies that automatic smoothing parameter selection be done
TRACEL	displays the trace of the smoothing matrix
Residuals and Confidence limits	
ALL	requests the following options: CLM, RESIDUAL, STD, SCALEDINDEP
CLM	displays $100(1-\alpha)\%$ confidence interval for the mean predicted value
RESIDUAL	displays residual statistics

Table 41.1. (continued)

Option	**Description**
STD	displays estimated prediction standard deviation
T	displays t statistics
INTERP=	specifies the degree of polynomials used in blending
Display Options	
DETAILS=	specifies which tables are to be displayed
Other options	
ALPHA=	sets significance value for confidence intervals
SCALE=	specifies the method used to scale the regressor variables
SCALEDINDEP	displays scaled independent variable coordinates

The following options are available in the MODEL statement after a slash (/).

ALL

requests all these options: CLM, RESIDUAL, SCALEDINDEP, STD, and T.

ALPHA=*number*

sets the significance level used for the construction of confidence intervals for the current MODEL statement. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals.

BUCKET=*number*

specifies the maximum number of points in the leaf nodes of the kd tree. The default value used is $s * n/5$, where s is a smoothing parameter value specified using the SMOOTH= option and n is the number of observations being used in the current BY group. The BUCKET= option is ignored if the DIRECT option is specified.

CLM

requests that $100(1-\alpha)$ confidence limits on the mean predicted value be added to the "Output Statistics" table. By default, 95% limits are computed; the ALPHA= option in the MODEL statement can be used to change the α-level. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

DEGREE= 1 | 2

sets the degree of the local polynomials to use for each local regression. The valid values are 1 for local linear fitting or 2 for local quadratic fitting, with 1 being the default.

DETAILS $<$ **(** *tables* **)** $>$

selects which tables to display, where *tables* is one or more of KDTREE, MODELSUMMARY, OUTPUTSTATISTICS, and PREDATVERTICES:

- KDTREE displays the kd tree structure.
- MODELSUMMARY displays the fit criteria for all smoothing parameter values that are specified in the SMOOTH= option in the MODEL statement, or which are fit with automatic smoothing parameter selection.

- OUTPUTSTATISTICS displays the predicted values and other requested statistics at the points in the input data set.
- PREDATVERTICES displays fitted values and coordinates of the kd tree vertices where the local least squares fitting is done.

The KDTREE and PREDATVERTICES specifications are ignored if the DIRECT option is specified in the MODEL statement. Specifying the option DETAILS with no qualifying list outputs all tables.

DFMETHOD= NONE | EXACT | APPROX < **(***approx-options***)** >

specifies the method used to calculate the "lookup" degrees of freedom used in performing statistical inference. The default is DFMETHOD=NONE, unless you specify any of the MODEL statement options ALL, CLM, or T, or any SCORE statement CLM option, in which case the default is DFMETHOD=EXACT.

You can specify the following *approx-options* in parentheses after the DFMETHOD=APPROX option:

QUANTILE=*number* specifies that the smallest 100(*number*)% of the nonzero coefficients in the smoothing matrix is set to zero in computing the approximate lookup degrees of freedom. The default value is QUANTILE=0.9.

CUTOFF=*number* specifies that coefficients in the smoothing matrix whose magnitude is less than the specified value are set to zero in computing the approximate lookup degrees of freedom. Using the CUTOFF= option overrides the QUANTILE= option.

See the "Sparse and Approximate Degrees of Freedom Computation" section on page 2246 for a description of the method used when the DFMETHOD=APPROX option is specified.

DIRECT

specifies that local least squares fits are to be done at every point in the input data set. When the direct option is not specified, a computationally faster method is used. This faster method performs local fitting at vertices of a kd tree decomposition of the predictor space followed by blending of the local polynomials to obtain a regression surface.

DROPSQUARE=(*variables***)**

specifies the quadratic monomials to exclude from the local quadratic fits. This option is ignored unless the DEGREE=2 option has been specified. For example,

```
model z=x y / degree=2 dropsquare=(y)
```

uses the monomials 1, x, y, x^2, and xy in performing the local fitting.

INTERP= LINEAR | CUBIC

specifies the degree of the interpolating polynomials used for blending local polynomial fits at the kd tree vertices. This option is ignored if the DIRECT option is specified in the model statement. INTERP=CUBIC is not supported for models with more than two regressors. The default is INTERP=LINEAR.

ITERATIONS=*number*

specifies the number of iterative reweighting steps to be done. Such iterations are appropriate when there are outliers in the data or when the error distribution is a symmetric long-tailed distribution. The default number of iterations is 1.

RESIDUAL | R

specifies that residuals are to be included in the "Output Statistics" table.

SCALE= NONE | SD $<$ **(***number***)** $>$

specifies the scaling method to be applied to scale the regressors. The default is NONE, in which case no scaling is applied. A specification of SD(*number*) indicates that a trimmed standard deviation is to be used as a measure of scale, where *number* is the trimming fraction. A specification of SD with no qualification defaults to 10% trimmed standard deviation.

SCALEDINDEP

specifies that scaled regressor coordinates be included in the output tables. This option is ignored if the SCALE= model option is not used or if SCALE=NONE is specified.

SELECT=*criterion* **< (<GLOBAL> <STEPS> <RANGE(***lower***,***upper***)>)>**
SELECT=*DFCriterion* **(** *target* **<GLOBAL> <STEPS> <RANGE(***lower***,***upper***)>)**

specifies that automatic smoothing parameter selection be done using the named *criterion* or *DFCriterion*. Valid values for the *criterion* are

AICC	specifies the AIC_C criterion (Hurvich, Simonoff, and Tsai 1998).
AICC1	specifies the AIC_{C_1} criterion (Hurvich, Simonoff, and Tsai 1998).
GCV	specifies the generalized cross-validation criterion (Craven and Wahba 1979).

The *DFCriterion* specifies the measure used to estimate the model degrees of freedom. The measures implemented in PROC LOESS all depend on prediction matrix L relating the observed and predicted values of the dependent variable. Valid values are

DF1	specifies Trace(L).
DF2	specifies Trace$(L^T L)$.
DF3	specifies 2Trace$(L) -$ Trace$(L^T L)$.

For both types of selection, the smoothing parameter value is selected to yield a minimum of an optimization criterion. If you specify *criterion* as one of AICC, AICC1, or GCV, the optimization criterion is the specified *criterion*. If you specify *DFCriterion* as one of DF1, DF2, or DF3, the optimization criterion is $|$ *DFCriterion* $-$ *target* $|$, where *target* is a specified target degree of freedom value. Note that if you specify a *DFCriterion*, then you must also specify a target value. See the section "Automatic Smoothing Parameter Selection" on page 2243 for definitions and properties of the selection criteria.

The selection is done as follows:

- If you specify the SMOOTH=*value-list* option, then PROC LOESS selects the largest value in this list that yields the global minimum of the specified optimization criterion.
- If you do not specify the SMOOTH= option, then PROC LOESS finds a local minimum of the specified optimization criterion using a golden section search of values less than or equal to one.

You can specify the following modifiers in parentheses after the specified criterion to alter the behavior of the SELECT= option:

GLOBAL — specifies that a global minimum be found within the range of smoothing parameter values examined. This modifier has no effect if you also specify the SMOOTH= option in the MODEL statement.

STEPS — specifies that all models evaluated in the selection process be displayed.

RANGE(*lower*,*upper*) — specifies that only smoothing parameter values greater than or equal to *lower* and less than or equal to *upper* be examined.

For models with one dependent variable, if you specify neither the SELECT= nor the SMOOTH= options in the MODEL statement, then PROC LOESS uses SELECT=AICC.

The following table summarizes how the smoothing parameter values are chosen for various combinations of the SMOOTH= option, the SELECT= option, and the SELECT= option modifiers.

Table 41.2. Smoothing Parameter Value(s) Used for Combinations of SMOOTH= and SELECT= OPTIONS for Models with One Dependent Variable

Syntax	**Search Method**	**Search Domain**
default	golden section using AICC	$(0, 1]$
SMOOTH=*list*	no selection	values in *list*
SMOOTH=*list* SELECT=*criterion*	global	values in *list*
SMOOTH=*list* SELECT=*criterion* (RANGE(l, u))	global	values in *list* within $[l, u]$
SELECT=*criterion*	golden section	$(0, 1]$
SELECT=*criterion* (RANGE(*l*,*u*))	golden section	$[l, u]$
SELECT=*criterion* (GLOBAL)	global	$(0, 1]$
SELECT=*criterion* (GLOBAL RANGE(l, u))	global	$[l, u]$

Some examples of using the SELECT= option are

SELECT=GCV	specifies selection using the GCV *criterion*.
SELECT=DF1(6.3)	specifies selection using the DF1 *DFCriterion* with target value 6.3.
SELECT=AICC(STEPS)	specifies selection using the AICC *criterion* showing all step details.
SELECT=DF2(7 GLOBAL)	specifies selection using the DF2 *DFCriterion* with target value 7 using a global search algorithm.

Note: The SELECT= option cannot be used for models with more than one dependent variable.

SMOOTH=*value-list*

specifies a list of positive smoothing parameter values. If you do not SELECT= option in the MODEL statement, then a separate fit is obtained for each SMOOTH= value specified. If you do specify the SELECT= option, then models with all values specified in the SMOOTH= list are examined, and PROC LOESS selects the value that minimizes the criterion specified in the SELECT= option.

For models with two or more dependent variables, if the SMOOTH= option is not specified in the MODEL statement, then SMOOTH=0.5 is used as a default.

STD

specifies that standardized errors are to be included in the “Output Statistics” table.

T

specifies that t statistics are to be included in the “Output Statistics” table.

TRACEL

specifies that the trace of the prediction matrix as well as the GCV and AICC statistics are to be included in the “FIT Summary” table. The use of any of the MODEL statement options ALL, CLM, DFMETHOD=EXACT, DIRECT, SELECT=, or T implicitly selects the TRACEL option.

SCORE Statement

SCORE *<DATA=SAS-data-set>* < *ID=(variable list)* > $<$ */ options* $>$ **;**

The fitted loess model is used to score the data in the specified SAS data set. This data set must contain all the regressor variables specified in the MODEL statement. Furthermore, when a BY statement is used, the score data set must also contain all the BY variables sorted in the order of the BY variables. A SCORE statement is optional, and more than one SCORE statement can be used. SCORE statements cannot be used if the DIRECT option is specified in the MODEL statement. The optional ID= (*variable list*) specifies ID variables to be included in the “Score Results” table.

You find the results of the SCORE statement in the “Score Results” table. This table contains all the data in the data set named in the SCORE statement, including observations with missing values. However, only those observations with nonmissing regressor variables are scored. If no data set is named in the SCORE statement, the data set named in the PROC LOESS statement is scored. You use the PRINT option in the SCORE statement to request that the “Score Results” table be displayed. You can place the “Score Results” table in an output data set using an ODS OUTPUT statement even if this table is not displayed.

The following options are available in the SCORE statement after a slash (/).

CLM

requests that $100(1-\alpha)$ confidence limits on the mean predicted value be added to the “Score Results” table. By default the 95% limits are computed; the ALPHA= option in the MODEL statement can be used to change the α-level. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

PRINT < (*variables* **)>**

specifies that the “Score Results” table is to be displayed. By default only the variables named in the MODEL statement, the variables listed in the ID list in the SCORE statement, and the scored dependent variables are displayed. The optional list in the PRINT option specifies additional variables in the score data set that are to be included in the displayed output. Note however that all columns in the SCORE data set are placed in the SCORE results table, even if you do not request that they be included in the displayed output.

SCALEDINDEP

specifies that scaled regressor coordinates be included in the SCORE results table. This option is ignored if the SCALE= model option is not used or if SCALE=NONE is specified.

STEPS

requests that all models evaluated during smoothing parameter value selection be scored, provided that the SELECT= option together with the STEPS modifier is specified in the MODEL statement. By default only the selected model is scored.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a variable in the input data set that contains values to be used as a priori weights for a loess fit.

The values of the weight variable must be nonnegative. If an observation’s weight is zero, negative, or missing, the observation is deleted from the analysis.

Details

Missing Values

PROC LOESS deletes any observation with missing values for any variable specified in the MODEL statement. This enables the procedure to reuse the kd tree for all the dependent variables that appear in the MODEL statement. If you have multiple dependent variables with different missing value structures for the same set of independent variables, you may want to use separate PROC LOESS steps for each dependent variable.

Output Data Sets

PROC LOESS assigns a name to each table it creates. You can use the ODS OUTPUT statement to place one or more of these tables in output data sets. See the section "ODS Table Names" on page 2248 for a list of the table names created by PROC LOESS. For detailed information on ODS, see Chapter 14, "Using the Output Delivery System."

For example, the following statements create an output data set named MyOutStats containing the OutputStatistics table and an output data set named MySummary containing the FitSummary table.

```
proc loess data=Melanoma;
   model Incidences=Year;
   ods output OutputStatistics = MyOutStats
              FitSummary       = MySummary;
run;
```

Often, a single MODEL statement describes more than one model. For example, the following statements fit eight different models (4 smoothing parameter values for each dependent variable).

```
proc loess data=notReal;
   model y1 y2 = x1 x2 x3/smooth =0.1 to 0.7 by 0.2;
   ods output OutputStatistics = MyOutStats;
run;
```

The eight OutputStatistics tables for these models are stacked in a single data set called MyOutStats. The data set contains a column named DepVarName and a column named SmoothingParameter that distinguish each model (see Figure 41.4 on page 2224 for an example). If you want the OutputStatistics table for each model to be in its own data set, you can do so by using the MATCH_ALL option in the ODS OUTPUT statement. The following statements create eight data sets named MyOutStats, MyOutStats1, ... , MyOutStats7.

```
proc loess data=notReal;
   model y1 y2 = x1 x2 x3/smooth =0.1 to 0.7 by 0.2;
   ods output OutputStatistics(match_all) = MyOutStats;
run;
```

For further options available in the ODS OUTPUT statement, see Chapter 14, “Using the Output Delivery System.”

Only the ScaleDetails and FitSummary tables are displayed by default. The other tables are optionally displayed by using the DETAILS option in the MODEL statement and the PRINT option in the SCORE statement. Note that it is not necessary to display a table in order for that table to be used in an ODS OUTPUT statement. For example, the following statements display the OutputStatistics and kdTree tables but place the OutputStatistics and PredAtVertices tables in output data sets.

```
proc loess data=Melanoma;
   model Incidences=Year/details(OutputStatistics kdTree);
   ods output OutputStatistics = MyOutStats
              PredAtVertices   = MyVerticesOut;
run;
```

Using the DETAILS option alone causes all tables to be displayed.

The MODEL statement options CLM, RESIDUAL, STD, SCALEDINDEP, and T control which optional columns are added to the OutputStatistics table. For example, to obtain an OutputStatistics output data set containing residuals and confidence limits in addition to the model variables and predicted value, you need to specify the RESIDUAL and CLM options in the MODEL statement as in the following example:

```
proc loess data=Melanoma;
   model Incidences=Year/residual clm;
   ods output OutputStatistics = MyOutStats;
run;
```

Finally, note that the ALL option in the MODEL statement includes all optional columns in the output. Also, ID columns can be added to the OutputStatistics table by using the ID statement.

Data Scaling

The loess algorithm to obtain a predicted value at a given point in the predictor space proceeds by doing a least squares fit using all data points that are close to the given point. Thus the algorithm depends critically on the metric used to define closeness. This has the consequence that if you have more than one predictor variable and these predictor variables have significantly different scales, then closeness depends almost entirely on the variable with the largest scaling. It also means that merely changing the units of one of your predictors can significantly change the loess model fit.

To circumvent this problem, it is necessary to standardize the scale of the independent variables in the loess model. The SCALE= option in the MODEL statement is provided for this purpose. PROC LOESS uses a symmetrically trimmed standard deviation as the scale estimate for each independent variable of the loess model. This is a robust scale estimator in that extreme values of a variable are discarded before estimating the data scaling. For example, to compute a 10% trimmed standard deviation of a sample, you discard the smallest and largest 5% of the data and compute the standard deviation of the remaining 90% of the data points. In this case, the trimming fraction is 0.1.

For example, the following statements specify that the variables Temperature and Catalyst are scaled before performing the loess fitting. In this case, because the trimming fraction is 0.1, the scale estimate used for each of these variables is a 10% trimmed standard deviation.

```
model Yield=Temperature Catalyst / scale = SD(0.1);
```

The default trimming fraction used by PROC LOESS is 0.1 and need not be specified by the SCALE= option. Thus the following MODEL statement is equivalent to the previous MODEL statement.

```
model Yield=Temperature Catalyst / scale = SD;
```

If the SCALE= option is not specified, no scaling of the independent variables is done. This is appropriate when there is only a single independent variable or when all the independent variables are a priori scaled similarly.

When the SCALE= option is specified, the scaling details for each independent variable are added to the ScaleDetails table (see Output 41.3.2 on page 2265 for an example). By default, this table contains only the minimum and maximum values of each independent variable in the model. Finally, note that when the SCALE= option is used, specifying the SCALEDINDEP option in the MODEL statement adds the scaled values of the independent variables to the OutputStatistics and PredAtVertices tables. If the SCALEDINDEP option is specified in the SCORE statement then scaled values of the independent variables are included in the ScoreResults table. By default, only the unscaled values are placed in these tables.

Direct versus Interpolated Fitting

Local regression to obtain a predicted value at a given point in the predictor space is done by doing a least squares fit using all data points in a local neighborhood of the given point. This method is computationally expensive because a local neighborhood must be determined and a least squares problem solved for each point at which a fitted value is required. A faster method is to obtain such fits at a representative sample of points in the predictor space and to obtain fitted values at all other points by interpolation.

PROC LOESS can fit models using either of these two paradigms. By default, PROC LOESS uses fitting at a sample of points and interpolation. The method fitting a local model at every data point is selected by specifying the DIRECT option in the MODEL statement.

kd Trees and Blending

PROC LOESS uses a kd tree to divide the box (also called the *initial cell* or *bucket*) enclosing all the predictor data points into rectangular cells. The vertices of these cells are the points at which local least squares fitting is done.

Starting from the initial cell, the direction of the longest cell edge is selected as the split direction. The median of this coordinate of the data in the cell is the split value. The data in the starting cell are partitioned into two child cells. The left child consists of all data from the parent cell whose coordinate in the split direction is less than the split value. The above procedure is repeated for each child cell that has more than a prespecified number of points, called the *bucket size* of the kd tree.

You can specify the bucket size with the BUCKET= option in the MODEL statement. If you do not specify the BUCKET= option, the default value used is the largest integer less than or equal to $ns/5$, where n is the number of observations and s is the value of the smoothing parameter. Note that if fitting is being done for a range of smoothing parameter values, the bucket size may change for each value.

The set of vertices of all the cells of the kd tree are the points at which PROC LOESS performs its local fitting. The fitted value at an original data point (or at any other point within the original data cell) is obtained by blending the fitted values at the vertices of the kd tree cell that contains that data point.

The univariate blending methods available in PROC LOESS are linear and cubic polynomial interpolation, with linear interpolation being the default. You can request cubic interpolation by specifying the INTERP=CUBIC option in the MODEL statement. In this case, PROC LOESS uses the unique cubic polynomial whose values and first derivatives match those of the fitted local polynomials evaluated at the two endpoints of the kd tree cell edge.

In the multivariate case, such univariate interpolating polynomials are computed on each edge of the kd-tree cells, and are combined using blending functions (Gordon 1971). In the case of two regressors, if you specify INTERP=CUBIC in the MODEL statement, PROC LOESS uses Hermite cubic polynomials as blending functions. If you do not specify INTERP=CUBIC, or if you specify a model with more than two regressors, then PROC LOESS uses linear polynomials as blending functions. In these cases, the blending method reduces to tensor product interpolation from the 2^p vertices of each kd tree cell, where p is the number of regressors.

While the details of the kd tree and the fitted values at the vertices of the kd tree are implementation details that seldom need to be examined, PROC LOESS does provide options for their display. Each kd tree subdivision of the data used by PROC LOESS is placed in the "kdTree" table. The predicted values at the vertices of each kd tree are placed in the "PredAtVertices" table. You can request these tables using the DETAILS option in the MODEL statement.

Local Weighting

The size of the local neighborhoods that PROC LOESS uses in performing local fitting is determined by the smoothing parameter value s. When $s < 1$, the local neighborhood used at a point x contains the s fraction of the data points closest to the point x. When $s \geq 1$, all data points are used.

Suppose q denotes the number of points in the local neighborhoods and $d_1, d_2, \ldots, d_q$ denote the distances in increasing order of the q points closest to x. The point at distance d_i from x is given a weight w_i in the local regression that decreases as the distance from x increases. PROC LOESS uses a tricube weight function to define

$$w_i = \frac{32}{5}\left(1 - (\frac{d_i}{d_q})^3\right)^3$$

If $s > 1$, then d_q is replaced by $d_q s^{1/p}$ in the previous formula, where p is the number of predictors in the model.

Finally, note that if a weight variable has been specified using a WEIGHT statement, then w_i is multiplied by the corresponding value of the specified weight variable.

Iterative Reweighting

PROC LOESS can do iterative reweighting to improve the robustness of the fit in the presence of outliers in the data. Iterative reweighting is also appropriate when statistical inference is requested and the error distribution is symmetric but not Gaussian.

The number of iterations is specified by the ITERATIONS= option in the MODEL statement. The default is ITERATIONS=1, which corresponds to no reweighting.

At iterations beyond the first iteration, the local weights w_i of the previous section are replaced by $r_i w_i$ where r_i is a weight that decreases as the residual of the fitted value at the previous iteration at the point corresponding to d_i increases. Refer to Cleveland and Grosse (1991) and Cleveland, Grosse, and Shyu (1992) for details.

Specifying the Local Polynomials

PROC LOESS uses linear or quadratic polynomials in doing the local least squares fitting. The option DEGREE = in the MODEL statement is used to specify the degree of the local polynomials used by PROC LOESS, with DEGREE = 1 being the default. In addition, when DEGREE = 2 is specified, the MODEL statement DROPSQUARE= option can be used to exclude specific monomials during the least squares fitting.

For example, the following statements use the monomials 1, x1, x2, x1*x2, and x2*x2 for the local least squares fitting.

```
proc loess data=notReal;
   model y= x1 x2/ degree=2 dropsquare=(x1);
run;
```

Statistical Inference

If you denote the ith measurement of the response by y_i and the corresponding measurement of predictors by x_i, then

$$y_i = g(x_i) + \epsilon_i$$

where g is the regression function and ϵ_i are independent random errors with mean zero. If the errors are normally distributed with constant variance, then you can obtain confidence intervals for the predictions from PROC LOESS. You can also obtain confidence limits in the case where ϵ_i is heteroscedastic but $a_i \epsilon_i$ has constant variance and a_i are a priori weights that are specified using the WEIGHT statement of PROC LOESS. You can do inference in the case in which the error distribution is symmetric by using iterative reweighting.

Formulae for doing statistical inference under the preceding conditions can be found in Cleveland and Grosse (1991) and Cleveland, Grosse, and Shyu (1992). The main result of their analysis is that a standardized residual for a loess model follows a t distribution with ρ degrees of freedom, where ρ is called the “lookup degrees of freedom.” ρ is a function of the smoothing matrix L, which defines the linear relationship between the fitted and observed dependent variable values of a loess model.

The determination of ρ is computationally expensive and is not done by default. It is computed if you specify the DFMETHOD=EXACT or DFMETHOD=APPROX option in the MODEL statement. It is also computed if you specify any of the options CLM, STD, or T in the MODEL statement.

If you specify the CLM option in the MODEL statement, confidence limits are added to the OutputStatistics table. By default, 95% limits are computed, but you can change this by using the ALPHA= option in the MODEL statement.

Automatic Smoothing Parameter Selection

There are several methodologies for automatic smoothing parameter selection. One class of methods chooses the smoothing parameter value to minimize a criterion that incorporates both the tightness of the fit and model complexity. Such a criterion can usually be written as a function of the error mean square, $\hat{\sigma}^2$, and a penalty function designed to decrease with increasing smoothness of the fit. This penalty function is usually defined in terms of the matrix L such that

$$\hat{y} = Ly$$

where y is the vector of observed values and $\hat{y}$ is the corresponding vector of predicted values of the dependent variable. Examples of specific criteria are generalized cross-validation (Craven and Wahba 1979) and the Akaike information criterion (Akaike 1973). These classical selectors have two undesirable properties when used with local polynomial and kernel estimators: they tend to undersmooth and tend to be non-robust in the sense that small variations of the input data can change the choice of smoothing parameter value significantly. Hurvich, Simonoff, and Tsai (1998) obtained several bias-corrected AIC criteria that limit these unfavorable properties and perform comparably with the *plug-in selectors* (Ruppert, Sheather, and Wand 1995). PROC LOESS provides automatic smoothing parameter selection using two of these bias-corrected AIC criteria, named AIC_{C_1} and AIC_C in Hurvich, Simonoff, and Tsai (1998), and generalized cross-validation, denoted by the acronym GCV.

The relevant formulae are

$$\begin{aligned}
AIC_{C_1} &= n\log(\hat{\sigma}^2) + n\frac{\delta_1/\delta_2(n+\nu_1)}{\delta_1^2/\delta_2 - 2} \\
AIC_C &= \log(\hat{\sigma}^2) + 1 + \frac{2\,(\text{Trace}(L)+1)}{n - \text{Trace}(L) - 2} \\
GCV &= \frac{n\hat{\sigma}^2}{(n - \text{Trace}(L))^2}
\end{aligned}$$

where n is the number of observations and

$$\begin{aligned}
\delta_1 &\equiv \text{Trace}(I-L)^T(I-L) \\
\delta_2 &\equiv \text{Trace}\left((I-L)^T(I-L)\right)^2 \\
\nu_1 &\equiv \text{Equivalent Number of Parameters} \\
&\equiv \text{Trace}(L^TL)
\end{aligned}$$

Note that the values of δ_1, δ_2, and the “Equivalent Number of Parameters” are reported in the “Fit Summary” table.

You invoke these methods for automatic smoothing parameter selection by specifying the SELECT=*criterion* option in the MODEL statement, where *criterion* is one of AICC1, AICC, or GCV. PROC LOESS evaluates the specified criterion for a sequence of smoothing parameter values and selects the value in this sequence that minimizes the specified criterion. If multiple values yield the optimum, then the largest of these values is selected.

A second class of methods seeks to set an approximate measure of model degrees of freedom to a specified target value. These methods are useful for making meaningful comparisons between loess fits and other nonparametric and parametric fits. The approximate model degrees of freedom in a nonparametric fit is a number that is analogous to the number of free parameters in a parametric model. There are three commonly used measures of model degrees of freedom in nonparametric models. These criteria are Trace(L), Trace(L^TL), and $2\text{Trace}L - \text{Trace}(L^TL)$. A discussion of their properties can be found in Hastie and Tibshirani (1990). You invoke these methods by specifying the SELECT=*DFCriterion*(*target*) option in the MODEL statement, where *DFCriterion* is one of DF1, DF2, or DF3. The criterion that is minimized is given in the following table:

Table 41.3. Minimization Criteria for Degree of Freedom Based Smoothing Parameter Value Selection

Syntax	**Minimization Criterion**
SELECT=DF1(*target*)	$\mid \text{Trace}(L) - \textit{target} \mid$
SELECT=DF2(*target*)	$\mid \text{Trace}(L^T L) - \textit{target} \mid$
SELECT=DF3(*target*)	$\mid 2\text{Trace}(L) - \text{Trace}(L^T L) - \textit{target} \mid$

The results are summarized in the “Smoothing Criterion” table. This table is displayed whenever automatic smoothing parameter selection is performed. You can obtain details of the sequence of models examined by specifying the DETAILS(MODELSUMMARY) option in the model statement to display the “Model Summary” table.

There are several ways in which you can control the sequence of models examined by PROC LOESS. If you specify the SMOOTH=*value-list* option in the MODEL statement, then only the values in this list are examined in performing the selection. For example, the following statements select the model that minimizes the AICC1 criterion among the three models with smoothing parameter values 0.1, 0.3, and 0.4:

```
proc loess data=notReal;
   model y= x1/ smooth=0.1 0.3 0.4 select=AICC1;
run;
```

If you do not specify the SMOOTH= option in the model statement, then by default PROC LOESS uses a golden section search method to find a local minimum of the specified criterion in the range $(0, 1]$. You can use the RANGE(*lower*,*upper*) modifier in the SELECT= option to change the interval in which the golden section search is performed. For example, the following statements request a golden section search to find a local minimizer of the GCV criterion for smoothing parameter values in the interval [0.1,0.5]:

```
proc loess data=notReal;
   model y= x1/select=GCV( range(0.1,0.5) );
run;
```

If you want to be sure of obtaining a global minimum in the range of smoothing parameter values examined, you can specify the GLOBAL modifier in the SELECT= option. For example, the following statements request that a global minimizer of the AICC criterion be obtained for smoothing parameter values in the interval $[0.2, 0.8]$:

```
proc loess data=notReal;
   model y= x1/select=AICC( global range(0.2,0.8) );
run;
```

Note that even though the smoothing parameter is a continuous variable, a given range of smoothing parameter values corresponds to a finite set of local models. For example, for a data set with 100 observations, the range $[0.2, 0.4]$ corresponds to models with $20, 21, 22, \ldots, 40$ points in the local neighborhoods. If the GLOBAL modifier is specified, all possible models in the range are evaluated sequentially.

Note that by default PROC LOESS displays a “Fit Summary” and other optionally requested tables only for the selected model. You can request that these tables be displayed for all models in the selection process by adding the STEPS modifier in the SELECT= option. Also note that by default scoring requested with SCORE statements is done only for the selected model. However, if you specify the STEPS in both the MODEL and SCORE statements, then all models evaluated in the selection process are scored.

In terms of computation, AIC_C, GCV, and DF1 depend on the smoothing matrix L only through its trace. In the direct method, this trace can be computed efficiently. In the interpolated method using kd trees, there is some additional computational cost but the overall work is not significant compared to the rest of the computation. In contrast, the quantities δ_1, δ_2, and ν_1, which appear in the AIC_{C_1} criterion, and the DF2 and DF3 criteria, depend on the entire L matrix and for this reason, the time needed to compute these quantities dominates the time required for the model fitting. Hence SELECT=AICC1, SELECT=DF2, and SELECT=DF3 are much more computationally expensive than SELECT=AICC, SELECT=GCV, and SELECT=DF1, especially when combined with the GLOBAL modifier. Hurvich, Simonoff, and Tsai (1998) note that AIC_C can be regarded as an approximation of AIC_{C_1} and that “the AIC_C selector generally performs well in all circumstances.”

For models with one dependent variable, PROC LOESS uses SELECT=AICC as its default, if you specify neither the SMOOTH= nor SELECT= options in the MODEL statement. With two or more dependent variables, automatic smoothing parameter selection needs to be done separately for each dependent variable. For this reason automatic smoothing parameter selection is not available for models with multiple dependent variables. In such cases you should use a separate PROC LOESS step for each dependent variable, if you want to use automatic smoothing parameter selection.

Sparse and Approximate Degrees of Freedom Computation

As noted in the “Statistical Inference” section on page 2243, obtaining confidence limits in loess models requires the computation of the “lookup degrees of freedom.” This in turn requires the computation of

$$\delta_2 \quad \equiv \quad \text{Trace}\left((I-L)^T(I-L)\right)^2$$

where L is the LOESS smoothing matrix (see the “Automatic Smoothing Parameter Selection” section on page 2243).

The work in a direct implementation of this formula grows as n^3, where n is the number of observations in analysis. For large n, this work dominates the time needed to fit the loess model itself. To alleviate this computational bottleneck, Cleveland

and Grosse (1991) and Cleveland, Grosse, and Shyu (1992) developed approximate methods for estimating this quantity in terms of more readily computable statistics. A different approach to obtaining a computationally cheap estimate of δ_2 has been implemented in PROC LOESS.

For large data sets with significant local structure, the LOESS model is often used with small values of the smoothing parameter. Recalling that the smoothing parameter defines the fraction of the data used in each local regression, this means that the loess fit at any point in regressor space depends on only a small fraction of the data. This is reflected in the smoothing matrix L whose (i, j)th entry is nonzero only if the ith and jth observations lie in at least one common local neighborhood. Hence the smoothing matrix is a sparse matrix (has mostly zero entries) in such cases. By exploiting this sparsity PROC LOESS now computes δ_2 orders of magnitude faster than in previous implementations.

When each local neighborhood contains a large subset of the data, i.e., when the smoothing parameter is large, then it is no longer true that the smoothing matrix is sparse. However, since a point in a local neighborhood is given a local weight that decreases with its distance from the center of the neighborhood, many of the coefficients in the smoothing matrix turn out to be nonzero but with orders of magnitude smaller than that of the larger coefficients in the matrix. The approximate method for computing δ_2 that has been implemented in PROC LOESS exploits these disparities in magnitudes of the elements in the smoothing matrix by setting the small elements to zero. This creates a sparse approximation of the smoothing matrix to which the fast sparse methods can be applied.

In order to decide the threshold at which elements in the smoothing matrix are set to zero, PROC LOESS samples the elements in the smoothing matrix to obtain the value of the element in a specified lower quantile in this sample. The magnitude of the element at this quantile is used as a cutoff value, and all elements in the smoothing matrix whose magnitude is less than this cutoff are set to zero for the approximate computation. By default all elements in the lower 90^{th} percentile are set to zero. You can use the DFMETHOD=APPROX(QUANTILE=) option in the MODEL statement to change this value. As you increase the value for the quantile to be zeroed, you speed up the degrees of freedom computation at the expense of increasing approximation errors. You can also use the DFMETHOD=APPROX(CUTOFF=) option in the MODEL statement to specify the cutoff value directly.

For small data sets, the approximate computation is not needed and would be rougher than for larger data sets. Hence PROC LOESS performs the exact computation for analyses with less than 500 points, even if DFMETHOD=APPROX is specified in the model statement. Also, for small values of the smoothing parameter, elements in the lower specified quantile may already all be zero. In such cases the approximate method is the same as the exact method. PROC LOESS labels as approximate any statistics that depend on the approximate computation of δ_2 only in the cases where the approximate computation was used and is different from the exact computation.

Scoring Data Sets

One or more SCORE statements can be used with PROC LOESS. A data set that includes all the variables specified in the MODEL and BY statements must be specified in each SCORE statement. Score results are placed in the ScoreResults table. This table is not displayed by default, but specifying the PRINT option in the SCORE statement produces this table. If you specify the CLM option in the SCORE statement, confidence intervals are included in the ScoreResults table.

Note that scoring is not supported when the DIRECT option is specified in the MODEL statement. Scoring at a point specified in a score data set is done by first finding the cell in the kd tree containing this point and then interpolating the scored value from the predicted values at the vertices of this cell. This methodology precludes scoring any points that are not contained in the box that surrounds the data used in fitting the loess model.

ODS Table Names

PROC LOESS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 41.4. ODS Tables Produced by PROC LOESS

ODS Table Name	Description	Statement	Option
FitSummary	Specified fit parameters and fit summary		default
kdTree	Structure of kd tree used	MODEL	DETAILS(kdTree)
ModelSummary	Summary of all models evaluated	MODEL	DETAILS(ModelSummary)
OutputStatistics	Coordinates and fit results at input data points	MODEL	DETAILS(OutputStatistics)
PredAtVertices	Coordinates and fitted values at kd tree vertices	MODEL	DETAILS(PredAtVertices)
ScaleDetails	Extent and scaling of the independent variables		default
ScoreResults	Coordinates and fit results at scoring points	SCORE	PRINT
SmoothingCriterion	Criterion value and selected smoothing parameter	MODEL	SELECT

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the LOESS procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

When the ODS GRAPHICS are in effect, the LOESS procedure produces a variety of plots. For models with multiple dependent variables, separate plots are produced for each dependent variable. For models where multiple smoothing parameters are requested with the SMOOTH= option in the MODEL statement and smoothing parameter value selection is not requested, then separate plots are produced for each smoothing parameter. If smoothing parameter value selection is requested with the SELECT=option in the MODEL statement, then the plots are produced for the selected model only. However, if the STEPS qualifier is included with the SELECT= option, then plots are produced for all smoothing parameters examined in the selection process.

The plots available are as follows:

- When smoothing parameter value selection is performed, the procedure displays a scatterplot of the value of SELECTION= criterion versus the smoothing parameter value for all smoothing parameter values examined in the selection process.
- With a single regressor, the procedure displays a scatterplot of the input data with the fitted LOESS curve overlayed. If the CLM option is specified in the model statement, then a confidence band at the significance level specified in the ALPHA= option is included in the plot. For each SCORE statement a scatterplot of the scored LOESS fit at the requested points is displayed.
- When you specify the RESIDUAL option in the MODEL statement, the procedure displays a panel containing plots of the residual versus each of the regressors in the model, and also a summary panel of fit diagnostics:

  - residuals versus the predicted values
  - histogram of the residuals
  - a normal quantile plot of the residuals
  - a "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals. This plot "shows how much variation in the data is explained by the fit and how much remains in the residuals" (Cleveland 1993).
  - dependent variable values versus the predicted values

  Note that plots in the Fit Diagnostics panel can be requested individually by specifying the PLOTS(UNPACKPANELS) option in the PROC LOESS statement.

PLOTS (*general-plot-options***)**

specifies characteristics of the graphics produced when you use the experimental ODS GRAPHICS statement. You can specify the following *general-plot-options* in parentheses after the PLOTS option:

UNPACK|UNPACKPANELS specifies that plots in the Fit Diagnostics Panel should be displayed separately. Use this option if you want to access individual diagnostic plots within the panel.

MAXPOINTS=*number* **| NONE** specifies that plots with elements that require processing more than *number* points are suppressed. The default is MAXPOINTS=5000. This cutoff is ignored if you specify MAXPOINTS=NONE.

ODS Graph Names

PROC LOESS assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 41.5.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 41.5. ODS Graphics Produced by PROC LOESS

ODS Graph Name	**Plot Description**	**PLOTS Option**
ActualByPredicted	Dependent variable versus LOESS fit	UNPACKPANELS
DiagnosticsPanel	Panel of fit diagnostics	
FitPlot	Loess curve overlayed on scatterplot of data	
QQPlot	Normal quantile plot residuals	UNPACKPANELS
ResidualByPredicted	Residuals versus LOESS fit	UNPACKPANELS
ResidualHistogram	Histogram of fit residuals	UNPACKPANELS
ResidualPanel*i*	Panel *i* of residuals versus regressors	
RFPlot	Side-by-side plots of quantiles of centered fit and residuals	UNPACKPANELS
ScorePlot	Loess fit evaluated at scoring points	
SelectionCriterionPlot	Selection criterion versus smoothing parameter	

Examples

Example 41.1. Engine Exhaust Emissions

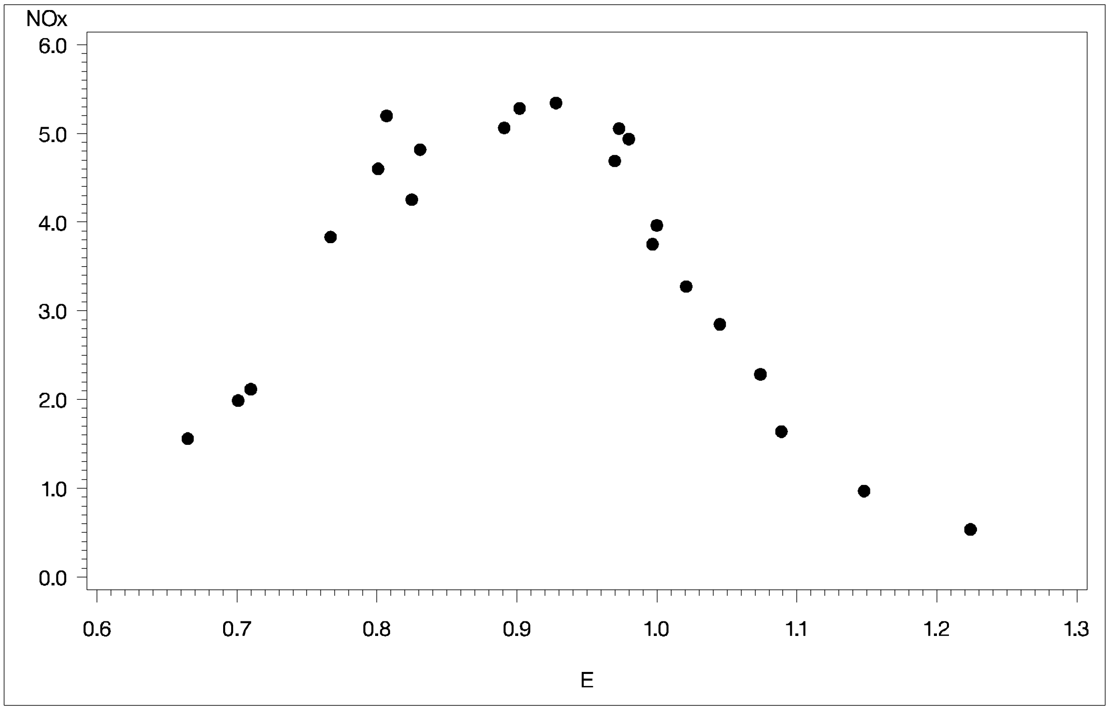
Investigators studied the exhaust emissions of a one cylinder engine (Brinkman 1981). The SAS data set Gas contains the results data. The dependent variable, NOx, measures the concentration, in micrograms per joule, of nitric oxide and nitrogen dioxide normalized by the amount of work of the engine. The independent variable, E, is a measure of the richness of the air and fuel mixture.

```
data Gas;
   input NOx E;
   format NOx f3.1;
   format E f3.1;
datalines;
4.818  0.831
2.849  1.045
3.275  1.021
4.691  0.97
4.255  0.825
5.064  0.891
2.118  0.71
4.602  0.801
2.286  1.074
0.97   1.148
3.965  1
5.344  0.928
3.834  0.767
1.99   0.701
5.199  0.807
5.283  0.902
3.752  0.997
0.537  1.224
1.64   1.089
5.055  0.973
4.937  0.98
1.561  0.665
;
```

The following PROC GPLOT statements produce the simple scatter plot of these data, displayed in Output 41.1.1.

```
symbol1 color=black value=dot ;
proc gplot data=Gas;
   plot NOx*E;
run;
```

Output 41.1.1. Scatter Plot of Gas Data



The following statements fit two loess models for these data. Because this is a small data set, it is reasonable to do direct fitting at every data point. As there is substantial curvature in the data, quadratic local polynomials are used. An ODS OUTPUT statement creates two output data sets containing the "Output Statistics" and "Fit Summary" tables.

```
proc loess data=Gas;
   ods output OutputStatistics = GasFit
              FitSummary=Summary;
   model NOx = E / degree=2 direct smooth = 0.6 1.0
                   alpha=.01 all details;
run;
```

The "Fit Summary" table for smoothing parameter value 0.6, shown in Output 41.1.2, records the fitting parameters specified and some overall fit statistics.

Output 41.1.2. Fit Summary Table

```
                        The LOESS Procedure
                      Smoothing Parameter: 0.6
                      Dependent Variable: NOx

                            Fit Summary

          Fit Method                                  Direct
          Number of Observations                          22
          Degree of Local Polynomials                      2
          Smoothing Parameter                        0.60000
          Points in Local Neighborhood                    13
          Residual Sum of Squares                    1.71852
          Trace[L]                                   6.42184
          GCV                                        0.00708
          AICC                                      -0.45637
          AICC1                                     -9.39715
          Delta1                                    15.12582
          Delta2                                    14.73089
          Equivalent Number of Parameters            5.96950
          Lookup Degrees of Freedom                 15.53133
          Residual Standard Error                    0.33707
```

The matrix L referenced in the “Fit Summary” table is the smoothing matrix. This matrix satisfies

$$\hat{y} = Ly$$

where y is the vector of observed values and $\hat{y}$ is the corresponding vector of predicted values of the dependent variable. The quantities

$$\begin{aligned}
\delta_1 &\equiv \text{Trace}(I-L)^T(I-L) \\
\delta_2 &\equiv \text{Trace}\left((I-L)^T(I-L)\right)^2 \\
\rho &\equiv \text{Lookup Degrees of Freedom} \\
&\equiv \delta_1^2/\delta_2
\end{aligned}$$

in the “Fit Summary” table are used in doing statistical inference.

The equivalent number of parameters and residual standard error in the “Fit Summary” table are defined by

$$\begin{aligned}
\text{Equivalent Number of Parameters} &\equiv \text{Trace}L^TL \\
\text{Residual Standard Error} &\equiv \sqrt{\text{Residual SS}/\delta_1}
\end{aligned}$$

The “Output Statistics” table for smoothing parameter value 0.6 is shown in Output 41.1.3. Note that, as the ALL option in the MODEL statement is specified, this table includes all the relevant optional columns. Furthermore, because the ALPHA=0.01 option is specified in the MODEL statement, the confidence limits in this table are 99% limits.

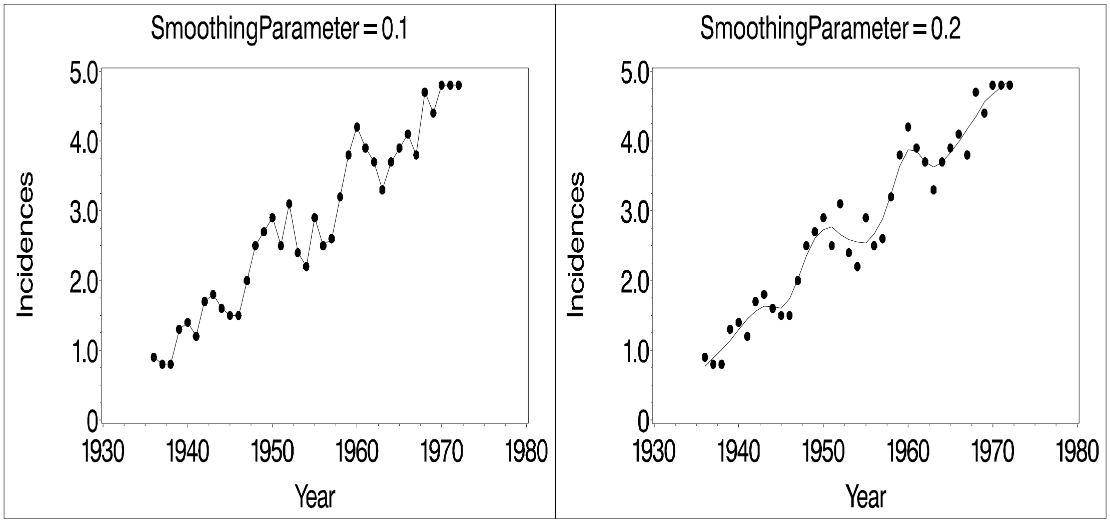
Output 41.1.3. Output Statistics Table

```
                              The LOESS Procedure
                            Smoothing Parameter: 0.6
                            Dependent Variable: NOx

                               Output Statistics

                                                 Estimated
                                 Predicted      Prediction
  Obs        E      NOx              NOx    Std Deviation       Residual    t Value

    1      0.8      4.8          4.87377          0.15528       -0.05577      -0.36
    2      1.0      2.8          2.81984          0.15380        0.02916       0.19
    3      1.0      3.3          3.48153          0.15187       -0.20653      -1.36
    4      1.0      4.7          4.73249          0.13923       -0.04149      -0.30
    5      0.8      4.3          4.82305          0.15278       -0.56805      -3.72
    6      0.9      5.1          5.18561          0.19337       -0.12161      -0.63
    7      0.7      2.1          2.51120          0.15528       -0.39320      -2.53
    8      0.8      4.6          4.48267          0.15285        0.11933       0.78
    9      1.1      2.3          2.12619          0.16683        0.15981       0.96
   10      1.1      1.0          0.97120          0.18134       -0.00120      -0.01
   11      1.0      4.0          4.09987          0.13477       -0.13487      -1.00
   12      0.9      5.3          5.31258          0.17283        0.03142       0.18
   13      0.8      3.8          3.84572          0.14929       -0.01172      -0.08
   14      0.7      2.0          2.26578          0.16712       -0.27578      -1.65
   15      0.8      5.2          4.58394          0.15363        0.61506       4.00
   16      0.9      5.3          5.24741          0.19319        0.03559       0.18
   17      1.0      3.8          4.16979          0.13478       -0.41779      -3.10
   18      1.2      0.5          0.53059          0.32170        0.00641       0.02
   19      1.1      1.6          1.83157          0.17127       -0.19157      -1.12
   20      1.0      5.1          4.66733          0.13735        0.38767       2.82
   21      1.0      4.9          4.52385          0.13556        0.41315       3.05
   22      0.7      1.6          1.19888          0.26774        0.36212       1.35

                               Output Statistics

                          Obs       99% Confidence Limits

                            1        4.41841        5.32912
                            2        2.36883        3.27085
                            3        3.03617        3.92689
                            4        4.32419        5.14079
                            5        4.37503        5.27107
                            6        4.61855        5.75266
                            7        2.05585        2.96655
                            8        4.03444        4.93089
                            9        1.63697        2.61541
                           10        0.43942        1.50298
                           11        3.70467        4.49507
                           12        4.80576        5.81940
                           13        3.40794        4.28350
                           14        1.77571        2.75584
                           15        4.13342        5.03445
                           16        4.68089        5.81393
                           17        3.77457        4.56502
                           18       -0.41278        1.47397
                           19        1.32933        2.33380
                           20        4.26456        5.07010
                           21        4.12632        4.92139
                           22        0.41375        1.98401
```

Plots of the data points and fitted models with 99% confidence bands are shown in Output 41.1.4.

```
proc sort data=GasFit;
   by SmoothingParameter E;
run;

symbol1 color=black value=dot ;
symbol2 color=black interpol=spline value=none;
symbol3 color=green interpol=spline value=none;
symbol4 color=green interpol=spline value=none;

%let opts=vaxis=axis1 hm=3 vm=3 overlay;

goptions nodisplay hsize=3.75;
axis1 label=(angle=90 rotate=0);

proc gplot data=GasFit;
   by SmoothingParameter;
   plot (DepVar Pred LowerCL UpperCL)*E/ &opts name='fitGas';
run; quit;

goptions display hsize=0 hpos=0;
proc greplay nofs tc=sashelp.templt template=h2;
    igout gseg;
    treplay 1:fitGas 2:fitGas1;
run; quit;
```

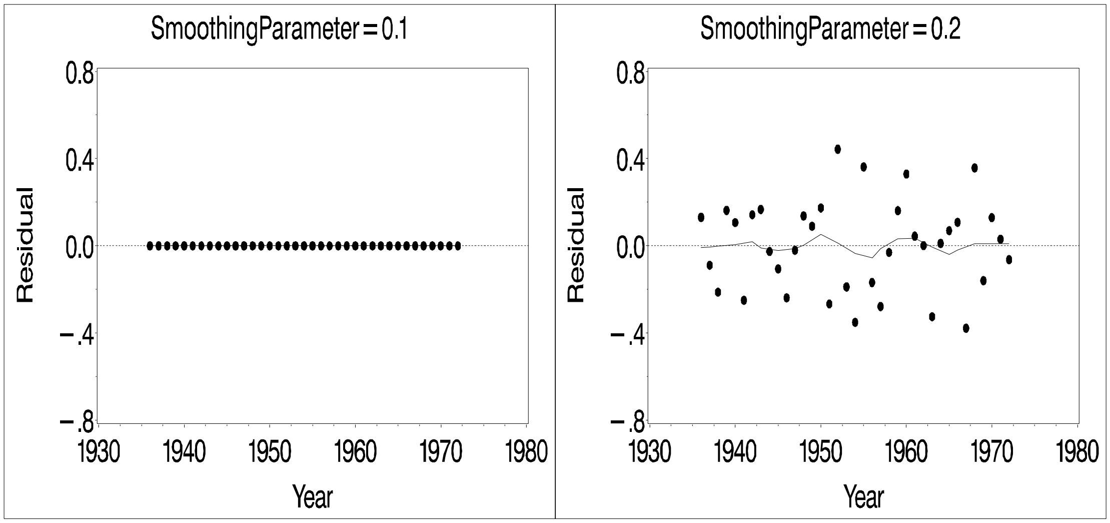
Output 41.1.4. Loess Fits with 99% Confidence Bands for Gas Data



It is evident from the preceding figure that the better fit is obtained with smoothing parameter value 0.6. Scatter plots of the fit residuals confirm this observation. Note also that PROC LOESS is again used to produce the Residual variable on these plots.

```
proc loess data=GasFit;
   by SmoothingParameter;
   ods output OutputStatistics=residout;
   model Residual=E;
run;

axis1 label = (angle=90 rotate=0)
      order = (-1 to 1 by 0.5);
goptions nodisplay hsize=3.75;
proc gplot data=residout;
   by SmoothingParameter;
   format DepVar 3.1;
   plot  DepVar*E Pred*E/ &opts vref=0 lv=2 vm=1
                           name='resGas';
run; quit;

goptions display hsize=0 hpos=0;
proc greplay nofs tc=sashelp.templt template=h2;
    igout gseg;
    treplay 1:resGas 2:resGas1;
run; quit;
```

Output 41.1.5. Scatter Plots of Loess Fit Residuals



The residual plots show that with smoothing parameter value 1, the loess model exhibits a lack of fit. Analysis of variance can be used to compare the model with smoothing parameter value 1, which serves as the null model, to the model with smoothing parameter value 0.6.

The statistic

$$F = \frac{(\text{rss}^{(n)} - \text{rss})/(\delta_1^{(n)} - \delta_1)}{\text{rss}/\delta_1}$$

has a distribution that is well approximated by an F distribution with

$$\nu = \frac{(\delta_1^{(n)} - \delta_1)^2}{\delta_2^{(n)} - \delta_2}$$

numerator degrees of freedom and ρ denominator degrees of freedom (Cleveland and Grosse 1991). Here quantities with superscript n refer to the null model, rss is the residual sum of squares, and δ_1, δ_2, and ρ are as previously defined.

The "Fit Summary" tables contain the information needed to carry out such an analysis. These tables have been captured in the output data set named Summary using an ODS OUTPUT statement. The following statements extract the relevant information from this data set and carry out the analysis of variance:

```
data h0 h1;
  set Summary(keep=SmoothingParameter Label1 nValue1
              where=(Label1 in ('Residual Sum of Squares','Delta1',
                     'Delta2','Lookup Degrees of Freedom')));
  if SmoothingParameter = 1 then output h0;
  else output h1;
run;

proc transpose data=h0(drop=SmoothingParameter Label1) out=h0;

data h0(drop=_NAME_); set h0;
  rename Col1 = RSSNull
         Col2 = delta1Null
         Col3 = delta2Null;

proc transpose data=h1(drop=SmoothingParameter Label1) out=h1;

data h1(drop=_NAME_); set h1;
  rename Col1 = RSS     Col2 = delta1
         Col3 = delta2  Col4 = rho;

data ftest; merge h0 h1;
  nu = (delta1Null - delta1)**2 / (delta2Null - delta2);
  Numerator = (RSSNull - RSS)/(delta1Null - delta1);
  Denominator = RSS/delta1;
  FValue = Numerator / Denominator;
  PValue = 1 - ProbF(FValue, nu, rho);
  label nu = 'Num DF'
        rho = 'Den DF'
        FValue = 'F Value'
        PValue = 'Pr > F';

proc print data=ftest label;
  var nu rho Numerator Denominator FValue PValue;
  format nu rho FValue 7.2 PValue 6.4;
run;
```

The results are shown in Output 41.1.6.

Output 41.1.6. Test ANOVA for LOESS MODELS of Gas Data

```
Obs      Num DF      Den DF     Numerator     Denominator     F Value     Pr > F

 1         2.67       15.53       1.05946         0.11362        9.32     0.0012
```

The highly significant p-value confirms that the loess model with smoothing parameter value 0.6 provides a better fit than the model with smoothing parameter value 1.

Example 41.2. Sulfate Deposits in the USA for 1990

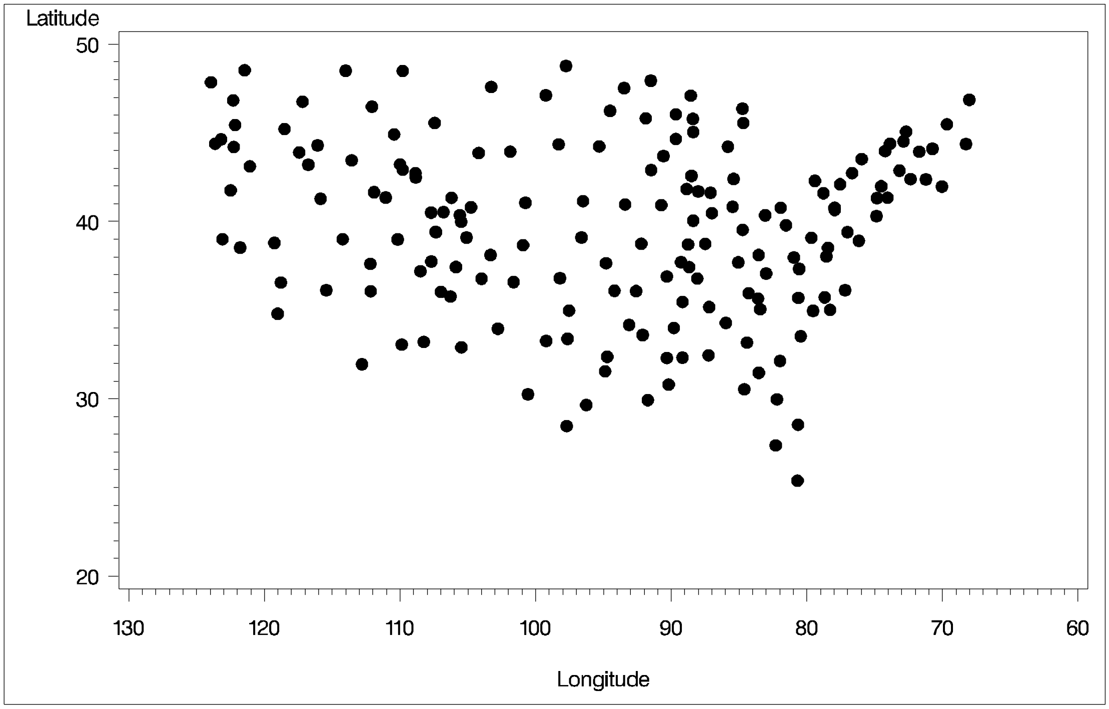
The following data set contains measurements in grams per square meter of sulfate (SO_4) deposits during 1990 at 179 sites throughout the 48 states.

```
data SO4;
   input Latitude Longitude SO4 @@;
datalines;
32.45833  87.24222 1.403 34.28778  85.96889 2.103
33.07139 109.86472 0.299 36.07167 112.15500 0.304
31.95056 112.80000 0.263 33.60500  92.09722 1.950
.
.      more data lines
.
42.92889 109.78667 0.182 43.22278 109.99111 0.161
43.87333 104.19222 0.306 44.91722 110.42028 0.210
45.07611  72.67556 2.646
;
```

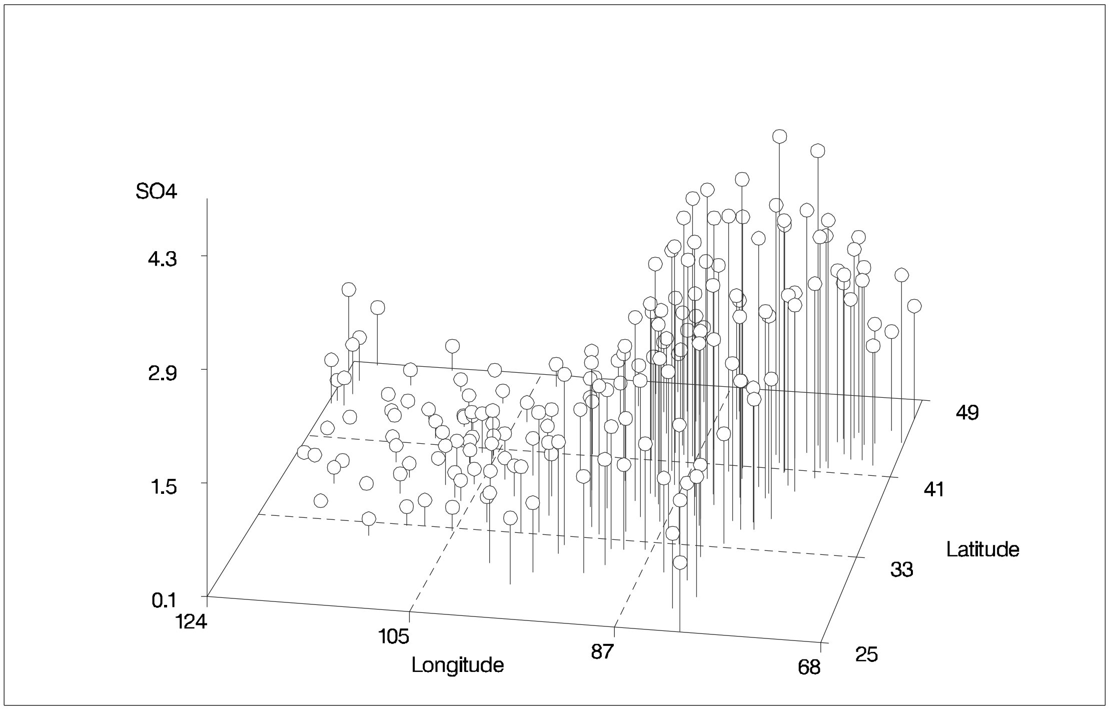
The following statements produce the two scatter plots of the SO4 data shown in Output 41.2.1 and Output 41.2.2:

```
symbol1 color=black value=dot ;
proc gplot data=SO4;
   plot Latitude*Longitude/hreverse;
run;

proc g3d data=SO4;
   format SO4 f4.1;
   scatter Longitude*Latitude=SO4 /
         shape='balloon'
         size=0.35
         rotate=80
         tilt=60;
run;
```

Output 41.2.1. Locations of Sulfate Measurements



Output 41.2.2. Scatter Plot of SO4 Data



From these scatter plots, it is clear that the largest concentrations are in the northeastern United States. These plots also indicate that a nonparametric surface, such as a loess fit, is appropriate for these data.

The sulfate measurements are irregularly spaced. The following statements create a SAS data set containing a regular grid of points that will be used in the SCORE statement:

```
data PredPoints;
   do Latitude = 26 to 46 by 1;
      do Longitude = 79 to 123 by 1;
         output;
      end;
   end;
```

The following statements fit loess models for two values of the smoothing parameter and save the results in output data sets:

```
proc loess data=SO4;
   ods Output ScoreResults=ScoreOut
              OutputStatistics=StatOut;
   model SO4=Latitude Longitude/smooth=0.15 0.4 residual;
   score data=PredPoints;
run;
```

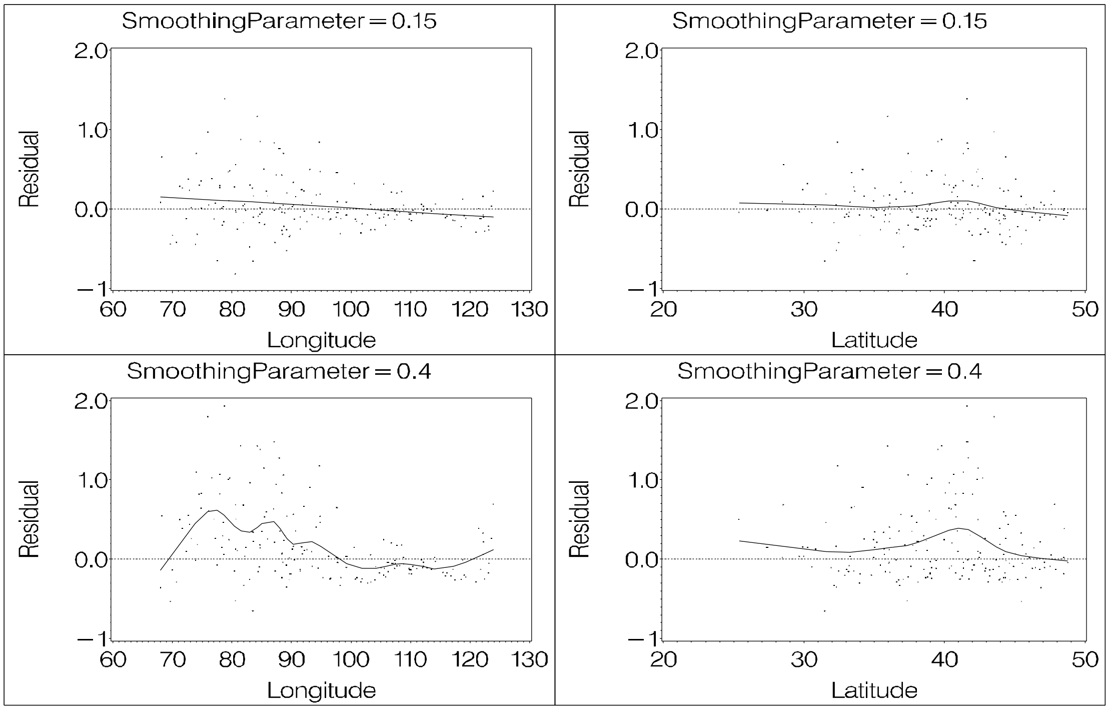
Notice that even though there are two predictors in the model, the SCALE= option is not appropriate because the predictors (Latitude and Longitude) are identically scaled.

Output 41.2.3 shows scatter plots of the fit residuals versus each of the predictors for the two smoothing parameter values specified. A loess fit of the residuals is also shown on these scatter plots and is obtained using PROC LOESS with the StatOut data set generated by the previous PROC LOESS step.

```
proc loess data=StatOut;
   by SmoothingParameter;
   ods output OutputStatistics=ResidLatOut;
   model residual=Latitude;
run;
proc loess data=StatOut;
   by SmoothingParameter;
   ods output OutputStatistics=ResidLongOut;
   model residual=Longitude;
run;
proc sort data=ResidLatOut;
   by SmoothingParameter Latitude;
run;
proc sort data=ResidLongOut;
   by SmoothingParameter Longitude;
run;
```

```
goptions nodisplay;
symbol1 color=black value=dot ;
symbol2 color=black interpol=join value=none;
%let opts = vaxis=axis1 overlay vref=0 lv=2;
axis1 label = (angle=90 rotate=0);

proc gplot data=ResidLatOut;
   by smoothingParameter;
   format DepVar 3.1;
   plot (DepVar Pred) * Latitude / &opts name='lat';
run;

proc gplot data=ResidLongOut;
   by smoothingParameter;
   format DepVar 3.1;
   plot (DepVar Pred) * Longitude / &opts name='long';
run;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
    igout gseg;
    treplay 1:long 2:long1 3:lat 4:lat1;
run; quit ;
```

Output 41.2.3. Scatter Plots of Loess Fit Residuals

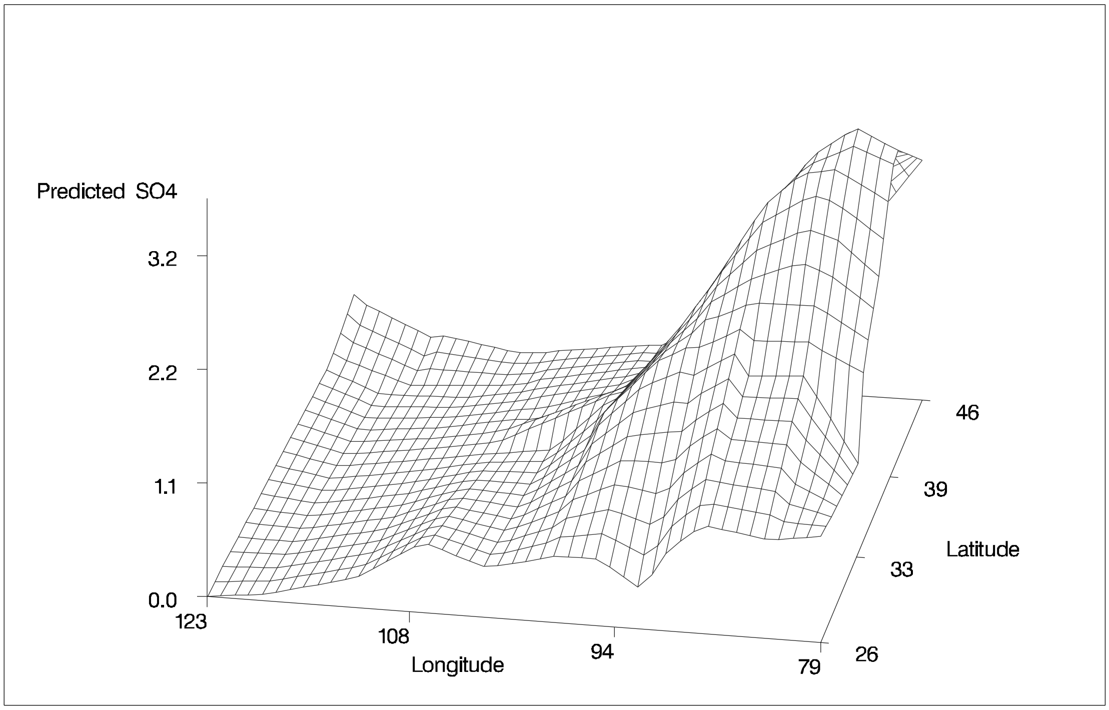


The scatter plots in Output 41.2.3 reveal that, with smoothing parameter value 0.4, there is significant information in the data that is not being captured by the loess model. By contrast, the residuals for the more localized smoothing parameter value 0.15 show a better fit.

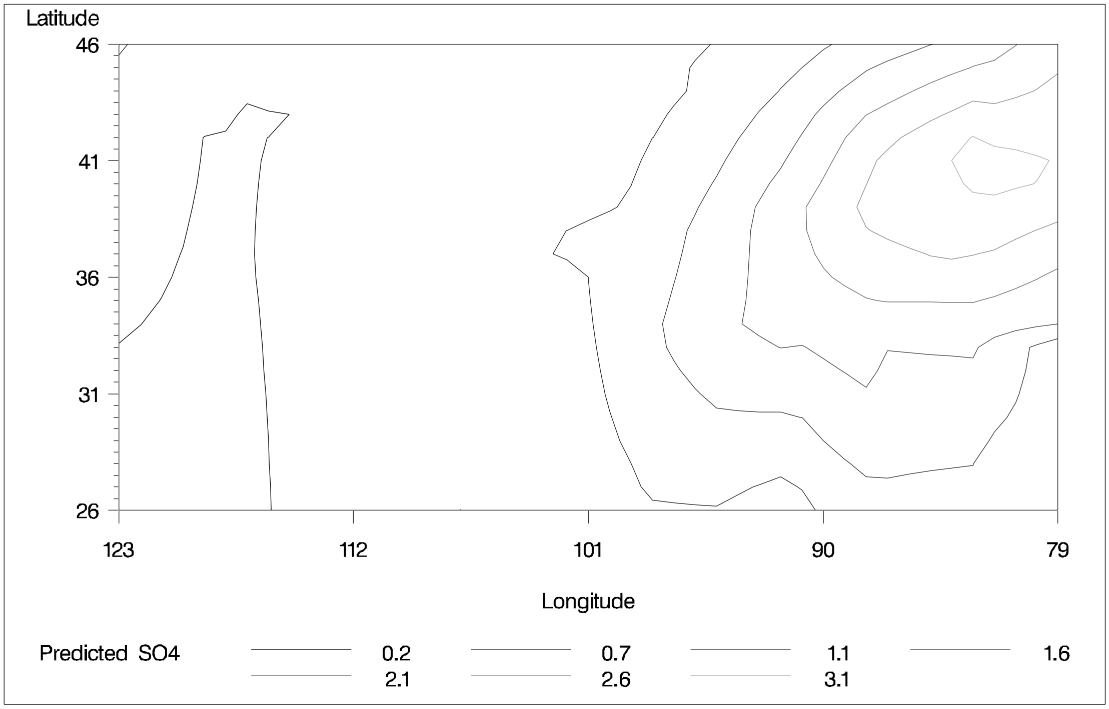
The ScoreOut data set contains the model predictions at the grid defined in the PredPoints data set. The following statements request a fitted surface and a contour plot of this surface with a smoothing parameter value 0.15:

```
proc g3d data=ScoreOut(where= (smoothingParameter=0.15));
   format Latitude f4.0;
   format Longitude f4.0;
   format p_SO4 f4.1;
   plot Longitude*Latitude=p_SO4/tilt=60 rotate=80;
run;

proc gcontour data=ScoreOut(where= (smoothingParameter=0.15));
   format latitude f4.0;
   format longitude f4.0;
   format p_SO4 f4.1;
   plot Latitude*Longitude = p_SO4/hreverse;
run;
```

Output 41.2.4. LOESS Fit of SO4 Data



Output 41.2.5. Contour Plot of LOESS Fit of SO4 Data



Example 41.3. Catalyst Experiment

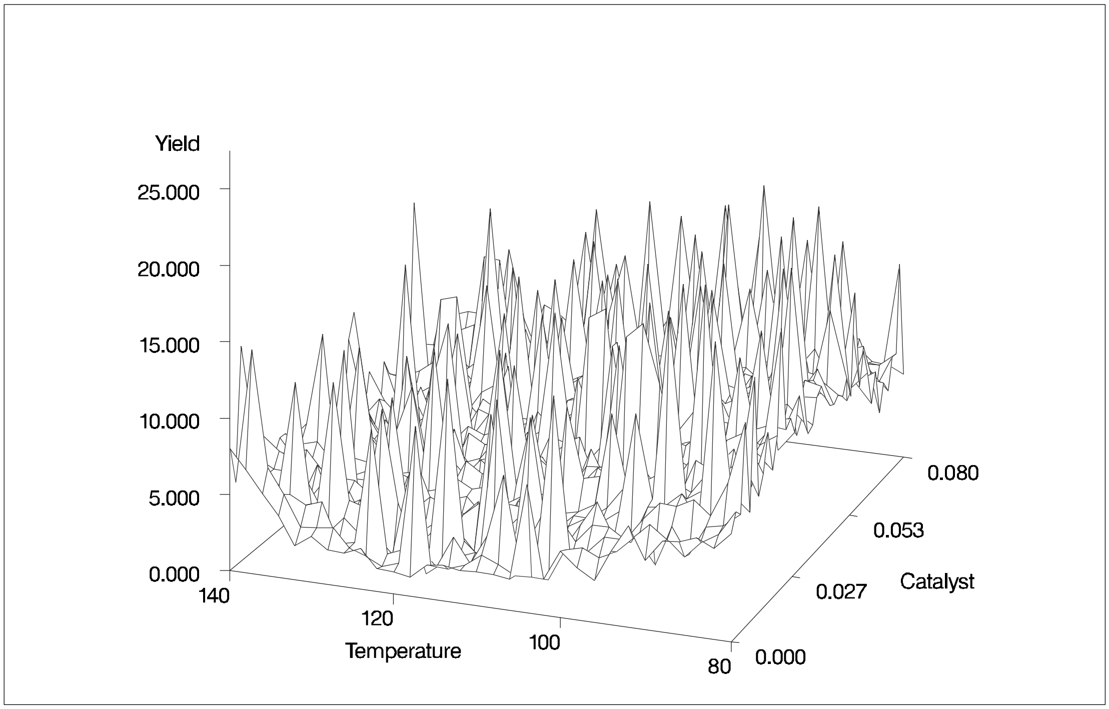
The following data set records the results of an experiment to determine how the yield of a chemical reaction varies with temperature and amount of a catalyst used.

```
data Experiment;
   input Temperature Catalyst Yield;
datalines;
  80      0.000      6.842
  80      0.002      7.944
  .
  .     more data lines
  .
 140      0.078      4.012
 140      0.080      5.212
;
```

Researchers know that about 10% of the yield measurements are corrupted due to an intermittent equipment problem. This can be seen in the surface plot of raw data shown in Output 41.3.1.

```
proc g3d data=Experiment;
   plot Temperature*Catalyst=Yield/zmin=0 zmax=25 zticknum=6;
run;
```

Output 41.3.1. Surface Plot of Experiment Data



A robust fitting method is needed to estimate the underlying surface in the presence of data outliers. The following statements invoke PROC LOESS with iterative reweighting to fit a surface to these data:

```
proc loess data=Experiment;
   ods output OutputStatistics=Results;
   model Yield = Temperature Catalyst /
                       scale=sd(0.1)
                       iterations=3;
run;
```

The ITERATIONS=3 option in the MODEL statement requests two iteratively reweighted iterations. Note the use of the SCALE=SD(0.1) option in the MODEL statement. This specifies that the independent variables in the model are to be divided by their respective 10% trimmed standard deviations before the fitted model is computed. This is appropriate as the independent variables Temperature and Catalyst are not similarly scaled. The “Scale Details” table produced by PROC LOESS is shown in Output 41.3.2.

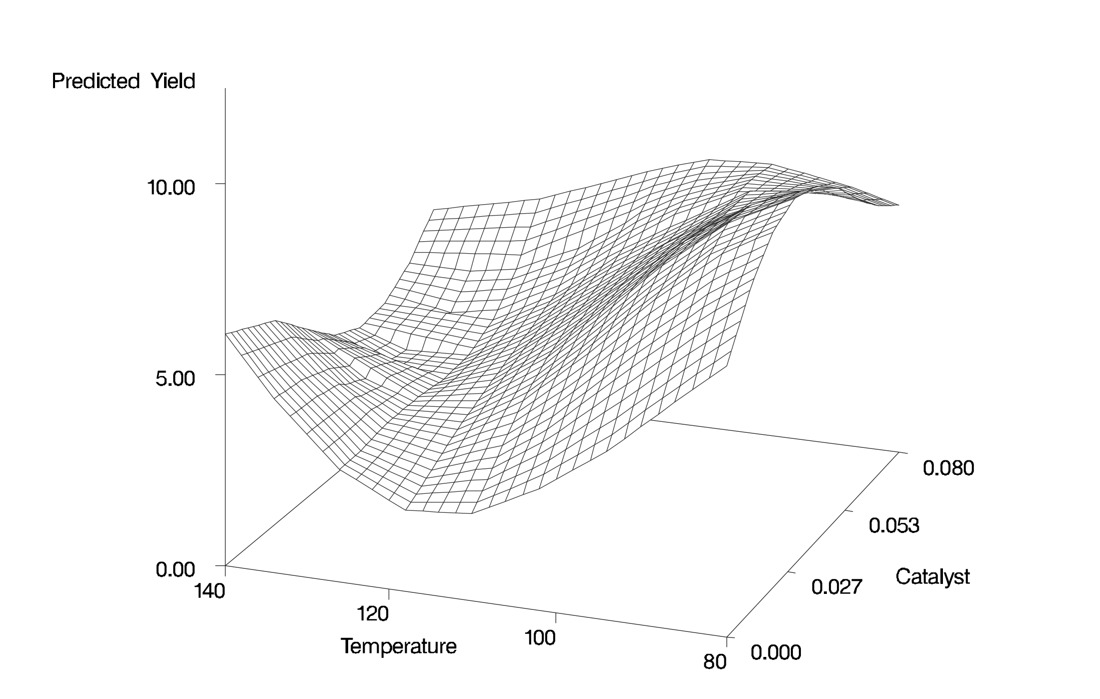
Output 41.3.2. Scale Details Table

```
                          The LOESS Procedure

                     Independent Variable Scaling

            Scaling applied: 10% trimmed standard deviation

         Statistic                     Temperature    Catalyst

         Minimum Value                          80       0.000
         Maximum Value                         140       0.080
         Trimmed Mean                          110       0.040
         Trimmed Standard Deviation             14       0.019
```

The following statements use the G3D procedure to plot the fitted surface shown in Output 41.3.3.

```
proc g3d data=Results;
   format Temperature f4.0;
   format Catalyst f6.3;
   format pred f5.2;
   plot Temperature*Catalyst=pred/zmin=0 zmax=10 zticknum=3;
run;
```

Output 41.3.3. Fitted Surface Plot for Experiment Data



Example 41.4. Automatic Smoothing Parameter Selection

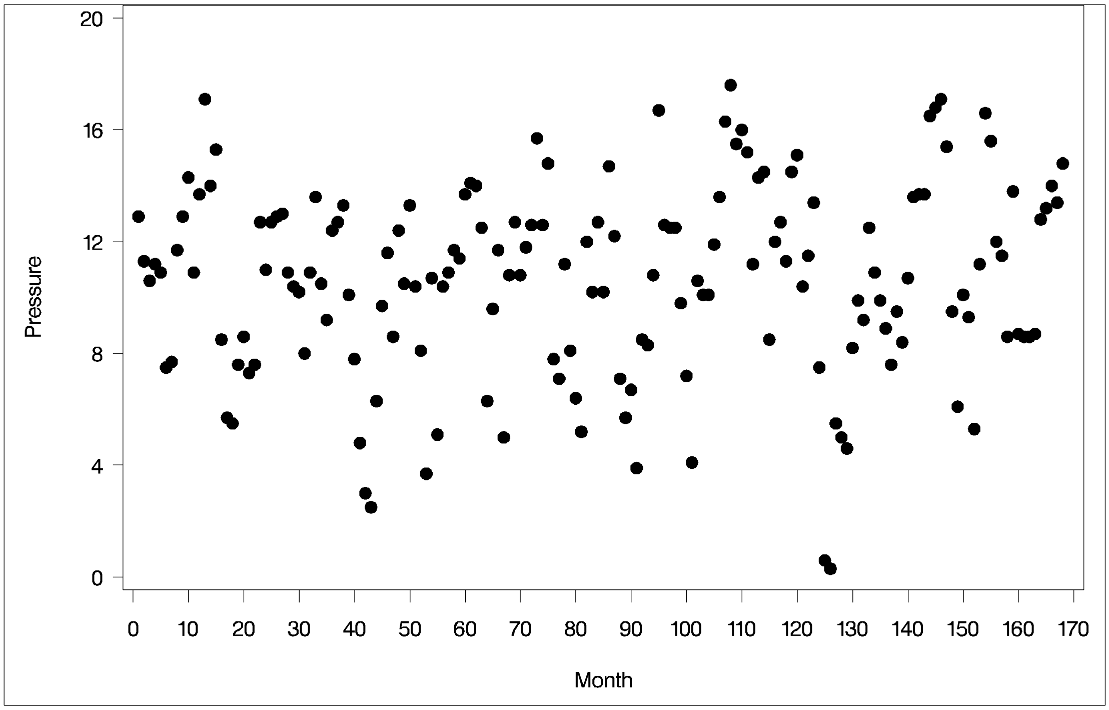
The following data set contains measurements of monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia, for a period of 168 months (National Institute of Standards and Technology 1998):

```
data ENSO;
  input Pressure @@;
  Month=_N_;
  format Pressure 4.1;
  format Month 3.0;
datalines;
12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7
12.9  14.3  10.9  13.7  17.1  14.0  15.3   8.5
 5.7   5.5   7.6   8.6   7.3   7.6  12.7  11.0
12.7  12.9  13.0  10.9  10.4  10.2   8.0  10.9
13.6  10.5   9.2  12.4  12.7  13.3  10.1   7.8
 4.8   3.0   2.5   6.3   9.7  11.6   8.6  12.4
10.5  13.3  10.4   8.1   3.7  10.7   5.1  10.4
10.9  11.7  11.4  13.7  14.1  14.0  12.5   6.3
 9.6  11.7   5.0  10.8  12.7  10.8  11.8  12.6
15.7  12.6  14.8   7.8   7.1  11.2   8.1   6.4
 5.2  12.0  10.2  12.7  10.2  14.7  12.2   7.1
 5.7   6.7   3.9   8.5   8.3  10.8  16.7  12.6
12.5  12.5   9.8   7.2   4.1  10.6  10.1  10.1
11.9  13.6  16.3  17.6  15.5  16.0  15.2  11.2
14.3  14.5   8.5  12.0  12.7  11.3  14.5  15.1
10.4  11.5  13.4   7.5   0.6   0.3   5.5   5.0
 4.6   8.2   9.9   9.2  12.5  10.9   9.9   8.9
 7.6   9.5   8.4  10.7  13.6  13.7  13.7  16.5
16.8  17.1  15.4   9.5   6.1  10.1   9.3   5.3
11.2  16.6  15.6  12.0  11.5   8.6  13.8   8.7
 8.6   8.6   8.7  12.8  13.2  14.0  13.4  14.8
;
```

The following PROC GPLOT statements produce the simple scatter plot of these data, displayed in Output 41.4.1:

```
symbol1 color=black value=dot ;
proc gplot data=ENSO;
   plot Pressure*Month /
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);;
run;
```

Output 41.4.1. Scatter Plot of ENSO Data



You can compute a loess fit and plot the results for these data using the following statements:

```
ods output OutputStatistics=ENSOstats;

proc loess data=ENSO;
   model Pressure=Month ;
run;

symbol1 color=black value=dot h=2.5 pct;
symbol2 color=black interpol=join value=none width=2;
proc gplot data=ENSOstats;
   plot (depvar pred)*Month / overlay
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);
run; quit;
```

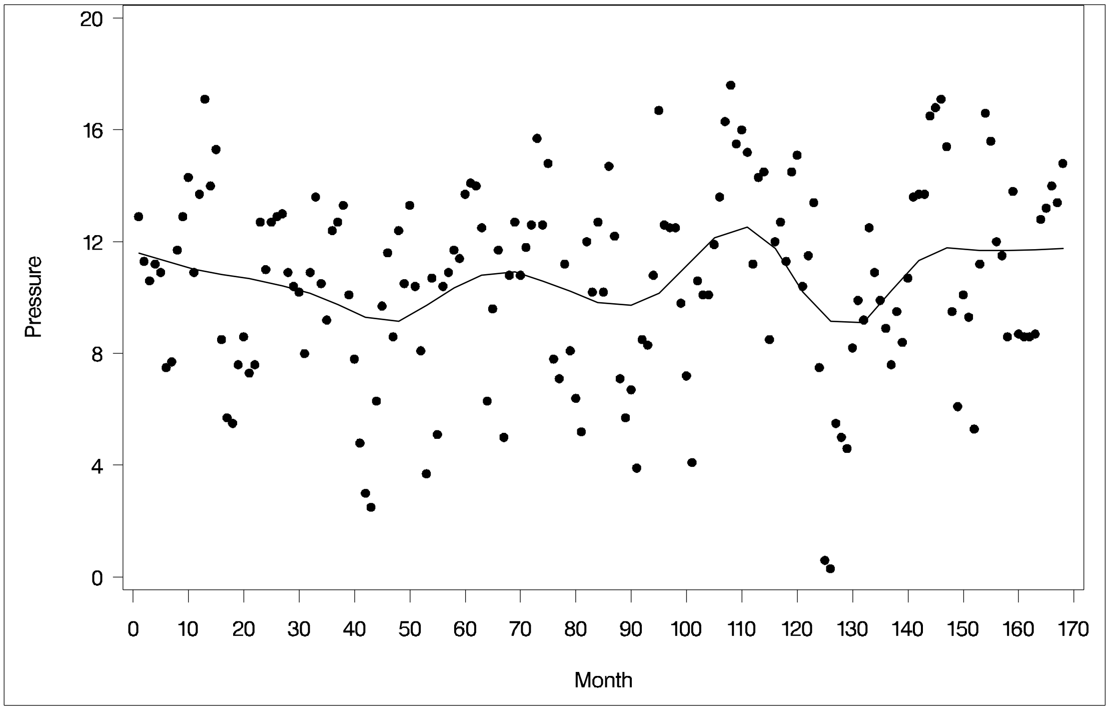
The “Smoothing Criterion” and “Fit Summary” tables are shown in Output 41.4.2 and the fit is plotted in Output 41.4.3.

Output 41.4.2. Output from PROC LOESS

```
                        The LOESS Procedure
                    Dependent Variable: Pressure

                          Optimal Smoothing
                              Criterion

                                     Smoothing
                          AICC       Parameter

                       3.41105         0.22321



                        The LOESS Procedure
                Selected Smoothing Parameter: 0.223
                    Dependent Variable: Pressure

                            Fit Summary

        Fit Method                              kd Tree
        Blending                                 Linear
        Number of Observations                      168
        Number of Fitting Points                     33
        kd Tree Bucket Size                           7
        Degree of Local Polynomials                   1
        Smoothing Parameter                     0.22321
        Points in Local Neighborhood                 37
        Residual Sum of Squares              1654.27725
        Trace[L]                                8.74180
        GCV                                     0.06522
        AICC                                    3.41105
```

The smoothing parameter value used for the loess fit shown in Output 41.4.3 was chosen using the default method of PROC LOESS, namely a golden section minimization of the AICC criterion over the interval $(0, 1]$. The fit seems to be oversmoothed. What accounts for this poor fit?

Output 41.4.3. Oversmoothed Loess Fit for the ENSO Data

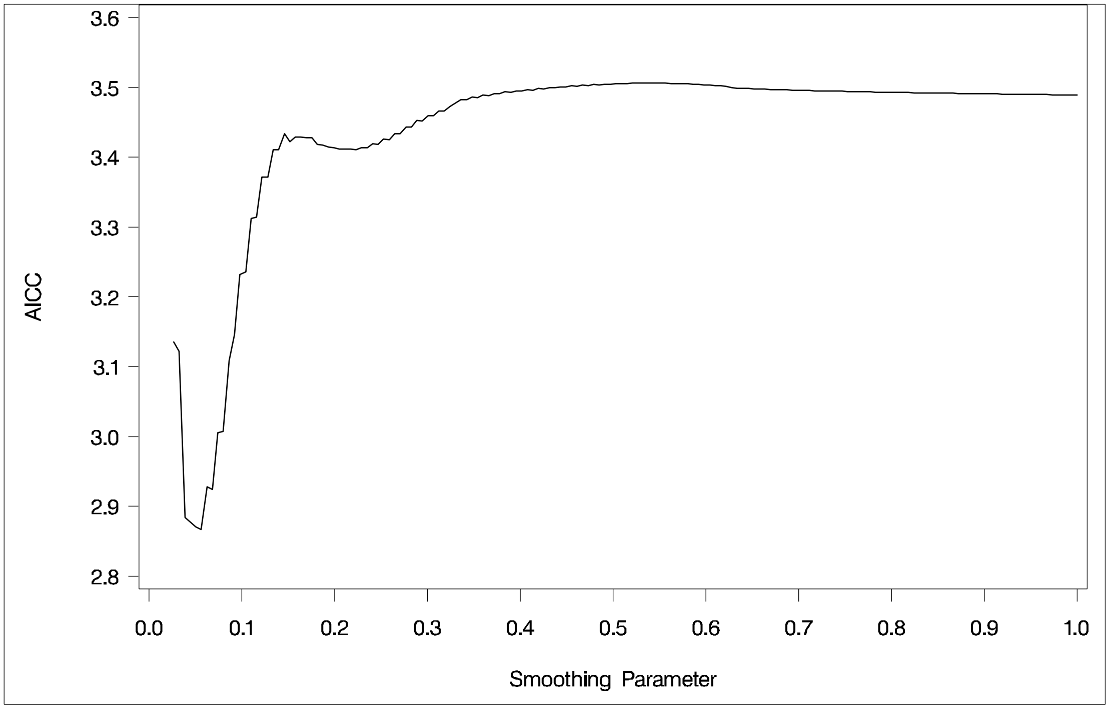


One possibility is that the golden section search has found a local rather than a global minimum of the AICC criterion. You can test this by redoing the fit requesting a global minimum. It is also helpful to plot the AICC criterion as a function of the smoothing parameter value used. You do this with the following statements:

```
ods output ModelSummary=ENSOsummary;

proc loess data=ENSO;
   model Pressure=Month/select=AICC(global);
run;

proc sort data=ENSOsummary;
    by smooth;
run;

symbol1 color=black interpol=join value=none width=2;
proc gplot data=ENSOsummary;
   format AICC f4.1; format smooth f4.1;
   plot AICC*Smooth /
        hminor = 0 vminor = 0 vaxis  = axis1 frame;
        axis1 label = ( r=0 a=90 );
run; quit;
```

The results are shown in Output 41.4.4.

Output 41.4.4. AICC versus Smoothing Parameter Showing Local Minima

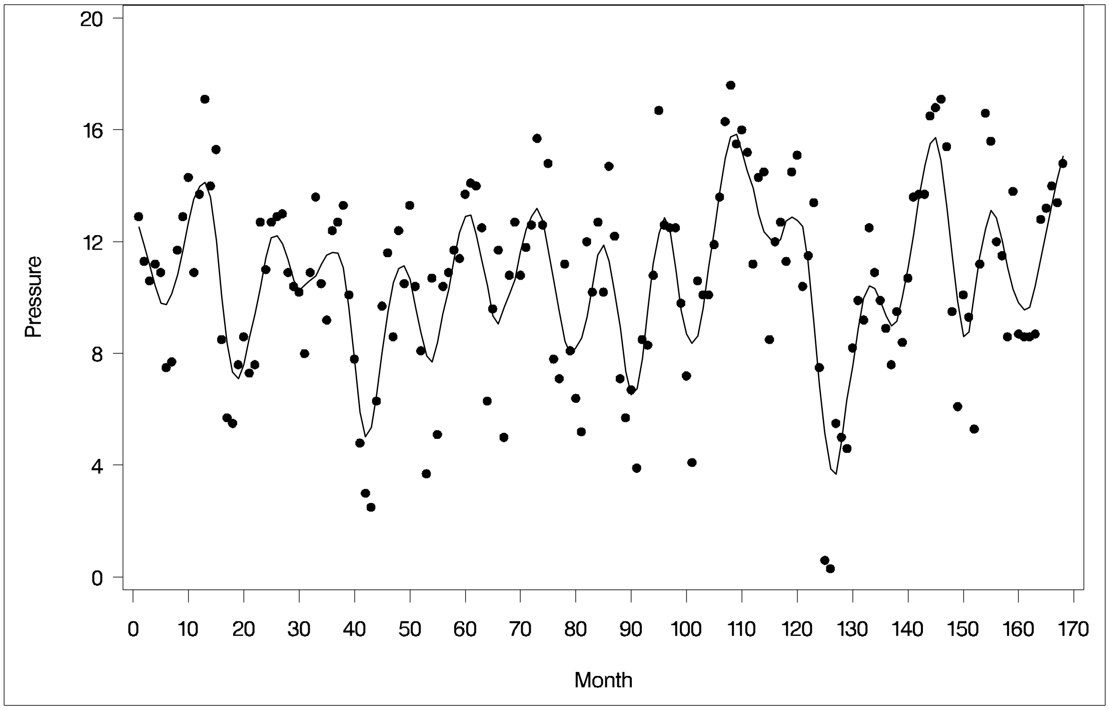


The explanation for the oversmoothed fit in Output 41.4.3 is now apparent. The golden section search algorithm found the local minimum that occurs near the value 0.22 of the smoothing parameter rather than the global minimum that occurs near 0.06. Note that if you restrict the range of smoothing parameter values examined to lie below 0.2, then the golden section search finds the global minimum as the following statements demonstrate:

```
ods output OutputStatistics=ENSOstats;

proc loess data=ENSO;
   model Pressure=Month/select=AICC( range(0.03,0.2) );
run;

symbol1 color=black value=dot h=2.5 pct;
symbol2 color=black interpol=join value=none width=2;
proc gplot data=ENSOstats;
   plot (depvar pred)*Month / overlay
        hminor = 0
        vminor = 0
        vaxis  = axis1
        frame;
        axis1 label = ( r=0 a=90 ) order=(0 to 20  by 4);
run; quit;
```

The fit obtained is shown in Output 41.4.5.

Output 41.4.5. Loess Fit for the ENSO Data



The loess fit shown in Output 41.4.5 clearly shows an annual cycle in the data. An interesting question is whether there is some phenomenon captured in the data that would explain the presence of the local minimum near 0.22 in the AICC curve. Note that there is some evidence of a cycle of about 42 months in the oversmoothed fit in Output 41.4.3. You can see this cycle because the strong annual cycle in Output 41.4.5 has been smoothed out. The physical phenomenon that accounts for the existence of this cycle has been identified as the periodic warming of the Pacific Ocean known as "El Niño."

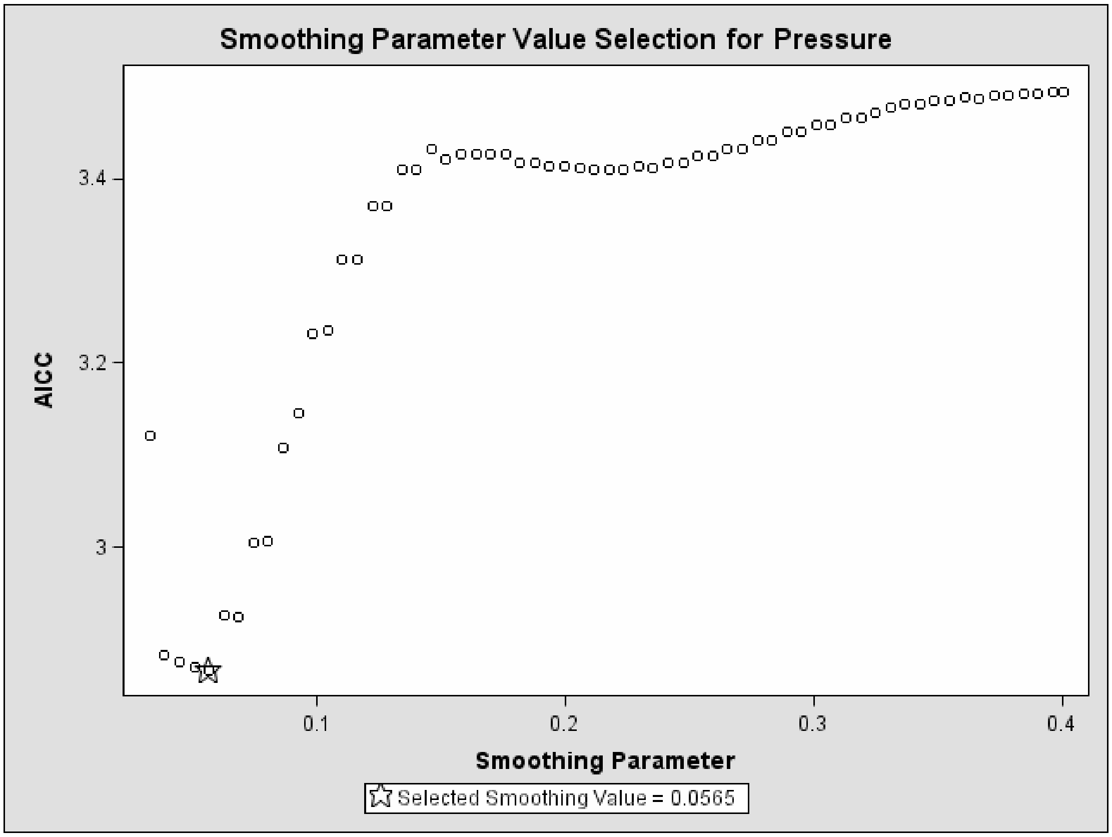
Example 41.5. ODS Graphics

This example highlights the use of ODS for creating statistical graphs with the LOESS procedure. The ENSO example is revisited to show how these graphics can be used to enrich the analysis and simplify the process for obtaining functionally equivalent plots to those previously presented with this example.

To request these plots you to need to specify the experimental ODS GRAPHICS statement. For general information about ODS graphics see Chapter 15, "Statistical Graphics Using ODS." The following statements produce the default plots:

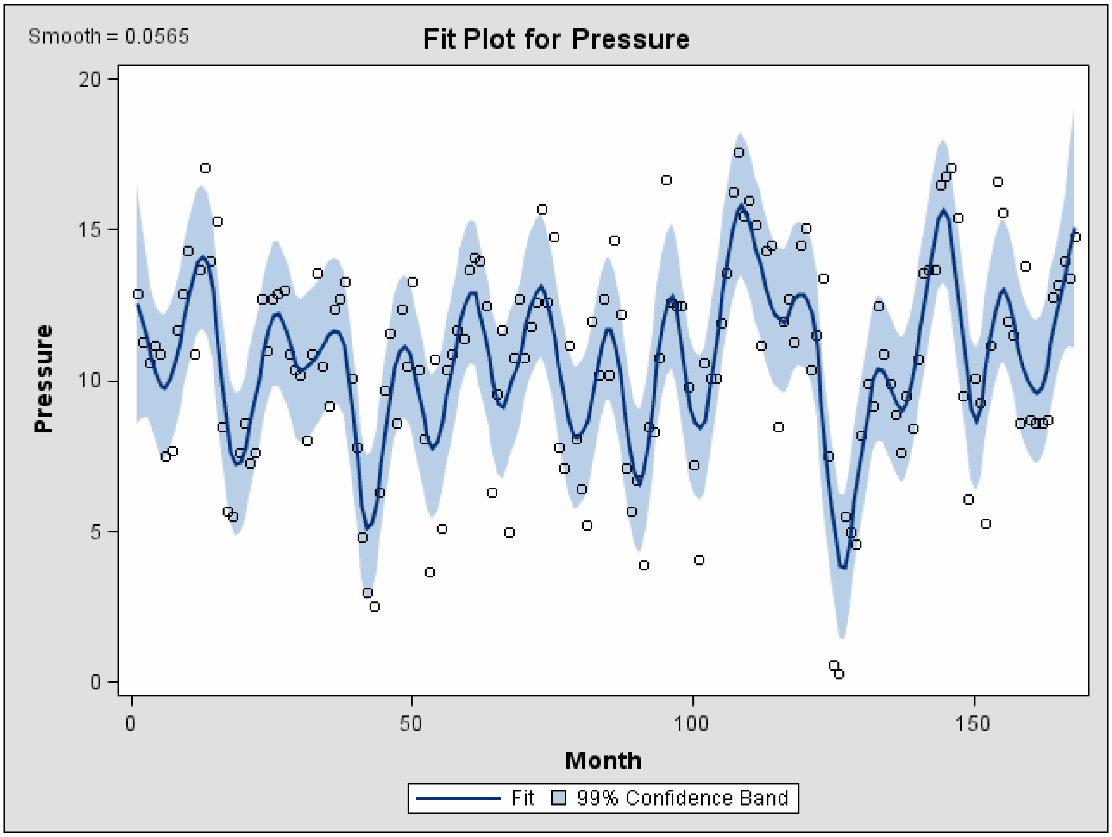
```
ods html;
ods graphics on;

proc loess data=ENSO;
   model Pressure=Month/select=AICC( range(0.03,0.4) global) clm
                             alpha=0.01;
run;

ods graphics off;
ods html close;
```

Output 41.5.1. Smoothing Parameter Selection (Experimental)



Note that Output 41.5.1 has points corresponding to all models with smoothing parameters between 0.03 and 0.4, as these models were requested in the SELECT= option. By default, only the models evaluated in the golden section search for the optimal smoothing parameter appear on this plot.

Output 41.5.2. LOESS Fit of ENSO Data (Experimental)

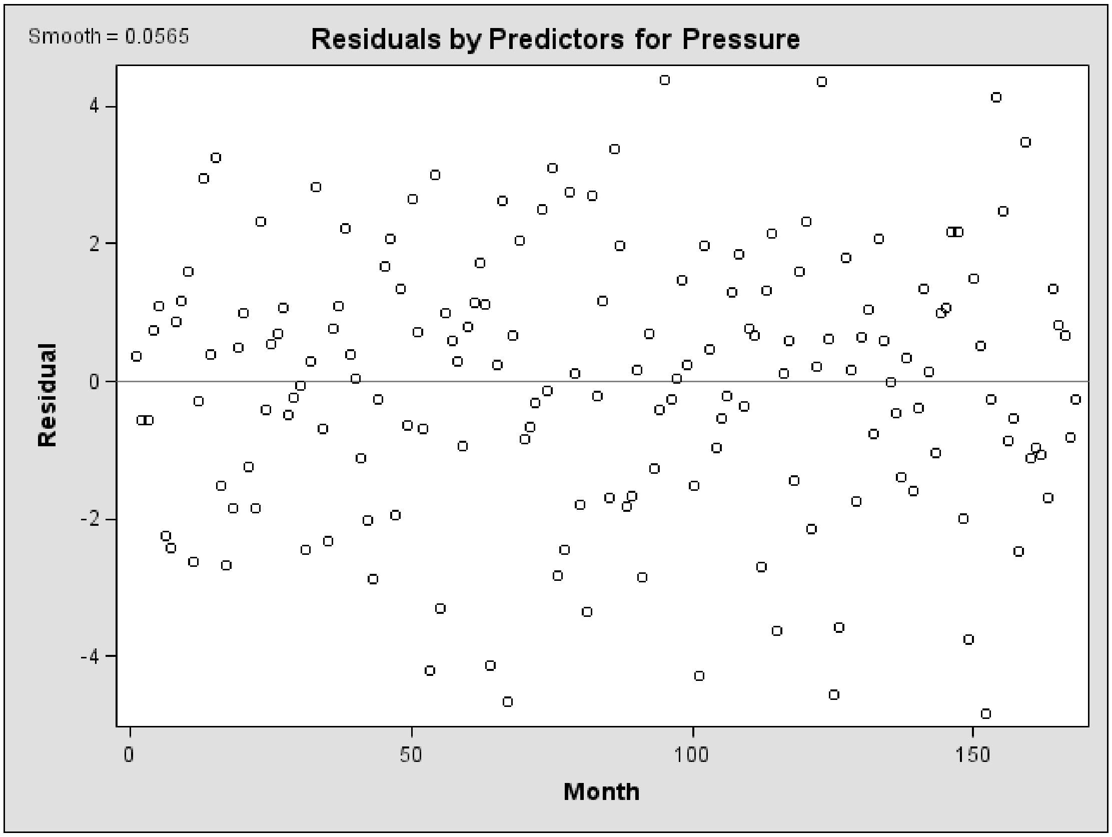


The fit plot shown in Output 41.5.2 is produced for models with a single regressor. Note that Output 41.5.2 includes the 99% confidence band that was requested in the MODEL statement using the CLM and ALPHA= options.

Diagnostic plots are produced when the RESIDUAL option is included in the model statement. The following statements produce these diagnostic plots:

```
ods html;
ods graphics on;

proc loess data=ENSO;
   model Pressure=Month/smooth = 0.0565 residual;
run;

ods graphics off;
ods html close;
```

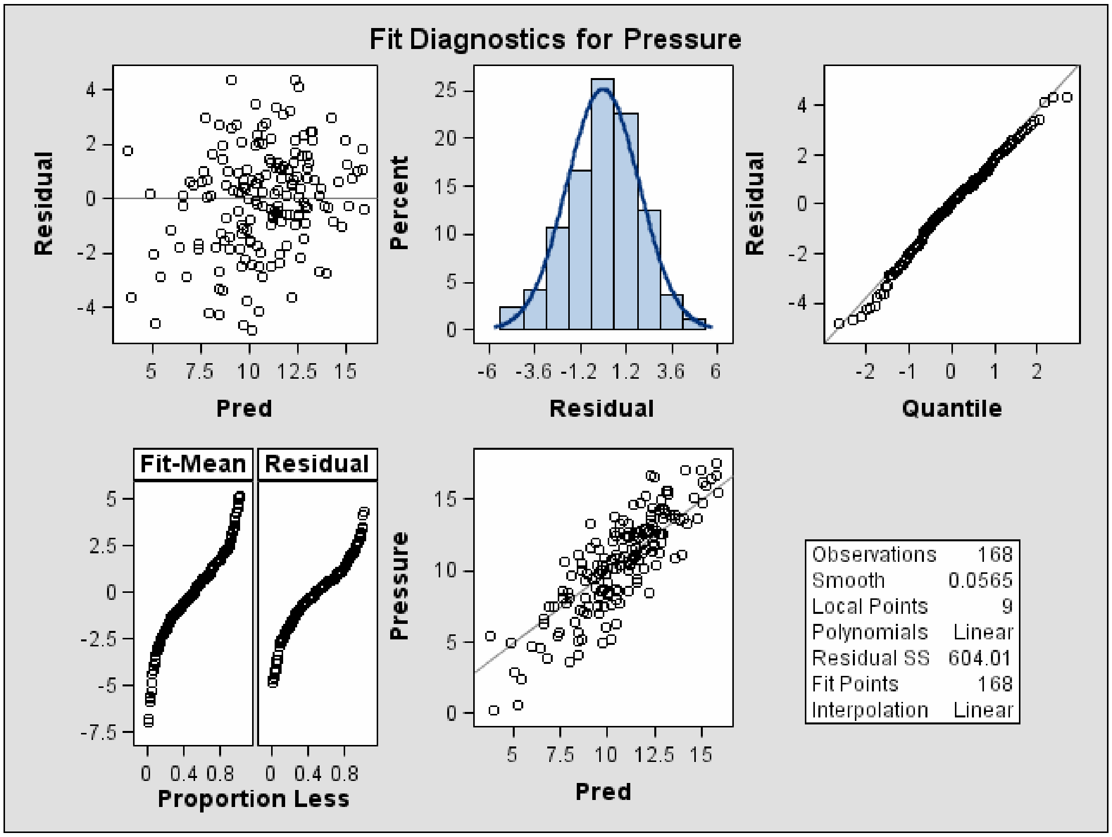
Output 41.5.3. Residuals by Regressors (Experimental)



The residuals in Output 41.5.3 do not exhibit any obvious patterns, suggesting that the signal in the data has been successfully modeled.

Additional information about the LOESS fit can be seen in the plots in the fit diagnostics panel that is also produced whenever the RESIDUAL option is specified in the MODEL statement. Output 41.5.4 shows this panel for the optimal smoothing parameter.

Output 41.5.4. Fit Diagnostics Panel (Experimental)



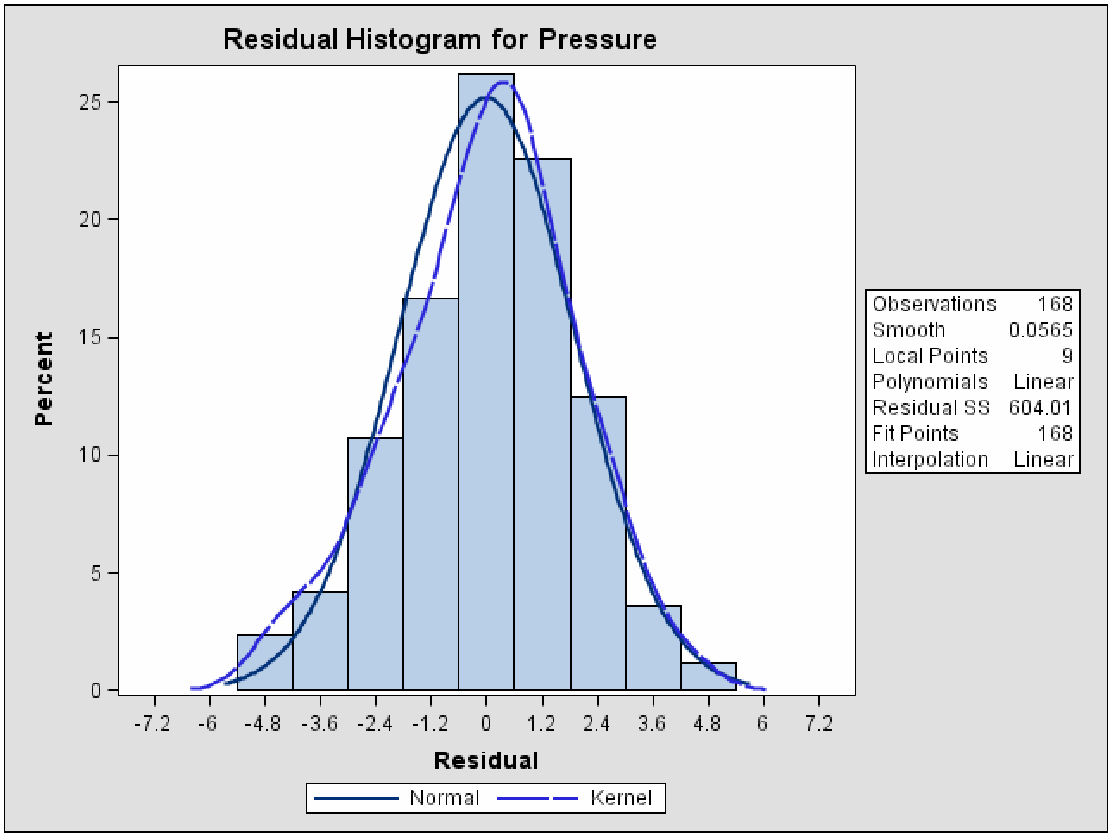
These diagnostic plots all suggest that the fit obtained is appropriate for these data:

- The plot of residuals versus predicted value shows no obvious pattern.
- The residual histogram with overlayed normal density supports the assumption of gaussian errors.
- The normal quantile plot is consistent with the assumption of gaussian errors.
- The “Residual-Fit” (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals shows that the spread in the residuals is no greater than the spread in the centered fit. For inappropriate models, the spread of the residuals in such a plot is often greater than the spread of the centered fit.
- The plot of the dependent variable versus the predicted values is centered around a 45 degree line and shows no obvious outliers.

If you want to obtain the plots in the Diagnostics Panel as individual plots, you can do so by specifying the PLOTS(UNPACKPANELS) option in the PROC LOESS statement. The following statements provide an example:

```
ods html;
ods graphics on;

proc loess data=ENSO plots(unpackpanels);
   model Pressure=Month/smooth = 0.0565 residual;
run;

ods graphics off;
ods html close;
```

Output 41.5.5. Residual Histogram (Experimental)



The residual histogram is shown in Output 41.5.5.

References

Akaike, H. (1973), “Information Theory and an Extension of the Maximum Likelihood Principle,” in Petrov and Csaki, eds., “Proceedings of the Second International Symposium on Information Theory,” 267–281.

Brinkman, N. D. (1981), “Ethanol Fuel—A Single-Cylinder Engine Study of Efficiency and Exhaust Emissions,” *Society of Automotive Engineers Transactions*, 90, 1410–1424.

Cleveland, W. (1993), *Visualizing Data*, Summit, NJ: Hobart Press.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988), “Regression by Local Fitting,” *Journal of Econometrics*, 37, 87–114.

Cleveland, W. S. and Grosse, E. (1991), “Computational Methods for Local Regression,” *Statistics and Computing*, 1, 47–62.

Cleveland, W. S., Grosse, E., and Shyu, M.-J. (1992), “A Package of C and Fortran Routines for Fitting Local Regression Models,” Unpublished.

Craven, P. and Wahba, G. (1979), “Smoothing Noisy Data with Spline Functions,” *Numerical Mathematics*, 31, 377–403.

Gordon, W. J. (1971), “Blending-function Methods of Bivariate and Multivariate Interpolation and Approximation,” *SIAM Journal of Numerical Analysis*, 8, No. 1, 158–177.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, New York: Chapman and Hall.

Houghton, A. N., Flannery, J., and Viola, M. V. (1980), “Malignant Melanoma in Connecticut and Denmark,” *International Journal of Cancer*, 25, 95–104.

Hurvich, C. M., Simonoff, J. S., and Tsai, C. L. (1998), “Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion,” *Journal of the Royal Statistical Society B*, 60, 271–293.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, http://www.nist.gov/itl/div898/strd: last accessed Jan 20, 1998.

Ruppert, D., Sheather, S. J., and Wand, M. P. (1995), “An Effective Bandwidth Selector for Local Least Squares Regression,” *Journal of the American Statistical Association*, 90, 1257–1270.

Chapter 42
The LOGISTIC Procedure

Chapter Contents

Chapter 42

The LOGISTIC Procedure

Overview

Binary responses (for example, success and failure), ordinal responses (for example, normal, mild, and severe), and nominal responses (for example, major TV networks viewed at a certain hour) arise in many fields of study. Logistic regression analysis is often used to investigate the relationship between these discrete responses and a set of explanatory variables. Several texts that discuss logistic regression are Collett (1991), Agresti (1990), Cox and Snell (1989), Hosmer and Lemeshow (2000), and Stokes, Davis, and Koch (2000).

For binary response models, the response, Y, of an individual or an experimental unit can take on one of two possible values, denoted for convenience by 1 and 2 (for example, $Y = 1$ if a disease is present, otherwise $Y = 2$). Suppose $\mathbf{x}$ is a vector of explanatory variables and $\pi = \Pr(Y = 1 \mid \mathbf{x})$ is the response probability to be modeled. The linear logistic model has the form

$$\text{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \boldsymbol{\beta}'\mathbf{x}$$

where α is the intercept parameter and $\boldsymbol{\beta}$ is the vector of parameters. Notice that the LOGISTIC procedure, by default, models the probability of the *lower* response levels.

The logistic model shares a common feature with a more general class of linear models, that a function $g = g(\mu)$ of the mean of the response variable is assumed to be linearly related to the explanatory variables. Since the mean μ implicitly depends on the stochastic behavior of the response, and the explanatory variables are assumed to be fixed, the function g provides the link between the random (stochastic) component and the systematic (deterministic) component of the response variable Y. For this reason, Nelder and Wedderburn (1972) refer to $g(\mu)$ as a link function. One advantage of the logit function over other link functions is that differences on the logistic scale are interpretable regardless of whether the data are sampled prospectively or retrospectively (McCullagh and Nelder 1989, Chapter 4). Other link functions that are widely used in practice are the probit function and the complementary log-log function. The LOGISTIC procedure enables you to choose one of these link functions, resulting in fitting a broader class of binary response models of the form

$$g(\pi) = \alpha + \boldsymbol{\beta}'\mathbf{x}$$

For ordinal response models, the response, Y, of an individual or an experimental unit may be restricted to one of a (usually small) number, $k + 1 (k \geq 1)$, of ordinal values, denoted for convenience by $1, \ldots, k, k + 1$. For example, the severity of coronary disease can be classified into three response categories as 1=no disease, 2=angina pectoris, and 3=myocardial infarction. The LOGISTIC procedure fits a common slopes cumulative model, which is a parallel lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$g(\Pr(Y \le i \mid \mathbf{x})) = \alpha_i + \boldsymbol{\beta}'\mathbf{x}, \quad i = 1, \ldots, k$$

where $\alpha_1, \ldots, \alpha_k$ are k intercept parameters, and $\boldsymbol{\beta}$ is the vector of parameters. This model has been considered by many researchers. Aitchison and Silvey (1957) and Ashford (1959) employ a probit scale and provide a maximum likelihood analysis; Walker and Duncan (1967) and Cox and Snell (1989) discuss the use of the log-odds scale. For the log-odds scale, the cumulative logit model is often referred to as the *proportional odds* model.

For nominal response logistic models, where the $k+1$ possible responses have no natural ordering, the logit model can also be extended to a *generalized logit* model, which has the form

$$\log\left(\frac{\Pr(Y = i \mid \mathbf{x})}{\Pr(Y = k+1 \mid \mathbf{x})}\right) = \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}, \quad i = 1, \ldots, k$$

where the $\alpha_1, \ldots, \alpha_k$ are k intercept parameters, and the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k$ are k vectors of parameters. These models were introduced by McFadden (1974) as the *discrete choice* model, and they are also known as *multinomial* models.

The LOGISTIC procedure fits linear logistic regression models for discrete response data by the method of maximum likelihood. It can also perform conditional logistic regression for binary response data and exact conditional logistic regression for binary and nominal response data. The maximum likelihood estimation is carried out with either the Fisher-scoring algorithm or the Newton-Raphson algorithm. You can specify starting values for the parameter estimates. The logit link function in the logistic regression models can be replaced by the probit function, the complementary log-log function, or the generalized logit function.

The LOGISTIC procedure enables you to specify categorical variables (also known as CLASS variables) or continuous variables as explanatory variables. You can also specify more complex model terms such as interactions and nested terms in the same way as in the GLM procedure. Any term specified in the model is referred to as an *effect*, whether it is a continuous variable, a CLASS variable, an interaction, or a nested term.

The LOGISTIC procedure allows either a full-rank parameterization or a less-than-full-rank parameterization. The full-rank parameterization offers eight coding methods: effect, reference, ordinal, polynomial, and orthogonalizations of these. The effect coding is the same method that is used in the CATMOD procedure. The less-than-full-rank parameterization is the same coding as that used in the GLM procedure.

The LOGISTIC procedure provides four effect selection methods: forward selection, backward elimination, stepwise selection, and best subset selection. The best subset selection is based on the likelihood score statistic. This method identifies a specified number of best models containing one, two, three effects, and so on, up to a single model containing effects for all the explanatory variables.

The LOGISTIC procedure has some additional options to control how to move effects in and out of a model with various model-building strategies such as forward selection, backward elimination, or stepwise selection. When there are no interaction terms, a main effect can enter or leave a model in a single step based on the p-value of the score or Wald statistic. When there are interaction terms, the selection process also depends on whether you want to preserve model hierarchy. These additional options enable you to specify whether model hierarchy is to be preserved, how model hierarchy is applied, and whether a single effect or multiple effects can be moved in a single step.

Odds ratio estimates are displayed along with parameter estimates. You can also specify the change in the explanatory variables for which odds ratio estimates are desired. Confidence intervals for the regression parameters and odds ratios can be computed based either on the profile likelihood function or on the asymptotic normality of the parameter estimators.

Various methods to correct for overdispersion are provided, including Williams' method for grouped binary response data. The adequacy of the fitted model can be evaluated by various goodness-of-fit tests, including the Hosmer-Lemeshow test for binary response data.

Like many procedures in SAS/STAT software that enable the specification of CLASS variables, the LOGISTIC procedure provides a CONTRAST statement for specifying customized hypothesis tests concerning the model parameters. The CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful for obtaining odds ratio estimates for various levels of the CLASS variables.

You can perform a conditional logistic regression on binary response data by specifying the STRATA statement. This enables you to perform matched-set and case-control analyses. The number of events and nonevents can vary across the strata. Many of the features available with the unconditional analysis are also available with a conditional analysis.

The LOGISTIC procedure enables you to perform exact conditional logistic regression using the method of Hirji, Mehta, and Patel (1987) and Mehta, Patel, and Senchaudhuri (1992) by specifying one or more EXACT statements. You can test individual parameters or conduct a joint test for several parameters. The procedure computes two exact tests: the exact conditional score test and the exact conditional probability test. You can request exact estimation of specific parameters and corresponding odds ratios where appropriate. Both point estimates and confidence intervals are provided.

Further features of the LOGISTIC procedure enable you to

- control the ordering of the response categories
- compute a generalized R^2 measure for the fitted model

- reclassify binary response observations according to their predicted response probabilities
- test linear hypotheses about the regression parameters
- create a data set for producing a receiver operating characteristic curve for each fitted model
- create a data set containing the estimated response probabilities, residuals, and influence diagnostics
- score a data set using a previously fitted model

Experimental graphics are now available with the LOGISTIC procedure. For more information, see the "ODS Graphics" section on page 2388.

The remaining sections of this chapter describe how to use PROC LOGISTIC and discuss the underlying statistical methodology. The "Getting Started" section introduces PROC LOGISTIC with an example for binary response data. The "Syntax" section (page 2289) describes the syntax of the procedure. The "Details" section (page 2329) summarizes the statistical technique employed by PROC LOGISTIC. The "Examples" section (page 2391) illustrates the use of the LOGISTIC procedure with 10 applications.

For more examples and discussion on the use of PROC LOGISTIC, refer to Stokes, Davis, and Koch (2000), Allison (1999), and SAS Institute Inc. (1995).

Getting Started

The LOGISTIC procedure is similar in use to the other regression procedures in the SAS System. To demonstrate the similarity, suppose the response variable y is binary or ordinal, and x1 and x2 are two explanatory variables of interest. To fit a logistic regression model, you can use a MODEL statement similar to that used in the REG procedure:

```
proc logistic;
   model y=x1 x2;
run;
```

The response variable y can be either character or numeric. PROC LOGISTIC enumerates the total number of response categories and orders the response levels according to the response variable option ORDER= in the MODEL statement. The procedure also allows the input of binary response data that are grouped:

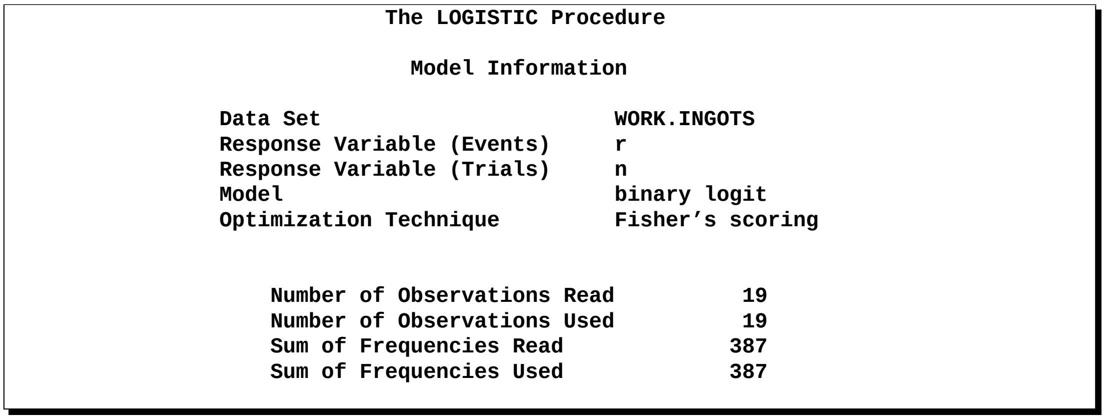
```
proc logistic;
   model r/n=x1 x2;
run;
```

Here, n represents the number of trials and r represents the number of events.

The following example illustrates the use of PROC LOGISTIC. The data, taken from Cox and Snell (1989, pp. 10–11), consist of the number, r, of ingots not ready for rolling, out of n tested, for a number of combinations of heating time and soaking time. The following invocation of PROC LOGISTIC fits the binary logit model to the grouped data:

```
data ingots;
   input Heat Soak r n @@;
   datalines;
7 1.0 0 10  14 1.0 0 31  27 1.0 1 56  51 1.0 3 13
7 1.7 0 17  14 1.7 0 43  27 1.7 4 44  51 1.7 0  1
7 2.2 0  7  14 2.2 2 33  27 2.2 0 21  51 2.2 0  1
7 2.8 0 12  14 2.8 0 31  27 2.8 1 22  51 4.0 0  1
7 4.0 0  9  14 4.0 0 19  27 4.0 1 16
;


proc logistic data=ingots;
   model r/n=Heat Soak;
run;
```

The results of this analysis are shown in the following tables.



```
                          The LOGISTIC Procedure

                            Model Information

             Data Set                            WORK.INGOTS
             Response Variable (Events)          r
             Response Variable (Trials)          n
             Model                               binary logit
             Optimization Technique              Fisher's scoring


                 Number of Observations Read          19
                 Number of Observations Used          19
                 Sum of Frequencies Read             387
                 Sum of Frequencies Used             387
```

Figure 42.1. Binary Logit Model

PROC LOGISTIC first lists background information in Figure 42.1 about the fitting of the model. Included are the name of the input data set, the response variable(s) used, the number of observations used, and the link function used.

```
                          Response Profile

               Ordered     Binary          Total
                 Value     Outcome     Frequency

                     1     Event              12
                     2     Nonevent          375


                     Model Convergence Status

          Convergence criterion (GCONV=1E-8) satisfied.
```

Figure 42.2. Response Profile with Events/Trials Syntax

The “Response Profile” table (Figure 42.2) lists the response categories (which are Event and Nonevent when grouped data are input), their ordered values, and their total frequencies for the given data.

```
                        Model Fit Statistics

                                           Intercept
                              Intercept          and
              Criterion            Only   Covariates

              AIC               108.988      101.346
              SC                112.947      113.221
              -2 Log L          106.988       95.346


              Testing Global Null Hypothesis: BETA=0

     Test                 Chi-Square       DF     Pr > ChiSq

     Likelihood Ratio        11.6428        2         0.0030
     Score                   15.1091        2         0.0005
     Wald                    13.0315        2         0.0015
```

Figure 42.3. Fit Statistics and Hypothesis Tests

The “Model Fit Statistics” table (Figure 42.3) contains the Akaike Information Criterion (AIC), the Schwarz Criterion (SC), and the negative of twice the log likelihood (-2 Log L) for the intercept-only model and the fitted model. AIC and SC can be used to compare different models, and the ones with smaller values are preferred. Results of the likelihood ratio test and the efficient score test for testing the joint significance of the explanatory variables (Soak and Heat) are included in the “Testing Global Null Hypothesis: BETA=0” table (Figure 42.3).

```
                    Analysis of Maximum Likelihood Estimates

                                    Standard          Wald
       Parameter    DF    Estimate     Error    Chi-Square    Pr > ChiSq

       Intercept     1     -5.5592    1.1197       24.6503        <.0001
       Heat          1      0.0820    0.0237       11.9454        0.0005
       Soak          1      0.0568    0.3312        0.0294        0.8639


                               Odds Ratio Estimates

                              Point          95% Wald
                    Effect    Estimate      Confidence Limits

                    Heat         1.085       1.036       1.137
                    Soak         1.058       0.553       2.026
```

Figure 42.4. Parameter Estimates and Odds Ratios

The "Analysis of Maximum Likelihood Estimates" table in Figure 42.4 lists the parameter estimates, their standard errors, and the results of the Wald test for individual parameters. The odds ratio for each effect parameter, estimated by exponentiating the corresponding parameter estimate, is shown in the "Odds Ratios Estimates" table (Figure 42.4), along with 95% Wald confidence intervals.

Using the parameter estimates, you can calculate the estimated logit of π as

$$-5.5592 + 0.082 \times \text{Heat} + 0.0568 \times \text{Soak}$$

If Heat$=7$ and Soak$=1$, then $\text{logit}(\hat{\pi}) = -4.9284$. Using this logit estimate, you can calculate $\hat{\pi}$ as follows:

$$\hat{\pi} = 1/(1 + \mathrm{e}^{4.9284}) = 0.0072$$

This gives the predicted probability of the event (ingot not ready for rolling) for Heat$=7$ and Soak$=1$. Note that PROC LOGISTIC can calculate these statistics for you; use the OUTPUT statement with the PREDICTED= option.

```
          Association of Predicted Probabilities and Observed Responses

                Percent Concordant     64.4    Somers' D    0.460
                Percent Discordant     18.4    Gamma        0.555
                Percent Tied           17.2    Tau-a        0.028
                Pairs                  4500    c            0.730
```

Figure 42.5. Association Table

Finally, the "Association of Predicted Probabilities and Observed Responses" table (Figure 42.5) contains four measures of association for assessing the predictive ability of a model. They are based on the number of pairs of observations with different response values, the number of concordant pairs, and the number of discordant pairs, which are also displayed. Formulas for these statistics are given in the "Rank Correlation of Observed Responses and Predicted Probabilities" section on page 2350.

To illustrate the use of an alternative form of input data, the following program creates the INGOTS data set with new variables NotReady and Freq instead of n and r. The variable NotReady represents the response of individual units; it has a value of 1 for units not ready for rolling (event) and a value of 0 for units ready for rolling (nonevent). The variable Freq represents the frequency of occurrence of each combination of Heat, Soak, and NotReady. Note that, compared to the previous data set, NotReady$=$1 implies Freq$=$r, and NotReady$=$0 implies Freq$=$n$-$r.

```
data ingots;
   input Heat Soak NotReady Freq @@;
   datalines;
7 1.0 0 10  14 1.0 0 31  14 4.0 0 19  27 2.2 0 21  51 1.0 1  3
7 1.7 0 17  14 1.7 0 43  27 1.0 1  1  27 2.8 1  1  51 1.0 0 10
7 2.2 0  7  14 2.2 1  2  27 1.0 0 55  27 2.8 0 21  51 1.7 0  1
7 2.8 0 12  14 2.2 0 31  27 1.7 1  4  27 4.0 1  1  51 2.2 0  1
7 4.0 0  9  14 2.8 0 31  27 1.7 0 40  27 4.0 0 15  51 4.0 0  1
;
```

The following SAS statements invoke PROC LOGISTIC to fit the same model using the alternative form of the input data set.

```
proc logistic data=ingots;
   model NotReady(event='1') = Soak Heat;
   freq Freq;
run;
```

Results of this analysis are the same as the previous one. The displayed output for the two runs are identical except for the background information of the model fit and the "Response Profile" table shown in Figure 42.6.

```
                    The LOGISTIC Procedure

                       Response Profile

              Ordered                       Total
                Value     NotReady      Frequency

                    1            0            375
                    2            1             12

              Probability modeled is NotReady=1.
```

Figure 42.6. Response Profile with Single-Trial Syntax

By default, Ordered Values are assigned to the sorted response values in ascending order, and PROC LOGISTIC models the probability of the response level that corresponds to the Ordered Value 1. There are several methods to change these defaults; the preceding statements specify the response variable option EVENT= to model the probability of NotReady=1 as displayed in Figure 42.6. See the "Response Level Ordering" section on page 2329 for more details.

Syntax

The following statements are available in PROC LOGISTIC:

```
PROC LOGISTIC < options >;
    BY variables ;
    CLASS variable <(v-options)> <variable <(v-options)>... >
              < / v-options >;
    CONTRAST 'label' effect values <,... effect values >< / options >;
    EXACT < 'label' >< Intercept >< effects >< / options > ;
    FREQ variable ;
    MODEL events/trials = < effects >< / options >;
    MODEL variable < (variable_options) > = < effects >< / options >;
    OUTPUT < OUT=SAS-data-set >
               < keyword=name. . .keyword=name >< / option >;
    SCORE < options >;
    STRATA effects < / options >;
    < label: > TEST equation1 < , . . . , < equationk >>< / option >;
    UNITS independent1=list1 < . . . independentk=listk >< / option > ;
    WEIGHT variable < / option >;
```

The PROC LOGISTIC and MODEL statements are required; only one MODEL statement can be specified. The CLASS statement (if used) must precede the MODEL statement, and the CONTRAST, EXACT, and STRATA statements (if used) must follow the MODEL statement. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC LOGISTIC statement. The remaining statements are covered in alphabetical order.

PROC LOGISTIC Statement

PROC LOGISTIC < *options* > **;**

The PROC LOGISTIC statement starts the LOGISTIC procedure and optionally identifies input and output data sets and suppresses the display of results.

ALPHA=α

specifies the level of significance α for $100(1-\alpha)$% confidence intervals. The value α must be between 0 and 1; the default value is 0.05, which results in 95% intervals. This value is used as the default confidence level for limits computed by the following options.

Statement	Options
CONTRAST	ESTIMATE=
EXACT	ESTIMATE=
MODEL	CLODDS= CLPARM=
OUTPUT	UCL= LCL=
SCORE	CLM

You can override the default in each of these cases by specifying the ALPHA= option for each statement individually.

COVOUT

adds the estimated covariance matrix to the OUTEST= data set. For the COVOUT option to have an effect, the OUTEST= option must be specified. See the section "OUTEST= Output Data Set" on page 2374 for more information.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set. The INMODEL= option cannot be specified with this option.

DESCENDING
DESC

reverses the sorting order for the levels of the response variable. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the levels according to the ORDER= option and then reverses that order. This option has the same effect as the response variable option DESCENDING in the MODEL statement. See the "Response Level Ordering" section on page 2329 for more detail.

EXACTONLY

requests only the exact analyses. The asymptotic analysis that PROC LOGISTIC usually performs is suppressed.

EXACTOPTIONS(*options*)

specifies options that apply to every EXACT statement in the program. The following options are available:

ADDTOBS adds the observed sufficient statistic to the sampled exact distribution if the statistic was not sampled. This option has no effect unless the METHOD=NETWORKMC option is specified and the ESTIMATE option is specified in the EXACT statement. If the observed statistic has not been sampled, then the parameter estimate does not exist; by specifying this option, you can produce (biased) estimates.

MAXTIME=*seconds* specifies the maximum clock time (in seconds) that PROC LOGISTIC can use to calculate the exact distributions. If the limit is exceeded, the procedure halts all computations and prints a note to the LOG. The default maximum clock time is seven days.

METHOD=*keyword* specifies which exact conditional algorithm to use for every EXACT statement specified. You can specify one of the following *keywords*:

DIRECT invokes the multivariate shift algorithm of Hirji, Mehta, and Patel (1987). This method directly builds the exact distribution, but it may require an excessive amount of memory in its intermediate stages. METHOD=DIRECT is invoked by default when you are conditioning out at most the intercept, or when the LINK=GLOGIT option is specified in the MODEL statement.

NETWORK invokes an algorithm similar to that described in Mehta, Patel, and Senchaudhuri (1992). This method builds a network for each parameter that you are conditioning out, combines the networks, then uses the multivariate shift algorithm to create the exact distribution. The NETWORK method can be faster and require less memory than the DIRECT method. The NETWORK method is invoked by default for most analyses.

NETWORKMC invokes the hybrid network and Monte Carlo algorithm of Mehta, Patel, and Senchaudhuri (2000). This method creates a network then samples from that network; this method does not reject any of the samples at the cost of using a large amount of memory to create the network. METHOD=NETWORKMC is most useful for producing parameter estimates for problems that are too large for the DIRECT and NETWORK methods to handle and for which asymptotic methods are invalid; for example, for sparse data on a large grid.

N=*n* specifies the number of Monte Carlo samples to take when METHOD=NETWORKMC. By default $n = 10,000$. If the procedure cannot obtain n samples due to a lack of memory, then a note is printed in the LOG (the number of valid samples is also reported in the listing) and the analysis continues.

Note that the number of samples used to produce any particular statistic may be smaller than *n*. For example, let $X1$ and $X2$ be continuous variables, denote their joint distribution by $f(X1, X2)$, and let $f(X1|X2 = x2)$ denote the marginal distribution of $X1$ conditioned on the observed value of $X2$. If you request the JOINT test of $X1$ and $X2$, then *n* samples are used to generate the estimate $\hat{f}(X1, X2)$ of $f(X1, X2)$, from which the test is computed. However, the parameter estimate for $X1$ is computed from the subset of $\hat{f}(X1, X2)$ having $X2 = x2$, and this subset need not contain *n* samples. Similarly, the distribution for each level of a classification variable is created by extracting the appropriate subset from the joint distribution for the CLASS variable. The sample sizes used to compute the statistics are written to the ODS OUTPUT data set of the tables.

In some cases, the marginal sample size may be too small to admit accurate estimation of a particular statistic; a note is printed in the LOG when a marginal sample size is less than 100. Increasing *n* will increase the number of samples used in a marginal distribution; however, if you want to control the sample size exactly, you can:

- Remove the JOINT option from the EXACT statement.
- Create dummy variables in a DATA step to represent the levels of a CLASS variable, and specify them as independent variables in the MODEL statement.

ONDISK uses disk-space instead of random access memory to build the exact conditional distribution. Use this option to handle larger problems at the cost of slower processing.

SEED=*n* specifies the initial seed for the random number generator used to take the Monte Carlo samples for METHOD=NETWORKMC. The value of the SEED= option must be an integer. If you do not specify a seed, or if you specify a value less than or equal to zero, then PROC LOGISTIC uses the time of day from the computer's clock to generate an initial seed. The seed is displayed in the "Model Information" table.

STATUSN=*n* prints a status line in the LOG after every n Monte Carlo samples for METHOD=NETWORKMC. The number of samples taken and the current exact p-value for testing the significance of the model are displayed. You can use this status line to track the progress of the computation of the exact conditional distributions.

STATUSTIME=*seconds* specifies the time interval (in seconds) for printing a status line in the LOG. You can use this status line to track the progress of the computation of the exact conditional distributions. The time interval you specify is approximate; the actual time interval will vary. By default, no status reports are produced.

INEST=*SAS-data-set*

names the SAS data set that contains initial estimates for all the parameters in the model. BY-group processing is allowed in setting up the INEST= data set. See the section "INEST= Input Data Set" on page 2376 for more information.

INMODEL=*SAS-data-set*

specifies the name of the SAS data set that contains the model information needed for scoring new data. This INMODEL= data set is the OUTMODEL= data set saved in a previous PROC LOGISTIC call. The DATA= option cannot be specified with this option; instead, specify the data sets to be scored in the SCORE statements.

When the INMODEL= data set is specified, FORMAT statements are not allowed; variables in the DATA= and PRIOR= data sets should be formatted within the data sets. If a SCORE statement is specified in the same run as fitting the model, FORMAT statements should be specified after the SCORE statement in order for the formats to apply to all the DATA= and PRIOR= data sets in the SCORE statement.

You can specify the BY statement provided the INMODEL= data set is created under the same BY-group processing.

The CLASS, EXACT, MODEL, OUTPUT, TEST, and UNIT statements are not available with the INMODEL= option.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOCOV

specifies that the covariance matrix is not saved in the OUTMODEL= data set. The covariance matrix is needed for computing the confidence intervals for the posterior probabilities in the OUT= data set in the SCORE statement. Specifying this option will reduce the size of the OUTMODEL= data set.

NOPRINT

suppresses all displayed output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
RORDER=DATA | FORMATTED | INTERNAL

specifies the sorting order for the levels of the response variable. See the response variable option ORDER= in the MODEL statement for more information.

OUTDESIGN=*SAS-data-set*

specifies the name of the data set that contains design matrix for the model. The data set contains the same number of observations as the corresponding DATA= data set and includes the response variable (with the same format as in the input data), the FREQ variable, the WEIGHT variable, the OFFSET variable, and the design variables for the covariates, including the Intercept variable of constant value 1 unless the NOINT option in the MODEL statement is specified.

OUTDESIGNONLY

suppresses the model fitting and only creates the OUTDESIGN= data set. This option is ignored if the OUTDESIGN= option is not specified.

OUTEST= *SAS-data-set*

creates an output SAS data set that contains the final parameter estimates and, optionally, their estimated covariances (see the preceding COVOUT option). The output data set also includes a variable named _LNLIKE_, which contains the log likelihood.

See the section "OUTEST= Output Data Set" on page 2374 for more information.

OUTMODEL=*SAS-data-set*

specifies the name of the SAS data set that contains the information about the fitted model. This data set contains sufficient information to score new data without having to refit the model. It is solely used as the input to the INMODEL= option in a subsequent PROC LOGISTIC call. **Note:** information is stored in this data set in a very compact form, hence you should not modify it manually.

SIMPLE

displays simple descriptive statistics (mean, standard deviation, minimum and maximum) for each continuous explanatory variable; and for each CLASS variable involved in the modeling, the frequency counts of the classification levels are displayed. The SIMPLE option generates a breakdown of the simple descriptive statistics or frequency counts for the entire data set and also for individual response categories.

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of CLASS, response, and strata variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases previous to Version 9. This option invokes the same option in the CLASS statement.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LOGISTIC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LOGISTIC procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

If a SCORE statement is specified, then define the *primary data set* to be the DATA= or the INMODEL=data set in the PROC LOGISTIC statement, and define the *secondary data set* to be the DATA= data set and PRIOR= data set in the SCORE statement. The primary data set contains all of the BY variables, and the secondary data set must contain either all of them or none of them. If the secondary data set contains all the BY-variables, matching is carried out between the primary and secondary data sets. If the secondary data set does not contain any of the BY-variables, the entire secondary data set is used for every BY-group in the primary data set and the BY-variables are added to the output data sets specified in the SCORE statement.

Caution: The order of your response and classification variables is determined by combining data across all BY groups; however, the observed levels may change between BY groups. This may affect the value of the reference level for these variables, and hence your interpretation of the model and the parameters.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable <(v-options)><variable <(v-options)>... >*
< / v-options > **;**

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *v-options* specified on any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *v-options* override the global *v-options*.

CPREFIX= *n*
specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding design variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING
DESC
reverses the sorting order of the classification variable. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the categories according to the ORDER= option and then reverses that order.

LPREFIX= *n*
specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding design variables. The default is $256 - \min(256, \max(2, f))$, where f is the formatted length of the CLASS variable.

MISSING

allows missing value ('.' for a numeric variable and blanks for a character variables) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST statement.

The following table shows how PROC LOGISTIC interprets values of the ORDER= option.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=EFFECT. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the "CLASS Variable Parameterization" section on page 2331 for further details.

EFFECT	specifies effect coding
GLM	specifies less-than-full-rank reference cell coding; this option can only be used as a global option
ORDINAL	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding
REFERENCE REF	specifies reference cell coding

ORTHEFFECT	orthogonalizes PARAM=EFFECT
ORTHORDINAL	orthogonalizes PARAM=ORDINAL
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL
ORTHREF	orthogonalizes PARAM=REFERENCE

The EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization.

REF=*'level'* **|** *keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference
LAST	designates the last ordered level as reference

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of CLASS, response, and strata variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases previous to Version 9. The TRUNCATE option is only available as a global option. This option invokes the same option in the PROC LOGISTIC statement.

CONTRAST Statement

CONTRAST *'label' row-description* <*,...row-description* >< */ options* > **;**

where a *row-description* is: *effect values* <*,...effect values*>

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST and ESTIMATE statements in PROC GLM and PROC CATMOD, depending on the coding schemes used with any classification variables involved.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\theta} = \mathbf{0}$, where $\boldsymbol{\theta}$ is the parameter vector. You must be familiar with the details of the model parameterization that PROC LOGISTIC uses (for more information, see the PARAM= option in the section “CLASS Statement” on page 2295).

Optionally, the CONTRAST statement enables you to estimate each row, $\boldsymbol{l}_i'\boldsymbol{\theta}$, of $\mathbf{L}\boldsymbol{\theta}$ and test the hypothesis $\boldsymbol{l}_i'\boldsymbol{\theta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. The name INTERCEPT can be used as an effect when one or more intercepts are included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the **L** matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The “Class Level Information” table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters.

The rows of **L** are specified in order and are separated by commas. Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*. For any of the full-rank parameterizations, if an effect is not specified in the CONTRAST statement, all of its coefficients in the **L** matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

When you use effect coding (by default or by specifying PARAM=EFFECT in the CLASS statement), all parameters are directly estimable (involve no other parameters). For example, suppose an effect coded CLASS variable A has four levels. Then there are three parameters ($\alpha_1, \alpha_2, \alpha_3$) representing the first three levels, and the fourth parameter is represented by

$$-\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\theta} = \mathbf{0}$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example,

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability. If PROC LOGISTIC finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC LOGISTIC handles missing level combinations of classification variables in the same manner as PROC GLM. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement. If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement, that is, the rank of **L**.

You can specify the following options after a slash (/).

ALPHA=α

specifies the level of significance α for the $100(1-\alpha)$% confidence interval for each contrast when the ESTIMATE option is specified. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

E

displays the $\mathbf{L}$ matrix.

ESTIMATE=*keyword*

requests that each individual contrast (that is, each row, $\boldsymbol{l}_i'\boldsymbol{\theta}$, of $\mathbf{L}\boldsymbol{\theta}$) or exponentiated contrast ($e^{\boldsymbol{l}_i'\boldsymbol{\theta}}$) be estimated and tested. PROC LOGISTIC displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the contrast or the exponentiated contrast ($e^{\boldsymbol{l}_i'\boldsymbol{\theta}}$), or both, by specifying one of the following *keywords*:

PARM	specifies that the contrast itself be estimated
EXP	specifies that the exponentiated contrast be estimated
BOTH	specifies that both the contrast and the exponentiated contrast be estimated

SINGULAR = *number*

tunes the estimability check. This option is ignored when the full-rank parameterization is used. If $\boldsymbol{v}$ is a vector, define ABS($\boldsymbol{v}$) to be the largest absolute value of the elements of $\boldsymbol{v}$. For a row vector $\boldsymbol{l}'$ of the contrast matrix $\mathbf{L}$, define c to be equal to ABS($\boldsymbol{l}$) if ABS($\boldsymbol{l}$) is greater than 0; otherwise, c equals 1. If ABS($\boldsymbol{l}' - \boldsymbol{l}'\boldsymbol{T}$) is greater than $c * number$, then $\boldsymbol{l}$ is declared nonestimable. The $\boldsymbol{T}$ matrix is the Hermite form matrix $\boldsymbol{I}_0^-\boldsymbol{I}_0$, where $\boldsymbol{I}_0^-$ represents a generalized inverse of the information matrix $\boldsymbol{I}_0$ of the null model. The value for $number$ must be between 0 and 1; the default value is 1E$-$4.

EXACT Statement

EXACT *<'label'>< Intercept >< effects >< / options >* **;**

The EXACT statement performs exact tests of the parameters for the specified effects and optionally estimates the parameters and outputs the exact conditional distributions. You can specify the keyword INTERCEPT and any effects in the MODEL statement. Inference on the parameters of the specified effects is performed by conditioning on the sufficient statistics of all the other model parameters (possibly including the intercept).

You can specify several EXACT statements, but they must follow the MODEL statement. Each statement can optionally include an identifying label. If several EXACT statements are specified, any statement without a label will be assigned a label of the form "Exact$n$", where "n" indicates the nth EXACT statement. The label is included in the headers of the displayed exact analysis tables.

If a STRATA statement is also specified, then a stratified exact conditional logistic regression is performed. The model contains a different intercept for each stratum, and these intercepts are conditioned out of the model along with any other nuisance parameters (essentially, any parameters specified in the MODEL statement which are not in the EXACT statement).

If the LINK=GLOGIT option is specified in the MODEL statement, then the EXACTOPTION option METHOD=DIRECT is invoked by default and a generalized logit model is fit. Since each effect specified in the MODEL statement adds k parameters to the model (where $k+1$ is the number of response levels), exact analysis of the generalized logit model using this method is limited to rather small problems.

The CONTRAST, OUTPUT, SCORE, TEST, and UNITS statements are not available with an exact analysis. Exact analyses are not performed when you specify a WEIGHT statement, a link other than LINK=LOGIT or LINK=GLOGIT, an offset variable, the NOFIT option, or a model-selection method. Exact estimation is not available for ordinal response models.

The following options can be specified in each EXACT statement after a slash (/):

ALPHA=α

specifies the level of significance α for $100(1-\alpha)$% confidence limits for the parameters or odds ratios. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

ESTIMATE $<$ **=*keyword*** $>$

estimates the individual parameters (conditional on all other parameters) for the effects specified in the EXACT statement. For each parameter, a point estimate, a confidence interval, and a p-value for a two-sided test that the parameter is zero are displayed. Note that the two-sided p-value is twice the one-sided p-value. You can optionally specify one of the following keywords:

PARM	specifies that the parameters be estimated. This is the default.
ODDS	specifies that the odds ratios be estimated. For classification variables, use of the reference parameterization is recommended.
BOTH	specifies that the parameters and odds ratios be estimated

JOINT

performs the joint test that all of the parameters are simultaneously equal to zero, individual hypothesis tests for the parameter of each continuous variable, and joint tests for the parameters of each classification variable. The joint test is indicated in the “Conditional Exact Tests” table by the label “Joint.”

JOINTONLY

performs only the joint test of the parameters. The test is indicated in the “Conditional Exact Tests” table by the label “Joint.” When this option is specified, individual tests for the parameters of each continuous variable and joint tests for the parameters of the classification variables are not performed.

CLTYPE=EXACT | MIDP

requests either the exact or mid-p confidence intervals for the parameter estimates. By default, the exact intervals are produced. The confidence coefficient can be specified with the ALPHA= option. The mid-p interval can be modified with the MIDPFACTOR= option. See the "Inference for a Single Parameter" section on page 2373 for details.

MIDPFACTOR=δ_1 **|** (δ_1, δ_2)

sets the tie factors used to produce the mid-p hypothesis statistics and the mid-p confidence intervals. δ_1 modifies both the hypothesis tests and confidence intervals, while δ_2 affects only the hypothesis tests. By default, $\delta_1 = 0.5$ and $\delta_2 = 1.0$. See the "Hypothesis Tests" section on page 2371 and the "Inference for a Single Parameter" section on page 2373 for details.

ONESIDED

requests one-sided confidence intervals and p-values for the individual parameter estimates and odds ratios. The one-sided p-value is the smaller of the left and right tail probabilities for the observed sufficient statistic of the parameter under the null hypothesis that the parameter is zero. The two-sided p-values (default) are twice the one-sided p-values. See the "Inference for a Single Parameter" section on page 2373 for more details.

OUTDIST=*SAS-data-set*

names the SAS data set containing the exact conditional distributions. This data set contains all of the exact conditional distributions required to process the corresponding EXACT statement. The data set contains the possible sufficient statistics for the parameters of the effects specified in the EXACT statement, the counts, and, when hypothesis tests are performed on the parameters, the probability of occurrence and the score value for each sufficient statistic. When you request an OUTDIST= data set, the observed sufficient statistics are displayed in the "Sufficient Statistics" table. See the "OUTDIST= Output Data Set" section on page 2377 for more information.

EXACT Statement Examples

- In the following example, two exact tests are computed: one for x1 and the other for x2. The test for x1 is based on the exact conditional distribution of the sufficient statistic for the x1 parameter given the observed values of the sufficient statistics for the intercept, x2, and x3 parameters; likewise, the test for x2 is conditional on the observed sufficient statistics for the intercept, x1, and x3:

```
proc logistic;
   model y= x1 x2 x3;
   exact 'lab1' x1 x2;
run;
```

- You can specify multiple EXACT statements in the same PROC LOGISTIC invocation. PROC LOGISTIC determines, from all the EXACT statements, the distinct conditional distributions that need to be evaluated. For example, there is only one exact conditional distribution for the following two EXACT statements, and it would be a waste of resources to compute the same exact conditional distribution twice:

```
exact 'One' x1 / estimate=parm;
exact 'Two' x1 / estimate=parm onesided;
```

- For each EXACT statement, individual tests for the parameters of the specified effects are computed unless the JOINTONLY option is specified. Consider the following EXACT statements:

  ```
  exact 'E12' x1 x2 / estimate;
  exact 'E1'  x1    / estimate;
  exact 'E2'  x2    / estimate;
  exact 'J12' x1 x2 / joint;
  ```

  In the E12 statement, the parameters for x1 and x2 are estimated and tested separately. Specifying the E12 statement is equivalent to specifying both the E1 and E2 statements. In the J12 statement, the joint test for the parameters of x1 and x2 is computed as well as the individual tests for x1 and x2.

  All exact conditional distributions for the tests and estimates computed in a single EXACT statement are output to the corresponding OUTDIST= data set. For example, consider the following EXACT statements:

  ```
  exact 'O1'   x1    /           outdist=o1;
  exact 'OJ12' x1 x2 / jointonly outdist=oj12;
  exact 'OA12' x1 x2 / joint     outdist=oa12;
  exact 'OE12' x1 x2 / estimate  outdist=oe12;
  ```

  The O1 statement outputs a single exact conditional distribution. The OJ12 statement outputs only the joint distribution for x1 and x2. The OA12 statement outputs three conditional distributions: one for x1, one for x2, and one jointly for x1 and x2. The OE12 statement outputs two conditional distributions: one for x1 and the other for x2. Data set oe12 contains both the x1 and x2 variables; the distribution for x1 has missing values in the x2 column while the distribution for x2 has missing values in the x1 column.

  See the "OUTDIST= Output Data Set" section on page 2377 for more information.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC LOGISTIC treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the model fitting. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

If a SCORE statement is specified, then the FREQ variable is used for computing fit statistics and the ROC curve, but they are not required for scoring. If the DATA= data set in the SCORE statement does not contain the FREQ variable, the frequency values are assumed to be 1 and a warning message is issued in the LOG. If you fit a model and perform the scoring in the same run, the same FREQ variable is used for fitting and scoring. If you fit a model in a previous run and input it with the INMODEL= option in the current run, then the FREQ variable can be different from the one used in the previous run; however, if a FREQ variable was not specified in the previous run you can still specify a FREQ variable in the current run.

MODEL Statement

MODEL *events/trials*= $<$ *effects* $><$ */ options* $>$ **;**

MODEL *variable* $<$ *(variable_options)* $>=$ $<$ *effects* $><$ */ options* $>$ **;**

The MODEL statement names the response variable and the explanatory effects, including covariates, main effects, interactions, and nested effects; see the section "Specification of Effects" on page 1784 of Chapter 32, "The GLM Procedure," for more information. If you omit the explanatory effects, the procedure fits an intercept-only model. Model options can be specified after a slash (/).

Two forms of the MODEL statement can be specified. The first form, referred to as *single-trial* syntax, is applicable to binary, ordinal, and nominal response data. The second form, referred to as *events/trials* syntax, is restricted to the case of binary response data. The *single-trial* syntax is used when each observation in the DATA= data set contains information on only a single trial, for instance, a single subject in an experiment. When each observation contains information on multiple binary-response trials, such as the counts of the number of subjects observed and the number responding, then *events/trials* syntax can be used.

In the *events/trials* syntax, you specify two variables that contain count data for a binomial experiment. These two variables are separated by a slash. The value of the first variable, *events*, is the number of positive responses (or events). The value of the second variable, *trials*, is the number of trials. The values of both *events* and ($trials - events$) must be nonnegative and the value of *trials* must be positive for the response to be valid.

In the *single-trial* syntax, you specify one variable (on the left side of the equal sign) as the response variable. This variable can be character or numeric. Options specific to the response variable can be specified immediately after the response variable with a pair of parentheses around them.

For both forms of the MODEL statement, explanatory *effects* follow the equal sign. Variables can be either continuous or classification variables. Classification variables can be character or numeric, and they must be declared in the CLASS statement. When an effect is a classification variable, the procedure enters a set of coded columns into the design matrix instead of directly entering a single column containing the values of the variable.

Response Variable Options

You can specify the following options by enclosing them in a pair of parentheses after the response variable.

DESCENDING | DESC

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the response categories according to the ORDER= option and then reverses that order. See the "Response Level Ordering" section on page 2329 for more detail.

EVENT='*category*' | *keyword*

specifies the event category for the binary response model. PROC LOGISTIC models the probability of the event category. The EVENT= option has no effect when there are more than two response categories. You can specify the value (formatted if a format is applied) of the event category in quotes or you can specify one of the following keywords. The default is EVENT=FIRST.

FIRST	designates the first ordered category as the event
LAST	designates the last ordered category as the event

One of the most common sets of response levels is {0,1}, with 1 representing the event for which the probability is to be modeled. Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. To specify the value 1 as the event category, use the MODEL statement

```
   model Y(event='1') = Exposure;
```

ORDER= DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the response variable. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows the interpretation of the ORDER= values.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

REFERENCE='*category*' | *keyword*
REF='*category*' | *keyword*

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each nonreference category is contrasted with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes or you can specify one of the following keywords. The default is REF=LAST.

FIRST	designates the first ordered category as the reference
LAST	designates the last ordered category as the reference

Model Options

Table 42.1 summarizes the options available in the MODEL statement, which can be specified after a slash (/).

Table 42.1. Model Statement Options

Option	Description
Model Specification Options	
LINK=	specifies link function
NOINT	suppresses intercept
NOFIT	suppresses model fitting
OFFSET=	specifies offset variable
SELECTION=	specifies effect selection method
Effect Selection Options	
BEST=	controls the number of models displayed for SCORE selection
DETAILS	requests detailed results at each step
FAST	uses fast elimination method
HIERARCHY=	specifies whether and how hierarchy is maintained and whether a single effect or multiple effects are allowed to enter or leave the model per step
INCLUDE=	specifies number of effects included in every model
MAXSTEP=	specifies maximum number of steps for STEPWISE selection
SEQUENTIAL	adds or deletes effects in sequential order
SLENTRY=	specifies significance level for entering effects
SLSTAY=	specifies significance level for removing effects
START=	specifies number of variables in first model
STOP=	specifies number of variables in final model
STOPRES	adds or deletes variables by residual chi-square criterion
Model-Fitting Specification Options	
ABSFCONV=	specifies absolute function convergence criterion
FCONV=	specifies relative function convergence criterion
GCONV=	specifies relative gradient convergence criterion

Table 42.1. (continued)

Option	Description
XCONV=	specifies relative parameter convergence criterion
MAXFUNCTION=	specifies maximum number of function calls for the conditional analysis
MAXITER=	specifies maximum number of iterations
NOCHECK	suppresses checking for infinite parameters
RIDGING=	specifies the technique used to improve the log-likelihood function when its value is worse than that of the previous step
SINGULAR=	specifies tolerance for testing singularity
TECHNIQUE=	specifies iterative algorithm for maximization
Options for Confidence Intervals	
ALPHA=	specifies α for the $100(1-\alpha)\%$ confidence intervals
CLPARM=	computes confidence intervals for parameters
CLODDS=	computes confidence intervals for odds ratios
PLCONV=	specifies profile likelihood convergence criterion
Options for Classifying Observations	
CTABLE	displays classification table
PEVENT=	specifies prior event probabilities
PPROB=	specifies probability cutpoints for classification
Options for Overdispersion and Goodness-of-Fit Tests	
AGGREGATE=	determines subpopulations for Pearson chi-square and deviance
SCALE=	specifies method to correct overdispersion
LACKFIT	requests Hosmer and Lemeshow goodness-of-fit test
Options for ROC Curves	
OUTROC=	names the output data set
ROCEPS=	specifies probability grouping criterion
Options for Regression Diagnostics	
INFLUENCE	displays influence statistics
IPLOTS	requests index plots
Options for Display of Details	
CORRB	displays correlation matrix
COVB	displays covariance matrix
EXPB	displays exponentiated values of estimates
ITPRINT	displays iteration history
NODUMMYPRINT	suppresses “Class Level Information” table
PARMLABEL	displays parameter labels
RSQUARE	displays generalized R^2
STB	displays standardized estimates
Computational Options	
NOLOGSCALE	performs calculations using normal scaling

The following list describes these options.

ABSFCONV=*value*

specifies the absolute function convergence criterion. Convergence requires a small change in the log-likelihood function in subsequent iterations,

$$|l_i - l_{i-1}| < value$$

where l_i is the value of the log-likelihood function at iteration i. See the section "Convergence Criteria" on page 2338.

AGGREGATE
AGGREGATE= *(variable-list)*

specifies the subpopulations on which the Pearson chi-square test statistic and the likelihood ratio chi-square test statistic (deviance) are calculated. Observations with common values in the given list of variables are regarded as coming from the same subpopulation. Variables in the list can be any variables in the input data set. Specifying the AGGREGATE option is equivalent to specifying the AGGREGATE= option with a variable list that includes all explanatory variables in the MODEL statement. The deviance and Pearson goodness-of-fit statistics are calculated only when the SCALE= option is specified. Thus, the AGGREGATE (or AGGREGATE=) option has no effect if the SCALE= option is not specified. See the section "Rescaling the Covariance Matrix" on page 2354 for more detail.

ALPHA=α

sets the level of significance α for $100(1-\alpha)$% confidence intervals for regression parameters or odds ratios. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if the option is not specified. This option has no effect unless confidence limits for the parameters or odds ratios are requested.

BEST=*n*

specifies that n models with the highest score chi-square statistics are to be displayed for each model size. It is used exclusively with the SCORE model selection method. If the BEST= option is omitted and there are no more than ten explanatory variables, then all possible models are listed for each model size. If the option is omitted and there are more than ten explanatory variables, then the number of models selected for each model size is, at most, equal to the number of explanatory variables listed in the MODEL statement.

CLODDS=PL | WALD | BOTH

requests confidence intervals for the odds ratios. Computation of these confidence intervals is based on the profile likelihood (CLODDS=PL) or based on individual Wald tests (CLODDS=WALD). By specifying CLODDS=BOTH, the procedure computes two sets of confidence intervals for the odds ratios, one based on the profile likelihood and the other based on the Wald tests. The confidence coefficient can be specified with the ALPHA= option.

CLPARM=PL | WALD | BOTH

requests confidence intervals for the parameters. Computation of these confidence intervals is based on the profile likelihood (CLPARM=PL) or individual Wald tests (CLPARM=WALD). By specifying CLPARM=BOTH, the procedure computes two sets of confidence intervals for the parameters, one based on the profile likelihood and the other based on individual Wald tests. The confidence coefficient can be specified with the ALPHA= option. See the "Confidence Intervals for Parameters" section on page 2345 for more information.

CORRB

displays the correlation matrix of the parameter estimates.

COVB

displays the covariance matrix of the parameter estimates.

CTABLE

classifies the input binary response observations according to whether the predicted event probabilities are above or below some cutpoint value z in the range $(0,1)$. An observation is predicted as an event if the predicted event probability exceeds z. You can supply a list of cutpoints other than the default list by using the PPROB= option (page 2315). The CTABLE option is ignored if the data have more than two response levels. Also, false positive and negative rates can be computed as posterior probabilities using Bayes' theorem. You can use the PEVENT= option to specify prior probabilities for computing these rates. For more information, see the "Classification Table" section on page 2352.

DETAILS

produces a summary of computational details for each step of the effect selection process. It produces the "Analysis of Effects Not in the Model" table before displaying the effect selected for entry for FORWARD or STEPWISE selection. For each model fitted, it produces the "Type 3 Analysis of Effects" table if the fitted model involves CLASS variables, the "Analysis of Maximum Likelihood Estimates" table, and measures of association between predicted probabilities and observed responses. For the statistics included in these tables, see the "Displayed Output" section on page 2381. The DETAILS option has no effect when SELECTION=NONE.

EXPB
EXPEST

displays the exponentiated values ($e^{\hat{\beta}_i}$) of the parameter estimates $\hat{\beta}_i$ in the "Analysis of Maximum Likelihood Estimates" table for the logit model. These exponentiated values are the estimated odds ratios for the parameters corresponding to the continuous explanatory variables.

FAST

uses a computational algorithm of Lawless and Singhal (1978) to compute a first-order approximation to the remaining slope estimates for each subsequent elimination of a variable from the model. Variables are removed from the model based on these approximate estimates. The FAST option is extremely efficient because the model is not refitted for every variable removed. The FAST option is used when SELECTION=BACKWARD and in the backward elimination steps when SELECTION=STEPWISE. The FAST option is ignored when SELECTION=FORWARD or SELECTION=NONE.

FCONV=*value*

specifies the relative function convergence criterion. Convergence requires a small relative change in the log-likelihood function in subsequent iterations,

$$\frac{|l_i - l_{i-1}|}{|l_{i-1}| + 1\mathrm{E}{-6}} < value$$

where l_i is the value of the log likelihood at iteration i. See the section "Convergence Criteria" on page 2338.

GCONV=*value*

specifies the relative gradient convergence criterion. Convergence requires that the normalized prediction function reduction is small,

$$\frac{\mathbf{g}_i' \mathbf{I}_i \mathbf{g}_i}{|l_i| + 1\mathrm{E}{-6}} < value$$

where l_i is the value of the log-likelihood function, $\mathbf{g}_i$ is the gradient vector, and $\mathbf{I}_i$ is the (expected) information matrix, all at iteration i. This is the default convergence criterion, and the default value is 1E$-$8. See the section "Convergence Criteria" on page 2338.

HIERARCHY=*keyword*
HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied and whether a single effect or multiple effects are allowed to enter or leave the model in one step. You can specify that only CLASS effects, or both CLASS and interval effects, be subject to the hierarchy requirement. The HIERARCHY= option is ignored unless you also specify one of the following options: SELECTION=FORWARD, SELECTION=BACKWARD, or SELECTION=STEPWISE.

Model hierarchy refers to the requirement that, for any term to be in the model, all effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are as follows:

NONE	Model hierarchy is not maintained. Any single effect can enter or leave the model at any given step of the selection process.
SINGLE	Only one effect can enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that you specify the main effects A and B and the interaction A*B in the model. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter

the model. The interaction effect can enter the model only when both main effects have already been entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.

SINGLECLASS This is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

MULTIPLE More than one effect can enter or leave the model at one time, subject to the model hierarchy requirement. In a forward selection step, a single main effect can enter the model, or an interaction can enter the model together with all the effects that are contained in the interaction. In a backward elimination step, an interaction itself, or the interaction together with all the effects that the interaction contains, can be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.

MULTIPLECLASS This is the same as HIERARCHY=MULTIPLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=SINGLE, which means that model hierarchy is to be maintained for all effects (that is, both CLASS and interval effects) and that only a single effect can enter or leave the model at each step.

INCLUDE=*n*

includes the first n effects in the MODEL statement in every model. By default, INCLUDE=0. The INCLUDE= option has no effect when SELECTION=NONE.

Note that the INCLUDE= and START= options perform different tasks: the INCLUDE= option includes the first n effects variables in every model, whereas the START= option only requires that the first n effects appear in the first model.

INFLUENCE

displays diagnostic measures for identifying influential observations in the case of a binary response model. It has no effect otherwise. For each observation, the INFLUENCE option displays the case number (which is the sequence number of the observation), the values of the explanatory variables included in the final model, and the regression diagnostic measures developed by Pregibon (1981). For a discussion of these diagnostic measures, see the "Regression Diagnostics" section on page 2359. When a STRATA statement is specified, the diagnostics are computed following Storer and Crowley (1985); see the "Regression Diagnostic Details" section on page 2367 for details.

IPLOTS

produces an index plot for each regression diagnostic statistic. An index plot is a scatterplot with the regression diagnostic statistic represented on the y-axis and the case number on the x-axis. See Example 42.6 on page 2422 for an illustration.

ITPRINT

displays the iteration history of the maximum-likelihood model fitting. The ITPRINT option also displays the last evaluation of the gradient vector and the final change in the -2 Log Likelihood.

LACKFIT
LACKFIT$<$**(***n***)**$>$

performs the Hosmer and Lemeshow goodness-of-fit test (Hosmer and Lemeshow 2000) for the case of a binary response model. The subjects are divided into approximately ten groups of roughly the same size based on the percentiles of the estimated probabilities. The discrepancies between the observed and expected number of observations in these groups are summarized by the Pearson chi-square statistic, which is then compared to a chi-square distribution with t degrees of freedom, where t is the number of groups minus n. By default, n=2. A small p-value suggests that the fitted model is not an adequate model. See the "The Hosmer-Lemeshow Goodness-of-Fit Test" section on page 2356 for more information.

LINK=*keyword*
L=*keyword*

specifies the link function linking the response probabilities to the linear predictors. You can specify one of the following keywords. The default is LINK=LOGIT.

CLOGLOG	the complementary log-log function. PROC LOGISTIC fits the binary complementary log-log model when there are two response categories and fits the cumulative complementary log-log model when there are more than two response categories. Aliases: CCLOGLOG, CCLL, CUMCLOGLOG.
GLOGIT	the generalized logit function. PROC LOGISTIC fits the generalized logit model where each nonreference category is contrasted with the reference category. You can use the response variable option REF= to specify the reference category.
LOGIT	the log odds function. PROC LOGISTIC fits the binary logit model when there are two response categories and fits the cumulative logit model when there are more than two response categories. Aliases: CLOGIT, CUMLOGIT.
PROBIT	the inverse standard normal distribution function. PROC LOGISTIC fits the binary probit model when there are two response categories and fits the cumulative probit model when there are more than two response categories. Aliases: NORMIT, CPROBIT, CUMPROBIT.

See the section "Link Functions and the Corresponding Distributions" on page 2334 for details.

MAXFUNCTION=*n*

specifies the maximum number of function calls to perform when maximizing the conditional likelihood. This option is only valid when a STRATA statement is specified. The default values are

- 125 when the number of parameters $p < 40$
- 500 when $40 \leq p < 400$
- 1000 when $p \geq 400$

Since the optimization is terminated only after completing a full iteration, the number of function calls that are actually performed can exceed n. If convergence is not attained, the displayed output and all output data sets created by the procedure contain results based on the last maximum likelihood iteration.

MAXITER=*n*

specifies the maximum number of iterations to perform. By default, MAXITER=25. If convergence is not attained in n iterations, the displayed output and all output data sets created by the procedure contain results that are based on the last maximum likelihood iteration.

MAXSTEP=*n*

specifies the maximum number of times any explanatory variable is added to or removed from the model when SELECTION=STEPWISE. The default number is twice the number of explanatory variables in the MODEL statement. When the MAXSTEP= limit is reached, the stepwise selection process is terminated. All statistics displayed by the procedure (and included in output data sets) are based on the last model fitted. The MAXSTEP= option has no effect when SELECTION=NONE, FORWARD, or BACKWARD.

NOCHECK

disables the checking process to determine whether maximum likelihood estimates of the regression parameters exist. If you are sure that the estimates are finite, this option can reduce the execution time if the estimation takes more than eight iterations. For more information, see the "Existence of Maximum Likelihood Estimates" section on page 2338.

NODUMMYPRINT
NODESIGNPRINT
NODP

suppresses the "Class Level Information" table, which shows how the design matrix columns for the CLASS variables are coded.

NOINT

suppresses the intercept for the binary response model, the first intercept for the ordinal response model (which forces all intercepts to be nonnegative), or all intercepts for the generalized logit model. This can be particularly useful in conditional logistic analysis; see Example 42.10 on page 2443.

NOFIT

performs the global score test without fitting the model. The global score test evaluates the joint significance of the effects in the MODEL statement. No further analyses are performed. If the NOFIT option is specified along with other MODEL statement options, NOFIT takes effect and all other options except LINK=, TECHNIQUE=, and OFFSET= are ignored.

NOLOGSCALE

specifies that computations for the conditional and exact conditional logistic model should be computed using normal scaling. Log-scaling can handle numerically larger problems than normal scaling; however, computations in the log-scale are slower than computations in normal-scale.

OFFSET= *name*

names the offset variable. The regression coefficient for this variable will be fixed at 1.

OUTROC=*SAS-data-set*
OUTR=*SAS-data-set*

creates, for binary response models, an output SAS data set that contains the data necessary to produce the receiver operating characteristic (ROC) curve. See the section "OUTROC= Output Data Set" on page 2378 for the list of variables in this data set.

PARMLABEL

displays the labels of the parameters in the "Analysis of Maximum Likelihood Estimates" table.

PEVENT= *value*
PEVENT= (*list* **)**

specifies one prior probability or a list of prior probabilities for the event of interest. The false positive and false negative rates are then computed as posterior probabilities by Bayes' theorem. The prior probability is also used in computing the rate of correct prediction. For each prior probability in the given list, a classification table of all observations is computed. By default, the prior probability is the total sample proportion of events. The PEVENT= option is useful for stratified samples. It has no effect if the CTABLE option is not specified. For more information, see the section "False Positive and Negative Rates Using Bayes' Theorem" on page 2353. Also see the PPROB= option for information on how the *list* is specified.

PLCL

is the same as specifying CLPARM=PL.

PLCONV= *value*

controls the convergence criterion for confidence intervals based on the profile likelihood function. The quantity *value* must be a positive number, with a default value of 1E−4. The PLCONV= option has no effect if profile likelihood confidence intervals (CLPARM=PL) are not requested.

PLRL

is the same as specifying CLODDS=PL.

PPROB=*value*
PPROB= (*list* **)**

specifies one critical probability value (or cutpoint) or a list of critical probability values for classifying observations with the CTABLE option. Each *value* must be between 0 and 1. A response that has a cross validated predicted probability greater than or equal to the current PPROB= value is classified as an event response. The PPROB= option is ignored if the CTABLE option is not specified.

A classification table for each of several cutpoints can be requested by specifying a list. For example,

```
pprob= (0.3, 0.5 to 0.8 by 0.1)
```

requests a classification of the observations for each of the cutpoints 0.3, 0.5, 0.6, 0.7, and 0.8. If the PPROB= option is not specified, the default is to display the classification for a range of probabilities from the smallest estimated probability (rounded down to the nearest 0.02) to the highest estimated probability (rounded up to the nearest 0.02) with 0.02 increments.

RIDGING=ABSOLUTE | RELATIVE | NONE

specifies the technique used to improve the log-likelihood function when its value in the current iteration is less than that in the previous iteration. If you specify the RIDGING=ABSOLUTE option, the diagonal elements of the negative (expected) Hessian are inflated by adding the ridge value. If you specify the RIDGING=RELATIVE option, the diagonal elements are inflated by a factor of 1 plus the ridge value. If you specify the RIDGING=NONE option, the crude line search method of taking half a step is used instead of ridging. By default, RIDGING=RELATIVE.

RISKLIMITS
RL
WALDRL

is the same as specifying CLODDS=WALD.

ROCEPS= *number*

specifies the criterion for grouping estimated event probabilities that are close to each other for the ROC curve. In each group, the difference between the largest and the smallest estimated event probabilities does not exceed the given value. The value for $number$ must be between 0 and 1; the default value is 1E$-$4. The smallest estimated probability in each group serves as a cutpoint for predicting an event response. The ROCEPS= option has no effect if the OUTROC= option is not specified.

RSQUARE
RSQ

requests a generalized R^2 measure for the fitted model. For more information, see the "Generalized Coefficient of Determination" section on page 2342.

SCALE= *scale*

enables you to supply the value of the dispersion parameter or to specify the method for estimating the dispersion parameter. It also enables you to display the "Deviance and Pearson Goodness-of-Fit Statistics" table. To correct for overdispersion or underdispersion, the covariance matrix is multiplied by the estimate of the dispersion parameter. Valid values for *scale* are as follows:

D | DEVIANCE
specifies that the dispersion parameter be estimated by the deviance divided by its degrees of freedom.

P | PEARSON
specifies that the dispersion parameter be estimated by the Pearson chi-square statistic divided by its degrees of freedom.

WILLIAMS <(*constant*)>
specifies that Williams' method be used to model overdispersion. This option can be used only with the *events/trials* syntax. An optional *constant* can be specified as the scale parameter; otherwise, a scale parameter is estimated under the full model. A set of weights is created based on this scale parameter estimate. These weights can then be used in fitting subsequent models of fewer terms than the full model. When fitting these submodels, specify the computed scale parameter as *constant*. See Example 42.9 on page 2438 for an illustration.

N | NONE
specifies that no correction is needed for the dispersion parameter; that is, the dispersion parameter remains as 1. This specification is used for requesting the deviance and the Pearson chi-square statistic without adjusting for overdispersion.

constant
sets the estimate of the dispersion parameter to be the square of the given *constant*. For example, SCALE=2 sets the dispersion parameter to 4. The value *constant* must be a positive number.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulations for calculating the Pearson chi-square statistic and the deviance. In the absence of the AGGREGATE (or AGGREGATE=) option, each observation is regarded as coming from a different subpopulation. For the *events/trials* syntax, each observation consists of n Bernoulli trials, where n is the value of the *trials* variable. For *single-trial* syntax, each observation consists of a single response, and for this setting it is not appropriate to carry out the Pearson or deviance goodness-of-fit analysis. Thus, PROC LOGISTIC ignores specifications SCALE=P, SCALE=D, and SCALE=N when *single-trial* syntax is specified without the AGGREGATE (or AGGREGATE=) option.

The "Deviance and Pearson Goodness-of-Fit Statistics" table includes the Pearson chi-square statistic, the deviance, their degrees of freedom, the ratio of each statistic divided by its degrees of freedom, and the corresponding p-value. For more information, see the "Overdispersion" section on page 2354.

SELECTION=BACKWARD | B
| FORWARD | F
| NONE | N
| STEPWISE | S
| SCORE

specifies the method used to select the variables in the model. BACKWARD requests backward elimination, FORWARD requests forward selection, NONE fits the complete model specified in the MODEL statement, and STEPWISE requests stepwise selection. SCORE requests best subset selection. By default, SELECTION=NONE. For more information, see the "Effect Selection Methods" section on page 2340.

SEQUENTIAL
SEQ

forces effects to be added to the model in the order specified in the MODEL statement or eliminated from the model in the reverse order specified in the MODEL statement. The model-building process continues until the next effect to be added has an insignificant adjusted chi-square statistic or until the next effect to be deleted has a significant Wald chi-square statistic. The SEQUENTIAL option has no effect when SELECTION=NONE.

SINGULAR=*value*

specifies the tolerance for testing the singularity of the Hessian matrix (Newton-Raphson algorithm) or the expected value of the Hessian matrix (Fisher-scoring algorithm). The Hessian matrix is the matrix of second partial derivatives of the log-likelihood function. The test requires that a pivot for sweeping this matrix be at least this number times a norm of the matrix. Values of the SINGULAR= option must be numeric. By default, *value* is the machine epsilon times 10^7, which is approximately 10^{-9} on most machines.

SLENTRY=*value*
SLE=*value*

specifies the significance level of the score chi-square for entering an effect into the model in the FORWARD or STEPWISE method. Values of the SLENTRY= option should be between 0 and 1, inclusive. By default, SLENTRY=0.05. The SLENTRY= option has no effect when SELECTION=NONE, SELECTION=BACKWARD, or SELECTION=SCORE.

SLSTAY=*value*
SLS=*value*

specifies the significance level of the Wald chi-square for an effect to stay in the model in a backward elimination step. Values of the SLSTAY= option should be between 0 and 1, inclusive. By default, SLSTAY=0.05. The SLSTAY= option has no effect when SELECTION=NONE, SELECTION=FORWARD, or SELECTION=SCORE.

START=*n*

begins the FORWARD, BACKWARD, or STEPWISE effect selection process with the first n effects listed in the MODEL statement. The value of n ranges from 0 to s, where s is the total number of effects in the MODEL statement. The default value of n is s for the BACKWARD method and 0 for the FORWARD and STEPWISE methods. Note that START=n specifies only that the first n effects appear in the first model, while INCLUDE=n requires that the first n effects be included in every model. For the SCORE method, START=n specifies that the smallest models contain n effects, where n ranges from 1 to s; the default value is 1. The START= option has no effect when SELECTION=NONE.

STB

displays the standardized estimates for the parameters for the continuous explanatory variables in the “Analysis of Maximum Likelihood Estimates” table. The standardized estimate of β_i is given by $\hat{\beta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

For the intercept parameters and parameters associated with a CLASS variable, the standardized estimates are set to missing.

STOP=*n*

specifies the maximum (FORWARD method) or minimum (BACKWARD method) number of effects to be included in the final model. The effect selection process is stopped when n effects are found. The value of n ranges from 0 to s, where s is the total number of effects in the MODEL statement. The default value of n is s for the FORWARD method and 0 for the BACKWARD method. For the SCORE method, STOP=n specifies that the largest models contain n effects, where n ranges from 1 to s; the default value of n is s. The STOP= option has no effect when SELECTION=NONE or STEPWISE.

STOPRES
SR

specifies that the removal or entry of effects be based on the value of the residual chi-square. If SELECTION=FORWARD, then the STOPRES option adds the effects into the model one at a time until the residual chi-square becomes insignificant (until the p-value of the residual chi-square exceeds the SLENTRY= *value*). If SELECTION=BACKWARD, then the STOPRES option removes effects from the model one at a time until the residual chi-square becomes significant (until the p-value of the residual chi-square becomes less than the SLSTAY= *value*). The STOPRES option has no effect when SELECTION=NONE or SELECTION=STEPWISE.

TECHNIQUE=FISHER | NEWTON
TECH=FISHER | NEWTON

specifies the optimization technique for estimating the regression parameters. NEWTON (or NR) is the Newton-Raphson algorithm and FISHER (or FS) is the

Fisher-scoring algorithm. Both techniques yield the same estimates, but the estimated covariance matrices are slightly different except for the case when the LOGIT link is specified for binary response data. The default is TECHNIQUE=FISHER. See the section "Iterative Algorithms for Model-Fitting" on page 2336 for details.

WALDCL
CL

is the same as specifying CLPARM=WALD.

XCONV=*value*

specifies the relative parameter convergence criterion. Convergence requires a small relative parameter change in subsequent iterations,

$$\max_j |\delta_j^{(i)}| < value$$

where

$$\delta_j^{(i)} = \begin{cases} \theta_j^{(i)} - \theta_j^{(i-1)} & |\theta_j^{(i-1)}| < 0.01 \\ \frac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}} & \text{otherwise} \end{cases}$$

and $\theta_j^{(i)}$ is the estimate of the jth parameter at iteration i. See the section "Convergence Criteria" on page 2338.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* >< *options* > **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors and their standard error estimates, the estimates of the cumulative or individual response probabilities, and the confidence limits for the cumulative probabilities. Regression diagnostic statistics and estimates of cross validated response probabilities are also available for binary response models. Formulas for the statistics are given in the "Linear Predictor, Predicted Probability, and Confidence Limits" section on page 2350, the "Regression Diagnostics" section on page 2359, and, for conditional logistic regression, in the "Conditional Logistic Regression" section on page 2365.

If you use the *single-trial* syntax, the data set also contains a variable named _LEVEL_, which indicates the level of the response that the given row of output is referring to. For instance, the value of the cumulative probability variable is the probability that the response variable is as large as the corresponding value of _LEVEL_. For details, see the section "OUT= Output Data Set in the OUTPUT Statement" on page 2376.

The estimated linear predictor, its standard error estimate, all predicted probabilities, and the confidence limits for the cumulative probabilities are computed for all observations in which the explanatory variables have no missing values, even if the response is missing. By adding observations with missing response values to the input data set, you can compute these statistics for new observations or for settings of the explanatory variables not present in the data without affecting the model fit.

OUT= *SAS-data-set*

names the output data set. If you omit the OUT= option, the output data set is created and given a default name using the DATAn convention.

The following sections explain options in the OUTPUT statement, divided into statistic options for any type of categorical responses, statistic options only for binary response, and other options. The statistic options specify the statistics to be included in the output data set and name the new variables that contain the statistics. If a STRATA statement is specified, only the PREDICTED=, DFBETAS=, and H= options are available; see the "Regression Diagnostic Details" section on page 2367 for details.

Statistic Options for Any Type of Categorical Response

LOWER=*name*
L=*name*

names the variable containing the lower confidence limits for π, where π is the probability of the event response if *events/trials* syntax or *single-trial* syntax with binary response is specified; for a cumulative model, π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_); for the generalized logit model, it is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_). See the ALPHA= option to set the confidence level.

PREDICTED=*name*
PRED=*name*
PROB=*name*
P=*name*

names the variable containing the predicted probabilities. For the *events/trials* syntax or *single-trial* syntax with binary response, it is the predicted event probability. For a cumulative model, it is the predicted cumulative probability (that is, the probability that the response variable is less than or equal to the value of _LEVEL_); and for the generalized logit model, it is the predicted individual probability (that is, the probability of the response category represented by the value of _LEVEL_).

PREDPROBS=(*keywords***)**

requests individual, cumulative, or cross validated predicted probabilities. Descriptions of the *keywords* are as follows.

INDIVIDUAL | I requests the predicted probability of each response level. For a response variable Y with three levels, 1, 2, and 3, the individual probabilities are Pr(Y=1), Pr(Y=2), and Pr(Y=3).

CUMULATIVE | C requests the cumulative predicted probability of each response level. For a response variable Y with three levels, 1,2, and 3, the cumulative probabilities are Pr(Y$\leq$1), Pr(Y$\leq$2), and Pr(Y$\leq$3). The cumulative probability for the last response level always has the constant value of 1. For generalized logit models, the cumulative predicted probabilities are not computed and are set to missing.

CROSSVALIDATE | XVALIDATE | X requests the cross validated individual predicted probability of each response level. These probabilities are derived from the leave-one-out principle; that is, dropping the data of one subject and reestimating the parameter estimates. PROC LOGISTIC uses a less expensive one-step approximation to compute the parameter estimates. This option is only valid for binary response models; for nominal and ordinal models, the cross validated probabilities are not computed and are set to missing.

See the "Details of the PREDPROBS= Option" section on page 2322 at the end of this section for further details.

STDXBETA=*name*

names the variable containing the standard error estimates of XBETA (the definition of which follows).

UPPER=*name*
U=*name*

names the variable containing the upper confidence limits for π, where π is the probability of the event response if *events/trials* syntax or *single-trial* syntax with binary response is specified; for a cumulative model, π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_); for the generalized logit model, it is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_). See the ALPHA= option to set the confidence level.

XBETA=*name*

names the variable containing the estimates of the linear predictor $\alpha_i + \boldsymbol{\beta}'\mathbf{x}$, where i is the corresponding ordered value of _LEVEL_.

Statistic Options Only for Binary Response

C=*name*

specifies the confidence interval displacement diagnostic that measures the influence of individual observations on the regression estimates.

CBAR=*name*

specifies the another confidence interval displacement diagnostic, which measures the overall change in the global regression estimates due to deleting an individual observation.

DFBETAS= _ALL_
DFBETAS=*var-list*

specifies the standardized differences in the regression estimates for assessing the effects of individual observations on the estimated regression parameters in the fitted model. You can specify a list of up to $s + 1$ variable names, where s is the number of explanatory variables in the MODEL statement, or you can specify just the keyword _ALL_. In the former specification, the first variable contains the standardized differences in the intercept estimate, the second variable contains the standardized differences in the parameter estimate for the first explanatory variable in the MODEL statement, and so on. In the latter specification, the DFBETAS statistics are named DFBETA_xxx, where xxx is the name of the regression parameter. For example, if the model contains two variables X1 and X2, the specification DFBETAS=_ALL_ produces three DFBETAS statistics: DFBETA_Intercept, DFBETA_X1, and DFBETA_X2. If an explanatory variable is not included in the final model, the corresponding output variable named in DFBETAS=*var-list* contains missing values.

DIFCHISQ=*name*

specifies the change in the chi-square goodness-of-fit statistic attributable to deleting the individual observation.

DIFDEV=*name*

specifies the change in the deviance attributable to deleting the individual observation.

H=*name*

specifies the diagonal element of the hat matrix for detecting extreme points in the design space.

RESCHI=*name*

specifies the Pearson (Chi) residual for identifying observations that are poorly accounted for by the model.

RESDEV=*name*

specifies the deviance residual for identifying poorly fitted observations.

Other Options

You can specify the following option after a slash.

ALPHA=α

sets the level of significance α for $100(1-\alpha)$% confidence limits for the appropriate response probabilities. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

Details of the PREDPROBS= Option

You can request any of the three given types of predicted probabilities. For example, you can request both the individual predicted probabilities and the cross validated probabilities by specifying PREDPROBS=(I X).

When you specify the PREDPROBS= option, two automatic variables _FROM_ and _INTO_ are included for the *single-trial* syntax and only one variable, _INTO_, is included for the *events/trials* syntax. The _FROM_ variable contains the formatted value of the observed response. The variable _INTO_ contains the formatted value of the response level with the largest individual predicted probability.

If you specify PREDPROBS=INDIVIDUAL, the OUTPUT data set contains k additional variables representing the individual probabilities, one for each response level, where k is the maximum number of response levels across all BY-groups. The names of these variables have the form IP_*xxx*, where *xxx* represents the particular level. The representation depends on the following situations.

- If you specify *events/trials* syntax, *xxx* is either ‘Event’ or ‘Nonevent’. Thus, the variable containing the event probabilities is named IP_Event and the variable containing the nonevent probabilities is named IP_Nonevent.
- If you specify the *single-trial* syntax with more than one BY group, *xxx* is 1 for the first ordered level of the response, 2 for the second ordered level of the response, . . ., and so forth, as given in the “Response Profile” table. The variable containing the predicted probabilities Pr(Y=1) is named IP_1, where Y is the response variable. Similarly, IP_2 is the name of the variable containing the predicted probabilities Pr(Y=2), and so on.
- If you specify the *single-trial* syntax with no BY-group processing, *xxx* is the left-justified formatted value of the response level (the value may be truncated so that IP_*xxx* does not exceed 32 characters.) For example, if Y is the response variable with response levels ‘None’, ‘Mild’, and ‘Severe’, the variables representing individual probabilities Pr(Y=’None’), P(Y=’Mild’), and P(Y=’Severe’) are named IP_None, IP_Mild, and IP_Severe, respectively.

If you specify PREDPROBS=CUMULATIVE, the OUTPUT data set contains k additional variables representing the cumulative probabilities, one for each response level, where k is the maximum number of response levels across all BY-groups. The names of these variables have the form CP_*xxx*, where *xxx* represents the particular response level. The naming convention is similar to that given by PREDPROBS=INDIVIDUAL. The PREDPROBS=CUMULATIVE values are the same as those output by the PREDICT=keyword, but are arranged in variables on each output observation rather than in multiple output observations.

If you specify PREDPROBS=CROSSVALIDATE, the OUTPUT data set contains k additional variables representing the cross validated predicted probabilities of the k response levels, where k is the maximum number of response levels across all BY-groups. The names of these variables have the form XP_*xxx*, where *xxx* represents the particular level. The representation is the same as that given by PREDPROBS=INDIVIDUAL except that for the *events/trials* syntax there are four variables for the cross validated predicted probabilities instead of two:

XP_EVENT_R1E is the cross validated predicted probability of an event when a current event trial is removed.

XP_NONEVENT_R1E is the cross validated predicted probability of a nonevent when a current event trial is removed.

XP_EVENT_R1N is the cross validated predicted probability of an event when a current nonevent trial is removed.

XP_NONEVENT_R1N is the cross validated predicted probability of a nonevent when a current nonevent trial is removed.

The cross validated predicted probabilities are precisely those used in the CTABLE option. See the "Predicted Probability of an Event for Classification" section on page 2352 for details of the computation.

SCORE Statement

SCORE $<$ *options* $>$ **;**

The SCORE statement creates a data set that contains all the data in the DATA= data set together with posterior probabilities and, optionally, prediction confidence intervals. Fit statistics are displayed on request. If you have binary response data, the SCORE statement can be used to create the OUTROC= data set containing data for the ROC curve. You can specify several SCORE statements. FREQ, WEIGHT, and BY statements can be used with the SCORE statements.

See the "Scoring Data Sets" section on page 2362 for more information, and see Example 42.13 on page 2462 for an illustration of how to use this statement.

You can specify the following options:

ALPHA=α

specifies the significance level α for $100(1-\alpha)$% confidence intervals. By default, α is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified. This option has no effect unless the CLM option in the SCORE statement is requested.

CLM

outputs the Wald-test-based confidence limits for the predicted probabilities. This option is not available when the INMODEL= data set is created with the NOCOV option.

DATA=*SAS-data-set*

names the SAS data set that you want to score. If you omit the DATA= option in the SCORE statement, then scoring is performed on the DATA= input data set in the PROC LOGISTIC statement, if specified; otherwise, the DATA=_LAST_ data set is used.

It is not necessary for the DATA= data set in the SCORE statement to contain the response variable unless you are specifying the FITSTAT or OUTROC= option.

Only those variables involved in the fitted model effects are required in the DATA= data set in the SCORE statement. For example, the following code uses forward selection to select effects.

```
proc logistic data=Neuralgia outmodel=sasuser.Model;
   class Treatment Sex;
   model Pain(event='Yes')= Treatment|Sex Age
         / selection=forward sle=.01;
run;
```

Suppose Treatment and Age are the effects selected for the final model. You can score a data set which does not contain the variable Sex since the effect Sex is not in the model that the scoring is based on.

```
proc logistic inmodel=sasuser.Model;
   score data=Neuralgia(drop=Sex);
run;
```

FITSTAT

displays a table of fit statistics. Four statistics are computed: total frequency, total weight, log likelihood, and misclassification rate.

OUT=*SAS-data-set*

names the SAS data set that contains the predicted information. If you omit the OUT= option, the output data set is created and given a default name using the DATAn convention.

OUTROC=*SAS-data-set*

names the SAS data set that contains the ROC curve for the DATA= data set. The ROC curve is computed only for binary response data. See the section "OUTROC= Output Data Set" on page 2378 for the list of variables in this data set.

PRIOR=*SAS-data-set*

names the SAS data set that contains the priors of the response categories. The priors may be values proportional to the prior probabilities; thus, they do not necessarily sum to one. This data set should include a variable named _PRIOR_ that contains the prior probabilities. For events/trials MODEL syntax, this data set should also include an _OUTCOME_ variable that contains the values EVENT and NONEVENT; for single-trial MODEL syntax, this data set should include the response variable that contains the unformatted response categories. See Example 42.13 on page 2462 for an example.

PRIOREVENT=*value*

specifies the prior event probability for a binary response model. If both PRIOR= and PRIOREVENT= options are specified, the PRIOR= option takes precedence.

ROCEPS=*value*

specifies the criterion for grouping estimated event probabilities that are close to each other for the ROC curve. In each group, the difference between the largest and the smallest estimated event probability does not exceed the given value. The $value$ must be between 0 and 1; the default value is 1E$-$4. The smallest estimated probability in each group serves as a cutpoint for predicting an event response. The ROCEPS= option has no effect if the OUTROC= option is not specified.

STRATA Statement

STRATA *variable <(option)>< variable <(option)>...>< / options >* **;**

The STRATA statement names the *variables* that define *strata* or *matched sets* to use in a *stratified conditional logistic regression* of binary response data. Observations having the same variable levels are in the same matched set. At least one variable must be specified to invoke the stratified analysis, and the usual unconditional asymptotic analysis is not performed. The stratified logistic model has the form

$$\text{logit}(\pi_{hi}) = \alpha_h + \mathbf{x}'_{hi}\boldsymbol{\beta}$$

where π_{hi} is the event probability for the ith observation in stratum h having covariates $\mathbf{x}_{hi}$, and where the stratum-specific intercepts α_h are the nuisance parameters which are to be conditioned out.

STRATA variables can also be specified in the MODEL statement as classification or continuous covariates; however, the effects are nondegenerate only when crossed with a non-stratification variable. Specifying several STRATA statements is the same as specifying one STRATA statement containing all the strata variables. The STRATA variables can be either character or numeric, and the formatted values of the STRATA variables determine the levels. Thus, you can use also use formats to group values into levels. See the discussion of the FORMAT procedure in the *SAS Procedures Guide*.

If an EXACT statement is also specified, then a stratified *exact* conditional logistic regression is performed.

The SCORE and WEIGHT statements are not available with a STRATA statement. The following MODEL options are also not supported with a STRATA statement: CLPARM=PL, CLODDS=PL, CTABLE, LACKFIT, LINK=, NOFIT, OUTMODEL=, OUTROC=, and SCALE=.

The “Strata Summary” table is displayed by default; it displays the number of strata which have a specific number of events and nonevents. For example, if you are analyzing a 1:5 matched study, this table enables you to verify that every stratum in the analysis has exactly one event and five non-events. Strata containing only events or only non-events are reported in this table, but such strata are uninformative and are not used in the analysis. (Note that you can use the response variable option EVENT= to identify the events; otherwise, the first ordered response category is the event.)

The following option can be specified for a stratification variable by enclosing the option in parentheses after the variable name, or it can be specified globally for all STRATA variables after a slash (/).

MISSING

treats missing values (‘.’, ‘.A’,...,‘.Z’ for numeric variables and blanks for character variables) as valid STRATA variable values.

The following strata options are also available after the slash.

NOSUMMARY

suppresses the display of the “Strata Summary” table.

INFO

displays the “Strata Information” table, which includes the stratum number, levels of the STRATA variables that define the stratum, the number of events, the number of nonevents, and the total frequency for each stratum. Since the number of strata can be very large, this table is only displayed on request.

TEST Statement

< *label:* > **TEST** *equation1* < , . . . , < *equationk* >>< / *option* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. The Wald test is used to test jointly the null hypotheses ($H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$) specified in a single TEST statement. When $\mathbf{c} = \mathbf{0}$ you should specify a CONTRAST statement instead.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\mathbf{c}$ vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output and should always be included. You can submit multiple TEST statements.

The form of an *equation* is as follows:

term < ±*term* **. . .** > < = ±*term* < ±*term* **. . .** >>

where *term* is a parameter of the model, or a constant, or a constant times a parameter. For a binary response model, the intercept parameter is named INTERCEPT; for an ordinal response model, the intercept parameters are named INTERCEPT, INTERCEPT2, INTERCEPT3, and so on. See the “Parameter Names in the OUTEST= Data Set” section on page 2375 for details on parameter naming conventions. When no equal sign appears, the expression is set to 0. The following code illustrates possible uses of the TEST statement:

```
proc logistic;
   model y= a1 a2 a3 a4;
   test1: test intercept + .5 * a2 = 0;
   test2: test intercept + .5 * a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

You can specify the following option in the TEST statement after a slash(/).

PRINT

displays intermediate calculations in the testing of the null hypothesis $H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$. This includes $\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}'$ bordered by $(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$ and $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$, where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator of $\boldsymbol{\theta}$ and $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\theta}}$.

For more information, see the "Testing Linear Hypotheses about the Regression Coefficients" section on page 2358.

UNITS Statement

UNITS *independent1 = list1* < . . . *independentk = listk* >< / *option* > **;**

The UNITS statement enables you to specify units of change for the continuous explanatory variables so that customized odds ratios can be estimated. An estimate of the corresponding odds ratio is produced for each unit of change specified for an explanatory variable. The UNITS statement is ignored for CLASS variables. If the CLODDS= option is specified in the MODEL statement, the corresponding confidence limits for the odds ratios are also displayed.

The term *independent* is the name of an explanatory variable and *list* represents a list of units of change, separated by spaces, that are of interest for that variable. Each unit of change in a list has one of the following forms:

- *number*
- SD or $-$SD
- *number* * SD

where *number* is any nonzero number, and SD is the sample standard deviation of the corresponding independent variable. For example, $X = -2$ requests an odds ratio that represents the change in the odds when the variable X is decreased by two units. $X = 2*$SD requests an estimate of the change in the odds when X is increased by two sample standard deviations.

You can specify the following option in the UNITS statement after a slash(/).

DEFAULT= *list*

gives a list of units of change for all explanatory variables that are not specified in the UNITS statement. Each unit of change can be in any of the forms described previously. If the DEFAULT= option is not specified, PROC LOGISTIC does not produce customized odds ratio estimates for any explanatory variable that is not listed in the UNITS statement.

For more information, see the "Odds Ratio Estimation" section on page 2347.

WEIGHT Statement

WEIGHT *variable* < / *option* > **;**

When a WEIGHT statement appears, each observation in the input data set is weighted by the value of the WEIGHT variable. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with negative, zero, or missing values for the WEIGHT variable are not used in the model fitting. When the WEIGHT statement is not specified, each observation is assigned a weight of 1.

If a SCORE statement is specified, then the WEIGHT variable is used for computing fit statistics and the ROC curve, but it is not required for scoring. If the DATA= data set in the SCORE statement does not contain the WEIGHT variable, the weights are assumed to be 1 and a warning message is issued in the LOG. If you fit a model and perform the scoring in the same run, the same WEIGHT variable is used for fitting and scoring. If you fit a model in a previous run and input it with the INMODEL= option in the current run, then the WEIGHT variable can be different from the one used in the previous run; however, if a WEIGHT variable was not specified in the previous run you can still specify a WEIGHT variable in the current run.

The following option can be added to the WEIGHT statement after a slash (/).

NORMALIZE
NORM

causes the weights specified by the WEIGHT variable to be normalized so that they add up to the actual sample size. With this option, the estimated covariance matrix of the parameter estimators is invariant to the scale of the WEIGHT variable.

Details

Missing Values

Any observation with missing values for the response, offset, strata, or explanatory variables is excluded from the analysis; however, missing values are valid for variables specified with the MISSING option in the CLASS or STRATA statements. The estimated linear predictor and its standard error estimate, the fitted probabilities and confidence limits, and the regression diagnostic statistics are not computed for any observation with missing offset or explanatory variable values. However, if only the response value is missing, the linear predictor, its standard error, the fitted individual and cumulative probabilities, and confidence limits for the cumulative probabilities can be computed and output to a data set using the OUTPUT statement.

Response Level Ordering

Response level ordering is important because, by default, PROC LOGISTIC models the probability of response levels with *lower Ordered Value*. Ordered Values are assigned to response levels in ascending sorted order (that is, the lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so on) and are displayed in the “Response Profiles” table. If your response variable Y takes values in $\{1, \ldots, k+1\}$, then, by default, the functions modeled with the cumulative model are

$$\text{logit}(\Pr(Y \le i|\mathbf{x})), \quad i = 1, \ldots, k$$

and for the generalized logit model the functions modeled are

$$\log\left(\frac{\Pr(Y = i|\mathbf{x})}{\Pr(Y = k+1|\mathbf{x})}\right), \quad i = 1, \ldots, k$$

where the highest Ordered Value $Y = k + 1$ is the reference level. You can change which probabilities are modeled by specifying the EVENT=, REF=, DESCENDING, or ORDER= response variable options in the MODEL statement.

For binary response data with event and nonevent categories, if your event category has a higher Ordered Value, then the nonevent is modeled and, since the default response function modeled is

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

where π is the probability of the response level assigned Ordered Value 1, and since

$$\text{logit}(\pi) = -\text{logit}(1-\pi)$$

the effect of reversing the order of the two response values is to change the signs of α and $\boldsymbol{\beta}$ in the model $\text{logit}(\pi) = \alpha + \boldsymbol{\beta}'\boldsymbol{x}$.

For example, suppose the binary response variable Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. By default, PROC LOGISTIC assigns Ordered Value 1 to response level Y=0, and Ordered Value 2 to response level Y=1. As a result, PROC LOGISTIC models the probability of the nonevent (Ordered Value=1) category. To model the event without changing the values of the variable Y, you can do the following:

- Explicitly state which response level is to be modeled using the response variable option EVENT= in the MODEL statement,

  ```
  model Y(event='1') = Exposure;
  ```

- Specify the response variable option REF= in the MODEL statement as the nonevent category for the response variable. This option is most useful for generalized logit models.

  ```
  model Y(ref='0') = Exposure;
  ```

- Specify the response variable option DESCENDING in the MODEL statement,

  ```
  model Y(descending)=Exposure;
  ```

- Assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. For this example, Y=1 is assigned formatted value ‘event’ and Y=0 is assigned formatted value ‘nonevent’. Since ORDER=FORMATTED by default, Ordered Value 1 is assigned to response level Y=1 so the procedure models the event.

```
proc format;
   value Disease 1='event' 0='nonevent';
run;
proc logistic;
   format Y Disease.;
   model Y=Exposure;
run;
```

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three design variables have a value of -1. For instance, if the reference level is 7 (REF='7'), the design matrix columns for A are as follows.

Effect Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all 4 levels.

Caution: PROC LOGISTIC initially parameterizes the CLASS variables by looking at the levels of the variables across the complete data set. If you have an *unbalanced* replication of levels across variables, then the design matrix and the parameter interpretation may be different from what you expect. For instance, suppose that in addition to the four-level variable A discussed above, you have another variable B with two levels, where the fourth level of A only occurs with the first level of B. If your model contains the effect A(B), then the design for A within the second level of B will not be a differential effect. In particular, the design will look like the following.

Effect Coding							
		Design Matrix					
		A(B=1)			**A(B=2)**		
B	**A**	**A1**	**A2**	**A5**	**A1**	**A2**	**A5**
1	1	1	0	0	0	0	0
1	2	0	1	0	0	0	0
1	5	0	0	1	0	0	0
1	7	−1	−1	−1	0	0	0
2	1	0	0	0	1	0	0
2	2	0	0	0	0	1	0
2	5	0	0	0	0	0	1

PROC LOGISTIC will then detect linear dependency among the last three design variables and set the parameter for A5(B=2) to zero, resulting in an interpretation of these parameters as if they were reference- or dummy-coded. The GLM or REFERENCE parameterization may be more appropriate for such problems.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects using the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three design variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects using the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, . . . according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three design variables have a value of 0. For instance, if the reference level is 7 (REF='7'), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	**AOEFF1**	**AOEFF2**	**AOEFF3**
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	**AOORD1**	**AOORD2**	**AOORD3**
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	**AOPOLY1**	**AOPOLY2**	**AOPOLY5**
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	**AOREF1**	**AOREF2**	**AOREF3**
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Link Functions and the Corresponding Distributions

Four link functions are available in the LOGISTIC procedure. The logit function is the default. To specify a different link function, use the LINK= option in the MODEL statement. The link functions and the corresponding distributions are as follows:

- The logit function

  $$g(p) = \log(p/(1-p))$$

  is the inverse of the cumulative logistic distribution function, which is

  $$F(x) = 1/(1+\exp(-x)) = \exp(x)/(1+\exp(x))$$

- The probit (or normit) function

  $$g(p) = \Phi^{-1}(p)$$

  is the inverse of the cumulative standard normal distribution function, which is

  $$F(x) = \Phi(x) = (2\pi)^{-1/2}\int_{-\infty}^{x} \exp(-z^2/2)dz$$

  Traditionally, the probit function contains the additive constant 5, but throughout PROC LOGISTIC, the terms probit and normit are used interchangeably.

- The complementary log-log function

  $$g(p) = \log(-\log(1-p))$$

  is the inverse of the cumulative extreme-value function (also called the Gompertz distribution), which is

  $$F(x) = 1 - \exp(-\exp(x))$$

- The generalized logit function extends the binary logit link to a vector of levels $(p_1, \ldots, p_{k+1})$ by contrasting each level with a fixed level

  $$g(p_i) = \log(p_i/p_{k+1}) \quad i = 1, \ldots, k$$

The variances of the normal, logistic, and extreme-value distributions are not the same. Their respective means and variances are

Distribution	Mean	Variance
Normal	0	1
Logistic	0	$\pi^2/3$
Extreme-value	$-\gamma$	$\pi^2/6$

where γ is the Euler constant. In comparing parameter estimates using different link functions, you need to take into account the different scalings of the corresponding distributions and, for the complementary log-log function, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates using the logit link function should be about $\pi/\sqrt{3}$ larger than the estimates from the probit link function.

Determining Observations for Likelihood Contributions

If you use *events/trials* MODEL syntax, each observation is split into two observations. One has response value 1 with a frequency equal to the frequency of the original observation (which is 1 if the FREQ statement is not used) times the value of the *events* variable. The other observation has response value 2 and a frequency equal to the frequency of the original observation times the value of (*trials* − *events*). These two observations will have the same explanatory variable values and the same FREQ and WEIGHT values as the original observation.

For either *single-trial* or *events/trials* syntax, let j index all observations. In other words, for *single-trial* syntax, j indexes the actual observations. And, for *events/trials* syntax, j indexes the observations after splitting (as described previously). If your data set has 30 observations and you use *single-trial* syntax, j has values from 1 to 30; if you use *events/trials* syntax, j has values from 1 to 60.

Suppose the response variable in a cumulative response model can take on the ordered values $1, \ldots, k, k{+}1$ where k is an integer ≥ 1. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector $\mathbf{x}_j$ is given by

$$L_j = \begin{cases} F(\alpha_1 + \boldsymbol{\beta}'\mathbf{x}_j) & y_j = 1 \\ F(\alpha_i + \boldsymbol{\beta}'\mathbf{x}_j) - F(\alpha_{i-1} + \boldsymbol{\beta}'\mathbf{x}_j) & 1 < y_j = i \leq k \\ 1 - F(\alpha_k + \boldsymbol{\beta}'\mathbf{x}_j) & y_j = k + 1 \end{cases}$$

where $F(\cdot)$ is the logistic, normal, or extreme-value distribution function, $\alpha_1, \ldots, \alpha_k$ are ordered intercept parameters, and $\boldsymbol{\beta}$ is the slope parameter vector.

For the generalized logit model, letting the $k + 1$st level be the reference level, the intercepts $\alpha_1, \ldots, \alpha_k$ are unordered and the slope vector $\boldsymbol{\beta}_i$ varies with each logit. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector $\mathbf{x}_j$ is given by

$$L_j = \Pr(Y = y_j|\mathbf{x}_j) = \begin{cases} \dfrac{e^{\alpha_i + \mathbf{x}_j'\boldsymbol{\beta}_i}}{1 + \sum_{m=1}^{k} e^{\alpha_m + \mathbf{x}_j'\boldsymbol{\beta}_m}} & 1 \leq y_j = i \leq k \\ \dfrac{1}{1 + \sum_{m=1}^{k} e^{\alpha_m + \mathbf{x}_j'\boldsymbol{\beta}_m}} & y_j = k + 1 \end{cases}$$

Iterative Algorithms for Model-Fitting

Two iterative maximum likelihood algorithms are available in PROC LOGISTIC. The default is the Fisher-scoring method, which is equivalent to fitting by iteratively reweighted least squares. The alternative algorithm is the Newton-Raphson method. Both algorithms give the same parameter estimates; however, the estimated covariance matrix of the parameter estimators may differ slightly. This is due to the fact that the Fisher-scoring method is based on the expected information matrix while the Newton-Raphson method is based on the observed information matrix. In the case of a binary logit model, the observed and expected information matrices are identical, resulting in identical estimated covariance matrices for both algorithms. For a generalized logit model, only the Newton-Raphson technique is available. You can use the TECHNIQUE= option to select a fitting algorithm.

Iteratively Reweighted Least-Squares Algorithm (Fisher Scoring)

Consider the multinomial variable $\mathbf{Z}_j = (Z_{1j}, \ldots, Z_{kj})'$ such that

$$Z_{ij} = \begin{cases} 1 & \text{if } Y_j = i \\ 0 & \text{otherwise} \end{cases}$$

With π_{ij} denoting the probability that the *j*th observation has response value *i*, the expected value of $\mathbf{Z}_j$ is $\boldsymbol{\pi}_j = (\pi_{1j}, \ldots, \pi_{kj})'$, and $\pi_{(k+1)j} = 1 - \sum_{i=1}^{k} \pi_{ij}$. The covariance matrix of $\mathbf{Z}_j$ is $\mathbf{V}_j$, which is the covariance matrix of a multinomial random variable for one trial with parameter vector $\boldsymbol{\pi}_j$. Let $\boldsymbol{\theta}$ be the vector of regression parameters; in other words, $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}')'$. Let $\mathbf{D}_j$ be the matrix of partial derivatives of $\boldsymbol{\pi}_j$ with respect to $\boldsymbol{\theta}$. The estimating equation for the regression parameters is

$$\sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z_j} - \boldsymbol{\pi}_\mathbf{j}) = \mathbf{0}$$

where $\mathbf{W}_j = w_j f_j \mathbf{V}_j^-$, w_j and f_j are the WEIGHT and FREQ values of the jth observation, and $\mathbf{V}_j^-$ is a generalized inverse of $\mathbf{V}_j$. PROC LOGISTIC chooses $\mathbf{V}_j^-$ as the inverse of the diagonal matrix with $\boldsymbol{\pi}_j$ as the diagonal.

With a starting value of $\boldsymbol{\theta}_0$, the maximum likelihood estimate of $\boldsymbol{\theta}$ is obtained iteratively as

$$\boldsymbol{\theta}_{m+1} = \boldsymbol{\theta}_m + (\sum_j \mathbf{D}_j' \mathbf{W_j} \mathbf{D}_j)^{-1} \sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z_j} - \boldsymbol{\pi}_\mathbf{j})$$

where $\mathbf{D}_j$, $\mathbf{W}_j$, and $\boldsymbol{\pi}_j$ are evaluated at $\boldsymbol{\theta}_m$. The expression after the plus sign is the step size. If the likelihood evaluated at $\boldsymbol{\theta}_{m+1}$ is less than that evaluated at $\boldsymbol{\theta}_m$, then $\boldsymbol{\theta}_{m+1}$ is recomputed by step-halving or ridging. The iterative scheme continues until convergence is obtained, that is, until $\boldsymbol{\theta}_{m+1}$ is sufficiently close to $\boldsymbol{\theta}_m$. Then the maximum likelihood estimate of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}_{m+1}$.

The covariance matrix of $\widehat{\boldsymbol{\theta}}$ is estimated by

$$\widehat{\text{cov}}(\widehat{\boldsymbol{\theta}}) = (\sum_j \widehat{\mathbf{D}}_j' \widehat{\mathbf{W}}_j \widehat{\mathbf{D}}_j)^{-1}$$

where $\widehat{\mathbf{D}}_j$ and $\widehat{\mathbf{W}}_j$ are, respectively, $\mathbf{D}_j$ and $\mathbf{W}_j$ evaluated at $\widehat{\boldsymbol{\theta}}$.

By default, starting values are zero for the slope parameters, and for the intercept parameters, starting values are the observed cumulative logits (that is, logits of the observed cumulative proportions of response). Alternatively, the starting values may be specified with the INEST= option.

Newton-Raphson Algorithm

For cumulative models, let the parameter vector be $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}')'$, and for the generalized logit model denote $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}'_1, \ldots, \boldsymbol{\beta}'_k)'$. The gradient vector and the Hessian matrix are given, respectively, by

$$\begin{aligned}\mathbf{g} &= \sum_j w_j f_j \frac{\partial l_j}{\partial \boldsymbol{\theta}} \\ \mathbf{H} &= \sum_j -w_j f_j \frac{\partial^2 l_j}{\partial \boldsymbol{\theta}^2}\end{aligned}$$

where $l_j = \log L_j$ is the log likelihood for the jth observation. With a starting value of $\boldsymbol{\theta}_0$, the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ is obtained iteratively until convergence is obtained:

$$\boldsymbol{\theta}_{m+1} = \boldsymbol{\theta}_m + \mathbf{H}^{-1}\mathbf{g}$$

where $\mathbf{H}$ and $\mathbf{g}$ are evaluated at $\boldsymbol{\theta}_m$. If the likelihood evaluated at $\boldsymbol{\theta}_{m+1}$ is less than that evaluated at $\boldsymbol{\theta}_m$, then $\boldsymbol{\theta}_{m+1}$ is recomputed by step-halving or ridging.

The covariance matrix of $\widehat{\boldsymbol{\theta}}$ is estimated by

$$\widehat{\mathrm{cov}}(\widehat{\boldsymbol{\theta}}) = \widehat{\mathbf{H}}^{-1}$$

where $\widehat{\mathbf{H}}$ is $\mathbf{H}$ evaluated at $\widehat{\boldsymbol{\theta}}$.

Convergence Criteria

Four convergence criteria are allowed, namely, ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E$-$8.

If you specify a STRATA statement, then all unspecified (or non-default) criteria are also compared to zero. For example, only specifying the criterion XCONV=1e$-$8 but attaining FCONV=0 terminates the optimization even if the XCONV= criterion is not satisfied, because the log likelihood has reached its maximum.

Existence of Maximum Likelihood Estimates

The likelihood equation for a logistic regression model does not always have a finite solution. Sometimes there is a nonunique maximum on the boundary of the parameter space, at infinity. The existence, finiteness, and uniqueness of maximum likelihood estimates for the logistic regression model depend on the patterns of data points in the observation space (Albert and Anderson 1984; Santner and Duffy 1986). The existence checks are not performed for conditional logistic regression.

Consider a binary response model. Let Y_j be the response of the ith subject and let $\mathbf{x}_j$ be the vector of explanatory variables (including the constant 1 associated with the intercept). There are three mutually exclusive and exhaustive types of data configurations: complete separation, quasi-complete separation, and overlap.

Complete Separation

There is a complete separation of data points if there exists a vector $\mathbf{b}$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \mathbf{b}'\mathbf{x}_j > 0 & Y_j = 1 \\ \mathbf{b}'\mathbf{x}_j < 0 & Y_j = 2 \end{cases}$$

This configuration gives nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the log likelihood diminishes to zero, and the dispersion matrix becomes unbounded.

Quasi-Complete Separation

The data are not completely separable but there is a vector $\mathbf{b}$ such that

$$\begin{cases} \mathbf{b}'\mathbf{x}_j \geq 0 & Y_j = 1 \\ \mathbf{b}'\mathbf{x}_j \leq 0 & Y_j = 2 \end{cases}$$

and equality holds for at least one subject in each response group. This configuration also yields nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the dispersion matrix becomes unbounded and the log likelihood diminishes to a nonzero constant.

Overlap

If neither complete nor quasi-complete separation exists in the sample points, there is an overlap of sample points. In this configuration, the maximum likelihood estimates exist and are unique.

Complete separation and quasi-complete separation are problems typically encountered with small data sets. Although complete separation can occur with any type of data, quasi-complete separation is not likely with truly continuous explanatory variables.

The LOGISTIC procedure uses a simple empirical approach to recognize the data configurations that lead to infinite parameter estimates. The basis of this approach is that any convergence method of maximizing the log likelihood must yield a solution giving complete separation, if such a solution exists. In maximizing the log likelihood, there is no checking for complete or quasi-complete separation if convergence is attained in eight or fewer iterations. Subsequent to the eighth iteration, the probability of the observed response is computed for each observation. If the probability of the observed response is one for all observations, there is a complete separation of data points and the iteration process is stopped. If the complete separation of data has not been determined and an observation is identified to have an extremely large probability ($\geq$0.95) of the observed response, there are two possible situations. First, there is overlap in the data set, and the observation is an atypical observation of its own group. The iterative process, if allowed to continue, will stop when a maximum is reached. Second, there is quasi-complete separation in the data set, and the asymptotic dispersion matrix is unbounded. If any of the diagonal elements of the dispersion matrix for the standardized observations vectors (all explanatory variables standardized to zero mean and unit variance) exceeds 5000, quasi-complete separation is declared and the iterative process is stopped. If either complete separation or quasi-complete separation is detected, a warning message is displayed in the procedure output.

Checking for quasi-complete separation is less foolproof than checking for complete separation. The NOCHECK option in the MODEL statement turns off the process of checking for infinite parameter estimates. In cases of complete or quasi-complete separation, turning off the checking process typically results in the procedure failing to converge. The presence of a WEIGHT statement also turns off the checking process.

Effect Selection Methods

Five effect-selection methods are available. The simplest method (and the default) is SELECTION=NONE, for which PROC LOGISTIC fits the complete model as specified in the MODEL statement. The other four methods are FORWARD for forward selection, BACKWARD for backward elimination, STEPWISE for stepwise selection, and SCORE for best subsets selection. These methods are specified with the SELECTION= option in the MODEL statement. Intercept parameters are forced to stay in the model unless the NOINT option is specified.

When SELECTION=FORWARD, PROC LOGISTIC first estimates parameters for effects forced into the model. These effects are the intercepts and the first n explanatory effects in the MODEL statement, where n is the number specified by the START= or INCLUDE= option in the MODEL statement (n is zero by default). Next, the procedure computes the score chi-square statistic for each effect not in the model and examines the largest of these statistics. If it is significant at the SLENTRY= level, the corresponding effect is added to the model. Once an effect is entered in the model, it is never removed from the model. The process is repeated until none of the remaining effects meet the specified level for entry or until the STOP= value is reached.

When SELECTION=BACKWARD, parameters for the complete model as specified in the MODEL statement are estimated unless the START= option is specified. In that case, only the parameters for the intercepts and the first n explanatory effects in the MODEL statement are estimated, where n is the number specified by the START= option. Results of the Wald test for individual parameters are examined. The least significant effect that does not meet the SLSTAY= level for staying in the model is removed. Once an effect is removed from the model, it remains excluded. The process is repeated until no other effect in the model meets the specified level for removal or until the STOP= value is reached. Backward selection is often less successful than forward or stepwise selection because the full model fit in the first step is the model most likely to result in a complete or quasi-complete separation of response values as described in the previous section.

The SELECTION=STEPWISE option is similar to the SELECTION=FORWARD option except that effects already in the model do not necessarily remain. Effects are entered into and removed from the model in such a way that each forward selection step may be followed by one or more backward elimination steps. The stepwise selection process terminates if no further effect can be added to the model or if the effect just entered into the model is the only effect removed in the subsequent backward elimination.

For SELECTION=SCORE, PROC LOGISTIC uses the branch and bound algorithm of Furnival and Wilson (1974) to find a specified number of models with the highest likelihood score (chi-square) statistic for all possible model sizes, from 1, 2, 3 effect models, and so on, up to the single model containing all of the explanatory effects. The number of models displayed for each model size is controlled by the BEST= option. You can use the START= option to impose a minimum model size, and you can use the STOP= option to impose a maximum model size. For instance, with BEST=3, START=2, and STOP=5, the SCORE selection method displays the best three models (that is, the three models with the highest score chi-squares) containing 2, 3, 4, and 5 effects. The SELECTION=SCORE option is not available for models with CLASS variables.

The options FAST, SEQUENTIAL, and STOPRES can alter the default criteria for entering or removing effects from the model when they are used with the FORWARD, BACKWARD, or STEPWISE selection methods.

Model Fitting Information

Suppose the model contains s explanatory effects. For the jth observation, let $\hat{\pi}_j$ be the estimated probability of the observed response. The three criteria displayed by the LOGISTIC procedure are calculated as follows:

- -2 Log Likelihood:

  $$-2 \text{ Log L} = -2 \sum_j w_j f_j \log(\hat{\pi}_j)$$

  where w_j and f_j are the weight and frequency values of the jth observation. For binary response models using *events/trials* MODEL syntax, this is equivalent to

  $$-2 \text{ Log L} = -2 \sum_j w_j f_j \{r_j \log(\hat{\pi}_j) + (n_j - r_j) \log(1 - \hat{\pi}_j)\}$$

  where r_j is the number of events, n_j is the number of trials, and $\hat{\pi}_j$ is the estimated event probability.

- Akaike Information Criterion:

  $$\text{AIC} = -2 \text{ Log L} + 2p$$

  where p is the number of parameters in the model. For cumulative response models, $p = k + s$ where k is the total number of response levels minus one, and s is the number of explanatory effects. For the generalized logit model, $p = k(s + 1)$.

- Schwarz Criterion:

  $$\text{SC} = -2 \text{ Log L} + p \log(\sum_j f_j)$$

  where p is as defined previously.

The -2 Log Likelihood statistic has a chi-square distribution under the null hypothesis (that all the explanatory effects in the model are zero) and the procedure produces a p-value for this statistic. The AIC and SC statistics give two different ways of adjusting the -2 Log Likelihood statistic for the number of terms in the model and the number of observations used. These statistics should be used when comparing different models for the same data (for example, when you use the METHOD=STEPWISE option in the MODEL statement); lower values of the statistic indicate a more desirable model.

Generalized Coefficient of Determination

Cox and Snell (1989, pp. 208–209) propose the following generalization of the coefficient of determination to a more general linear model:

$$R^2 = 1 - \left\{ \frac{L(\mathbf{0})}{L(\widehat{\boldsymbol{\theta}})} \right\}^{\frac{2}{n}}$$

where $L(\mathbf{0})$ is the likelihood of the intercept-only model, $L(\widehat{\boldsymbol{\theta}})$ is the likelihood of the specified model, and n is the sample size. The quantity R^2 achieves a maximum of less than one for discrete models, where the maximum is given by

$$R^2_{\text{max}} = 1 - \{L(\mathbf{0})\}^{\frac{2}{n}}$$

Nagelkerke (1991) proposes the following adjusted coefficient, which can achieve a maximum value of one:

$$\tilde{R}^2 = \frac{R^2}{R^2_{\text{max}}}$$

Properties and interpretation of R^2 and $\tilde{R}^2$ are provided in Nagelkerke (1991). In the "Testing Global Null Hypothesis: BETA=0" table, R^2 is labeled as "RSquare" and $\tilde{R}^2$ is labeled as "Max-rescaled RSquare." Use the RSQUARE option to request R^2 and $\tilde{R}^2$.

Score Statistics and Tests

To understand the general form of the score statistics, let $\mathbf{U}(\boldsymbol{\theta})$ be the vector of first partial derivatives of the log likelihood with respect to the parameter vector $\boldsymbol{\theta}$, and let $\mathbf{H}(\boldsymbol{\theta})$ be the matrix of second partial derivatives of the log likelihood with respect to $\boldsymbol{\theta}$. That is, $\mathbf{U}(\boldsymbol{\theta})$ is the gradient vector, and $\mathbf{H}(\boldsymbol{\theta})$ is the Hessian matrix. Let $\mathbf{I}(\boldsymbol{\theta})$ be either $-\mathbf{H}(\boldsymbol{\theta})$ or the expected value of $-\mathbf{H}(\boldsymbol{\theta})$. Consider a null hypothesis H_0. Let $\hat{\boldsymbol{\theta}}_0$ be the MLE of $\boldsymbol{\theta}$ under H_0. The chi-square score statistic for testing H_0 is defined by

$$\mathbf{U}'(\hat{\boldsymbol{\theta}}_0)\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_0)\mathbf{U}(\hat{\boldsymbol{\theta}}_0)$$

and it has an asymptotic χ^2 distribution with r degrees of freedom under H_0, where r is the number of restrictions imposed on $\boldsymbol{\theta}$ by H_0.

Residual Chi-Square

When you use SELECTION=FORWARD, BACKWARD, or STEPWISE, the procedure calculates a residual score chi-square score statistic and reports the statistic, its degrees of freedom, and the p-value. This section describes how the statistic is calculated.

Suppose there are s explanatory effects of interest. The full cumulative response model has a parameter vector

$$\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_s)'$$

where $\alpha_1, \ldots, \alpha_k$ are intercept parameters, and $\beta_1, \ldots, \beta_s$ are the common slope parameters for the explanatory effects, and the full generalized logit model has a parameter vector

$$\begin{array}{rcl}\boldsymbol{\theta} & = & (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}'_1, \ldots, \boldsymbol{\beta}'_k)' \quad \text{with} \\ \boldsymbol{\beta}'_i & = & (\beta_{i1}, \ldots, \beta_{is}), \quad i = 1, \ldots, k\end{array}$$

where β_{ij} is the slope parameter for the jth effect in the ith logit.

Consider the null hypothesis $H_0\colon \beta_{t+1} = \ldots = \beta_s = 0$ where $t < s$ for the cumulative response model, and $H_0\colon \beta_{i,t+1} = \ldots = \beta_{is} = 0, t < s, i = 1, \ldots, k$ for the generalized logit model. For the reduced model with t explanatory effects, let $\hat{\alpha}_1, \ldots, \hat{\alpha}_k$ be the MLEs of the unknown intercept parameters, let $\hat{\beta}_1, \ldots, \hat{\beta}_t$ be the MLEs of the unknown slope parameters, and let $\hat{\boldsymbol{\beta}}'_{i(t)} = (\hat{\beta}_{i1}, \ldots, \hat{\beta}_{it}), i = 1, \ldots, k$ be those for the generalized logit model. The residual chi-square is the chi-square score statistic testing the null hypothesis H_0; that is, the residual chi-square is

$$\mathbf{U}'(\hat{\boldsymbol{\theta}}_0)\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_0)\mathbf{U}(\hat{\boldsymbol{\theta}}_0)$$

where for the cumulative response model $\hat{\boldsymbol{\theta}}_0 = (\hat{\alpha}_1, \ldots, \hat{\alpha}_k, \hat{\beta}_1, \ldots, \hat{\beta}_t, 0, \ldots, 0)'$, and for the generalized logit model $\hat{\boldsymbol{\theta}}_0 = (\hat{\alpha}_1, \ldots, \hat{\alpha}_k, \hat{\boldsymbol{\beta}}'_{1(t)}, \mathbf{0}'_{(s-t)}, \ldots \hat{\boldsymbol{\beta}}'_{k(t)}, \mathbf{0}'_{(s-t)})'$. where $\mathbf{0}_{(s-t)}$ denote a vector of $s-t$ zeros.

The residual chi-square has an asymptotic chi-square distribution with $s-t$ degrees of freedom ($k(s-t)$ for the generalized logit model). A special case is the global score chi-square, where the reduced model consists of the k intercepts and no explanatory effects. The global score statistic is displayed in the "Testing Global Null Hypothesis: BETA=0" table. The table is not produced when the NOFIT option is used, but the global score statistic is displayed.

Testing Individual Effects Not in the Model

These tests are performed in the FORWARD or STEPWISE method, and are displayed when the DETAILS option is specified. In the displayed output, the tests are labeled "Score Chi-Square" in the "Analysis of Effects Not in the Model" table and in the "Summary of Stepwise (Forward) Selection" table. This section describes how the tests are calculated.

Suppose that k intercepts and t explanatory variables (say $v_1, \ldots, v_t$) have been fitted to a model and that v_{t+1} is another explanatory variable of interest. Consider a full model with the k intercepts and $t+1$ explanatory variables ($v_1, \ldots, v_t, v_{t+1}$) and a reduced model with v_{t+1} excluded. The significance of v_{t+1} adjusted for $v_1, \ldots, v_t$ can be determined by comparing the corresponding residual chi-square with a chi-square distribution with one degree of freedom (k degrees of freedom for the generalized logit model).

Testing the Parallel Lines Assumption

For an ordinal response, PROC LOGISTIC performs a test of the parallel lines assumption. In the displayed output, this test is labeled "Score Test for the Equal Slopes Assumption" when the LINK= option is NORMIT or CLOGLOG. When LINK=LOGIT, the test is labeled as "Score Test for the Proportional Odds Assumption" in the output. For small sample sizes, this test may be too liberal (Stokes, Davis, and Koch 2000). This section describes the methods used to calculate the test.

For this test the number of response levels, $k+1$, is assumed to be strictly greater than 2. Let Y be the response variable taking values $1, \ldots, k, k+1$. Suppose there are s explanatory variables. Consider the general cumulative model without making the parallel lines assumption

$$g(\Pr(Y \le i \mid \mathbf{x})) = (1, \mathbf{x}')\boldsymbol{\theta}_i, \quad 1 \le i \le k$$

where $g(\cdot)$ is the link function, and $\boldsymbol{\theta}_i = (\alpha_i, \beta_{i1}, \ldots, \beta_{is})'$ is a vector of unknown parameters consisting of an intercept α_i and s slope parameters $\beta_{i1}, \ldots, \beta_{is}$. The parameter vector for this general cumulative model is

$$\boldsymbol{\theta} = (\boldsymbol{\theta}'_1, \ldots, \boldsymbol{\theta}'_k)'$$

Under the null hypothesis of parallelism $H_0\colon \beta_{1m} = \beta_{2m} = \cdots = \beta_{km}, 1 \le m \le s$, there is a single common slope parameter for each of the s explanatory variables. Let $\beta_1, \ldots, \beta_s$ be the common slope parameters. Let $\hat{\alpha}_1, \ldots, \hat{\alpha}_k$ and $\hat{\beta}_1, \ldots, \hat{\beta}_s$ be the MLEs of the intercept parameters and the common slope parameters. Then, under H_0, the MLE of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}_0 = (\hat{\boldsymbol{\theta}}_1', \ldots, \hat{\boldsymbol{\theta}}_k')' \quad \text{with} \quad \hat{\boldsymbol{\theta}}_i = (\hat{\alpha}_i, \hat{\beta}_1, \ldots, \hat{\beta}_s)' \quad 1 \le i \le k$$

and the chi-squared score statistic $\mathbf{U}'(\hat{\boldsymbol{\theta}}_0)\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_0)\mathbf{U}(\hat{\boldsymbol{\theta}}_0)$ has an asymptotic chi-square distribution with $s(k-1)$ degrees of freedom. This tests the parallel lines assumption by testing the equality of separate slope parameters simultaneously for all explanatory variables.

Confidence Intervals for Parameters

There are two methods of computing confidence intervals for the regression parameters. One is based on the profile likelihood function, and the other is based on the asymptotic normality of the parameter estimators. The latter is not as time-consuming as the former, since it does not involve an iterative scheme; however, it is not thought to be as accurate as the former, especially with small sample size. You use the CLPARM= option to request confidence intervals for the parameters.

Likelihood Ratio-Based Confidence Intervals

The likelihood ratio-based confidence interval is also known as the profile likelihood confidence interval. The construction of this interval is derived from the asymptotic χ^2 distribution of the generalized likelihood ratio test (Venzon and Moolgavkar 1988). Suppose that the parameter vector is $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_s)'$ and you want to compute a confidence interval for β_j. The profile likelihood function for $\beta_j = \gamma$ is defined as

$$l_j^*(\gamma) = \max_{\boldsymbol{\beta} \in \mathcal{B}_j(\gamma)} l(\boldsymbol{\beta})$$

where $\mathcal{B}_j(\gamma)$ is the set of all $\boldsymbol{\beta}$ with the jth element fixed at γ, and $l(\boldsymbol{\beta})$ is the log-likelihood function for $\boldsymbol{\beta}$. If $l_{\max} = l(\widehat{\boldsymbol{\beta}})$ is the log likelihood evaluated at the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$, then $2(l_{\max} - l_j^*(\beta_j))$ has a limiting chi-square distribution with one degree of freedom if β_j is the true parameter value. Let $l_0 = l_{\max} - .5\chi_1^2(1-\alpha)$, where $\chi_1^2(1-\alpha)$ is the $100(1-\alpha)$ percentile of the chi-square distribution with one degree of freedom. A $100(1-\alpha)$% confidence interval for β_j is

$$\{\gamma : l_j^*(\gamma) \ge l_0\}$$

The endpoints of the confidence interval are found by solving numerically for values of β_j that satisfy equality in the preceding relation. To obtain an iterative algorithm for computing the confidence limits, the log-likelihood function in a neighborhood of $\boldsymbol{\beta}$ is approximated by the quadratic function

$$\tilde{l}(\boldsymbol{\beta}+\boldsymbol{\delta}) = l(\boldsymbol{\beta}) + \boldsymbol{\delta}'\mathbf{g} + \frac{1}{2}\boldsymbol{\delta}'\mathbf{V}\boldsymbol{\delta}$$

where $\mathbf{g} = \mathbf{g}(\boldsymbol{\beta})$ is the gradient vector and $\mathbf{V} = \mathbf{V}(\boldsymbol{\beta})$ is the Hessian matrix. The increment $\boldsymbol{\delta}$ for the next iteration is obtained by solving the likelihood equations

$$\frac{d}{d\boldsymbol{\delta}}\{\tilde{l}(\boldsymbol{\beta}+\boldsymbol{\delta}) + \lambda(\mathbf{e}_j'\boldsymbol{\delta} - \gamma)\} = \mathbf{0}$$

where λ is the Lagrange multiplier, $\mathbf{e}_j$ is the jth unit vector, and γ is an unknown constant. The solution is

$$\boldsymbol{\delta} = -\mathbf{V}^{-1}(\mathbf{g} + \lambda\mathbf{e}_j)$$

By substituting this $\boldsymbol{\delta}$ into the equation $\tilde{l}(\boldsymbol{\beta}+\boldsymbol{\delta}) = l_0$, you can estimate λ as

$$\lambda = \pm\left(\frac{2(l_0 - l(\boldsymbol{\beta}) + \frac{1}{2}\mathbf{g}'\mathbf{V}^{-1}\mathbf{g})}{\mathbf{e}_j'\mathbf{V}^{-1}\mathbf{e}_j}\right)^{\frac{1}{2}}$$

The upper confidence limit for β_j is computed by starting at the maximum likelihood estimate of $\boldsymbol{\beta}$ and iterating with positive values of λ until convergence is attained. The process is repeated for the lower confidence limit using negative values of λ.

Convergence is controlled by value ϵ specified with the PLCONV= option in the MODEL statement (the default value of ϵ is 1E−4). Convergence is declared on the current iteration if the following two conditions are satisfied:

$$|l(\boldsymbol{\beta}) - l_0| \le \epsilon$$

and

$$(\mathbf{g} + \lambda\mathbf{e}_j)'\mathbf{V}^{-1}(\mathbf{g} + \lambda\mathbf{e}_j) \le \epsilon$$

Wald Confidence Intervals

Wald confidence intervals are sometimes called the normal confidence intervals. They are based on the asymptotic normality of the parameter estimators. The $100(1-\alpha)\%$ Wald confidence interval for β_j is given by

$$\widehat{\beta}_j \pm z_{1-\alpha/2}\widehat{\sigma}_j$$

where z_p is the $100p$th percentile of the standard normal distribution, $\widehat{\beta}_j$ is the maximum likelihood estimate of β_j, and $\widehat{\sigma}_j$ is the standard error estimate of $\widehat{\beta}_j$.

Odds Ratio Estimation

Consider a dichotomous response variable with outcomes *event* and *nonevent*. Consider a dichotomous risk factor variable X that takes the value 1 if the risk factor is present and 0 if the risk factor is absent. According to the logistic model, the log odds function, $g(X)$, is given by

$$g(X) \equiv \log\left(\frac{\Pr(\textit{event} \mid X)}{\Pr(\textit{nonevent} \mid X)}\right) = \beta_0 + \beta_1 X$$

The odds ratio ψ is defined as the ratio of the odds for those with the risk factor ($X = 1$) to the odds for those without the risk factor ($X = 0$). The log of the odds ratio is given by

$$\log(\psi) \equiv \log(\psi(X=1, X=0)) = g(X=1) - g(X=0) = \beta_1$$

The parameter, β_1, associated with X represents the change in the log odds from $X = 0$ to $X = 1$. So, the odds ratio is obtained by simply exponentiating the value of the parameter associated with the risk factor. The odds ratio indicates how the odds of *event* change as you change X from 0 to 1. For instance, $\psi = 2$ means that the odds of an event when $X = 1$ are twice the odds of an event when $X = 0$.

Suppose the values of the dichotomous risk factor are coded as constants a and b instead of 0 and 1. The odds when $X = a$ become $\exp(\beta_0 + a\beta_1)$, and the odds when $X = b$ become $\exp(\beta_0 + b\beta_1)$. The odds ratio corresponding to an increase in X from a to b is

$$\psi = \exp[(b-a)\beta_1] = [\exp(\beta_1)]^{b-a} \equiv [\exp(\beta_1)]^c$$

Note that for any a and b such that $c = b - a = 1, \psi = \exp(\beta_1)$. So the odds ratio can be interpreted as the change in the odds for any increase of one unit in the corresponding risk factor. However, the change in odds for some amount other than one unit is often of greater interest. For example, a change of one pound in body weight may be too small to be considered important, while a change of 10 pounds may be more meaningful. The odds ratio for a change in X from a to b is estimated by raising the odds ratio estimate for a unit change in X to the power of $c = b - a$ as shown previously.

For a polytomous risk factor, the computation of odds ratios depends on how the risk factor is parameterized. For illustration, suppose that Race is a risk factor with four categories: White, Black, Hispanic, and Other.

For the effect parameterization scheme (PARAM=EFFECT) with White as the reference group, the design variables for Race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	-1	-1	-1

The log odds for Black is

$$\begin{aligned} g(\text{Black}) &= \beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) \\ &= \beta_0 + \beta_1 \end{aligned}$$

The log odds for White is

$$\begin{aligned} g(\text{White}) &= \beta_0 + \beta_1(X_1 = -1) + \beta_2(X_2 = -1) + \beta_3(X_3 = -1)) \\ &= \beta_0 - \beta_1 - \beta_2 - \beta_3 \end{aligned}$$

Therefore, the log odds ratio of Black versus White becomes

$$\begin{aligned} \log(\psi(\text{Black}, \text{White})) &= g(\text{Black}) - g(\text{White}) \\ &= 2\beta_1 + \beta_2 + \beta_3 \end{aligned}$$

For the reference cell parameterization scheme (PARAM=REF) with White as the reference cell, the design variables for race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	0	0	0

The log odds ratio of Black versus White is given by

$$\begin{aligned} &\log(\psi(\text{Black}, \text{White})) \\ &\quad = \quad g(\text{Black}) - g(\text{White}) \\ &\quad = \quad (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0)) + \beta_3(X_3 = 0)) - \\ &\qquad\quad\; (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0)) \\ &\quad = \quad \beta_1 \end{aligned}$$

For the GLM parameterization scheme (PARAM=GLM), the design variables are as follows.

	Design Variables			
Race	X_1	X_2	X_3	X_4
Black	1	0	0	0
Hispanic	0	1	0	0
Other	0	0	1	0
White	0	0	0	1

The log odds ratio of Black versus White is

$$\begin{aligned}
&\log(\psi(\text{Black}, \text{White})) \\
&\quad = \; g(\text{Black}) - g(\text{White}) \\
&\quad = \; (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 0)) - \\
&\qquad\;\; (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 1)) \\
&\quad = \; \beta_1 - \beta_4
\end{aligned}$$

Consider the hypothetical example of heart disease among race in Hosmer and Lemeshow (2000, p 56). The entries in the following contingency table represent counts.

	Race			
Disease Status	White	Black	Hispanic	Other
Present	5	20	15	10
Absent	20	10	10	10

The computation of odds ratio of Black versus White for various parameterization schemes is tabulated in the following table.

Odds Ratio of Heart Disease Comparing Black to White

	Parameter Estimates				
PARAM	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	Odds Ratio Estimation
EFFECT	0.7651	0.4774	0.0719		$\exp(2 \times 0.7651 + 0.4774 + 0.0719) = 8$
REF	2.0794	1.7917	1.3863		$\exp(2.0794) = 8$
GLM	2.0794	1.7917	1.3863	0.0000	$\exp(2.0794) = 8$

Since the log odds ratio ($\log(\psi)$) is a linear function of the parameters, the Wald confidence interval for $\log(\psi)$ can be derived from the parameter estimates and the estimated covariance matrix. Confidence intervals for the odds ratios are obtained by exponentiating the corresponding confidence intervals for the log odd ratios. In the displayed output of PROC LOGISTIC, the “Odds Ratio Estimates” table contains the odds ratio estimates and the corresponding 95% Wald confidence intervals. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.

To customize odds ratios for specific units of change for a continuous risk factor, you can use the UNITS statement to specify a list of relevant units for each explanatory variable in the model. Estimates of these customized odds ratios are given in a separate table. Let (L_j, U_j) be a confidence interval for $\log(\psi)$. The corresponding lower and upper confidence limits for the customized odds ratio $\exp(c\beta_j)$ are $\exp(cL_j)$ and $\exp(cU_j)$, respectively (for $c > 0$), or $\exp(cU_j)$ and $\exp(cL_j)$, respectively (for

$c < 0$). You use the CLODDS= option to request the confidence intervals for the odds ratios.

For a generalized logit model, odds ratios are computed similarly, except k odds ratios are computed for each effect, corresponding to the k logits in the model.

Rank Correlation of Observed Responses and Predicted Probabilities

The predicted mean score of an observation is the sum of the Ordered Values (shown in the Response Profile table) minus one, weighted by the corresponding predicted probabilities for that observation; that is, the predicted means score= $\sum_{i=1}^{k+1}(i-1)\hat{\pi}_i$, where $k+1$ is the number of response levels and $\hat{\pi}_i$ is the predicted probability of the ith (ordered) response.

A pair of observations with different observed responses is said to be *concordant* if the observation with the lower ordered response value has a lower predicted mean score than the observation with the higher ordered response value. If the observation with the lower ordered response value has a higher predicted mean score than the observation with the higher ordered response value, then the pair is *discordant*. If the pair is neither concordant nor discordant, it is a *tie*. Enumeration of the total numbers of concordant and discordant pairs is carried out by categorizing the predicted mean score into intervals of length $k/500$ and accumulating the corresponding frequencies of observations.

Let N be the sum of observation frequencies in the data. Suppose there is a total of t pairs with different responses, n_c of them are concordant, n_d of them are discordant, and $t - n_c - n_d$ of them are tied. PROC LOGISTIC computes the following four indices of rank correlation for assessing the predictive ability of a model:

$$\begin{aligned}
c &= (n_c + 0.5(t - n_c - n_d))/t \\
\text{Somers' } D &= (n_c - n_d)/t \\
\text{Goodman-Kruskal Gamma} &= (n_c - n_d)/(n_c + n_d) \\
\text{Kendall's Tau-}a &= (n_c - n_d)/(0.5N(N-1))
\end{aligned}$$

Note that c also gives an estimate of the area under the receiver operating characteristic (ROC) curve when the response is binary (Hanley and McNeil 1982).

For binary responses, the predicted mean score is equal to the predicted probability for Ordered Value 2. As such, the preceding definition of concordance is consistent with the definition used in previous releases for the binary response model.

Linear Predictor, Predicted Probability, and Confidence Limits

This section describes how predicted probabilities and confidence limits are calculated using the maximum likelihood estimates (MLEs) obtained from PROC LOGISTIC. For a specific example, see the "Getting Started" section on page 2284. Predicted probabilities and confidence limits can be output to a data set with the OUTPUT statement.

Cumulative Response Models

For a vector of explanatory variables $\mathbf{x}$, the linear predictor

$$\eta_i = g(\Pr(Y \le i \mid \mathbf{x})) = \alpha_i + \boldsymbol{\beta}'\mathbf{x} \quad 1 \le i \le k$$

is estimated by

$$\hat{\eta}_i = \hat{\alpha}_i + \hat{\boldsymbol{\beta}}'\mathbf{x}$$

where $\hat{\alpha}_i$ and $\hat{\boldsymbol{\beta}}$ are the MLEs of α_i and $\boldsymbol{\beta}$. The estimated standard error of η_i is $\hat{\sigma}(\hat{\eta}_i)$, which can be computed as the square root of the quadratic form $(1, \mathbf{x}')\hat{\mathbf{V}}_{\mathbf{b}}(1, \mathbf{x}')'$ where $\hat{\mathbf{V}}_{\mathbf{b}}$ is the estimated covariance matrix of the parameter estimates. The asymptotic $100(1-\alpha)\%$ confidence interval for η_i is given by

$$\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile point of a standard normal distribution.

The predicted value and the $100(1-\alpha)\%$ confidence limits for $\Pr(Y \le i \mid \mathbf{x})$ are obtained by back-transforming the corresponding measures for the linear predictor.

Link	**Predicted Probability**	**100(1-α) Confidence Limits**
LOGIT	$1/(1+e^{-\hat{\eta}_i})$	$1/(1+\mathrm{e}^{-\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)})$
PROBIT	$\Phi(\hat{\eta}_i)$	$\Phi(\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i))$
CLOGLOG	$1-\mathrm{e}^{-\mathrm{e}^{\hat{\eta}_i}}$	$1-\mathrm{e}^{-\mathrm{e}^{\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)}}$

Generalized Logit Model

For a vector of explanatory variables $\mathbf{x}$, let π_i denote the probability of obtaining the response value i:

$$\pi_i = \begin{cases} \pi_{k+1} e^{\alpha_i + \mathbf{x}'\boldsymbol{\beta}_i} & 1 \le i \le k \\ \dfrac{1}{1+\sum_{i=1}^{k} e^{\alpha_i + \mathbf{x}'\boldsymbol{\beta}_i}} & i = k+1 \end{cases}$$

By the *delta method*,

$$\sigma^2(\pi_i) = \left(\frac{\partial \pi_i}{\partial \boldsymbol{\theta}}\right)' \mathbf{V}(\boldsymbol{\theta}) \frac{\partial \pi_i}{\partial \boldsymbol{\theta}}$$

A 100(1$-\alpha$)% confidence level for π_i is given by

$$\hat{\pi}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\pi}_i)$$

where $\hat{\pi}_i$ is the estimated expected probability of response i, and $\hat{\sigma}(\hat{\pi}_i)$ is obtained by evaluating $\sigma(\pi_i)$ at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

Classification Table

For binary response data, the response is either an *event* or a *nonevent*. In PROC LOGISTIC, the response with Ordered Value 1 is regarded as the *event*, and the response with Ordered Value 2 is the *nonevent*. PROC LOGISTIC models the probability of the *event*. From the fitted model, a predicted *event* probability can be computed for each observation. The method to compute a reduced-bias estimate of the predicted probability is given in the "Predicted Probability of an Event for Classification" section on page 2352, which follows. If the predicted *event* probability exceeds some cutpoint value $z \in [0, 1]$, the observation is predicted to be an *event* observation; otherwise, it is predicted as a *nonevent*. A 2×2 frequency table can be obtained by cross-classifying the observed and predicted responses. The CTABLE option produces this table, and the PPROB= option selects one or more cutpoints. Each cutpoint generates a classification table. If the PEVENT= option is also specified, a classification table is produced for each combination of PEVENT= and PPROB= values.

The accuracy of the classification is measured by its *sensitivity* (the ability to predict an *event* correctly) and specificity (the ability to predict a *nonevent* correctly). *Sensitivity* is the proportion of *event* responses that were predicted to be *events*. *Specificity* is the proportion of *nonevent* responses that were predicted to be *nonevents*. PROC LOGISTIC also computes three other conditional probabilities: *false positive rate*, *false negative rate*, and *rate of correct classification*. The *false positive rate* is the proportion of predicted *event* responses that were observed as *nonevents*. The *false negative rate* is the proportion of predicted *nonevent* responses that were observed as *events*. Given prior probabilities specified with the PEVENT= option, these conditional probabilities can be computed as posterior probabilities using Bayes' theorem.

Predicted Probability of an Event for Classification

When you classify a set of binary data, if the same observations used to fit the model are also used to estimate the classification error, the resulting error-count estimate is biased. One way of reducing the bias is to remove the binary observation to be classified from the data, reestimate the parameters of the model, and then classify the observation based on the new parameter estimates. However, it would be costly to fit the model leaving out each observation one at a time. The LOGISTIC procedure provides a less expensive one-step approximation to the preceding parameter estimates. Let $\mathbf{b}$ be the MLE of the parameter vector $(\alpha, \boldsymbol{\beta}')'$ based on all observations. Let $\mathbf{b}_j$ denote the MLE computed without the jth observation. The one-step estimate of $\mathbf{b}_j$ is given by

$$\mathbf{b}_j^1 = \mathbf{b} - \frac{w_j(y_j - \hat{\pi}_j)}{1 - h_{jj}} \hat{\mathbf{V}}_{\mathbf{b}} \begin{pmatrix} 1 \\ \mathbf{x}_j \end{pmatrix}$$

where

y_j is 1 for an event response and 0 otherwise

w_j is the WEIGHT value

$\hat{\pi}_j$ is the predicted event probability based on $\mathbf{b}$

h_{jj} is the hat diagonal element (defined on page 2359) with $n_j = 1$ and $r_j = y_j$

$\hat{\mathbf{V}}_\mathbf{b}$ is the estimated covariance matrix of $\mathbf{b}$

False Positive and Negative Rates Using Bayes' Theorem

Suppose n_1 of n individuals experience an event, for example, a disease. Let this group be denoted by $\mathcal{C}_1$, and let the group of the remaining $n_2 = n - n_1$ individuals who do not have the disease be denoted by $\mathcal{C}_2$. The jth individual is classified as giving a positive response if the predicted probability of disease ($\hat{\pi}^*_j$) is large. The probability $\hat{\pi}^*_j$ is the reduced-bias estimate based on a one-step approximation given in the preceding section. For a given cutpoint z, the jth individual is predicted to give a positive response if $\hat{\pi}^*_j \geq z$.

Let B denote the event that a subject has the disease and $\bar{B}$ denote the event of not having the disease. Let A denote the event that the subject responds positively, and let $\bar{A}$ denote the event of responding negatively. Results of the classification are represented by two conditional probabilities, $\Pr(A|B)$ and $\Pr(A|\bar{B})$, where $\Pr(A|B)$ is the sensitivity, and $\Pr(A|\bar{B})$ is one minus the specificity.

These probabilities are given by

$$\Pr(A|B) = \frac{\sum_{j \in \mathcal{C}_1} I(\hat{\pi}^*_j \geq z)}{n_1}$$

$$\Pr(A|\bar{B}) = \frac{\sum_{j \in \mathcal{C}_2} I(\hat{\pi}^*_j \geq z)}{n_2}$$

where $I(\cdot)$ is the indicator function.

Bayes' theorem is used to compute the error rates of the classification. For a given prior probability $\Pr(B)$ of the disease, the false positive rate P_{F+} and the false negative rate P_{F-} are given by Fleiss (1981, pp. 4–5) as follows:

$$P_{F+} = \Pr(\bar{B}|A) = \frac{\Pr(A|\bar{B})[1 - \Pr(B)]}{\Pr(A|\bar{B}) + \Pr(B)[\Pr(A|B) - \Pr(A|\bar{B})]}$$

$$P_{F-} = \Pr(B|\bar{A}) = \frac{[1 - \Pr(A|B)]\Pr(B)}{1 - \Pr(A|\bar{B}) - \Pr(B)[\Pr(A|B) - \Pr(A|\bar{B})]}$$

The prior probability $\Pr(B)$ can be specified by the PEVENT= option. If the PEVENT= option is not specified, the sample proportion of diseased individuals is used; that is, $\Pr(B) = n_1/n$. In such a case, the false positive rate and the false negative rate reduce to

$$P_{F+} = \frac{\sum_{j \in \mathcal{C}_2} I(\hat{\pi}^*_j \geq z)}{\sum_{j \in \mathcal{C}_1} I(\hat{\pi}^*_j \geq z) + \sum_{j \in \mathcal{C}_2} I(\hat{\pi}^*_j \geq z)}$$

$$P_{F-} = \frac{\sum_{j \in \mathcal{C}_1} I(\hat{\pi}^*_j < z)}{\sum_{j \in \mathcal{C}_1} I(\hat{\pi}^*_j < z) + \sum_{j \in \mathcal{C}_2} I(\hat{\pi}^*_j < z)}$$

Note that for a stratified sampling situation in which n_1 and n_2 are chosen a priori, n_1/n is not a desirable estimate of $\Pr(B)$. For such situations, the PEVENT= option should be specified.

Overdispersion

For a correctly specified model, the Pearson chi-square statistic and the deviance, divided by their degrees of freedom, should be approximately equal to one. When their values are much larger than one, the assumption of binomial variability may not be valid and the data are said to exhibit overdispersion. Underdispersion, which results in the ratios being less than one, occurs less often in practice.

When fitting a model, there are several problems that can cause the goodness-of-fit statistics to exceed their degrees of freedom. Among these are such problems as outliers in the data, using the wrong link function, omitting important terms from the model, and needing to transform some predictors. These problems should be eliminated before proceeding to use the following methods to correct for overdispersion.

Rescaling the Covariance Matrix

One way of correcting overdispersion is to multiply the covariance matrix by a dispersion parameter. This method assumes that the sample sizes in each subpopulation are approximately equal. You can supply the value of the dispersion parameter directly, or you can estimate the dispersion parameter based on either the Pearson chi-square statistic or the deviance for the fitted model.

The Pearson chi-square statistic χ_P^2 and the deviance χ_D^2 are given by

$$\begin{aligned} \chi_P^2 &= \sum_{i=1}^{m} \sum_{j=1}^{k+1} \frac{(r_{ij} - n_i\hat{\pi}_{ij})^2}{n_i\hat{\pi}_{ij}} \\ \chi_D^2 &= 2\sum_{i=1}^{m} \sum_{j=1}^{k+1} r_{ij} \log\left(\frac{r_{ij}}{n_i\hat{\pi}_{ij}}\right) \end{aligned}$$

where m is the number of subpopulation profiles, $k+1$ is the number of response levels, r_{ij} is the total weight (sum of the product of the frequencies and the weights) associated with jth level responses in the ith profile, $n_i = \sum_{j=1}^{k+1} r_{ij}$, and $\hat{\pi}_{ij}$ is the fitted probability for the jth level at the ith profile. Each of these chi-square statistics has $mk - p$ degrees of freedom, where p is the number of parameters estimated. The dispersion parameter is estimated by

$$\widehat{\sigma}^2 = \begin{cases} \chi_P^2/(mk-p) & \text{SCALE=PEARSON} \\ \chi_D^2/(mk-p) & \text{SCALE=DEVIANCE} \\ (\textit{constant})^2 & \text{SCALE=}\textit{constant} \end{cases}$$

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the subpopulations. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulation profiles. If you do not specify this option, each observation is regarded as coming from a separate subpopulation. For *events/trials* syntax, each observation represents n Bernoulli trials, where n is the value of the *trials* variable; for *single-trial* syntax, each observation represents a single trial. Without the AGGREGATE (or AGGREGATE=) option, the Pearson chi-square statistic and the deviance are calculated only for *events/trials* syntax.

Note that the parameter estimates are not changed by this method. However, their standard errors are adjusted for overdispersion, affecting their significance tests.

Williams' Method

Suppose that the data consist of n binomial observations. For the ith observation, let r_i/n_i be the observed proportion and let $\mathbf{x}_i$ be the associated vector of explanatory variables. Suppose that the response probability for the ith observation is a random variable P_i with mean and variance

$$E(P_i) = \pi_i \quad \text{and} \quad V(P_i) = \phi\pi_i(1-\pi_i)$$

where p_i is the probability of the event, and ϕ is a nonnegative but otherwise unknown scale parameter. Then the mean and variance of r_i are

$$E(r_i) = n_i\pi_i \quad \text{and} \quad V(r_i) = n_i\pi_i(1-\pi_i)[1+(n_i-1)\phi]$$

Williams (1982) estimates the unknown parameter ϕ by equating the value of Pearson's chi-square statistic for the full model to its approximate expected value. Suppose w_i^* is the weight associated with the ith observation. The Pearson chi-square statistic is given by

$$\chi^2 = \sum_{i=1}^{n} \frac{w_i^*(r_i - n_i\hat{\pi}_i)^2}{n_i\hat{\pi}_i(1-\hat{\pi}_i)}$$

Let $g'(\cdot)$ be the first derivative of the link function $g(\cdot)$. The approximate expected value of χ^2 is

$$E_{\chi^2} = \sum_{i=1}^{n} w_i^*(1 - w_i^* v_i d_i)[1+\phi(n_i-1)]$$

where $v_i = n_i/(\pi_i(1-\pi_i)[g'(\pi_i)]^2)$ and d_i is the variance of the linear predictor $\hat{\alpha}_i + \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$. The scale parameter ϕ is estimated by the following iterative procedure.

At the start, let $w_i^* = 1$ and let π_i be approximated by r_i/n_i, $i = 1, 2, \ldots, n$. If you apply these weights and approximated probabilities to χ^2 and E_{χ^2} and then equate them, an initial estimate of ϕ is therefore

$$\hat{\phi}_0 = \frac{\chi^2 - (n-p)}{\sum_i (n_i - 1)(1 - v_i d_i)}$$

where p is the total number of parameters. The initial estimates of the weights become $\hat{w}_{i0}^* = [1 + (n_i - 1)\hat{\phi}_0]^{-1}$. After a weighted fit of the model, $\widehat{\boldsymbol{\beta}}$ is recalculated, and so is χ^2. Then a revised estimate of ϕ is given by

$$\hat{\phi}_1 = \frac{\chi^2 - \sum_i w_i^*(1 - w_i^* v_i d_i)}{w_i^*(n_i - 1)(1 - w_i^* v_i d_i)}$$

The iterative procedure is repeated until χ^2 is very close to its degrees of freedom.

Once ϕ has been estimated by $\hat{\phi}$ under the full model, weights of $(1 + (n_i - 1)\hat{\phi})^{-1}$ can be used in fitting models that have fewer terms than the full model. See Example 42.9 on page 2438 for an illustration.

The Hosmer-Lemeshow Goodness-of-Fit Test

Sufficient replication within subpopulations is required to make the Pearson and deviance goodness-of-fit tests valid. When there are one or more continuous predictors in the model, the data are often too sparse to use these statistics. Hosmer and Lemeshow (2000) proposed a statistic that they show, through simulation, is distributed as chi-square when there is no replication in any of the subpopulations. This test is only available for binary response models.

First, the observations are sorted in increasing order of their estimated event probability. The event is the response level specified in the response variable option EVENT=, or the response level which is not specified in the REF= option, or, if neither of these options were specified, then the event is the response level identified in the “Response Profiles” table as “Ordered Value 1”. The observations are then divided into approximately ten groups according to the following scheme. Let N be the total number of subjects. Let M be the target number of subjects for each group given by

$$M = [0.1 \times N + 0.5]$$

where $[x]$ represents the integral value of x. If the *single-trial* syntax is used, blocks of subjects are formed of observations with identical values of the explanatory variables. Blocks of subjects are not divided when being placed into groups.

Suppose there are n_1 subjects in the first block and n_2 subjects in the second block. The first block of subjects is placed in the first group. Subjects in the second block are added to the first group if

$$n_1 < M \quad \text{and} \quad n_1 + [0.5 \times n_2] \leq M$$

Otherwise, they are placed in the second group. In general, suppose subjects of the (j-1)th block have been placed in the kth group. Let c be the total number of subjects currently in the kth group. Subjects for the jth block (containing n_j subjects) are also placed in the kth group if

$$c < M \quad \text{and} \quad c + [0.5 \times n_j] \le M$$

Otherwise, the n_j subjects are put into the next group. In addition, if the number of subjects in the last group does not exceed $[0.05 \times N]$ (half the target group size), the last two groups are collapsed to form only one group.

Note that the number of groups, g, may be smaller than 10 if there are fewer than 10 patterns of explanatory variables. There must be at least three groups in order for the Hosmer-Lemeshow statistic to be computed.

The Hosmer-Lemeshow goodness-of-fit statistic is obtained by calculating the Pearson chi-square statistic from the $2 \times g$ table of observed and expected frequencies, where g is the number of groups. The statistic is written

$$\chi^2_{HL} = \sum_{i=1}^{g} \frac{(O_i - N_i\bar{\pi}_i)^2}{N_i\bar{\pi}_i(1-\bar{\pi}_i)}$$

where N_i is the total frequency of subjects in the ith group, O_i is the total frequency of event outcomes in the ith group, and $\bar{\pi}_i$ is the average estimated predicted probability of an event outcome for the ith group. The Hosmer-Lemeshow statistic is then compared to a chi-square distribution with $(g - n)$ degrees of freedom, where the value of n can be specified in the LACKFIT option in the MODEL statement. The default is $n = 2$. Large values of χ^2_{HL} (and small p-values) indicate a lack of fit of the model.

Receiver Operating Characteristic Curves

In a sample of n individuals, suppose n_1 individuals are observed to have a certain condition or event. Let this group be denoted by $\mathcal{C}_1$, and let the group of the remaining $n_2 = n - n_1$ individuals who do not have the condition be denoted by $\mathcal{C}_2$. Risk factors are identified for the sample, and a logistic regression model is fitted to the data. For the jth individual, an estimated probability $\hat{\pi}_j$ of the event of interest is calculated. Note that the $\hat{\pi}_j$ are computed as shown in the "Linear Predictor, Predicted Probability, and Confidence Limits" section on page 2350 and are not the cross validated estimates discussed in the "Classification Table" section on page 2352.

Suppose the n individuals undergo a test for predicting the event and the test is based on the estimated probability of the event. Higher values of this estimated probability are assumed to be associated with the event. A receiver operating characteristic (ROC) curve can be constructed by varying the cutpoint that determines which estimated event probabilities are considered to predict the event. For each cutpoint z, the following measures can be output to a data set using the OUTROC= option in the

MODEL statement or the OUTROC= option in the SCORE statement:

$$\begin{aligned}
\_\text{POS}\_(z) &= \sum_{i \in \mathcal{C}_1} I(\hat{\pi}_i \geq z) \\
\_\text{NEG}\_(z) &= \sum_{i \in \mathcal{C}_2} I(\hat{\pi}_i < z) \\
\_\text{FALPOS}\_(z) &= \sum_{i \in \mathcal{C}_2} I(\hat{\pi}_i \geq z) \\
\_\text{FALNEG}\_(z) &= \sum_{i \in \mathcal{C}_1} I(\hat{\pi}_i < z) \\
\_\text{SENSIT}\_(z) &= \frac{\_\text{POS}\_(z)}{n_1} \\
\_\text{1MSPEC}\_(z) &= \frac{\_\text{FALPOS}\_(z)}{n_2}
\end{aligned}$$

where $I(\cdot)$ is the indicator function.

Note that _POS_(z) is the number of correctly predicted event responses, _NEG_(z) is the number of correctly predicted nonevent responses, _FALPOS_(z) is the number of falsely predicted event responses, _FALNEG_(z) is the number of falsely predicted nonevent responses, _SENSIT_(z) is the sensitivity of the test, and _1MSPEC_(z) is one minus the specificity of the test.

A plot of the ROC curve can be constructed by using the PLOT or GPLOT procedure with the OUTROC= data set and plotting sensitivity (_SENSIT_) against 1-specificity (_1MSPEC_); see Example 42.7 on page 2429 for an illustration. The area under the ROC curve, as determined by the trapezoidal rule, is estimated by the statistic c in the “Association of Predicted Probabilities and Observed Responses” table.

Testing Linear Hypotheses about the Regression Coefficients

Linear hypotheses for $\boldsymbol{\theta}$ are expressed in matrix form as

$$H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\mathbf{c}$ is a vector of constants. The vector of regression coefficients $\boldsymbol{\theta}$ includes slope parameters as well as intercept parameters. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = (\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})'[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$$

where $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})$ is the estimated covariance matrix. Under H_0, χ^2_W has an asymptotic chi-square distribution with r degrees of freedom, where r is the rank of $\mathbf{L}$.

Regression Diagnostics

For binary response data, regression diagnostics developed by Pregibon (1981) can be requested by specifying the INFLUENCE option. For diagnostics available with conditional logistic regression, see the "Regression Diagnostic Details" section on page 2367.

This section uses the following notation:

r_j, n_j r_j is the number of event responses out of n_j trials for the jth observation. If *events/trials* syntax is used, r_j is the value of *events* and n_j is the value of *trials*. For *single-trial* syntax, $n_j = 1$, and $r_j = 1$ if the ordered response is 1, and $r_j = 0$ if the ordered response is 2.

w_j is the weight of the jth observation.

π_j is the probability of an event response for the jth observation given by $\pi_j = F(\alpha + \boldsymbol{\beta}'\mathbf{x}_j)$, where $F(\cdot)$ is the inverse link function defined on page 2335.

b is the maximum likelihood estimate (MLE) of $(\alpha, \boldsymbol{\beta}')'$.

$\hat{\mathbf{V}}_{\mathbf{b}}$ is the estimated covariance matrix of **b**.

$\hat{p}_j, \hat{q}_j$ $\hat{p}_j$ is the estimate of π_j evaluated at **b**, and $\hat{q}_j = 1 - \hat{p}_j$.

Pregibon suggests using the index plots of several diagnostic statistics to identify influential observations and to quantify the effects on various aspects of the maximum likelihood fit. In an index plot, the diagnostic statistic is plotted against the observation number. In general, the distributions of these diagnostic statistics are not known, so cutoff values cannot be given for determining when the values are large. However, the IPLOTS and INFLUENCE options provide displays of the diagnostic values allowing visual inspection and comparison of the values across observations. In these plots, if the model is correctly specified and fits all observations well, then no extreme points should appear.

The next five sections give formulas for these diagnostic statistics.

Hat Matrix Diagonal

The diagonal elements of the hat matrix are useful in detecting extreme points in the design space where they tend to have larger values. The jth diagonal element is

$$h_{jj} = \begin{cases} \widetilde{w}_j(1, \mathbf{x}_j')\hat{\mathbf{V}}_{\mathbf{b}}(1, \mathbf{x}_j')' & \text{Fisher-Scoring} \\ \widehat{w}_j(1, \mathbf{x}_j')\hat{\mathbf{V}}_{\mathbf{b}}(1, \mathbf{x}_j')' & \text{Newton-Raphson} \end{cases}$$

where

$$\begin{aligned} \widetilde{w}_j &= \frac{w_j n_j}{\hat{p}_j \hat{q}_j [g'(\hat{p}_j)]^2} \\ \widehat{w}_j &= \widetilde{w}_j + \frac{w_j(r_j - n_j\hat{p}_j)[\hat{p}_j\hat{q}_j g''(\hat{p}_j) + (\hat{q}_j - \hat{p}_j)g'(\hat{p}_j)]}{(\hat{p}_j\hat{q}_j)^2[g'(\hat{p}_j)]^3} \end{aligned}$$

and $g'(\cdot)$ and $g''(\cdot)$ are the first and second derivatives of the link function $g(\cdot)$, respectively.

For a binary response logit model, the hat matrix diagonal elements are

$$h_{jj} = w_j n_j \hat{p}_j \hat{q}_j (1, \mathbf{x}_j') \hat{\mathbf{V}}_\mathbf{b} \begin{pmatrix} 1 \\ \mathbf{x}_j \end{pmatrix}$$

If the estimated probability is extreme (less than 0.1 and greater than 0.9, approximately), then the hat diagonal may be greatly reduced in value. Consequently, when an observation has a very large or very small estimated probability, its hat diagonal value is not a good indicator of the observation's distance from the design space (Hosmer and Lemeshow 2000, p 171).

Pearson Residuals and Deviance Residuals

Pearson and Deviance residuals are useful in identifying observations that are not explained well by the model. Pearson residuals are components of the Pearson chi-square statistic and deviance residuals are components of the deviance. The Pearson residual for the jth observation is

$$\chi_j = \frac{\sqrt{w_j}(r_j - n_j\hat{p}_j)}{\sqrt{n_j \hat{p}_j \hat{q}_j}}$$

The Pearson chi-square statistic is the sum of squares of the Pearson residuals. The deviance residual for the jth observation is

$$d_j = \begin{cases} -\sqrt{-2w_j n_j \log(\hat{q}_j)} & \text{if } r_j = 0 \\ \pm\sqrt{2w_j[r_j \log(\frac{r_j}{n_j\hat{p}_j}) + (n_j - r_j)\log(\frac{n_j - r_j}{n_j \hat{q}_j})]} & \text{if } 0 < r_j < n_j \\ \sqrt{-2w_j n_j \log(\hat{p}_j)} & \text{if } r_j = n_j \end{cases}$$

where the plus (minus) in ± is used if r_j/n_j is greater (less) than $\hat{p}_j$. The deviance is the sum of squares of the deviance residuals.

DFBETAS

For each parameter estimate, the procedure calculates a DFBETAS diagnostic for each observation. The DFBETAS diagnostic for an observation is the standardized difference in the parameter estimate due to deleting the observation, and it can be used to assess the effect of an individual observation on each estimated parameter of the fitted model. Instead of re-estimating the parameter every time an observation is deleted, PROC LOGISTIC uses the one-step estimate. See the section "Predicted Probability of an Event for Classification" on page 2352. For the jth observation, the DFBETAS are given by

$$\text{DFBETAS}i_j = \mathbf{\Delta}_i \mathbf{b}_j^1 / \hat{\sigma}(b_i)$$

where $i = 0, 1, \ldots, s$, $\hat{\sigma}(b_i)$ is the standard error of the ith component of $\mathbf{b}$, and $\mathbf{\Delta}_i\mathbf{b}_j^1$ is the ith component of the one-step difference

$$\mathbf{\Delta b}_j^1 = \frac{w_j(r_j - n_j\hat{p}_j)}{1 - h_{jj}}\hat{\mathbf{V}}_{\mathbf{b}}\left(\begin{array}{c} 1 \\ \mathbf{x}_j \end{array}\right)$$

$\mathbf{\Delta b}_j^1$ is the approximate change $(\mathbf{b} - \mathbf{b}_j^1)$ in the vector of parameter estimates due to the omission of the jth observation. The DFBETAS are useful in detecting observations that are causing instability in the selected coefficients.

C and CBAR

C and CBAR are confidence interval displacement diagnostics that provide scalar measures of the influence of individual observations on $\mathbf{b}$. These diagnostics are based on the same idea as the Cook distance in linear regression theory, and by using the one-step estimate, C and CBAR for the jth observation are computed as

$$C_j = \chi_j^2 h_{jj}/(1 - h_{jj})^2$$

and

$$\overline{C}_j = \chi_j^2 h_{jj}/(1 - h_{jj})$$

respectively.

Typically, to use these statistics, you plot them against an index (as the IPLOT option does) and look for outliers.

DIFDEV and DIFCHISQ

DIFDEV and DIFCHISQ are diagnostics for detecting ill-fitted observations; in other words, observations that contribute heavily to the disagreement between the data and the predicted values of the fitted model. DIFDEV is the change in the deviance due to deleting an individual observation while DIFCHISQ is the change in the Pearson chi-square statistic for the same deletion. By using the one-step estimate, DIFDEV and DIFCHISQ for the jth observation are computed as

$$\text{DIFDEV} = d_j^2 + \overline{C}_j$$

and

$$\text{DIFCHISQ} = \overline{C}_j/h_{jj}$$

Scoring Data Sets

Scoring a data set, which is especially important for predictive modeling, means applying a previously fitted model to a new data set in order to compute the conditional, or *posterior*, probabilities of each response category given the values of the explanatory variables in each observation.

The SCORE statement enables you to score new data sets and output the scored values and, optionally, the corresponding confidence limits into a SAS data set. If the response variable is included in the new data set, then you can request fit statistics for the data, which is especially useful for test or validation data. If the response is binary, you can also create a SAS data set containing the *receiver operating characteristic* (ROC) curve. You can specify multiple SCORE statements in the same invocation of PROC LOGISTIC.

By default, the posterior probabilities are based on implicit prior probabilities that are proportional to the frequencies of the response categories in the *training data* (the data used to fit the model). Explicit prior probabilities should be specified when the sample proportions of the response categories in the training data differ substantially from the operational data to be scored. For example, to detect a rare category, it is common practice to use a training set in which the rare categories are over-represented; without prior probabilities that reflect the true incidence rate, the predicted posterior probabilities for the rare category will be too high. By specifying the correct priors, the posterior probabilities are adjusted appropriately.

The model fit to the DATA= data set in the PROC LOGISTIC statement is the default model used for the scoring. Alternatively, you can save a fit model on one run of PROC LOGISTIC and use it to score new data on a subsequent run. The OUTMODEL= option in the PROC LOGISTIC statement saves the model information in a SAS data set. Specifying this data set in the INMODEL= option of a new PROC LOGISTIC run will score the DATA= data set in the SCORE statement without refitting the model.

The rest of this section provides some computational details about the scoring.

Posterior Probabilities and Confidence Limits

Let F be the inverse link function. That is,

$$F(t) = \begin{cases} \frac{1}{1+\exp(-t)} & \text{logistic} \\ \Phi(t) & \text{normal} \\ 1-\exp(-\exp(t)) & \text{complementary log-log} \end{cases}$$

The first derivative of F is given by

$$F'(t) = \begin{cases} \frac{\exp(-t)}{(1+\exp(-t))^2} & \text{logistic} \\ \phi(t) & \text{normal} \\ \exp(t)\exp(-\exp(t)) & \text{complementary log-log} \end{cases}$$

Suppose there are $k+1$ response categories. Let Y be the response variable with levels $1, \ldots, k+1$. Let $\mathbf{x} = (x_0, x_1, \ldots, x_p)'$ be a $(p+1)$-vector of covariates, with $x_0 \equiv 1$. Let $\boldsymbol{\theta}$ be the vector of regression parameters.

Posterior probabilities are given by

$$\mathrm{P}_n(i) = \frac{\mathrm{P}_o(i)\frac{\widetilde{\mathrm{p}_n}(i)}{\mathrm{p}_o(i)}}{\sum_j \mathrm{P}_o(j)\frac{\widetilde{\mathrm{p}_n}(j)}{\mathrm{p}_o(j)}} \qquad i = 1, \ldots, k+1$$

where the old posterior probabilities (P_o) are the conditional probabilities of the response categories given $\mathbf{x}$, and the old priors (p_o) are the sample proportions of response categories of the training data. To simplify notation, absorb the old priors into the new priors; that is

$$\mathrm{p}_n(i) = \frac{\widetilde{\mathrm{p}_n}(i)}{\mathrm{p}_o(i)} \qquad i = 1, \ldots, k+1$$

The posterior probabilities are functions of $\boldsymbol{\theta}$ and their estimates are obtained by substituting $\boldsymbol{\theta}$ by its MLE $\widehat{\boldsymbol{\theta}}$. The variances of the estimated posterior probabilities are given by the *delta method* as follows:

$$Var(\widehat{\mathrm{P}_n}(i)) = \left[\frac{\partial \mathrm{P}_n(i)}{\partial \boldsymbol{\theta}}\right]' Var(\widehat{\boldsymbol{\theta}}) \left[\frac{\partial \mathrm{P}_n(i)}{\partial \boldsymbol{\theta}}\right]$$

where

$$\frac{\partial \mathrm{P}_n(i)}{\partial \boldsymbol{\theta}} = \frac{\frac{\partial \mathrm{P}_o(i)}{\partial \boldsymbol{\theta}}\mathrm{p}_n(i)}{\sum_j \mathrm{P}_o(j)\mathrm{p}_n(j)} - \frac{\mathrm{P}_o(i)\mathrm{p}_n(i)\sum_j \frac{\partial \mathrm{P}_o(j)}{\partial \boldsymbol{\theta}}\mathrm{p}_n(j)}{[\sum_j \mathrm{P}_o(j)\mathrm{p}_n(j)]^2}$$

A 100(1-α) percent confidence interval for $\mathrm{P}_n(i)$ is

$$\widehat{\mathrm{P}_n}(i) \pm z_{1-\alpha/2}\sqrt{\widehat{Var}(\widehat{\mathrm{P}_n}(i))}$$

where z_τ is the upper 100τ percentile of the standard normal distribution.

Cumulative Response Model

Let $\alpha_1, \ldots, \alpha_k$ be the intercept parameters and let $\boldsymbol{\beta}$ be the vector of slope parameters. Denote $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}')'$. Let

$$\eta_i = \eta_i(\boldsymbol{\theta}) = \alpha_i + \mathbf{x}'\boldsymbol{\beta}, \qquad i = 1, \ldots, k$$

Estimates of $\eta_1, \ldots, \eta_k$ are obtained by substituting the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$.

The predicted probabilities of the responses are

$$\widehat{\mathrm{P}}_o(i) = \widehat{\Pr}(Y=i) = \begin{cases} F(\hat\eta_1) & i=1 \\ F(\hat\eta_i) - F(\hat\eta_{i-1}) & i=2,\ldots,k \\ 1-F(\hat\eta_k) & i=k+1 \end{cases}$$

For $i=1,\ldots,k$, let $\delta_i(x)$ be a $(k+1)$ column vector with ith entry equal to 1, $k+1$th entry equal to x, and all other entries 0. The derivative of $\mathrm{P}_o(i)$ with respect to $\boldsymbol{\theta}$ are

$$\frac{\partial \mathrm{P}_o(i)}{\partial \boldsymbol{\theta}} = \begin{cases} F'(\alpha_1 + x'\beta)\delta_1(x) & i=1 \\ F'(\alpha_i + x'\beta)\delta_i(x) - F'(\alpha_{i-1} + x'\beta)\delta_{i-1}(x) & i=2,\ldots,k \\ -F'(\alpha_k + x'\beta)\delta_k(x) & i=k+1 \end{cases}$$

Generalized Logit Model

Consider the last response level (Y=k+1) as the reference. Let $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_k$ be the parameter vectors for the first k logits, respectively. Denote $\boldsymbol{\theta} = (\boldsymbol{\beta}_1',\ldots,\boldsymbol{\beta}_k')'$. Let $\eta = (\eta_1,\ldots,\eta_k)'$ with

$$\eta_i = \eta_i(\boldsymbol{\theta}) = \mathbf{x}'\boldsymbol{\beta}_i \quad i=1,\ldots,k$$

Estimates of $\eta_1,\ldots,\eta_k$ are obtained by substituting the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ for $\boldsymbol{\theta}$.

The predicted probabilities are

$$\begin{aligned} \widehat{\mathrm{P}}_o(k+1) \equiv \Pr(Y=k+1|\mathbf{x}) &= \frac{1}{1+\sum_{l=1}^k \exp(\hat\eta_l)} \\ \widehat{\mathrm{P}}_o(i) \equiv \Pr(Y=i|\mathbf{x}) &= \widehat{\mathrm{P}}_o(k+1)\exp(\eta_i), \quad i=1,\ldots,k \end{aligned}$$

The derivative of $\mathrm{P}_o(i)$ with respect to $\boldsymbol{\theta}$ are

$$\begin{aligned} \frac{\partial \mathrm{P}_o(i)}{\partial \boldsymbol{\theta}} &= \frac{\partial \eta}{\partial \boldsymbol{\theta}} \frac{\partial \mathrm{P}_o(i)}{\partial \eta} \\ &= (I_k \otimes \mathbf{x}) \left(\frac{\partial \mathrm{P}_o(i)}{\partial \eta_1}, \cdots, \frac{\partial \mathrm{P}_o(i)}{\partial \eta_k} \right)' \end{aligned}$$

where

$$\frac{\partial \mathrm{P}_o(i)}{\partial \eta_j} = \begin{cases} \mathrm{P}_o(i)(1-\mathrm{P}_o(i)) & j=i \\ -\mathrm{P}_o(i)\mathrm{P}_o(j) & \text{otherwise} \end{cases}$$

Special Case of Binary Response Model with No Priors

Let $\boldsymbol{\beta}$ be the vector of regression parameters. Let

$$\eta = \eta(\boldsymbol{\beta}) = \mathbf{x}'\boldsymbol{\beta}$$

The variance of $\hat{\eta}$ is given by

$$Var(\hat{\eta}) = \mathbf{x}'Var(\hat{\boldsymbol{\beta}})\mathbf{x}$$

A 100(1-α) percent confidence interval for η is

$$\hat{\eta} \pm z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\eta})}$$

Estimates of $\mathrm{P}_o(1)$ and confidence intervals for the $\mathrm{P}_o(1)$ are obtained by back-transforming $\hat{\eta}$ and the confidence intervals for η, respectively. That is,

$$\widehat{\mathrm{P}_o}(1) = F(\hat{\eta})$$

and the confidence intervals are

$$F\left(\hat{\eta} \pm z_{1-\alpha/2}\sqrt{\widehat{Var}(\hat{\eta})}\right)$$

Conditional Logistic Regression

The method of maximum likelihood described in the preceding sections relies on large-sample asymptotic normality for the validity of estimates and especially of their standard errors. When you do not have a large sample size compared to the number of parameters, this approach may be inappropriate resulting in biased inferences. This situation typically arises when your data are stratified and you fit intercepts to each stratum so that the number of parameters is of the same order as the sample size. For example, in a 1:1 matched pairs study with n pairs and p covariates, you would estimate $n-1$ intercept parameters and p slope parameters. Taking the stratification into account by "conditioning out" (and not estimating) the stratum-specific intercepts gives consistent and asymptotically normal MLEs for the slope coefficients. See Breslow and Day (1980) and Stokes, Davis, and Koch (2000) for more information. If your nuisance parameters are not just stratum-specific intercepts, you can perform an exact conditional logistic regression.

Computational Details

For each stratum h, $h = 1,\ldots,H$, number the observations as $i = 1,\ldots,n_h$ so that hi indexes the ith observation in the hth stratum. Denote the p covariates for observation hi as $\mathbf{x}_{hi}$ and its binary response as y_{hi}, let $\mathbf{y} = (y_{11},\ldots,y_{1n_1},\ldots,y_{H1},\ldots,y_{Hn_H})'$, $\mathbf{X}_h = (\mathbf{x}_{h1}\ldots\mathbf{x}_{hn_h})'$, and $\mathbf{X} = (\mathbf{X}'_1\ldots\mathbf{X}'_H)'$. Let the dummy variables $z_h, h = 1,\ldots,H$, be indicator functions for the strata ($z_h = 1$ if the observation is in stratum h), denote $\mathbf{z}_{hi} = (z_1, \ldots, z_H)$ for observation hi, $\mathbf{Z}_h = (\mathbf{z}_{h1} \ldots \mathbf{z}_{hn_h})'$, and $\mathbf{Z} = (\mathbf{Z}'_1 \ldots \mathbf{Z}'_H)'$. Denote $\mathbf{X}^* = (\mathbf{Z}||\mathbf{X})$ and $\mathbf{x}^*_{hi} = (\mathbf{z}'_{hi}||\mathbf{x}'_{hi})'$. Arrange the observations in each stratum h so that $y_{hi} = 1$ for $i = 1, \ldots, m_h$, and $y_{hi} = 0$ for $i = m_{h+1}, \ldots, n_h$. Suppose all observations have unit frequency.

Consider the binary logistic regression model on page 2405 written as

$$\text{logit}(\boldsymbol{\pi}) = \mathbf{X}^* \boldsymbol{\theta}$$

where the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\alpha}', \boldsymbol{\beta}')'$ consists of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_H)'$, α_h is the intercept for stratum $h, h = 1, \ldots, H$, and $\boldsymbol{\beta}$ is the parameter vector for the p covariates.

From the “Determining Observations for Likelihood Contributions” section on page 2336, you can write the likelihood contribution of observation $hi, i = 1, \ldots, n_h, h = 1, \ldots, H$, as

$$L_{hi}(\boldsymbol{\theta}) = \frac{e^{y_{hi}\mathbf{x}^{*\prime}_{hi}\boldsymbol{\theta}}}{1 + e^{\mathbf{x}^{*\prime}_{hi}\boldsymbol{\theta}}}$$

where $y_{hi} = 1$ when the response takes Ordered Value 1, and $y_{hi} = 0$ otherwise.

The full likelihood is

$$L(\boldsymbol{\theta}) = \prod_{h=1}^{H} \prod_{i=1}^{n_h} L_{hi}(\boldsymbol{\theta}) = \frac{e^{\mathbf{y}'\mathbf{X}^*\boldsymbol{\theta}}}{\prod_{h=1}^{H} \prod_{i=1}^{n_h} \left(1 + e^{\mathbf{x}^{*\prime}_{hi}\boldsymbol{\theta}}\right)}$$

Unconditional likelihood inference is based on maximizing this likelihood function.

When your nuisance parameters are the stratum-specific intercepts $(\alpha_1, \ldots, \alpha_H)'$, and $\boldsymbol{\beta}$ are your parameters of interest, “conditioning out” the nuisance parameters produces the following conditional likelihood (Lachin 2000)

$$L(\boldsymbol{\beta}) = \prod_{h=1}^{H} L_h(\boldsymbol{\beta}) = \prod_{h=1}^{H} \frac{\prod_{i=1}^{m_h} \exp(\mathbf{x}'_{hi}\boldsymbol{\beta})}{\sum \prod_{j=j_1}^{j_{m_h}} \exp(\mathbf{x}'_{hj}\boldsymbol{\beta})}$$

where the summation is over all $\binom{n_h}{m_h}$ subsets $\{j_1, \ldots, j_{m_h}\}$ of m_h observations chosen from the n_h observations in stratum h. Note that the nuisance parameters have been factored out of this equation.

For conditional asymptotic inference, maximum likelihood estimates $\widehat{\boldsymbol{\beta}}$ of the regression parameters are obtained by maximizing the conditional likelihood, and asymptotic results are applied to the conditional likelihood function and the maximum likelihood estimators. A relatively fast method for computing this conditional likelihood and its derivatives is given by Gail, Lubin, and Rubinstein (1981) and Howard (1972). The default optimization techniques, which are the same as those implemented by the NLP procedure in SAS/OR software, are

- Newton-Raphson with ridging when the number of parameters $p < 40$
- quasi-Newton when $40 \leq p < 400$
- conjugate gradient when $p \geq 400$

Sometimes the log likelihood converges but the estimates diverge. This condition is flagged by having inordinately large standard errors for some of your parameter estimates, and can be monitored by specifying the ITPRINT option. Unfortunately, broad existence criteria such as those discussed in the "Existence of Maximum Likelihood Estimates" section on page 2338 do not exist for this model. It may be possible to circumvent such a problem by standardizing your independent variables before fitting the model.

Regression Diagnostic Details

Diagnostics are used to indicate observations that may have undue influence on the model fit, or which may be outliers. Further investigation should be performed before removing such an observation from the data set.

The derivations in this section follow Storer and Crowley's (1985) method of augmenting the logistic regression model, which provides an estimate of the "one-step" DFBETAS estimates advocated by Pregibon (1984). The method also provides estimates of conditional stratum-specific predicted values, residuals, and leverage for each observation.

Following Storer and Crowley (1985), the log-likelihood contribution can be written as

$$\begin{aligned} l_h &= \log(L_h) = \mathbf{y}'_h \boldsymbol{\gamma}_h - a(\boldsymbol{\gamma}_h) \quad \text{where} \\ a(\boldsymbol{\gamma}_h) &= \log\left[\sum \prod_{j=j_1}^{j_{m_h}} \exp(\boldsymbol{\gamma}_{hj})\right] \end{aligned}$$

and the h subscript on matrices indicates the submatrix for the stratum, $\boldsymbol{\gamma}'_h = (\gamma_{h1}, \ldots, \gamma_{hn_h})$, and $\gamma_{hi} = \mathbf{x}'_{hi}\boldsymbol{\beta}$. Then the gradient and information matrix are

$$\begin{aligned} \mathbf{g}(\boldsymbol{\beta}) &= \left\{\frac{\partial l_h}{\partial \boldsymbol{\beta}}\right\}_{h=1}^{H} = \mathbf{X}'(\mathbf{y} - \boldsymbol{\pi}) \\ \boldsymbol{\Lambda}(\boldsymbol{\beta}) &= \left\{\frac{\partial^2 l_h}{\partial \boldsymbol{\beta}^2}\right\}_{h=1}^{H} = \mathbf{X}'\text{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_H)\mathbf{X} \end{aligned}$$

where

$$\begin{aligned} \pi_{hi} &= \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi}} = \frac{\sum_{j(i)} \prod_{j=j_1}^{j_{m_h}} \exp(\gamma_{hj})}{\sum \prod_{j=j_1}^{j_{m_h}} \exp(\gamma_{hj})} \\ \boldsymbol{\pi}_h &= (\pi_{h1}, \ldots, \pi_{hn_h}) \end{aligned}$$

$$\begin{aligned}
\mathbf{U}_h &= \frac{\partial^2 a(\boldsymbol{\gamma}_h)}{\partial \boldsymbol{\gamma}_h^2} = \left\{ \frac{\partial^2 \mathbf{a}(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi} \partial \gamma_{hj}} \right\} = \{a_{ij}\} \\
a_{ij} &= \frac{\sum_{k(i,j)} \prod_{k=k_1}^{k_{m_h}} \exp(\gamma_{hk})}{\sum \prod_{k=k_1}^{k_{m_h}} \exp(\gamma_{hk})} - \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi}} \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hj}} = \pi_{hij} - \pi_{hi}\pi_{hj}
\end{aligned}$$

where π_{hi} is the conditional stratum-specific probability that subject i in stratum h is a case, the summation on $j(i)$ is over all subsets from $\{1, \ldots, n_h\}$ of size m_h that contain the index i, and the summation on $k(i,j)$ is over all subsets from $\{1, \ldots, n_h\}$ of size m_h that contain the indices i and j.

To produce the true one-step estimate $\boldsymbol{\beta}_{hi}^1$, start at the MLE $\widehat{\boldsymbol{\beta}}$, delete the hith observation, and take one-step of the Newton-Raphson algorithm using the reduced data set. Note that if there is only one event or one nonevent in a stratum, deletion of that single observation is equivalent to deletion of the entire stratum. The augmentation method does not take this into account.

The augmented model is

$$\text{logit}(\Pr(y_{hi} = 1 | \mathbf{x}_{hi})) = \mathbf{x}_{hi}' \boldsymbol{\beta} + \mathbf{z}_{hi}' \gamma$$

where $\mathbf{z}_{hi} = (0, \ldots, 0, 1, 0, \ldots, 0)'$ has a 1 in the hith coordinate, and use $\boldsymbol{\beta}^0 = (\widehat{\boldsymbol{\beta}}, 0)'$ as the initial estimate for $(\boldsymbol{\beta}, \gamma)'$. The gradient and information matrix before the step are

$$\begin{aligned}
\mathbf{g}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{X}' \\ \mathbf{z}_{hi}' \end{bmatrix} (\mathbf{y} - \boldsymbol{\pi}) = \begin{bmatrix} \mathbf{0} \\ y_{hi} - \pi_{hi} \end{bmatrix} \\
\boldsymbol{\Lambda}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{X}' \\ \mathbf{z}_{hi}' \end{bmatrix} \mathbf{U} \, [\mathbf{X} \quad \mathbf{z}_{hi}] = \begin{bmatrix} \boldsymbol{\Lambda}(\boldsymbol{\beta}) & \mathbf{X}'\mathbf{U}\mathbf{z}_{hi} \\ \mathbf{z}_{hi}'\mathbf{U}\mathbf{X} & \mathbf{z}_{hi}'\mathbf{U}\mathbf{z}_{hi} \end{bmatrix}
\end{aligned}$$

Inserting the $\boldsymbol{\beta}^0$ and $(\mathbf{X}', \mathbf{z}_{hi}')'$ into the Gail, Lubin, and Rubinstein (1981) algorithm provides the appropriate estimates of $\mathbf{g}(\boldsymbol{\beta}^0)$ and $\boldsymbol{\Lambda}(\boldsymbol{\beta}^0)$. Indicate these estimates with $\widehat{\boldsymbol{\pi}} = \boldsymbol{\pi}(\widehat{\boldsymbol{\beta}})$, $\widehat{\mathbf{U}} = \mathbf{U}(\widehat{\boldsymbol{\beta}})$, $\widehat{\mathbf{g}}$, and $\widehat{\boldsymbol{\Lambda}}$.

DFBETA is computed from the information matrix as

$$\begin{aligned}
\Delta_{hi}\boldsymbol{\beta} &= \boldsymbol{\beta}^0 - \boldsymbol{\beta}_{hi}^1 \\
&= -\widehat{\boldsymbol{\Lambda}}^{-1}(\boldsymbol{\beta}^0)\widehat{\mathbf{g}}(\boldsymbol{\beta}^0) \\
&= -\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\mathbf{z}_{hi})\mathbf{M}^{-1}\mathbf{z}_{hi}'(\mathbf{y} - \widehat{\boldsymbol{\pi}}) \quad \text{where} \\
\mathbf{M} &= (\mathbf{z}_{hi}'\widehat{\mathbf{U}}\mathbf{z}_{hi}) - (\mathbf{z}_{hi}'\widehat{\mathbf{U}}\mathbf{X})\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\mathbf{z}_{hi})
\end{aligned}$$

For each observation in the dataset, a DFBETA statistic is computed for each parameter β_j, $1 \leq j \leq p$, and standardized by the standard error of β_j from the full data set to produce the estimate of DFBETAS.

The estimated residuals $e_{hi} = y_{hi} - \widehat{\pi}_{hi}$ are obtained from $\widehat{\mathbf{g}}(\boldsymbol{\beta}^0)$, and the weights, or predicted probabilities, are then $\widehat{\pi}_{hi} = y_{hi} - e_{hi}$. The residuals are standardized and reported as (estimated) Pearson residuals:

$$\frac{r_{hi} - n_{hi}\widehat{\pi}_{hi}}{\sqrt{n_{hi}\widehat{\pi}_{hi}(1 - \widehat{\pi}_{hi})}}$$

where r_{hi} is the number of events in the observation and n_{hi} is the number of trials.

The estimated leverage is defined as

$$\mathsf{h}_{hi} = \frac{\mathrm{trace}\{(\mathbf{z}'_{hi}\widehat{\mathbf{U}}\mathbf{X})\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\mathbf{z}_{hi})\}}{\mathrm{trace}\{\mathbf{z}'_{hi}\widehat{\mathbf{U}}\mathbf{z}_{hi}\}}$$

This definition of leverage produces different values from those defined by Pregibon (1984), Moolgavkar, Lustbader, and Venzon (1985), and Hosmer and Lemeshow (2000); however, it has the advantage that no extra computations beyond those for the DFBETAS are required.

For events/trials MODEL syntax, treat each observation as two observations (the first for the nonevents and the second for the events) with frequencies $f_{h,2i-1} = n_{hi} - r_{hi}$ and $f_{h,2i} = r_{hi}$, and augment the model with a matrix $\mathbf{Z}_{hi} = [\mathbf{z}_{h,2i-1}\mathbf{z}_{h,2i}]$ instead of a single $\mathbf{z}_{hi}$ vector. Writing $\gamma_{hi} = \mathbf{x}'_{hi}\boldsymbol{\beta} f_{hi}$ in the preceding section results in the following gradient and information matrix.

$$\begin{aligned}
\mathbf{g}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{0} \\ f_{h,2i-1}(y_{h,2i-1} - \pi_{h,2i-1}) \\ f_{h,2i}(y_{h,2i} - \pi_{h,2i}) \end{bmatrix} \\
\boldsymbol{\Lambda}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \boldsymbol{\Lambda}(\boldsymbol{\beta}) & \mathbf{X}'\mathrm{diag}(\mathbf{f})\mathbf{U}\mathrm{diag}(\mathbf{f})\mathbf{Z}_{hi} \\ \mathbf{Z}'_{hi}\mathrm{diag}(\mathbf{f})\mathbf{U}\mathrm{diag}(\mathbf{f})\mathbf{X} & \mathbf{Z}'_{hi}\mathrm{diag}(\mathbf{f})\mathbf{U}\mathrm{diag}(\mathbf{f})\mathbf{Z}_{hi} \end{bmatrix}
\end{aligned}$$

The predicted probabilities are then $\widehat{\pi}_{hi} = y_{h,2i} - e_{h,2i}/r_{h,2i}$, while the leverage and the DFBETAs are produced from $\boldsymbol{\Lambda}(\boldsymbol{\beta}^0)$ in a similar fashion as for the preceding single-trial equations.

Exact Conditional Logistic Regression

The theory of exact conditional logistic regression analysis was originally laid out by Cox (1970), and the computational methods employed in PROC LOGISTIC are described in Hirji, Mehta, and Patel (1987), Hirji (1992), and Mehta, Patel, and Senchaudhuri (1992). Other useful references for the derivations include Cox and Snell (1989), Agresti (1990), and Mehta and Patel (1995).

Exact conditional inference is based on generating the conditional distribution for the sufficient statistics of the parameters of interest. This distribution is called the *permutation* or *exact conditional* distribution. Using the notation in the “Computational

Details” section on page 2365, follow Mehta and Patel (1995) and first note that the sufficient statistics $\mathbf{T} = (T_1, \ldots, T_p)$ for $\boldsymbol{\theta}$ are

$$T_j = \sum_{i=1}^{n} y_i x_{ij}, \quad j = 1, \ldots, p$$

Denote a vector of observable sufficient statistics as $\mathbf{t} = (t_1, \ldots, t_p)'$.

The probability density function (pdf) for $\mathbf{T}$ can be created by summing over all binary sequences $\mathbf{y}$ that generate an observable $\mathbf{t}$ and letting $C(\mathbf{t}) = ||\{\mathbf{y} : \mathbf{y}'\mathbf{X} = \mathbf{t}'\}||$ denote the number of sequences $\mathbf{y}$ that generate $\mathbf{t}$

$$\Pr(\mathbf{T} = \mathbf{t}) = \frac{C(\mathbf{t}) \exp(\mathbf{t}'\boldsymbol{\theta})}{\prod_{i=1}^{n} [1 + \exp(\mathbf{x}_i'\boldsymbol{\theta})]}$$

In order to condition out the stratum parameters, partition the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_0', \boldsymbol{\theta}_1')'$, where $\boldsymbol{\theta}_0$ is a $p_0 \times 1$ vector of the nuisance parameters, and $\boldsymbol{\theta}_1$ is the parameter vector for the remaining $p_1 = p - p_0$ parameters of interest. Likewise, partition $\mathbf{X}$ into $\mathbf{X}_0$ and $\mathbf{X}_1$, $\mathbf{T}$ into $\mathbf{T}_0$ and $\mathbf{T}_1$, and $\mathbf{t}$ into $\mathbf{t}_0$ and $\mathbf{t}_1$. The nuisance parameters can be removed from the analysis by conditioning on their sufficient statistics to create the conditional likelihood of $\mathbf{T}_1$ given $\mathbf{T}_0 = \mathbf{t}_0$

$$\begin{aligned} \Pr(\mathbf{T}_1 = \mathbf{t}_1 | \mathbf{T}_0 = \mathbf{t}_0) &= \frac{\Pr(\mathbf{T} = \mathbf{t})}{\Pr(\mathbf{T}_0 = \mathbf{t}_0)} \\ = f_{\boldsymbol{\theta}_1}(\mathbf{t}_1 | \mathbf{t}_0) &= \frac{C(\mathbf{t}_0, \mathbf{t}_1) \exp(\mathbf{t}_1'\boldsymbol{\theta}_1)}{\sum_u C(\mathbf{t}_0, \mathbf{u}) \exp(\mathbf{u}'\boldsymbol{\theta}_1)} \end{aligned}$$

where $C(\mathbf{t}_0, \mathbf{u})$ is the number of vectors $\mathbf{y}$ such that $\mathbf{y}'\mathbf{X}_0 = \mathbf{t}_0$ and $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$. Note that the nuisance parameters have factored out of this equation, and that $C(\mathbf{t}_0, \mathbf{t}_1)$ is a constant.

The goal of the exact conditional analysis is to determine how likely the observed response $\mathbf{y}_0$ is with respect to all 2^n possible responses $\mathbf{y} = (y_1, \ldots, y_n)'$. One way to proceed is to generate every $\mathbf{y}$ vector for which $\mathbf{y}'\mathbf{X}_0 = \mathbf{t}_0$, and count the number of vectors $\mathbf{y}$ for which $\mathbf{y}'\mathbf{X}_1$ is equal to each unique $\mathbf{t}_1$. Generating the conditional distribution from complete enumeration of the joint distribution is conceptually simple; however, this method becomes computationally infeasible very quickly. For example, if you had only 30 observations, you’d have to scan through 2^{30} different $\mathbf{y}$ vectors.

Several algorithms are available in PROC LOGISTIC to generate the exact distribution. All of the algorithms are based on the following observation. Given any $\mathbf{y} = (y_1, \ldots, y_n)'$ and a design $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)'$, let $\mathbf{y}_{(i)} = (y_1, \ldots, y_i)'$ and $\mathbf{X}_{(i)} = (\mathbf{x}_1, \ldots, \mathbf{x}_i)'$ be the first i rows of each matrix. Write the sufficient statistic based on these i rows as $\mathbf{t}_{(i)}' = \mathbf{y}_{(i)}'\mathbf{X}_{(i)}$. A recursion relation results: $\mathbf{t}_{(i+1)} = \mathbf{t}_{(i)} + y_{i+1}\mathbf{x}_{i+1}$.

The following methods are available.

- The *multivariate shift algorithm* developed by Hirji, Mehta, and Patel (1987) steps through the recursion relation by adding one observation at a time and building an intermediate distribution at each step. If it determines that $\mathbf{t}_{(i)}$ for the nuisance parameters could eventually equal $\mathbf{t}$, then $\mathbf{t}_{(i)}$ is added to the intermediate distribution.
- Hirji (1992) extends the multivariate shift algorithm to generalized logit models. Since the generalized logit model fits a new set of parameters to each logit, the number of parameters in the model can easily get too large for this algorithm to handle. Note for these models that the hypothesis tests for each effect are computed across the logit functions, while individual parameters are estimated for each logit function.
- A network algorithm described in Mehta, Patel, and Senchaudhuri (1992) builds a network for each parameter that you are conditioning out in order to identify feasible y_i for the $\mathbf{y}$ vector. These networks are combined and the set of feasible y_i is further reduced, then the multivariate shift algorithm uses this knowledge to build the exact distribution without adding as many intermediate $\mathbf{t}_{(i+1)}$ as the multivariate shift algorithm does.
- Mehta, Patel, and Senchaudhuri (2000) devised a hybrid Monte-Carlo and network algorithm that extends their 1992 algorithm by sampling from the combined network to build the exact distribution.

The bulk of the computation time and memory for these algorithms is consumed by the creation of the networks and the exact joint distribution. After the joint distribution for a set of effects is created, the computational effort required to produce hypothesis tests and parameter estimates for any subset of the effects is (relatively) trivial.

Hypothesis Tests

Consider testing the null hypothesis $H_0\colon \boldsymbol{\beta}_1 = \mathbf{0}$ against the alternative $H_A\colon \boldsymbol{\beta}_1 \neq \mathbf{0}$, conditional on $\mathbf{T}_0 = \mathbf{t}_0$. Under the null hypothesis, the test statistic for the *exact probability test* is just $f_{\beta_1=0}(\mathbf{t}_1|\mathbf{t}_0)$, while the corresponding p-value is the probability of getting a less likely (more extreme) statistic,

$$p(\mathbf{t}_1|\mathbf{t}_0) = \sum_{u \in \Omega_p} f_0(\mathbf{u}|\mathbf{t}_0)$$

where $\Omega_p = \{\mathbf{u}$: there exist $\mathbf{y}$ with $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$, $\mathbf{y}'\mathbf{X}_0 = \mathbf{t}_0$, and $f_0(\mathbf{u}|\mathbf{t}_0) \leq f_0(\mathbf{t}_1|\mathbf{t}_0)\}$.

For the *exact conditional scores test,* the conditional mean $\boldsymbol{\mu}_1$ and variance matrix $\boldsymbol{\Sigma}_1$ of the $\mathbf{T}_1$ (conditional on $\mathbf{T}_0 = \mathbf{t}_0$) are calculated, and the score statistic for the observed value,

$$s = (\mathbf{t}_1 - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\mathbf{t}_1 - \boldsymbol{\mu}_1)$$

is compared to the score for each member of the distribution

$$S(\mathbf{T}_1) = (\mathbf{T}_1 - \boldsymbol{\mu}_1)'\boldsymbol{\Sigma}_1^{-1}(\mathbf{T}_1 - \boldsymbol{\mu}_1)$$

The resulting p-value is

$$p(\mathbf{t}_1|\mathbf{t}_0) = Pr(S \geq s) = \sum_{u \in \Omega_s} f_0(\mathbf{u}|\mathbf{t}_0)$$

where $\Omega_s = \{\mathbf{u}$: there exist $\mathbf{y}$ with $\mathbf{y}'\mathbf{X}_1 = \mathbf{u}$, $\mathbf{y}'\mathbf{X}_0 = \mathbf{t}_0$, and $S(\mathbf{u}) \geq s\}$.

The mid-p statistic, defined as

$$p(\mathbf{t}_1|\mathbf{t}_0) - \frac{1}{2} f_0(\mathbf{t}_1|\mathbf{t}_0)$$

was proposed by Lancaster (1961) to compensate for the discreteness of a distribution. Refer to Agresti (1992) for more information. However, to allow for more flexibility in handling ties, you can write the mid-p statistic as (based on a suggestion by LaMotte 2002 and generalizing Vollset, Hirji, and Afifi 1991)

$$\sum_{u \in \Omega_<} f_0(\mathbf{u}|\mathbf{t}_0) + \delta_1 f_0(\mathbf{t}_1|\mathbf{t}_0) + \delta_2 \sum_{u \in \Omega_=} f_0(\mathbf{u}|\mathbf{t}_0)$$

where, for $i \in \{p, s\}$, $\Omega_<$ is Ω_i using strict inequalities, and $\Omega_=$ is Ω_i using equalities with the added restriction that $\mathbf{u} \neq \mathbf{t}_1$. Letting $(\delta_1, \delta_2) = (0.5, 1.0)$ yields Lancaster's mid-p.

Caution: When the exact distribution has ties and METHOD=NETWORKMC is specified, the Monte Carlo algorithm estimates $p(\mathbf{t}|\mathbf{t}_0)$ with error, and hence it cannot determine precisely which values contribute to the reported p-values. For example, if the exact distribution has densities $\{0.2, 0.2, 0.2, 0.4\}$ and if the observed statistic has probability 0.2, then the exact probability p-value is exactly 0.6. Under Monte Carlo sampling, if the densities after N samples are $\{0.18, 0.21, 0.23, 0.38\}$ and the observed probability is 0.21, then the resulting p-value is 0.39. Therefore, the exact probability test p-value for this example fluctuates between 0.2, 0.4, and 0.6, and the reported p-values are actually lower bounds for the true p-values. If you need more precise values, you can specify the OUTDIST= option, determine appropriate cutoff values for the observed probability and score, then construct the true p-value estimates from the OUTDIST= data set using the following statements.

```
data _null_;
   set outdist end=end;
   retain pvalueProb 0 pvalueScore 0;
   if prob < ProbCutOff then pvalueProb+prob;
   if score > ScoreCutOff then pvalueScore+prob;
   if end then put pvalueProb pvalueScore;
run;
```

Inference for a Single Parameter

Exact parameter estimates are derived for a single parameter β_i by regarding all the other parameters $\boldsymbol{\beta}_0 = (\beta_1, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots, \beta_{p+q})'$ as nuisance parameters. The appropriate sufficient statistics are $\mathbf{T}_1 = T_i$ and $\mathbf{T}_0 = (T_1, \ldots, T_{i-1}, T_{i+1}, \ldots, T_{p+q})'$, with their observed values denoted by the lowercase t. Hence, the conditional pdf used to create the parameter estimate for β_i is

$$f_{\beta_i}(t_i|\mathbf{t}_0) = \frac{C(\mathbf{t}_0, t_i)\exp(t_i\beta_i)}{\sum_{u\in\Omega} C(\mathbf{t}_0, u)\exp(u\beta_i)}$$

for $\Omega = \{u\colon$ there exist $\mathbf{y}$ with $T_i = u$ and $\mathbf{T}_0 = \mathbf{t}_0\}$.

The maximum exact conditional likelihood estimate is the quantity $\widehat{\beta}_i$, which maximizes the conditional pdf. A Newton-Raphson algorithm is used to perform this search. However, if the observed t_i attains either its maximum or minimum value in the exact distribution (that is, either $t_i = \min\{u : u \in \Omega\}$ or $t_i = \max\{u : u \in \Omega\}$), then the conditional pdf is monotonically increasing in β_i and cannot be maximized. In this case, a median unbiased estimate (Hirji, Tsiatis, and Mehta 1989) $\widehat{\beta}_i$ is produced that satisfies $f_{\widehat{\beta}_i}(t_i|\mathbf{t}_0) = 0.5$, and a Newton-Raphson-type algorithm is used to perform the search.

Likelihood ratio tests based on the conditional pdf are used to test the null $H_0\colon \beta_i = 0$ against the alternative $H_A\colon \beta_i > 0$. The critical region for this UMP test consists of the upper tail of values for T_i in the exact distribution. Thus, the one-sided significance level $p_+(t_i; 0)$ is

$$p_+(t_i; 0) = \sum_{u \ge t_i} f_0(u|\mathbf{t}_0)$$

Similarly, the one-sided significance level $p_-(t_i; 0)$ against $H_A\colon \beta_i < 0$ is

$$p_-(t_i; 0) = \sum_{u \le t_i} f_0(u|\mathbf{t}_0)$$

The two-sided significance level $p(t_i; 0)$ against $H_A\colon \beta_i \ne 0$ is calculated as

$$p(t_i; 0) = 2\min[p_-(t_i; 0), p_+(t_i; 0)]$$

An upper $100(1-2\epsilon)$% exact confidence limit for $\widehat{\beta}_i$ corresponding to the observed t_i is the solution $\beta_U(t_i)$ of $\epsilon = p_-(t_i, \beta_U(t_i))$, while the lower exact confidence limit is the solution $\beta_L(t_i)$ of $\epsilon = p_+(t_i, \beta_L(t_i))$. Again, a Newton-Raphson procedure is used to search for the solutions.

Specifying the ONESIDED option displays only one p-value and one confidence interval, because small values of $p_+(t_i; 0)$ and $p_-(t_i; 0)$ support different alternative hypotheses and only one of these p-values can be less than 0.50.

The mid-p confidence limits are the solutions to $\min\{p_-(t_i, \beta(t_i)), p_+(t_i, \beta(t_i))\} - (1 - \delta_1) f_{\beta(t_i)}(u|\mathbf{t}_0) = \epsilon$ for $\epsilon = \alpha/2, 1 - \alpha/2$ (Vollset, Hirji, and Afifi 1991). $\delta_1 = 1$ produces the usual exact (or *max-p*) confidence interval, $\delta_1 = 0.5$ yields the mid-p interval, and $\delta_1 = 0$ gives the *min-p* interval. The mean of the endpoints of the max-p and min-p intervals provides the *mean-p* interval as defined by Hirji, Mehta, and Patel (1988).

Estimates and confidence intervals for the odds-ratios are produced by exponentiating the estimates and interval endpoints for the parameters.

OUTEST= Output Data Set

The OUTEST= data set contains one observation for each BY group containing the maximum likelihood estimates of the regression coefficients. If you also use the COVOUT option in the PROC LOGISTIC statement, there are additional observations containing the rows of the estimated covariance matrix. If you use the FORWARD, BACKWARD, or STEPWISE selection method, only the estimates of the parameters and covariance matrix for the final model are output to the OUTEST= data set.

Variables in the OUTEST= Data Set

The OUTEST= data set contains the following variables:

- any BY variables specified
- _LINK_, a character variable of length 8 with four possible values: CLOGLOG for the complementary log-log function, LOGIT for the logit function, NORMIT for the probit (alias normit) function, and GLOGIT for the generalized logit function.
- _TYPE_, a character variable of length 8 with two possible values: PARMS for parameter estimates or COV for covariance estimates. If an EXACT statement is also specified, then two other values are possible: EPARMMLE for the exact maximum likelihood estimates and EPARMMUE for the exact median unbiased estimates.
- _NAME_, a character variable containing the name of the response variable when _TYPE_=PARMS, EPARMMLE, and EPARMMUE, or the name of a model parameter when _TYPE_=COV
- _STATUS_, a character variable that indicates whether the estimates have converged
- one variable for each intercept parameter
- one variable for each slope parameter and one variable for the offset variable if the OFFSET= option if specified. If an effect is not included in the final model in a model building process, the corresponding parameter estimates and covariances are set to missing values.
- _LNLIKE_, the log likelihood

Parameter Names in the OUTEST= Data Set

If there are only two response categories in the entire data set, the intercept parameter is named Intercept. If there are more than two response categories in the entire data set, the intercept parameters are named Intercept_xxx, where xxx is the value (formatted if a format is applied) of the corresponding response category.

For continuous explanatory variables, the names of the parameters are the same as the corresponding variables. For class variables, the parameter names are obtained by concatenating the corresponding CLASS variable name with the CLASS category; see the PARAM= option in the CLASS statement and the "CLASS Variable Parameterization" section on page 2331 for more details. For interaction and nested effects, the parameter names are created by concatenating the names of each effect.

For the generalized logit model, names of parameters corresponding to each nonreference category contain _xxx as the suffix, where xxx is the value (formatted if a format is applied) of the corresponding nonreference category. For example, suppose the variable Net3 represents the television network (ABC, CBS, and NBC) viewed at a certain time. The following code fits a generalized logit model with Age and Gender (a CLASS variable with values Female and Male) as explanatory variables.

```
proc logistic;
   class Gender;
   model Net3 = Age Gender / link=glogit;
run;
```

There are two logit functions, one contrasting ABC with NBC and the other contrasting CBS with NBC. For each logit, there are three parameters: an intercept parameter, a slope parameter for Age, and a slope parameter for Gender (since there are only two gender levels and the EFFECT parameterization is used by default). The names of the parameters and their descriptions are as follows.

Intercept_ABC	intercept parameter for the logit contrasting ABC with NBC
Intercept_CBS	intercept parameter for the logit contrasting CBS with NBC
Age_ABC	Age slope parameter for the logit contrasting ABC with NBC
Age_CBS	Age slope parameter for the logit contrasting CBS with NBC
GenderFemale_ABC	Gender=Female slope parameter for the logit contrasting ABC with NBC
GenderFemale_CBS	Gender=Female slope parameter for the logit contrasting CBS with NBC

INEST= Input Data Set

You can specify starting values for the iterative algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set.

The INEST= data set must contain the intercept variables (named Intercept for binary response models and Intercept, Intercept2, Intercept3, and so forth, for ordinal and nominal response models) and all explanatory variables in the MODEL statement. If BY processing is used, the INEST= data set should also include the BY variables, and there must be one observation for each BY group. If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values.

OUT= Output Data Set in the OUTPUT Statement

The OUT= data set in the OUTPUT statement contains all the variables in the input data set along with statistics you request using *keyword=name* options or the PREDPROBS= option in the OUTPUT statement. In addition, if you use the *single-trial* syntax and you request any of the XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= options, the OUT= data set contains the automatic variable _LEVEL_. The value of _LEVEL_ identifies the response category upon which the computed values of XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= are based.

When there are more than two response levels, only variables named by the XBETA=, STDXBETA=, PREDICTED=, LOWER=, and UPPER= options and the variables given by PREDPROBS=(INDIVIDUAL CUMULATIVE) have their values computed; the other variables have missing values. If you fit a generalized logit model, the cumulative predicted probabilities are not computed.

When there are only two response categories, each input observation produces one observation in the OUT= data set.

If there are more than two response categories and you only specify the PREDPROBS= option, then each input observation produces one observation in the OUT= data set. However, if you fit an ordinal (cumulative) model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as one fewer than the number of response levels, and the predicted probabilities and their confidence limits correspond to the cumulative predicted probabilities. If you fit a generalized logit model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as the number of response categories; the predicted probabilities and their confidence limits correspond to the probabilities of individual response categories.

For observations in which only the response variable is missing, values of the XBETA=, STDXBETA=, PREDICTED=, UPPER=, LOWER=, and the PREDPROBS= options are computed even though these observations do not affect the model fit. This enables, for instance, predicted probabilities to be computed for new observations.

OUT= Output Data Set in a SCORE Statement

The OUT= data set in a SCORE statement contains all the variables in the data set being scored. The data set being scored can be either the input DATA= data set in the PROC LOGISTIC statement or the DATA= data set in the SCORE statement. The DATA= data set in the SCORE statement may not contain a response variable.

If the data set being scored contains a response variable, then denote the *normalized* levels (left justified formatted values of 16 characters or less) of your response variable Y by $Y_1, \ldots, Y_{k+1}$. For each response level, the OUT= data set also contains:

- F_Y, the normalized levels of the response variable Y in the data set being scored. If the *events*/*trials* syntax is used, the F_Y variable is not created.
- I_Y, the normalized levels that the observations are classified into. Note that an observation is classified into the level with the largest probability. If the *events*/*trials* syntax is used, the _INTO_ variable is created instead and it contains the values EVENT and NONEVENT.
- P_Y_i, the posterior probabilities of the normalized response level Y_i.
- If the CLM option is specified in the SCORE statement, the OUT= data set also includes:
  - LCL_Y_i, the lower 100(1-α)% confidence limits for P_Y_i
  - UCL_Y_i, the upper 100(1-α)% confidence limits for P_Y_i

OUTDIST= Output Data Set

The OUTDIST= data set contains every exact conditional distribution necessary to process the EXACT statement. For example, the following statements create one distribution for the x1 parameter and another for the x2 parameters, and produces the data set dist shown in Figure 42.7:

```
proc logistic;
   class x2 / param=ref;
   model y=x1 x2;
   exact x1 x2/ outdist=dist;
proc print data=dist;
run;
```

```
                   Obs    x1    x20    x21    Count      Score        Prob

                     1     .     0      0        3     5.81151     0.03333
                     2     .     0      1       15     1.66031     0.16667
                     3     .     0      2        9     3.12728     0.10000
                     4     .     1      0       15     1.46523     0.16667
                     5     .     1      1       18     0.21675     0.20000
                     6     .     1      2        6     4.58644     0.06667
                     7     .     2      0       19     1.61869     0.21111
                     8     .     2      1        2     3.27293     0.02222
                     9     .     3      0        3     6.27189     0.03333
                    10     2     .      .        6     3.03030     0.12000
                    11     3     .      .       12     0.75758     0.24000
                    12     4     .      .       11     0.00000     0.22000
                    13     5     .      .       18     0.75758     0.36000
                    14     6     .      .        3     3.03030     0.06000
```

Figure 42.7. OUTDIST

The first nine observations in the dist data set contain a exact distribution for the parameters of the x2 effect (hence the values for the x1 parameter are missing), and the remaining five observations are for the x1 parameter. If a joint distribution was created, there would be observations with values for both the x1 and x2 parameters. For CLASS variables, the corresponding parameters in the dist data set are identified by concatenating the variable name with the appropriate classification level.

The data set contains the possible sufficient statistics of the parameters for the effects specified in the EXACT statement, and the Count variable contains the number of different responses that yield these statistics. For example, there were 6 possible response vectors $\mathbf{y}$ for which the dot product $\mathbf{y}'\mathbf{x1}$ was equal to 2, and for which $\mathbf{y}'\mathbf{x20}$, $\mathbf{y}'\mathbf{x21}$, and $\mathbf{y}'\mathbf{1}$ were equal to their actual observed values (displayed in the "Sufficient Statistics" table). When hypothesis tests are performed on the parameters, the Prob variable contains the probability of obtaining that statistic (which is just the count divided by the total count), and the Score variable contains the score for that statistic. For more information, see the section "EXACT Statement Examples" on page 2302.

OUTROC= Output Data Set

The OUTROC= data set contains data necessary for producing the ROC curve, and can be created by specifying the OUTROC= option in the MODEL statement or the OUTROC= option in the SCORE statement: It has the following variables:

- any BY variables specified
- _STEP_, the model step number. This variable is not included if model selection is not requested.
- _PROB_, the estimated probability of an event. These estimated probabilities serve as cutpoints for predicting the response. Any observation with an estimated event probability that exceeds or equals _PROB_ is predicted to be an event; otherwise, it is predicted to be a nonevent. Predicted probabilities

that are close to each other are grouped together, with the maximum allowable difference between the largest and smallest values less than a constant that is specified by the ROCEPS= option. The smallest estimated probability is used to represent the group.

- _POS_, the number of correctly predicted event responses
- _NEG_, the number of correctly predicted nonevent responses
- _FALPOS_, the number of falsely predicted event responses
- _FALNEG_, the number of falsely predicted nonevent responses
- _SENSIT_, the sensitivity, which is the proportion of event observations that were predicted to have an event response
- _1MSPEC_, one minus specificity, which is the proportion of nonevent observations that were predicted to have an event response

Note that none of these statistics are affected by the bias-correction method discussed in the "Classification Table" section on page 2352. An ROC curve is obtained by plotting _SENSIT_ against _1MSPEC_. For more information, see the section "Receiver Operating Characteristic Curves" on page 2357.

Computational Resources

The memory needed to fit an unconditional model is approximately $24(p+2)^2$ bytes, where p is the number of parameters estimated. For cumulative response models with more than two response levels, a test of the parallel lines assumption requires an additional memory of approximately $4k^2(m+1)^2+24(m+2)^2$ bytes, where k is the number of response levels and m is the number of slope parameters. However, if this additional memory is not available, the procedure skips the test and finishes the other computations. You may need more memory if you use the SELECTION= option for model building.

The data that consist of relevant variables (including the design variables for model effects) and observations for fitting the model are stored in the utility file. If sufficient memory is available, such data will also be kept in memory; otherwise, the data are reread from the utility file for each evaluation of the likelihood function and its derivatives, with the resulting execution time of the procedure substantially increased.

If a conditional logistic regression is performed, then approximately $4(m^2+m+4)\max_h(m_h)+(8s_H+36)H+12s_H$ additional bytes of memory are needed, where m_h is the number of events in stratum h, H is the total number of strata, and s_H is the number of variables used to define the strata.

Computational Resources for Exact Conditional Logistic Regression

Many problems require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For such problems, consider whether exact methods are really necessary. Stokes, Davis, and Koch (2000) suggest looking at exact p-values when the sample size is small and the approximate p-values from the unconditional analysis are less than 0.10, and they provide *rules of thumb* for determining when various models are valid.

A formula does not exist that can predict the amount of time and memory necessary to generate the exact conditional distributions for a particular problem. The time and memory required depends on several factors, including the total sample size, the number of parameters of interest, the number of nuisance parameters, and the order in which the parameters are processed. To provide a feel for how these factors affect performance, 19 data sets containing Nobs $\in \{10, \ldots, 500\}$ observations consisting of up to 10 independent uniform binary covariates (X1,. . .,*XN*) and a binary response variable (Y), are generated and exact conditional distributions are created for X1 conditional on the other covariates using the default METHOD=NETWORK. Figure 42.8 displays results obtained on a 400Mhz PC with 768MB RAM running Microsoft Windows NT.

```
data one;
   do obs=1 to HalfNobs;
      do Y=0 to 1;
         X1=round(ranuni(0));
         ...
         XN=round(ranuni(0));
         output;
      end;
   end;
options fullstimer;
proc logistic exactonly exactoptions(method=network maxtime=1200);
   class X1 ... XN / param=ref;
   model Y=X1 ... XN;
   exact X1 / outdist=dist;
run;
```

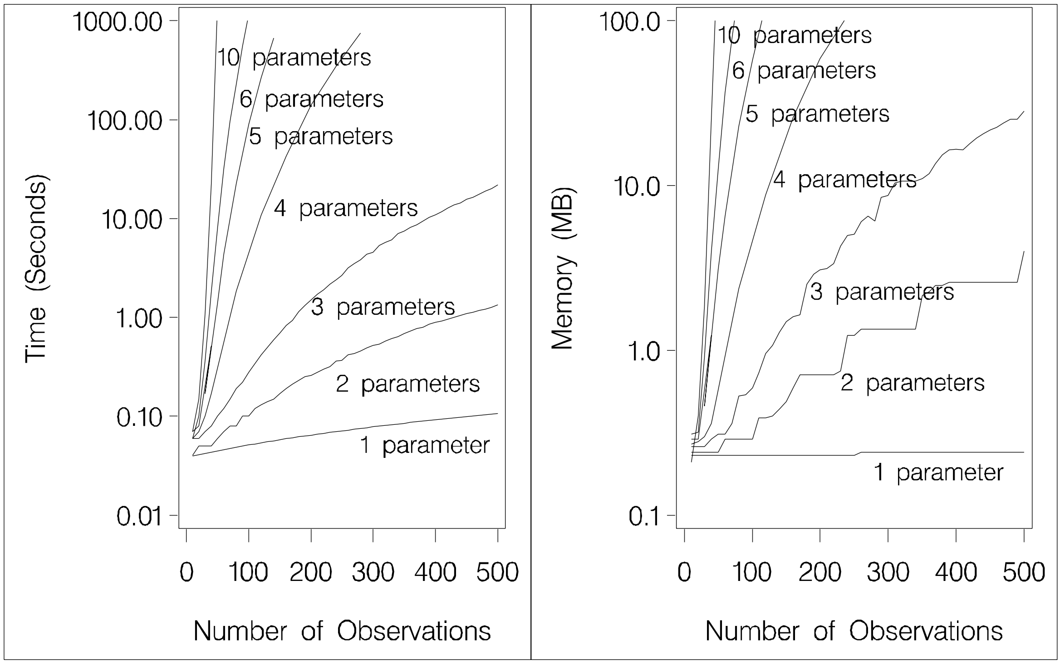


Figure 42.8. Mean Time and Memory Required

At any time while PROC LOGISTIC is deriving the distributions, you can terminate the computations by pressing the system interrupt key sequence (refer to the SAS Companion for your system) and choosing to stop computations. If you run out of memory, refer to the SAS Companion for your system to see how to allocate more.

You can use the EXACTOPTIONS option MAXTIME= to limit the total amount of time PROC LOGISTIC uses to derive all of the exact distributions. If PROC LOGISTIC does not finish within that time, the procedure terminates.

Calculation of frequencies are performed in the log-scale by default. This reduces the need to check for excessively large frequencies but can be slower than not scaling. You can turn off the log-scaling by specifying the NOLOGSCALE option in the MODEL statement. If a frequency in the exact distribution is larger than the largest integer that can be held in double-precision, a warning is printed to the LOG, but since inaccuracies due to adding small numbers to these large frequencies may have little-or-no effect on the statistics, the exact computations continue.

You can monitor the progress of the procedure by submitting your program with the EXACTOPTIONS option STATUSTIME=. If the procedure is too slow, you can try another method by specifying the EXACTOPTIONS option METHOD=, you can try reordering the variables in the MODEL statement (note that CLASS variables are always processed before continuous covariates), or you can try reparameterizing your classification variables, for example:

```
   class  class-variables / param=ref ref=first order=freq;
```

Displayed Output

If you use the NOPRINT option in the PROC LOGISTIC statement, the procedure does not display any output. Otherwise, the displayed output of the LOGISTIC procedure includes the following:

- “Model Information” table, which gives
  - name of the input Data Set
  - name and label of the Response Variable, if the *single-trial* syntax is used
  - number of Response Levels, if the *single-trial* syntax is used
  - name of the Events Variable, if the *events/trials* syntax is used
  - name of the Trials Variable, if the *events/trials* syntax is used
  - name of the Offset Variable, if the OFFSET= option is specified
  - name of the Frequency Variable, if the FREQ statement is specified
  - name of the Weight Variable, if the WEIGHT statement is specified
  - Number of Strata, if the STRATA statement is specified
  - Number of Strata Ignored and the total Frequency Ignored, if the STRATA statement is specified and at least one stratum has no events or no nonevents
  - Link Function

  - Optimization Technique
  - seed, if METHOD=NETWORKMC is specified

- “Number of Observations” table, which gives

  - Number of Observations read from the input data set
  - Number of Observations used in the analysis
  - Sum of Frequencies of all the observations read from the input data set
  - Sum of Frequencies of all the observations used in the analysis
  - Sum of Weights of all the observations read from the input data set
  - Sum of Weights of all the observations used in the analysis
  - Normalized Sum of Weights of all the observations used in the analysis, if the SCALE=WILLIAMS option is specified in the MODEL statement or the NORMALIZE option is specified in the WEIGHT statement.

  An ODS OUTPUT data set created from this table contains all of the information in every row.

- “Response Profile” table, which gives, for each response level, the ordered value (an integer between one and the number of response levels, inclusive); the value of the response variable if the *single-trial* syntax is used or the values “Event” and “Nonevent” if the *events/trials* syntax is used; the count or frequency; and the sum of weights if the WEIGHT statement is specified
- “Class Level Information” table, which gives the level and the design variables for each CLASS explanatory variable
- “Descriptive Statistics for Continuous Explanatory Variables” table for continuous explanatory variables, the “Frequency Distribution of Class Variables,” and the “Weight Distribution of Class Variables” tables (if the WEIGHT statement is specified), if you specify the SIMPLE option in the PROC LOGISTIC statement. The “Descriptive Statistics for Continuous Explanatory Variables” table contains the mean, standard deviation, maximum and minimum of each continuous variable specified in the MODEL statement.
- “Maximum Likelihood Iterative Phase” table, if you use the ITPRINT option in the MODEL statement. This table gives the iteration number, the step size (in the scale of 1.0, .5, .25, and so on) or the ridge value, -2 log likelihood, and parameter estimates for each iteration. Also displayed are the last evaluation of the gradient vector and the last change in the -2 log likelihood.
- Pearson and deviance goodness-of-fit statistics, if you use the SCALE= option in the MODEL statement
- score test result for testing the parallel lines assumption, if an ordinal response model is fitted. If LINK=CLOGLOG or LINK=PROBIT, this test is labeled “Score Test for the Parallel Slopes Assumption.” The proportion odds assumption is a special case of the parallel lines assumption when LINK=LOGIT. In this case, the test is labeled “Score Test for the Proportional Odds Assumption”.
- “Model Fit Statistics” and “Testing Global Null Hypothesis: BETA=0” tables, which give the various criteria (-2 Log L, AIC, SC) based on the likelihood

for fitting a model with intercepts only and for fitting a model with intercepts and explanatory variables. If you specify the NOINT option, these statistics are calculated without considering the intercept parameters. The third column of the table gives the chi-square statistics and p-values for the -2 Log L statistic and for the Score statistic. These test the joint effect of the explanatory variables included in the model. The Score criterion is always missing for the models identified by the first two columns of the table. Note also that the first two rows of the Chi-Square column are always missing, since tests cannot be performed for AIC and SC.

- generalized R^2 measures for the fitted model, if you specify the RSQUARE option in the MODEL statement
- "Type 3 Analysis of Effects" table, if the model contains an effect involving a CLASS variable. This table gives the Wald Chi-square statistic, the degrees of freedom, and the p-value for each effect in the model
- "Analysis of Maximum Likelihood Estimates" table, which includes
  - parameter name, which also identifies the CLASS variable level and, for generalized logit models, a response variable column to identify the corresponding logit by displaying the nonreference level of the logit
  - maximum likelihood estimate of the parameter
  - estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
  - Wald chi-square statistic, computed by squaring the ratio of the parameter estimate divided by its standard error estimate
  - p-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom
  - standardized estimate for the slope parameter, if you specify the STB option in the MODEL statement. This estimate is given by $\hat{\beta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

    $$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

    Standardized estimates of the intercept parameters are set to missing.
  - $\mathrm{e}^{\hat{\beta}_i}$ for each slope parameter β_i, if you specify the EXPB option in the MODEL statement. For continuous variables, this is equivalent to the estimated odds ratio for a 1 unit change.
  - label of the variable, if you specify the PARMLABEL option in the MODEL statement and if space permits. Due to constraints on the line size, the variable label may be suppressed in order to display the table in one panel. Use the SAS system option LINESIZE= to specify a larger line size to accommodate variable labels. A shorter line size can break the table into two panels allowing labels to be displayed.

- "Odds Ratio Estimates" table, which contains the odds ratio estimates and the corresponding 95% Wald confidence intervals. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.
- "Association of Predicted Probabilities and Observed Responses" table, which includes a breakdown of the number of pairs with different responses, and four rank correlation indexes: Somers' *D*, Goodman-Kruskal Gamma, and Kendall's Tau-*a*, and *c*
- confidence intervals for all the parameters, if you use the CLPARM= option in the MODEL statement
- confidence intervals for all the odds ratios, if you use the CLODDS= option in the MODEL statement
- a summary of the model-building process, if you use a FORWARD, BACKWARD, or STEPWISE selection method. This summary gives the step number, the explanatory variables entered or removed at each step, the chi-square statistic, and the corresponding p-value on which the entry or removal of the variable is based (the score chi-square is used to determine entry; the Wald chi-square is used to determine removal)
- "Analysis of Variables Removed by Fast Backward Elimination" table, if you specify the FAST option in the MODEL statement. This table gives the approximate chi-square statistic for the variable removed, the corresponding p-value with respect to a chi-square distribution with one degree of freedom, the residual chi-square statistic for testing the joint significance of the variable and the preceding ones, the degrees of freedom, and the p-value of the residual chi-square with respect to a chi-square distribution with the corresponding degrees of freedom
- "Analysis of Effects not in the Model" table, if you specify the DETAILS option in the MODEL statement. This table gives the score chi-square statistic for testing the significance of each variable not in the model after adjusting for the variables already in the model, and the p-value of the chi-square statistic with respect to a chi-square distribution with one degree of freedom
- classification table, if you use the CTABLE option in the MODEL statement. For each prior event probability (labeled "Prob Event") specified by the PEVENT= option and each cutpoint specified in the PPROB= option, the table gives the four entries of the 2×2 table of observed and predicted responses and the percentages of correct classification, sensitivity, specificity, false positive, and false negative. The columns labeled "Correct" give the number of correctly classified events and nonevents. "Incorrect Event" gives the number of nonevents incorrectly classified as events. "Incorrect Nonevent" gives the number of nonevents incorrectly classified as events.
- estimated covariance matrix of the parameter estimates, if you use the COVB option in the MODEL statement
- estimated correlation matrix of the parameter estimates, if you use the CORRB option in the MODEL statement
- "Contrast Test Results" table, if you specify a CONTRAST statement. This table gives the result of the Wald test for each CONTRAST specified. If you

specify the E option in the CONTRAST statement, then the contrast matrix is displayed. If you specify the ESTIMATE= option in the CONTRAST statement, then estimates and Wald tests for each contrast (row of the contrast matrix) or exponentiated contrast are produced.

- "Linear Hypothesis Testing" table, if you specify a TEST statement. This table gives the result of the Wald test for each TEST statement specified. If you specify the PRINT option in the TEST statement, then matrices used in the intermediate calculations are also displayed.
- results of the Hosmer and Lemeshow test for the goodness of fit of the fitted model, if you use the LACKFIT option in the MODEL statement
- "Regression Diagnostics" table, if you use the INFLUENCE option in the MODEL statement. This table gives, for each observation, the case number (which is the observation number), the values of the explanatory variables included in the model, the Pearson residual, the deviance residual, the diagonal element of the hat matrix, the standardized difference in the estimate for each parameter (*name* DFBETA, where *name* is either Intercept or the name of an explanatory variable), two confidence interval displacement diagnostics (C and CBAR), the change in the Pearson chi-square statistic (DIFCHISQ), and the change in the deviance (DIFDEV)

  If you also specify the STRATA statement, then this table contains the case number (which is the observation number), the values of the explanatory variables included in the model, the estimated one-step Pearson residual, the estimated one-step diagonal element of the hat matrix, and the estimated one-step standardized difference in the estimate for each parameter.
- index plots of regression diagnostics, if you specify the IPLOTS option in the MODEL statement. These include plots of
  - Pearson residuals
  - deviance residuals
  - diagonal elements of the hat matrix
  - standardized differences in parameter estimates, DFBETA0 for the intercept estimate, DFBETA1 for the slope estimate of the first explanatory variable in the MODEL statement, and so on
  - confidence interval displacement diagnostics C
  - confidence interval displacement diagnostics CBAR
  - changes in the Pearson chi-square statistic
  - changes in the deviance
- if you specify a STRATA statement
  - "Strata Summary" table, which displays a pattern of the number of events and the number of non-events in a stratum, the number of strata having that pattern, and the total number of observations contained in those strata
  - "Strata Information" table, if you specify the INFO option on the STRATA statement. This table displays each stratum, its frequency, and the number of events and non-events in that stratum.

- if you specify an EXACT statement
  - “Sufficient Statistics” table, if you request an OUTDIST= data set. This table is displayed before printing any of the exact analysis results and lists the parameters and their observed sufficient statistics.
  - “Conditional Exact Tests” table, which provides two tests for the null hypothesis that the parameters for the specified effects are zero: the Exact Probability Test and the Exact Conditional Scores test. For each test, the test statistic, an exact p-value (the probability of obtaining a more extreme statistic than the observed, assuming the null hypothesis), and a mid p-value (which adjusts for the discreteness of the distribution) are displayed.
  - “Exact Parameter Estimates” table, if you specify the ESTIMATE, ESTIMATE=PARM, or ESTIMATE=BOTH options. This table gives individual parameter estimates for each variable (conditional on the values of all the other parameters in the model), confidence limits, and a two-sided p-value (twice the one-sided p-value) for testing that the parameter is zero.
  - “Exact Odds Ratios” table, if you specify the ESTIMATE=ODDS or ESTIMATE=BOTH options. This table gives odds ratio estimates for the individual parameters, confidence limits, and a two-sided p-value for testing that the odds ratio is 1.

ODS Table Names

PROC LOGISTIC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 42.2. ODS Tables Produced in PROC LOGISTIC

ODS Table Name	**Description**	**Statement**	**Option**
Association	Association of predicted probabilities and observed responses	MODEL	default
BestSubsets	Best subset selection	MODEL	SELECTION=SCORE
ClassFreq	Frequency breakdown of CLASS variables	PROC	Simple (with CLASS vars)
ClassLevelInfo	CLASS variable levels and design variables	MODEL	default (with CLASS vars)
Classification	Classification table	MODEL	CTABLE
ClassWgt	Weight breakdown of CLASS variables	PROC, WEIGHT	Simple (with CLASS vars)
CLOddsPL	Profile likelihood confidence limits for odds ratios	MODEL	CLODDS=PL
CLOddsWald	Wald’s confidence limits for odds ratios	MODEL	CLODDS=WALD

Table 42.2. (continued)

ODS Table Name	Description	Statement	Option
CLParmPL	Profile likelihood confidence limits for parameters	MODEL	CLPARM=PL
CLParmWald	Wald's confidence limits for parameters	MODEL	CLPARM=WALD
ContrastCoeff	L matrix from CONTRAST	CONTRAST	E
ContrastEstimate	Estimates from CONTRAST	CONTRAST	ESTIMATE=
ContrastTest	Wald test for CONTRAST	CONTRAST	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB
CumulativeModelTest	Test of the cumulative model assumption	MODEL	(ordinal response)
EffectNotInModel	Test for effects not in model	MODEL	SELECTION=S/F

ExactOddsRatio	Exact Odds Ratios	EXACT	ESTIMATE=ODDS,
			ESTIMATE=BOTH

ExactParmEst	Parameter Estimates	EXACT	ESTIMATE,
			ESTIMATE=PARM,
			ESTIMATE=BOTH

ExactTests	Conditional Exact Tests	EXACT	default
FastElimination	Fast backward elimination	MODEL	SELECTION=B,FAST
FitStatistics	Model fit statistics	MODEL	default
GlobalScore	Global score test	MODEL	NOFIT
GlobalTests	Test for global null hypothesis	MODEL	default
GoodnessOfFit	Pearson and deviance goodness-of-fit tests	MODEL	SCALE
IndexPlots	Batch capture of the index plots	MODEL	IPLOTS
Influence	Regression diagnostics	MODEL	INFLUENCE
IterHistory	Iteration history	MODEL	ITPRINT
LackFitChiSq	Hosmer-Lemeshow chi-square test results	MODEL	LACKFIT
LackFitPartition	Partition for the Hosmer-Lemeshow test	MODEL	LACKFIT
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
LogLikeChange	Final change in the log likelihood	MODEL	ITPRINT
ModelBuildingSummary	Summary of model building	MODEL	SELECTION=B/F/S
ModelInfo	Model information	PROC	default
NObs	Number of Observations	PROC	default
OddsRatios	Odds ratios	MODEL	default

Table 42.2. (continued)

ODS Table Name	**Description**	**Statement**	**Option**
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
RSquare	R-square	MODEL	RSQUARE
ResidualChiSq	Residual chi-square	MODEL	SELECTION=F/B
ResponseProfile	Response profile	PROC	default
SimpleStatistics	Summary statistics for explanatory variables	PROC	SIMPLE
StrataSummary	Number of strata with specific response frequencies	STRATA	default
StrataInfo	Event and non-event frequencies for each stratum	STRATA	INFO
SuffStats	Sufficient Statistics	EXACT	OUTDIST=
TestPrint1	**L**[cov(**b**)]**L**’ and **Lb-c**	TEST	PRINT
TestPrint2	Ginv(**L**[cov(**b**)]**L**’) and Ginv(**L**[cov(**b**)]**L**’)(**Lb-c**)	TEST	PRINT
TestStmts	Linear hypotheses testing results	TEST	default
Type3	Type 3 tests of effects	MODEL	default (with CLASS variables)

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the LOGISTIC procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement in addition to options on the MODEL or GRAPHICS statement as described in the following sections. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

MODEL Statement Options

If the INFLUENCE or IPLOTS option is specified in the MODEL statement, then the lineprinter plots are suppressed and ODS GRAPHICS versions of the plots are produced.

If you specify the OUTROC= option, and if ROCEPS= is not specified, then ROC curves are produced. If you also specify a SELECTION= method then an overlaid plot of all the ROC curves for each step of the selection process is displayed.

GRAPHICS Statement and Options

GRAPHICS *options* **;**

The GRAPHICS statement provides options for requesting and modifying certain graphical displays. This statement has no effect unless ODS GRAPHICS ON has been specified. The functionality of this statement may be replaced by alternative syntax in a future release.

The following options are available.

DFBETAS displays the DFBETAS versus Case Number plots. This acts like DFBETAS=_ALL_ in the OUTPUT statement. These plots are produced by default when the GRAPHICS statement is specified.

HATDIAG displays plots of DIFCHISQ, DIFDEV, and DFBETAS (when the DFBETAS option is specified) versus the hat diagonals.

INFLUENCE | INDEX displays the INFLUENCE plots with no DFBETAS. These plots are produced by default when the GRAPHICS statement is specified.

PHAT displays plots of DIFCHISQ, DIFDEV, and DFBETAS (when the DFBETAS option is specified) versus the predicted event probability.

ALL invokes the DFBETAS, HATDIAG, INFLUENCE, and PHAT options.

NOINFLUENCE suppresses the default INFLUENCE and DFBETAS plots.

NOPANELS unpanels the graphical displays and produces a series of plots which form the panelled display.

ROC displays the ROC curve. If the ROCEPS= option is specified on the MODEL statement then it must be equal to zero, otherwise no ROC curve is produced. If you also specify a SELECTION= method then an overlaid plot of all the ROC curves for each step of the selection process is displayed.

ESTPROB(*fit-options***)** displays the fit curves for the model when only one continuous covariate is specified in the model. If you use events/trials syntax, then this displays the estimated event probability and the prediction limits versus the covariate with the observed proportions overlaid on the graph. If you use single-trial syntax, this displays the estimated event probability and the prediction limits versus the covariate with the observed responses overlaid on the graph. If you specify a polytomous logit model, then the estimated probabilities for each possible response level are graphed. If you have an OFFSET= variable with more than one value, then the prediction curves are replaced with error bars and the estimated probabilities are displayed at the observed covariate values.

The following *fit-options* are available with the ESTPROB option.

ALPHA=α specifies the size of the prediction interval. The ALPHA= value specified on the PROC statement is the default. If neither ALPHA= value is specified, then ALPHA=0.05 by default.

GRIDSIZE=n specifies the number of equally-spaced points at which the fit curve is computed. By default, GRIDSIZE=50.

OBSERVE specifies that the fit curve should be computed at the observed values only.

See Example 42.6 on page 2422 and Example 42.7 on page 2429 for examples of the ODS graphical displays.

ODS Graph Names

PROC LOGISTIC assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 42.3.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 42.3. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 42.3. ODS Graphics Produced by PROC LOGISTIC

ODS Graph Name	Plot Description	Statement	Option
InfluencePlots	Panel of influence statistics vs. case number	GRAPHICS or MODEL	INFLUENCE INFLUENCE or IPLOTS
PearsonChisquarePlot	Pearson chi-square residual vs. case number	GRAPHICS	INFLUENCE NOPANELS
DevianceResidualPlot	Deviance residual vs. case number	GRAPHICS	INFLUENCE NOPANELS
HatPlot	Hat diagonal vs. case number	GRAPHICS	INFLUENCE NOPANELS
CPlot	CI displacement C vs. case number	GRAPHICS	INFLUENCE NOPANELS
CBarPlot	CI displacement Cbar vs. case number	GRAPHICS	INFLUENCE NOPANELS
DeltaChisqPlot	Difchisq vs. case number	GRAPHICS	INFLUENCE NOPANELS
DeltaDeviancePlot	Difdev vs. case number	GRAPHICS	INFLUENCE NOPANELS
DFBetasPlot	DFBetas vs. case number	GRAPHICS	DFBETAS NOPANELS
EstProbPlots	Panel of estimated probability vs. influence	GRAPHICS	PHAT
PhatDifChisqPlot	Estimated probability vs. difchisq	GRAPHICS	PHAT NOPANELS
PhatDifDevPlot	Estimated probability vs. difdev	GRAPHICS	PHAT NOPANELS
PhatDFBetasPlot	Estimated probability vs. dfbetas	GRAPHICS	PHAT NOPANELS
HatDiagPlots	Panel of hat diagonals vs. influence statistics	GRAPHICS	HATDIAG

Table 42.3. (continued)

ODS Graph Name	Plot Description	Statement	Option
HatDiagDifChisqPlot	Hat diagonals vs. difchisq	GRAPHICS	HATDIAG NOPANELS
HatDiagDifDevPlot	Hat diagonals vs. difdev	GRAPHICS	HATDIAG NOPANELS
HatDiagDFBetasPlot	Hat diagonals vs. dfbetas	GRAPHICS	HATDIAG NOPANELS
ROCCurve	Receiver operating characteristics curve	GRAPHICS or MODEL	ROC OUTROC=
ROCOverlay	ROC curves for model selection steps	GRAPHICS and MODEL	ROC SELECTION=
FitCurve	Estimated probability vs. one continuous covariate	GRAPHICS	ESTPROB

Examples

Example 42.1. Stepwise Logistic Regression and Predicted Values

Consider a study on cancer remission (Lee 1974). The data, consisting of patient characteristics and whether or not cancer remission occurred, are saved in the data set Remission.

```
data Remission;
   input remiss cell smear infil li blast temp;
   label remiss='Complete Remission';
   datalines;
1   .8   .83  .66  1.9  1.1     .996
1   .9   .36  .32  1.4   .74    .992
0   .8   .88  .7    .8   .176   .982
0  1     .87  .87   .7  1.053   .986
1   .9   .75  .68  1.3   .519   .98
0  1     .65  .65   .6   .519   .982
1   .95  .97  .92  1    1.23    .992
0   .95  .87  .83  1.9  1.354  1.02
0  1     .45  .45   .8   .322   .999
0   .95  .36  .34   .5  0      1.038
0   .85  .39  .33   .7   .279   .988
0   .7   .76  .53  1.2   .146   .982
0   .8   .46  .37   .4   .38   1.006
0   .2   .39  .08   .8   .114   .99
0  1     .9   .9   1.1  1.037   .99
1  1     .84  .84  1.9  2.064  1.02
0   .65  .42  .27   .5   .114  1.014
0  1     .75  .75  1    1.322  1.004
0   .5   .44  .22   .6   .114   .99
1  1     .63  .63  1.1  1.072   .986
0  1     .33  .33   .4   .176  1.01
0   .9   .93  .84   .6  1.591  1.02
1  1     .58  .58  1     .531  1.002
0   .95  .32  .3   1.6   .886   .988
1  1     .6   .6   1.7   .964   .99
```

```
1  1     .69  .69   .9   .398   .986
0  1     .73  .73   .7   .398   .986
;
```

The data set Remission contains seven variables. The variable remiss is the cancer remission indicator variable with a value of 1 for remission and a value of 0 for nonremission. The other six variables are the risk factors thought to be related to cancer remission.

The following invocation of PROC LOGISTIC illustrates the use of stepwise selection to identify the prognostic factors for cancer remission. A significance level of 0.3 (SLENTRY=0.3) is required to allow a variable into the model, and a significance level of 0.35 (SLSTAY=0.35) is required for a variable to stay in the model. A detailed account of the variable selection process is requested by specifying the DETAILS option. The Hosmer and Lemeshow goodness-of-fit test for the final selected model is requested by specifying the LACKFIT option. The OUTEST= and COVOUT options in the PROC LOGISTIC statement create a data set that contains parameter estimates and their covariances for the final selected model. The response variable option EVENT= sets remiss=1 (remission) to be Ordered Value 1 so that the probability of remission is modeled. The OUTPUT statement creates a data set that contains the cumulative predicted probabilities and the corresponding confidence limits, and the individual and cross validated predicted probabilities for each observation.

```
title 'Stepwise Regression on Cancer Remission Data';
proc logistic data=Remission outest=betas covout;
   model remiss(event='1')=cell smear infil li blast temp
                / selection=stepwise
                  slentry=0.3
                  slstay=0.35
                  details
                  lackfit;
   output out=pred p=phat lower=lcl upper=ucl
          predprob=(individual crossvalidate);
run;

proc print data=betas;
   title2 'Parameter Estimates and Covariance Matrix';
run;

proc print data=pred;
   title2 'Predicted Probabilities and 95% Confidence Limits';
run;
```

In stepwise selection, an attempt is made to remove any insignificant variables from the model before adding a significant variable to the model. Each addition or deletion of a variable to or from a model is listed as a separate step in the displayed output, and at each step a new model is fitted. Details of the model selection steps are shown in Output 42.1.1 –Output 42.1.5.

Output 42.1.1. Startup Model

```
                  Stepwise Regression on Cancer Remission Data

                             The LOGISTIC Procedure

                               Model Information

     Data Set                      WORK.REMISSION
     Response Variable             remiss               Complete Remission
     Number of Response Levels     2
     Model                         binary logit
     Optimization Technique        Fisher's scoring


                    Number of Observations Read          27
                    Number of Observations Used          27


                                Response Profile

                       Ordered                    Total
                         Value       remiss     Frequency

                             1            0            18
                             2            1             9

                        Probability modeled is remiss=1.


                          Stepwise Selection Procedure


Step  0. Intercept entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                   Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

      Intercept     1     -0.6931      0.4082        2.8827        0.0895


                            Residual Chi-Square Test

                       Chi-Square       DF     Pr > ChiSq

                           9.4609        6         0.1493


                     Analysis of Effects Eligible for Entry

                                          Score
                   Effect      DF    Chi-Square    Pr > ChiSq

                   cell         1        1.8893        0.1693
                   smear        1        1.0745        0.2999
                   infil        1        1.8817        0.1701
                   li           1        7.9311        0.0049
                   blast        1        3.5258        0.0604
                   temp         1        0.6591        0.4169
```

Output 42.1.2. Step 1 of the Stepwise Analysis

```
Step  1. Effect li entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                                   Intercept
                                    Intercept            and
                     Criterion           Only     Covariates

                     AIC               36.372         30.073
                     SC                37.668         32.665
                     -2 Log L          34.372         26.073


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         8.2988        1         0.0040
            Score                    7.9311        1         0.0049
            Wald                     5.9594        1         0.0146


                   Analysis of Maximum Likelihood Estimates

                                           Standard          Wald
      Parameter    DF    Estimate           Error    Chi-Square    Pr > ChiSq

      Intercept     1     -3.7771          1.3786        7.5064        0.0061
      li            1      2.8973          1.1868        5.9594        0.0146


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     84.0    Somers' D    0.710
               Percent Discordant     13.0    Gamma        0.732
               Percent Tied            3.1    Tau-a        0.328
               Pairs                   162    c            0.855


                            Residual Chi-Square Test

                       Chi-Square       DF     Pr > ChiSq

                           3.1174        5         0.6819


NOTE: No effects for the model in Step 1 are removed.


                     Analysis of Effects Eligible for Entry

                                         Score
                   Effect      DF    Chi-Square    Pr > ChiSq

                   cell         1        1.1183        0.2903
                   smear        1        0.1369        0.7114
                   infil        1        0.5715        0.4497
                   blast        1        0.0932        0.7601
                   temp         1        1.2591        0.2618
```

Output 42.1.3. Step 2 of the Stepwise Analysis

```
Step  2. Effect temp entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                                 Intercept
                                   Intercept           and
                     Criterion          Only    Covariates

                     AIC              36.372        30.648
                     SC               37.668        34.535
                     -2 Log L         34.372        24.648


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         9.7239        2         0.0077
            Score                    8.3648        2         0.0153
            Wald                     5.9052        2         0.0522


                   Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

      Intercept     1     47.8448     46.4381        1.0615        0.3029
      li            1      3.3017      1.3593        5.9002        0.0151
      temp          1    -52.4214     47.4897        1.2185        0.2697


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     87.0    Somers' D    0.747
               Percent Discordant     12.3    Gamma        0.752
               Percent Tied            0.6    Tau-a        0.345
               Pairs                   162    c            0.873


                            Residual Chi-Square Test

                       Chi-Square       DF     Pr > ChiSq

                           2.1429        4         0.7095


NOTE: No effects for the model in Step 2 are removed.


                     Analysis of Effects Eligible for Entry

                                          Score
                   Effect      DF    Chi-Square    Pr > ChiSq

                   cell         1        1.4700        0.2254
                   smear        1        0.1730        0.6775
                   infil        1        0.8274        0.3630
                   blast        1        1.1013        0.2940
```

Output 42.1.4. Step 3 of the Stepwise Analysis

```
Step  3. Effect cell entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                                  Intercept
                                   Intercept            and
                     Criterion          Only     Covariates

                     AIC              36.372         29.953
                     SC               37.668         35.137
                     -2 Log L         34.372         21.953


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        12.4184        3         0.0061
            Score                    9.2502        3         0.0261
            Wald                     4.8281        3         0.1848


                   Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

      Intercept     1     67.6339     56.8875        1.4135        0.2345
      cell          1      9.6521      7.7511        1.5507        0.2130
      li            1      3.8671      1.7783        4.7290        0.0297
      temp          1    -82.0737     61.7124        1.7687        0.1835


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     88.9    Somers' D    0.778
               Percent Discordant     11.1    Gamma        0.778
               Percent Tied            0.0    Tau-a        0.359
               Pairs                   162    c            0.889


                            Residual Chi-Square Test

                       Chi-Square       DF     Pr > ChiSq

                           0.1831        3         0.9803


NOTE: No effects for the model in Step 3 are removed.


                     Analysis of Effects Eligible for Entry

                                          Score
                   Effect      DF    Chi-Square    Pr > ChiSq

                   smear        1        0.0956        0.7572
                   infil        1        0.0844        0.7714
                   blast        1        0.0208        0.8852


NOTE: No (additional) effects met the 0.3 significance level for entry into the
      model.
```

Output 42.1.5. Summary of the Stepwise Selection

```
                          Summary of Stepwise Selection

               Effect                    Number       Score        Wald
   Step   Entered   Removed     DF          In   Chi-Square  Chi-Square   Pr > ChiSq

      1   li                     1           1       7.9311                   0.0049
      2   temp                   1           2       1.2591                   0.2618
      3   cell                   1           3       1.4700                   0.2254
```

Prior to the first step, the intercept-only model is fitted and individual score statistics for the potential variables are evaluated (Output 42.1.1). In Step 1 (Output 42.1.2), variable li is selected into the model since it is the most significant variable among those to be chosen ($p = 0.0049 < 0.3$). The intermediate model that contains an intercept and li is then fitted. li remains significant ($p = 0.0146 < 0.35$) and is not removed. In Step 2 (Output 42.1.3), variable temp is added to the model. The model then contains an intercept and variables li and temp. Both li and temp remain significant at 0.035 level; therefore, neither li nor temp is removed from the model. In Step 4 (Output 42.1.4), variable cell is added to the model. The model then contains an intercept and variables li, temp, and cell. None of these variables are removed from the model since all are significant at the 0.35 level. Finally, none of the remaining variables outside the model meet the entry criterion, and the stepwise selection is terminated. A summary of the stepwise selection is displayed in Output 42.1.5.

Output 42.1.6. Display of the LACKFIT Option

```
                    Partition for the Hosmer and Lemeshow Test

                                      remiss = 1              remiss = 0
       Group         Total     Observed     Expected     Observed     Expected

           1             3            0         0.00            3         3.00
           2             3            0         0.01            3         2.99
           3             3            0         0.19            3         2.81
           4             3            0         0.56            3         2.44
           5             4            1         1.09            3         2.91
           6             3            2         1.35            1         1.65
           7             3            2         1.84            1         1.16
           8             3            3         2.15            0         0.85
           9             2            1         1.80            1         0.20


                     Hosmer and Lemeshow Goodness-of-Fit Test

                        Chi-Square       DF     Pr > ChiSq

                            6.2983        7         0.5054
```

Results of the Hosmer and Lemeshow test are shown in Output 42.1.6. There is no evidence of a lack of fit in the selected model $(p = 0.5054)$.

Output 42.1.7. Data Set of Estimates and Covariances

```
                  Stepwise Regression on Cancer Remission Data
                   Parameter Estimates and Covariance Matrix

Obs    _LINK_    _TYPE_     _STATUS_      _NAME_       Intercept        cell

 1     LOGIT     PARMS     0 Converged    remiss           67.63       9.652
 2     LOGIT     COV       0 Converged    Intercept      3236.19     157.097
 3     LOGIT     COV       0 Converged    cell          157.10      60.079
 4     LOGIT     COV       0 Converged    smear              .            .
 5     LOGIT     COV       0 Converged    infil              .            .
 6     LOGIT     COV       0 Converged    li             64.57       6.945
 7     LOGIT     COV       0 Converged    blast              .            .
 8     LOGIT     COV       0 Converged    temp         -3483.23    -223.669

Obs    smear    infil        li       blast        temp     _LNLIKE_

 1       .        .       3.8671        .        -82.07     -10.9767
 2       .        .      64.5726        .      -3483.23     -10.9767
 3       .        .       6.9454        .       -223.67     -10.9767
 4       .        .        .            .           .       -10.9767
 5       .        .        .            .           .       -10.9767
 6       .        .       3.1623        .        -75.35     -10.9767
 7       .        .        .            .           .       -10.9767
 8       .        .     -75.3513        .       3808.42     -10.9767
```

The data set betas created by the OUTEST= and COVOUT options is displayed in Output 42.1.7. The data set contains parameter estimates and the covariance matrix for the final selected model. Note that all explanatory variables listed in the MODEL statement are included in this data set; however, variables that are not included in the final model have all missing values.

Output 42.1.8. Predicted Probabilities and Confidence Intervals

```
                                   Stepwise Regression on Cancer Remission Data
                                Predicted Probabilities and 95% Confidence Limits

                                                                                      _
    r                                       _  _                                      L
    e          s     i          b           F  I                                      E
    m    c     m     n          l      t    R  N     I        I        X        X     V     p
 O  i    e     e     f          a      e    O  T     P        P        P        P     E     h        l        u
 b  s    l     a     i    l     s      m    M  O     _        _        _        _     L     a        c        c
 s  s    l     r     l    i     t      p    _  _     0        1        0        1     _     t        l        l

 1  1  0.80  0.83  0.66  1.9  1.100  0.996  1  1  0.27735  0.72265  0.43873  0.56127  1  0.72265  0.16892  0.97093
 2  1  0.90  0.36  0.32  1.4  0.740  0.992  1  1  0.42126  0.57874  0.47461  0.52539  1  0.57874  0.26788  0.83762
 3  0  0.80  0.88  0.70  0.8  0.176  0.982  0  0  0.89540  0.10460  0.87060  0.12940  1  0.10460  0.00781  0.63419
 4  0  1.00  0.87  0.87  0.7  1.053  0.986  0  0  0.71742  0.28258  0.67259  0.32741  1  0.28258  0.07498  0.65683
 5  1  0.90  0.75  0.68  1.3  0.519  0.980  1  1  0.28582  0.71418  0.36901  0.63099  1  0.71418  0.25218  0.94876
 6  0  1.00  0.65  0.65  0.6  0.519  0.982  0  0  0.72911  0.27089  0.67269  0.32731  1  0.27089  0.05852  0.68951
 7  1  0.95  0.97  0.92  1.0  1.230  0.992  1  0  0.67844  0.32156  0.72923  0.27077  1  0.32156  0.13255  0.59516
 8  0  0.95  0.87  0.83  1.9  1.354  1.020  0  1  0.39277  0.60723  0.09906  0.90094  1  0.60723  0.10572  0.95287
 9  0  1.00  0.45  0.45  0.8  0.322  0.999  0  0  0.83368  0.16632  0.80864  0.19136  1  0.16632  0.03018  0.56123
10  0  0.95  0.36  0.34  0.5  0.000  1.038  0  0  0.99843  0.00157  0.99840  0.00160  1  0.00157  0.00000  0.68962
11  0  0.85  0.39  0.33  0.7  0.279  0.988  0  0  0.92715  0.07285  0.91723  0.08277  1  0.07285  0.00614  0.49982
12  0  0.70  0.76  0.53  1.2  0.146  0.982  0  0  0.82714  0.17286  0.63838  0.36162  1  0.17286  0.00637  0.87206
13  0  0.80  0.46  0.37  0.4  0.380  1.006  0  0  0.99654  0.00346  0.99644  0.00356  1  0.00346  0.00001  0.46530
14  0  0.20  0.39  0.08  0.8  0.114  0.990  0  0  0.99982  0.00018  0.99981  0.00019  1  0.00018  0.00000  0.96482
15  0  1.00  0.90  0.90  1.1  1.037  0.990  0  1  0.42878  0.57122  0.35354  0.64646  1  0.57122  0.25303  0.83973
16  1  1.00  0.84  0.84  1.9  2.064  1.020  1  1  0.28530  0.71470  0.47213  0.52787  1  0.71470  0.15362  0.97189
17  0  0.65  0.42  0.27  0.5  0.114  1.014  0  0  0.99938  0.00062  0.99937  0.00063  1  0.00062  0.00000  0.62665
18  0  1.00  0.75  0.75  1.0  1.322  1.004  0  0  0.77711  0.22289  0.73612  0.26388  1  0.22289  0.04483  0.63670
19  0  0.50  0.44  0.22  0.6  0.114  0.990  0  0  0.99846  0.00154  0.99842  0.00158  1  0.00154  0.00000  0.79644
20  1  1.00  0.63  0.63  1.1  1.072  0.986  1  1  0.35089  0.64911  0.42053  0.57947  1  0.64911  0.26305  0.90555
21  0  1.00  0.33  0.33  0.4  0.176  1.010  0  0  0.98307  0.01693  0.98170  0.01830  1  0.01693  0.00029  0.50475
22  0  0.90  0.93  0.84  0.6  1.591  1.020  0  0  0.99378  0.00622  0.99348  0.00652  1  0.00622  0.00003  0.56062
23  1  1.00  0.58  0.58  1.0  0.531  1.002  1  0  0.74739  0.25261  0.84423  0.15577  1  0.25261  0.06137  0.63597
24  0  0.95  0.32  0.30  1.6  0.886  0.988  0  1  0.12989  0.87011  0.03637  0.96363  1  0.87011  0.40910  0.98481
25  1  1.00  0.60  0.60  1.7  0.964  0.990  1  1  0.06868  0.93132  0.08017  0.91983  1  0.93132  0.44114  0.99573
26  1  1.00  0.69  0.69  0.9  0.398  0.986  1  0  0.53949  0.46051  0.62312  0.37688  1  0.46051  0.16612  0.78529
27  0  1.00  0.73  0.73  0.7  0.398  0.986  0  0  0.71742  0.28258  0.67259  0.32741  1  0.28258  0.07498  0.65683
```

The data set pred created by the OUTPUT statement is displayed in Output 42.1.8. It contains all the variables in the input data set, the variable phat for the (cumulative) predicted probability, the variables lcl and ucl for the lower and upper confidence limits for the probability, and four other variables (viz., IP_1, IP_0, XP_1, and XP_0) for the PREDPROBS= option. The data set also contains the variable _LEVEL_, indicating the response value to which phat, lcl, and ucl refer. For instance, for the first row of the OUTPUT data set, the values of _LEVEL_ and phat, lcl, and ucl are 1, 0.72265, 0.16892 and 0.97093, respectively; this means that the estimated probability that remiss≤1 is 0.723 for the given explanatory variable values, and the corresponding 95% confidence interval is (0.16892, 0.97093). The variables IP_1 and IP_0 contain the predicted probabilities that remiss=1 and remiss=0, respectively. Note that values of phat and IP_1 are identical since they both contain the probabilities that remiss=1. The variables XP_1 and XP_0 contain the cross validated predicted probabilities that remiss=1 and remiss=0, respectively.

Next, a different variable selection method is used to select prognostic factors for cancer remission, and an efficient algorithm is employed to eliminate insignificant variables from a model. The following SAS statements invoke PROC LOGISTIC to perform the backward elimination analysis.

```
title ’Backward Elimination on Cancer Remission Data’;
proc logistic data=Remission;
   model remiss(event=’1’)=temp cell li smear blast
         / selection=backward fast slstay=0.2 ctable;
run;
```

The backward elimination analysis (SELECTION=BACKWARD) starts with a model that contains all explanatory variables given in the MODEL statement. By specifying the FAST option, PROC LOGISTIC eliminates insignificant variables without refitting the model repeatedly. This analysis uses a significance level of 0.2 (SLSTAY=0.2) to retain variables in the model, which is different from the previous stepwise analysis where SLSTAY=.35. The CTABLE option is specified to produce classifications of input observations based on the final selected model.

Output 42.1.9. Initial Step in Backward Elimination

```
                 Backward Elimination on Cancer Remission Data

                             The LOGISTIC Procedure

                               Model Information

     Data Set                      WORK.REMISSION
     Response Variable             remiss               Complete Remission
     Number of Response Levels     2
     Model                         binary logit
     Optimization Technique        Fisher's scoring


                    Number of Observations Read          27
                    Number of Observations Used          27


                                Response Profile

                       Ordered                      Total
                         Value       remiss     Frequency

                             1            0            18
                             2            1             9

                        Probability modeled is remiss=1.


                         Backward Elimination Procedure

Step  0. The following effects were entered:

Intercept  temp  cell  li  smear  blast


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                                  Intercept
                                   Intercept            and
                     Criterion          Only     Covariates

                     AIC              36.372         33.857
                     SC               37.668         41.632
                     -2 Log L         34.372         21.857


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        12.5146        5         0.0284
            Score                    9.3295        5         0.0966
            Wald                     4.7284        5         0.4499
```

Output 42.1.10. Fast Elimination Step

```
Step  1. Fast Backward Elimination:


           Analysis of Effects Removed by Fast Backward Elimination

                                                                       Pr >
 Effect                                            Residual          Residual
 Removed    Chi-Square      DF    Pr > ChiSq     Chi-Square      DF      ChiSq

 blast          0.0008       1        0.9768         0.0008       1     0.9768
 smear          0.0951       1        0.7578         0.0959       2     0.9532
 cell           1.5134       1        0.2186         1.6094       3     0.6573
 temp           0.6535       1        0.4189         2.2628       4     0.6875



                          Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                            Model Fit Statistics

                                              Intercept
                                Intercept           and
                   Criterion         Only    Covariates

                   AIC             36.372        30.073
                   SC              37.668        32.665
                   -2 Log L        34.372        26.073


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         8.2988        1         0.0040
            Score                    7.9311        1         0.0049
            Wald                     5.9594        1         0.0146


                          Residual Chi-Square Test

                      Chi-Square       DF     Pr > ChiSq

                          2.8530        4         0.5827


                       Summary of Backward Elimination

                Effect               Number          Wald
          Step    Removed      DF          In    Chi-Square    Pr > ChiSq

             1    blast         1           4        0.0008        0.9768
             1    smear         1           3        0.0951        0.7578
             1    cell          1           2        1.5134        0.2186
             1    temp          1           1        0.6535        0.4189
```

Output 42.1.10. (continued)

```
                    Analysis of Maximum Likelihood Estimates

                                      Standard          Wald
       Parameter     DF     Estimate     Error    Chi-Square    Pr > ChiSq

       Intercept      1      -3.7771    1.3786        7.5064        0.0061
       li             1       2.8973    1.1868        5.9594        0.0146


          Association of Predicted Probabilities and Observed Responses

                Percent Concordant     84.0    Somers' D    0.710
                Percent Discordant     13.0    Gamma        0.732
                Percent Tied            3.1    Tau-a        0.328
                Pairs                   162    c            0.855
```

Results of the fast elimination analysis are shown in Output 42.1.9 and Output 42.1.10. Initially, a full model containing all six risk factors is fit to the data (Output 42.1.9). In the next step (Output 42.1.10), PROC LOGISTIC removes blast, smear, cell, and temp from the model all at once. This leaves li and the intercept as the only variables in the final model. Note that in this analysis, only parameter estimates for the final model are displayed because the DETAILS option has not been specified.

Note that you can also use the FAST option when SELECTION=STEPWISE. However, the FAST option operates only on backward elimination steps. In this example, the stepwise process only adds variables, so the FAST option would not be useful.

Output 42.1.11. Classifying Input Observations

```
                          Classification Table

          Correct      Incorrect                Percentages
 Prob          Non-          Non-           Sensi-  Speci-  False  False
Level  Event  Event  Event  Event  Correct  tivity  ficity   POS    NEG

0.060      9      0     18      0     33.3   100.0     0.0   66.7     .
0.080      9      2     16      0     40.7   100.0    11.1   64.0    0.0
0.100      9      4     14      0     48.1   100.0    22.2   60.9    0.0
0.120      9      4     14      0     48.1   100.0    22.2   60.9    0.0
0.140      9      7     11      0     59.3   100.0    38.9   55.0    0.0
0.160      9     10      8      0     70.4   100.0    55.6   47.1    0.0
0.180      9     10      8      0     70.4   100.0    55.6   47.1    0.0
0.200      8     13      5      1     77.8    88.9    72.2   38.5    7.1
0.220      8     13      5      1     77.8    88.9    72.2   38.5    7.1
0.240      8     13      5      1     77.8    88.9    72.2   38.5    7.1
0.260      6     13      5      3     70.4    66.7    72.2   45.5   18.8
0.280      6     13      5      3     70.4    66.7    72.2   45.5   18.8
0.300      6     13      5      3     70.4    66.7    72.2   45.5   18.8
0.320      6     14      4      3     74.1    66.7    77.8   40.0   17.6
0.340      5     14      4      4     70.4    55.6    77.8   44.4   22.2
0.360      5     14      4      4     70.4    55.6    77.8   44.4   22.2
0.380      5     15      3      4     74.1    55.6    83.3   37.5   21.1
0.400      5     15      3      4     74.1    55.6    83.3   37.5   21.1
0.420      5     15      3      4     74.1    55.6    83.3   37.5   21.1
0.440      5     15      3      4     74.1    55.6    83.3   37.5   21.1
0.460      4     16      2      5     74.1    44.4    88.9   33.3   23.8
0.480      4     16      2      5     74.1    44.4    88.9   33.3   23.8
0.500      4     16      2      5     74.1    44.4    88.9   33.3   23.8
0.520      4     16      2      5     74.1    44.4    88.9   33.3   23.8
0.540      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.560      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.580      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.600      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.620      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.640      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.660      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.680      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.700      3     16      2      6     70.4    33.3    88.9   40.0   27.3
0.720      2     16      2      7     66.7    22.2    88.9   50.0   30.4
0.740      2     16      2      7     66.7    22.2    88.9   50.0   30.4
0.760      2     16      2      7     66.7    22.2    88.9   50.0   30.4
0.780      2     16      2      7     66.7    22.2    88.9   50.0   30.4
0.800      2     17      1      7     70.4    22.2    94.4   33.3   29.2
0.820      2     17      1      7     70.4    22.2    94.4   33.3   29.2
0.840      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.860      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.880      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.900      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.920      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.940      0     17      1      9     63.0     0.0    94.4  100.0   34.6
0.960      0     18      0      9     66.7     0.0   100.0     .    33.3
```

Results of the CTABLE option are shown in Output 42.1.11. Each row of the “Classification Table” corresponds to a cutpoint applied to the predicted probabilities, which is given in the Prob Level column. The 2×2 frequency tables of observed and predicted responses are given by the next four columns. For example, with a cutpoint of 0.5, 4 events and 16 nonevents were classified correctly. On the other hand, 2 nonevents were incorrectly classified as events and 5 events were incorrectly classified as nonevents. For this cutpoint, the correct classification rate is 20/27 (=74.1%), which is given in the sixth column. Accuracy of the classification is summarized by the sensitivity, specificity, and false positive and negative rates, which are displayed in the last four columns. You can control the number of cutpoints used, and their values, by using the PPROB= option.

Example 42.2. Logistic Modeling with Categorical Predictors

Consider a study of the analgesic effects of treatments on elderly patients with neuralgia. Two test treatments and a placebo are compared. The response variable is whether the patient reported pain or not. Researchers recorded age and gender of the patients and the duration of complaint before the treatment began. The data, consisting of 60 patients, are contained in the data set Neuralgia.

```
Data Neuralgia;
   input Treatment $ Sex $ Age Duration Pain $ @@;
   datalines;
P  F  68   1  No   B  M  74  16  No  P  F  67  30  No
P  M  66  26  Yes  B  F  67  28  No  B  F  77  16  No
A  F  71  12  No   B  F  72  50  No  B  F  76   9  Yes
A  M  71  17  Yes  A  F  63  27  No  A  F  69  18  Yes
B  F  66  12  No   A  M  62  42  No  P  F  64   1  Yes
A  F  64  17  No   P  M  74   4  No  A  F  72  25  No
P  M  70   1  Yes  B  M  66  19  No  B  M  59  29  No
A  F  64  30  No   A  M  70  28  No  A  M  69   1  No
B  F  78   1  No   P  M  83   1  Yes B  F  69  42  No
B  M  75  30  Yes  P  M  77  29  Yes P  F  79  20  Yes
A  M  70  12  No   A  F  69  12  No  B  F  65  14  No
B  M  70   1  No   B  M  67  23  No  A  M  76  25  Yes
P  M  78  12  Yes  B  M  77   1  Yes B  F  69  24  No
P  M  66   4  Yes  P  F  65  29  No  P  M  60  26  Yes
A  M  78  15  Yes  B  M  75  21  Yes A  F  67  11  No
P  F  72  27  No   P  F  70  13  Yes A  M  75   6  Yes
B  F  65   7  No   P  F  68  27  Yes P  M  68  11  Yes
P  M  67  17  Yes  B  M  70  22  No  A  M  65  15  No
P  F  67   1  Yes  A  M  67  10  No  P  F  72  11  Yes
A  F  74   1  No   B  M  80  21  Yes A  F  69   3  No
;
```

The data set Neuralgia contains five variables: Treatment, Sex, Age, Duration, and Pain. The last variable, Pain, is the response variable. A specification of Pain=Yes indicates there was pain, and Pain=No indicates no pain. The variable Treatment is a categorical variable with three levels: A and B represent the two test treatments, and P represents the placebo treatment. The gender of the patients is given by the categorical variable Sex. The variable Age is the age of the patients, in years, when treatment began. The duration of complaint, in months, before the treatment began is given by the variable Duration. The following statements use the LOGISTIC procedure to fit a two-way logit with interaction model for the effect of Treatment and Sex, with Age and Duration as covariates. The categorical variables Treatment and Sex are declared in the CLASS statement.

```
proc logistic data=Neuralgia;
   class Treatment Sex;
   model Pain= Treatment Sex Treatment*Sex Age Duration / expb;
run;
```

In this analysis, PROC LOGISTIC models the probability of no pain (Pain=No). By default, effect coding is used to represent the CLASS variables. Two design variables are created for Treatment and one for Sex, as shown in Output 42.2.1.

Output 42.2.1. Effect Coding of CLASS Variables

```
                         The LOGISTIC Procedure

                        Class Level Information

                                           Design
                    Class         Value     Variables

                    Treatment     A           1      0
                                  B           0      1
                                  P          -1     -1

                    Sex           F           1
                                  M          -1
```

PROC LOGISTIC displays a table of the Type 3 analysis of effects based on the Wald test (Output 42.2.2). Note that the Treatment*Sex interaction and the duration of complaint are not statistically significant (p =0.9318 and p =0.8752, respectively). This indicates that there is no evidence that the treatments affect pain differently in men and women, and no evidence that the pain outcome is related to the duration of pain.

Output 42.2.2. Wald Tests of Individual Effects

```
                       Type 3 Analysis of Effects

                                              Wald
              Effect               DF    Chi-Square    Pr > ChiSq

              Treatment             2       11.9886        0.0025
              Sex                   1        5.3104        0.0212
              Treatment*Sex         2        0.1412        0.9318
              Age                   1        7.2744        0.0070
              Duration              1        0.0247        0.8752
```

Parameter estimates are displayed in Output 42.2.3. The Exp(Est) column contains the exponentiated parameter estimates requested with the EXPB option. These values may, but do not necessarily, represent odds ratios for the corresponding variables. For continuous explanatory variables, the Exp(Est) value corresponds to the odds ratio for a unit increase of the corresponding variable. For CLASS variables using the effect coding, the Exp(Est) values have no direct interpretation as a comparison of levels.

However, when the reference coding is used, the Exp(Est) values represent the odds ratio between the corresponding level and the last level. Following the parameter estimates table, PROC LOGISTIC displays the odds ratio estimates for those variables that are not involved in any interaction terms. If the variable is a CLASS variable, the odds ratio estimate comparing each level with the last level is computed regardless of the coding scheme. In this analysis, since the model contains the Treatment*Sex interaction term, the odds ratios for Treatment and Sex were not computed. The odds ratio estimates for Age and Duration are precisely the values given in the Exp(Est) column in the parameter estimates table.

Output 42.2.3. Parameter Estimates with Effect Coding

```
                    Analysis of Maximum Likelihood Estimates

                                Standard        Wald
Parameter          DF  Estimate    Error  Chi-Square  Pr > ChiSq  Exp(Est)

Intercept           1   19.2236   7.1315      7.2661      0.0070   2.232E8
Treatment     A     1    0.8483   0.5502      2.3773      0.1231     2.336
Treatment     B     1    1.4949   0.6622      5.0956      0.0240     4.459
Sex           F     1    0.9173   0.3981      5.3104      0.0212     2.503
Treatment*Sex A F   1   -0.2010   0.5568      0.1304      0.7180     0.818
Treatment*Sex B F   1    0.0487   0.5563      0.0077      0.9302     1.050
Age                 1   -0.2688   0.0996      7.2744      0.0070     0.764
Duration            1   0.00523   0.0333      0.0247      0.8752     1.005


                           Odds Ratio Estimates

                                  Point          95% Wald
            Effect             Estimate      Confidence Limits

            Age                   0.764       0.629       0.929
            Duration              1.005       0.942       1.073
```

The following PROC LOGISTIC statements illustrate the use of forward selection on the data set Neuralgia to identify the effects that differentiate the two Pain responses. The option SELECTION=FORWARD is specified to carry out the forward selection. The term Treatment|Sex@2 illustrates another way to specify main effects and two-way interaction as is available in other procedures such as PROC GLM. (Note that, in this case, the “@2” is unnecessary because no interactions besides the two-way interaction are possible).

```
proc logistic data=Neuralgia;
   class Treatment Sex;
   model Pain=Treatment|Sex@2 Age Duration
         /selection=forward expb;
run;
```

Results of the forward selection process are summarized in Output 42.2.4. The variable Treatment is selected first, followed by Age and then Sex. The results are consistent with the previous analysis (Output 42.2.2) in which the Treatment*Sex interaction and Duration are not statistically significant.

Output 42.2.4. Effects Selected into the Model

```
                              The LOGISTIC Procedure

                           Summary of Forward Selection

             Effect                            Number         Score
      Step   Entered              DF               In    Chi-Square    Pr > ChiSq

         1   Treatment             2                1       13.7143        0.0011
         2   Age                   1                2       10.6038        0.0011
         3   Sex                   1                3        5.9959        0.0143
```

Output 42.2.5 shows the Type 3 analysis of effects, the parameter estimates, and the odds ratio estimates for the selected model. All three variables, Treatment, Age, and Sex, are statistically significant at the 0.05 level (p =0.0011, p =0.0011, and p =0.0143, respectively). Since the selected model does not contain the Treatment*Sex interaction, odds ratios for Treatment and Sex are computed. The estimated odds ratio is 24.022 for treatment A versus placebo, 41.528 for Treatment B versus placebo, and 6.194 for female patients versus male patients. Note that these odds ratio estimates are not the same as the corresponding values in the Exp(Est) column in the parameter estimates table because effect coding was used. From Output 42.2.5, it is evident that both Treatment A and Treatment B are better than the placebo in reducing pain; females tend to have better improvement than males; and younger patients are faring better than older patients.

Output 42.2.5. Type 3 Effects and Parameter Estimates with Effect Coding

```
                           Type 3 Analysis of Effects

                                         Wald
                Effect             DF    Chi-Square    Pr > ChiSq

                Treatment           2       12.6928        0.0018
                Sex                 1        5.3013        0.0213
                Age                 1        7.6314        0.0057


                    Analysis of Maximum Likelihood Estimates

                                      Standard          Wald
Parameter          DF    Estimate        Error    Chi-Square    Pr > ChiSq    Exp(Est)

Intercept           1     19.0804       6.7882        7.9007        0.0049    1.9343E8
Treatment     A     1      0.8772       0.5274        2.7662        0.0963       2.404
Treatment     B     1      1.4246       0.6036        5.5711        0.0183       4.156
Sex           F     1      0.9118       0.3960        5.3013        0.0213       2.489
Age                 1     -0.2650       0.0959        7.6314        0.0057       0.767


                               Odds Ratio Estimates

                                      Point          95% Wald
               Effect              Estimate      Confidence Limits

               Treatment A vs P      24.022       3.295     175.121
               Treatment B vs P      41.528       4.500     383.262
               Sex       F vs M       6.194       1.312      29.248
               Age                    0.767       0.636       0.926
```

Finally, PROC LOGISTIC is invoked to refit the previously selected model using reference coding for the CLASS variables. Two CONTRAST statements are specified. The one labeled 'Pairwise' specifies three rows in the contrast matrix, L, for all the pairwise comparisons between the three levels of Treatment. The contrast labeled 'Female vs Male' compares female to male patients. The option ESTIMATE=EXP is specified in both CONTRAST statements to exponentiate the estimates of $\mathbf{L}'\beta$. With the given specification of contrast coefficients, the first row of the 'Pairwise' CONTRAST statement corresponds to the odds ratio of A versus P, the second row corresponds to B versus P, and the third row corresponds to A versus B. There is only one row in the 'Female vs Male' CONTRAST statement, and it corresponds to the odds ratio comparing female to male patients.

```
proc logistic data=Neuralgia;
   class Treatment Sex /param=ref;
   model Pain= Treatment Sex age;
   contrast 'Pairwise' Treatment 1  0,
                       Treatment 0  1,
                       Treatment 1 -1 / estimate=exp;
   contrast 'Female vs Male' Sex 1 / estimate=exp;
run;
```

Output 42.2.6. Reference Coding of CLASS Variables

```
      The LOGISTIC Procedure

     Class Level Information

                          Design
Class         Value     Variables

Treatment     A          1      0
              B          0      1
              P          0      0

Sex           F          1
              M          0
```

The reference coding is shown in Output 42.2.6. The Type 3 analysis of effects, the parameter estimates for the reference coding, and the odds ratio estimates are displayed in Output 42.2.7. Although the parameter estimates are different (because of the different parameterizations), the “Type 3 Analysis of Effects” table and the “Odds Ratio” table remain the same as in Output 42.2.5. With effect coding, the treatment A parameter estimate (0.8772) estimates the effect of treatment A compared to the average effect of treatments A, B, and placebo. The treatment A estimate (3.1790) under the reference coding estimates the difference in effect of treatment A and the placebo treatment.

Output 42.2.7. Type 3 Effects and Parameter Estimates with Reference Coding

```
                         Type 3 Analysis of Effects

                                            Wald
                  Effect           DF    Chi-Square    Pr > ChiSq

                  Treatment         2       12.6928        0.0018
                  Sex               1        5.3013        0.0213
                  Age               1        7.6314        0.0057


                    Analysis of Maximum Likelihood Estimates

                                            Standard          Wald
      Parameter       DF     Estimate         Error    Chi-Square    Pr > ChiSq

      Intercept        1      15.8669        6.4056        6.1357        0.0132
      Treatment A      1       3.1790        1.0135        9.8375        0.0017
      Treatment B      1       3.7264        1.1339       10.8006        0.0010
      Sex       F      1       1.8235        0.7920        5.3013        0.0213
      Age              1      -0.2650        0.0959        7.6314        0.0057


                               Odds Ratio Estimates

                                      Point          95% Wald
               Effect              Estimate      Confidence Limits

               Treatment A vs P      24.022        3.295     175.121
               Treatment B vs P      41.528        4.500     383.262
               Sex       F vs M       6.194        1.312      29.248
               Age                    0.767        0.636       0.926
```

Output 42.2.8 contains two tables: the “Contrast Test Results” table and the “Contrast Rows Estimation and Testing Results” table. The former contains the overall Wald test for each CONTRAST statement. Although three rows are specified in the ’Pairwise’ CONTRAST statement, there are only two degrees of freedom, and the Wald test result is identical to the Type 3 analysis of Treatment in Output 42.2.7. The latter table contains estimates and tests of individual contrast rows. The estimates for the first two rows of the ’Pairwise’ CONTRAST statement are the same as those given in the “Odds Ratio Estimates” table (in Output 42.2.7). Both treatments A and B are highly effective over placebo in reducing pain. The third row estimates the odds ratio comparing A to B. The 95% confidence interval for this odds ratio is (0.0932, 3.5889), indicating that the pain reduction effects of these two test treatments are not that different. Again, the ’Female vs Male’ contrast shows that female patients fared better in obtaining relief from pain than male patients.

Output 42.2.8. Results of CONTRAST Statements

```
                              Contrast Test Results

                                                  Wald
               Contrast            DF     Chi-Square    Pr > ChiSq

               Pairwise             2        12.6928        0.0018
               Female vs Male       1         5.3013        0.0213


                  Contrast Rows Estimation and Testing Results

                                                Standard
 Contrast         Type        Row   Estimate       Error    Alpha   Confidence Limits

 Pairwise         EXP           1    24.0218     24.3473     0.05     3.2951     175.1
 Pairwise         EXP           2    41.5284     47.0877     0.05     4.4998     383.3
 Pairwise         EXP           3     0.5784      0.5387     0.05     0.0932    3.5889
 Female vs Male   EXP           1     6.1937      4.9053     0.05     1.3116   29.2476

                  Contrast Rows Estimation and Testing Results

                                                   Wald
            Contrast         Type        Row   Chi-Square    Pr > ChiSq

            Pairwise         EXP           1       9.8375        0.0017
            Pairwise         EXP           2      10.8006        0.0010
            Pairwise         EXP           3       0.3455        0.5567
            Female vs Male   EXP           1       5.3013        0.0213
```

Example 42.3. Ordinal Logistic Regression

Consider a study of the effects on taste of various cheese additives. Researchers tested four cheese additives and obtained 52 response ratings for each additive. Each response was measured on a scale of nine categories ranging from strong dislike (1) to excellent taste (9). The data, given in McCullagh and Nelder (1989, p. 175) in the form of a two-way frequency table of additive by rating, are saved in the data set Cheese.

```
data Cheese;
   do Additive = 1 to 4;
      do y = 1 to 9;
         input freq @@;
         output;
      end;
   end;
   label y='Taste Rating';
   datalines;
0  0  1  7  8  8 19  8  1
6  9 12 11  7  6  1  0  0
1  1  6  8 23  7  5  1  0
0  0  0  1  3  7 14 16 11
;
```

The data set Cheese contains the variables y, Additive, and freq. The variable y contains the response rating. The variable Additive specifies the cheese additive (1, 2, 3, or 4). The variable freq gives the frequency with which each additive received each rating.

The response variable y is ordinally scaled. A cumulative logit model is used to investigate the effects of the cheese additives on taste. The following SAS statements invoke PROC LOGISTIC to fit this model with y as the response variable and three indicator variables as explanatory variables, with the fourth additive as the reference level. With this parameterization, each Additive parameter compares an additive to the fourth additive. The COVB option produces the estimated covariance matrix.

```
proc logistic data=Cheese;
   freq freq;
   class Additive (param=ref ref='4');
   model y=Additive / covb;
   title1 'Multiple Response Cheese Tasting Experiment';
run;
```

Results of the analysis are shown in Output 42.3.1, and the estimated covariance matrix is displayed in Output 42.3.2.

Since the strong dislike (y=1) end of the rating scale is associated with lower Ordered Values in the Response Profile table, the probability of disliking the additives is modeled.

The score chi-square for testing the proportional odds assumption is 17.287, which is not significant with respect to a chi-square distribution with 21 degrees of freedom $(p = 0.694)$. This indicates that the proportional odds model adequately fits the data. The positive value (1.6128) for the parameter estimate for Additive1 indicates a tendency towards the lower-numbered categories of the first cheese additive relative to the fourth. In other words, the fourth additive is better in taste than the first additive. Each of the second and the third additives is less favorable than the fourth additive. The relative magnitudes of these slope estimates imply the preference ordering: fourth, first, third, second.

Output 42.3.1. Proportional Odds Model Regression Analysis

```
           Multiple Response Cheese Tasting Experiment

                      The LOGISTIC Procedure

                        Model Information

 Data Set                      WORK.CHEESE
 Response Variable             y                    Taste Rating
 Number of Response Levels     9
 Frequency Variable            freq
 Model                         cumulative logit
 Optimization Technique        Fisher’s scoring


             Number of Observations Read          36
             Number of Observations Used          28
             Sum of Frequencies Read             208
             Sum of Frequencies Used             208


                         Response Profile

                Ordered                      Total
                  Value            y     Frequency

                      1            1             7
                      2            2            10
                      3            3            19
                      4            4            27
                      5            5            41
                      6            6            28
                      7            7            39
                      8            8            25
                      9            9            12

Probabilities modeled are cumulated over the lower Ordered Values.


                     Model Convergence Status

          Convergence criterion (GCONV=1E-8) satisfied.


          Score Test for the Proportional Odds Assumption

                Chi-Square       DF     Pr > ChiSq

                   17.2866       21         0.6936
```

Output 42.3.1. (continued)

```
                  Multiple Response Cheese Tasting Experiment

                              Model Fit Statistics

                                                   Intercept
                                      Intercept          and
                     Criterion             Only   Covariates

                     AIC                875.802      733.348
                     SC                 902.502      770.061
                     -2 Log L           859.802      711.348


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio       148.4539        3         <.0001
            Score                  111.2670        3         <.0001
            Wald                   115.1504        3         <.0001


                   Analysis of Maximum Likelihood Estimates

                                         Standard          Wald
     Parameter      DF    Estimate          Error    Chi-Square    Pr > ChiSq

     Intercept 1     1     -7.0801         0.5624      158.4851        <.0001
     Intercept 2     1     -6.0249         0.4755      160.5500        <.0001
     Intercept 3     1     -4.9254         0.4272      132.9484        <.0001
     Intercept 4     1     -3.8568         0.3902       97.7087        <.0001
     Intercept 5     1     -2.5205         0.3431       53.9704        <.0001
     Intercept 6     1     -1.5685         0.3086       25.8374        <.0001
     Intercept 7     1     -0.0669         0.2658        0.0633        0.8013
     Intercept 8     1      1.4930         0.3310       20.3439        <.0001
     Additive  1     1      1.6128         0.3778       18.2265        <.0001
     Additive  2     1      4.9645         0.4741      109.6427        <.0001
     Additive  3     1      3.3227         0.4251       61.0931        <.0001


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     67.6    Somers' D    0.578
               Percent Discordant      9.8    Gamma        0.746
               Percent Tied           22.6    Tau-a        0.500
               Pairs                 18635    c            0.789
```

Output 42.3.2. Estimated Covariance Matrix

```
                   Multiple Response Cheese Tasting Experiment

                           Estimated Covariance Matrix

              Intercept_   Intercept_   Intercept_   Intercept_   Intercept_
 Parameter             1            2            3            4            5

 Intercept_1    0.316291     0.219581     0.176278     0.147694     0.114024
 Intercept_2    0.219581     0.226095     0.177806     0.147933      0.11403
 Intercept_3    0.176278     0.177806     0.182473     0.148844     0.114092
 Intercept_4    0.147694     0.147933     0.148844     0.152235     0.114512
 Intercept_5    0.114024      0.11403     0.114092     0.114512     0.117713
 Intercept_6    0.091085     0.091081     0.091074     0.091109     0.091821
 Intercept_7    0.057814     0.057813     0.057807      0.05778     0.057721
 Intercept_8    0.041304     0.041304       0.0413     0.041277     0.041162
 Additive1      -0.09419     -0.09421     -0.09427     -0.09428     -0.09246
 Additive2      -0.18686     -0.18161      -0.1687     -0.14717     -0.11415
 Additive3      -0.13565     -0.13569      -0.1352     -0.13118     -0.11207

                           Estimated Covariance Matrix

             Intercept_  Intercept_  Intercept_
Parameter             6           7           8  Additive1  Additive2  Additive3

Intercept_1    0.091085    0.057814    0.041304   -0.09419   -0.18686   -0.13565
Intercept_2    0.091081    0.057813    0.041304   -0.09421   -0.18161   -0.13569
Intercept_3    0.091074    0.057807      0.0413   -0.09427    -0.1687    -0.1352
Intercept_4    0.091109     0.05778    0.041277   -0.09428   -0.14717   -0.13118
Intercept_5    0.091821    0.057721    0.041162   -0.09246   -0.11415   -0.11207
Intercept_6     0.09522    0.058312    0.041324   -0.08521   -0.09113   -0.09122
Intercept_7    0.058312     0.07064     0.04878   -0.06041   -0.05781   -0.05802
Intercept_8    0.041324     0.04878    0.109562   -0.04436    -0.0413   -0.04143
Additive1      -0.08521    -0.06041    -0.04436   0.142715   0.094072   0.092128
Additive2      -0.09113    -0.05781     -0.0413   0.094072    0.22479   0.132877
Additive3      -0.09122    -0.05802    -0.04143   0.092128   0.132877   0.180709
```

Example 42.4. Nominal Response Data: Generalized Logits Model

Over the course of one school year, third graders from three different schools are exposed to three different styles of mathematics instruction: a self-paced computer-learning style, a team approach, and a traditional class approach. The students are asked which style they prefer and their responses, classified by the type of program they are in (a regular school day versus a regular day supplemented with an afternoon school program) are displayed in Table 42.4. The data set is from Stokes, Davis, and Koch (2000), and is also analyzed in the "Generalized Logits Model" section on page 824 of Chapter 22, "The CATMOD Procedure."

Table 42.4. School Program Data

		Learning Style Preference		
School	**Program**	**Self**	**Team**	**Class**
1	Regular	10	17	26
1	Afternoon	5	12	50
2	Regular	21	17	26
2	Afternoon	16	12	36
3	Regular	15	15	16
3	Afternoon	12	12	20

The levels of the response variable (self, team, and class) have no essential ordering, so a logistic regression is performed on the generalized logits. The model to be fit is

$$\log\left(\frac{\pi_{hij}}{\pi_{hir}}\right) = \alpha_j + \mathbf{x}'_{hi}\boldsymbol{\beta}_j$$

where π_{hij} is the probability that a student in school h and program i prefers teaching style j, $j \neq r$, and style r is the baseline style (in this case, class). There are separate sets of intercept parameters α_j and regression parameters $\boldsymbol{\beta}_j$ for each logit, and the matrix $\mathbf{x}_{hi}$ is the set of explanatory variables for the hith population. Thus, two logits are modeled for each school and program combination: the logit comparing self to class and the logit comparing team to class.

The following statements create the data set school and request the analysis. The LINK=GLOGIT option forms the generalized logits. The response variable option ORDER=DATA means that the response variable levels are ordered as they exist in the data set: self, team, and class; thus, the logits are formed by comparing self to class and by comparing team to class. The ODS statement suppresses the display of the maximum likelihood estimates. The results of this analysis are shown in Output 42.4.1 through Output 42.4.4.

```
data school;
   length Program $ 9;
   input School Program $ Style $ Count @@;
   datalines;
1 regular   self 10  1 regular   team 17  1 regular   class 26
1 afternoon self  5  1 afternoon team 12  1 afternoon class 50
2 regular   self 21  2 regular   team 17  2 regular   class 26
2 afternoon self 16  2 afternoon team 12  2 afternoon class 36
3 regular   self 15  3 regular   team 15  3 regular   class 16
3 afternoon self 12  3 afternoon team 12  3 afternoon class 20
;
```

```
proc logistic data=school;
   freq Count;
   class School Program(ref=first);
   model Style(order=data)=School Program School*Program
         / link=glogit;
run;
```

Output 42.4.1. Analysis of Saturated Model

```
                   The LOGISTIC Procedure

                     Model Information

      Data Set                      WORK.SCHOOL
      Response Variable             Style
      Number of Response Levels     3
      Frequency Variable            Count
      Model                         generalized logit
      Optimization Technique        Fisher's scoring


          Number of Observations Read          18
          Number of Observations Used          18
          Sum of Frequencies Read             338
          Sum of Frequencies Used             338


                      Response Profile

             Ordered                      Total
               Value     Style        Frequency

                   1     self                79
                   2     team                85
                   3     class              174

Logits modeled use Style='class' as the reference category.


                  Class Level Information

                                         Design
            Class       Value          Variables

            School      1               1      0
                        2               0      1
                        3              -1     -1

            Program     afternoon      -1
                        regular         1
```

Output 42.4.2. Fit Statistics

```
                          Model Convergence Status

               Convergence criterion (GCONV=1E-8) satisfied.


                            Model Fit Statistics

                                                      Intercept
                                       Intercept            and
                  Criterion                 Only     Covariates

                  AIC                    699.404        689.156
                  SC                     707.050        735.033
                  -2 Log L               695.404        665.156
```

Output 42.4.3. Tests

```
                Testing Global Null Hypothesis: BETA=0

         Test                 Chi-Square       DF     Pr > ChiSq

         Likelihood Ratio        30.2480       10         0.0008
         Score                   28.3738       10         0.0016
         Wald                    25.6828       10         0.0042


                       Type 3 Analysis of Effects

                                               Wald
            Effect                DF     Chi-Square    Pr > ChiSq

            School                 4        14.5522        0.0057
            Program                2        10.4815        0.0053
            School*Program         4         1.7439        0.7827
```

Output 42.4.4. Estimates

```
                            Analysis of Maximum Likelihood Estimates

                                                            Standard          Wald
Parameter                       Style     DF    Estimate       Error    Chi-Square    Pr > ChiSq

Intercept                       self       1     -0.8097      0.1488       29.5989        <.0001
Intercept                       team       1     -0.6585      0.1366       23.2449        <.0001
School         1                self       1     -0.8194      0.2281       12.9066        0.0003
School         1                team       1     -0.2675      0.1881        2.0233        0.1549
School         2                self       1      0.2974      0.1919        2.4007        0.1213
School         2                team       1     -0.1033      0.1898        0.2961        0.5863
Program        regular          self       1      0.3985      0.1488        7.1684        0.0074
Program        regular          team       1      0.3537      0.1366        6.7071        0.0096
School*Program 1         regular   self       1      0.2751      0.2281        1.4547        0.2278
School*Program 1         regular   team       1      0.1474      0.1881        0.6143        0.4332
School*Program 2         regular   self       1     -0.0998      0.1919        0.2702        0.6032
School*Program 2         regular   team       1     -0.0168      0.1898        0.0079        0.9293
```

The “Type 3 Analysis of Effects” table in Output 42.4.3 shows that the interaction effect is clearly nonsignificant, so a main effects model is fit with the following statements.

```
proc logistic data=school;
   freq Count;
   class School Program(ref=first);
   model Style(order=data)=School Program / link=glogit;
run;
```

Output 42.4.5. Analysis of Main Effects Model

```
                             The LOGISTIC Procedure

                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                                  Intercept
                                   Intercept            and
                     Criterion          Only     Covariates

                     AIC             699.404        682.934
                     SC              707.050        713.518
                     -2 Log L        695.404        666.934


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        28.4704        6         <.0001
            Score                   27.1190        6         0.0001
            Wald                    25.5881        6         0.0003


                          Type 3 Analysis of Effects

                                           Wald
                  Effect       DF    Chi-Square    Pr > ChiSq

                  School        4       14.8424        0.0050
                  Program       2       10.9160        0.0043
```

All of the global fit tests in Output 42.4.5 suggest the model is significant, and the Type 3 tests show that the school and program effects are also significant.

Output 42.4.6. Estimates

```
                              Analysis of Maximum Likelihood Estimates

                                                             Standard          Wald
      Parameter                 Style     DF     Estimate       Error    Chi-Square    Pr > ChiSq

      Intercept                 self       1      -0.7978      0.1465       29.6502        <.0001
      Intercept                 team       1      -0.6589      0.1367       23.2300        <.0001
      School     1              self       1      -0.7992      0.2198       13.2241        0.0003
      School     1              team       1      -0.2786      0.1867        2.2269        0.1356
      School     2              self       1       0.2836      0.1899        2.2316        0.1352
      School     2              team       1      -0.0985      0.1892        0.2708        0.6028
      Program    regular        self       1       0.3737      0.1410        7.0272        0.0080
      Program    regular        team       1       0.3713      0.1353        7.5332        0.0061


                                       Odds Ratio Estimates

                                                              Point          95% Wald
           Effect                                 Style     Estimate      Confidence Limits

           School  1 vs 3                         self         0.269         0.127        0.570
           School  1 vs 3                         team         0.519         0.267        1.010
           School  2 vs 3                         self         0.793         0.413        1.522
           School  2 vs 3                         team         0.622         0.317        1.219
           Program regular    vs afternoon        self         2.112         1.215        3.670
           Program regular    vs afternoon        team         2.101         1.237        3.571
```

The parameter estimates, tests for individual parameters, and odds ratios are displayed in Output 42.4.6. The Program variable has nearly the same effect on both logits, while School=1 has the largest effect of the schools.

Example 42.5. Stratified Sampling

Consider the hypothetical example in Fleiss (1981, pp. 6–7) in which a test is applied to a sample of 1,000 people known to have a disease and to another sample of 1,000 people known not to have the same disease. In the diseased sample, 950 test positive; in the nondiseased sample, only 10 test positive. If the true disease rate in the population is 1 in 100, specifying PEVENT=0.01 results in the correct false positive and negative rates for the stratified sampling scheme. Omitting the PEVENT= option is equivalent to using the overall sample disease rate (1000/2000 = 0.5) as the value of the PEVENT= option, which would ignore the stratified sampling.

The SAS code is as follows:

```
data Screen;
   do Disease='Present','Absent';
      do Test=1,0;
         input Count @@;
         output;
      end;
   end;
   datalines;
950  50
 10 990
;
```

```
proc logistic data=Screen;
   freq Count;
   model Disease(event='Present')=Test
         / pevent=.5 .01 ctable pprob=.5;
run;
```

The response variable option EVENT= indicates that Disease='Present' is the event. The CTABLE option is specified to produce a classification table. Specifying PPROB=0.5 indicates a cutoff probability of 0.5. A list of two probabilities, 0.5 and 0.01, is specified for the PEVENT= option; 0.5 corresponds to the overall sample disease rate, and 0.01 corresponds to a true disease rate of 1 in 100.

The classification table is shown in Output 42.5.1.

Output 42.5.1. False Positive and False Negative Rates

```
                                The LOGISTIC Procedure

                                 Classification Table

                   Correct      Incorrect             Percentages
 Prob   Prob             Non-           Non-          Sensi-  Speci-  False  False
Event  Level   Event    Event   Event  Event  Correct tivity  ficity   POS    NEG

0.500  0.500     950      990      10     50     97.0   95.0    99.0    1.0    4.8

0.010  0.500     950      990      10     50     99.0   95.0    99.0   51.0    0.1
```

In the classification table, the column “Prob Level” represents the cutoff values (the settings of the PPROB= option) for predicting whether an observation is an event. The “Correct” columns list the numbers of subjects that are correctly predicted as events and nonevents, respectively, and the “Incorrect” columns list the number of nonevents incorrectly predicted as events and the number of events incorrectly predicted as nonevents, respectively. For PEVENT=0.5, the false positive rate is 1% and the false negative rate is 4.8%. These results ignore the fact that the samples were stratified and incorrectly assume that the overall sample proportion of disease (which is 0.5) estimates the true disease rate. For a true disease rate of 0.01, the false positive rate and the false negative rate are 51% and 0.1%, respectively, as shown on the second line of the classification table.

Example 42.6. Logistic Regression Diagnostics

In a controlled experiment to study the effect of the rate and volume of air inspired on a transient reflex vaso-constriction in the skin of the digits, 39 tests under various combinations of rate and volume of air inspired were obtained (Finney 1947). The end point of each test is whether or not vaso-constriction occurred. Pregibon (1981) uses this set of data to illustrate the diagnostic measures he proposes for detecting influential observations and to quantify their effects on various aspects of the maximum likelihood fit.

The vaso-constriction data are saved in the data set vaso:

```
data vaso;
   length Response $12;
   input Volume Rate Response @@;
   LogVolume=log(Volume);
   LogRate=log(Rate);
   datalines;
3.70  0.825  constrict       3.50  1.09   constrict
1.25  2.50   constrict       0.75  1.50   constrict
0.80  3.20   constrict       0.70  3.50   constrict
0.60  0.75   no_constrict    1.10  1.70   no_constrict
0.90  0.75   no_constrict    0.90  0.45   no_constrict
0.80  0.57   no_constrict    0.55  2.75   no_constrict
0.60  3.00   no_constrict    1.40  2.33   constrict
0.75  3.75   constrict       2.30  1.64   constrict
3.20  1.60   constrict       0.85  1.415  constrict
1.70  1.06   no_constrict    1.80  1.80   constrict
0.40  2.00   no_constrict    0.95  1.36   no_constrict
1.35  1.35   no_constrict    1.50  1.36   no_constrict
1.60  1.78   constrict       0.60  1.50   no_constrict
1.80  1.50   constrict       0.95  1.90   no_constrict
1.90  0.95   constrict       1.60  0.40   no_constrict
2.70  0.75   constrict       2.35  0.03   no_constrict
1.10  1.83   no_constrict    1.10  2.20   constrict
1.20  2.00   constrict       0.80  3.33   constrict
0.95  1.90   no_constrict    0.75  1.90   no_constrict
1.30  1.625  constrict
;
```

In the data set vaso, the variable Response represents the outcome of a test. The variable LogVolume represents the log of the volume of air intake, and the variable LogRate represents the log of the rate of air intake.

The following SAS statements invoke PROC LOGISTIC to fit a logistic regression model to the vaso-constriction data, where Response is the response variable, and LogRate and LogVolume are the explanatory variables. The INFLUENCE option and the IPLOTS option are specified to display the regression diagnostics and the index plots.

```
ods html;
ods graphics on;

title ’Occurrence of Vaso-Constriction’;
proc logistic data=vaso;
   model Response=LogRate LogVolume/influence iplots;
run;

ods graphics off;
ods html close;
```

Results of the model fit are shown in Output 42.6.1. Both LogRate and LogVolume are statistically significant to the occurrence of vaso-constriction ($p = 0.0131$ and $p = 0.0055$, respectively). Their positive parameter estimates indicate that a higher inspiration rate or a larger volume of air intake is likely to increase the probability of vaso-constriction.

Output 42.6.1. Logistic Regression Analysis for Vaso-Constriction Data

```
                 Occurrence of Vaso-Constriction

                      The LOGISTIC Procedure

                        Model Information

          Data Set                      WORK.VASO
          Response Variable             Response
          Number of Response Levels     2
          Model                         binary logit
          Optimization Technique        Fisher's scoring


             Number of Observations Read          39
             Number of Observations Used          39


                         Response Profile

              Ordered                          Total
                Value     Response         Frequency

                    1     constrict               20
                    2     no_constrict            19

           Probability modeled is Response='constrict'.


                     Model Convergence Status

          Convergence criterion (GCONV=1E-8) satisfied.
```

Output 42.6.1. (continued)

```
                        Occurrence of Vaso-Constriction

                              Model Fit Statistics

                                                     Intercept
                                       Intercept           and
                     Criterion              Only    Covariates

                     AIC                  56.040        35.227
                     SC                   57.703        40.218
                     -2 Log L             54.040        29.227


                    Testing Global Null Hypothesis: BETA=0

             Test                 Chi-Square       DF     Pr > ChiSq

             Likelihood Ratio        24.8125        2         <.0001
             Score                   16.6324        2         0.0002
             Wald                     7.8876        2         0.0194


                   Analysis of Maximum Likelihood Estimates

                                        Standard          Wald
       Parameter    DF     Estimate        Error    Chi-Square    Pr > ChiSq

       Intercept     1      -2.8754       1.3208        4.7395        0.0295
       LogRate       1       4.5617       1.8380        6.1597        0.0131
       LogVolume     1       5.1793       1.8648        7.7136        0.0055


          Association of Predicted Probabilities and Observed Responses

                Percent Concordant     93.7    Somers' D    0.874
                Percent Discordant      6.3    Gamma        0.874
                Percent Tied            0.0    Tau-a        0.448
                Pairs                   380    c            0.937
```

The INFLUENCE option displays the values of the explanatory variables (LogRate and LogVolume) for each observation, a column for each diagnostic produced, and the *case number* which represents the sequence number of the observation (Output 42.6.2). Also produced (but not shown here) is a lineprinter plot where the vertical axis represents the case number and the horizontal axis represents the value of the diagnostic statistic.

The index plots produced by the IPLOTS option are essentially the same lineprinter plots as those produced by the INFLUENCE option with a 90-degree rotation and perhaps on a more refined scale. This version of the plots are not displayed here. The vertical axis of an index plot represents the value of the diagnostic and the horizontal axis represents the sequence (case number) of the observation. The index plots are useful for identification of extreme values.

Since the experimental ODS GRAPHICS statement is also specified, the lineprinter plots from the INFLUENCE and IPLOTS options are suppressed and graphical displays are produced as shown in Output 42.6.3 through Output 42.6.5. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the LOGISTIC procedure, see the “ODS Graphics” section on page 2388.

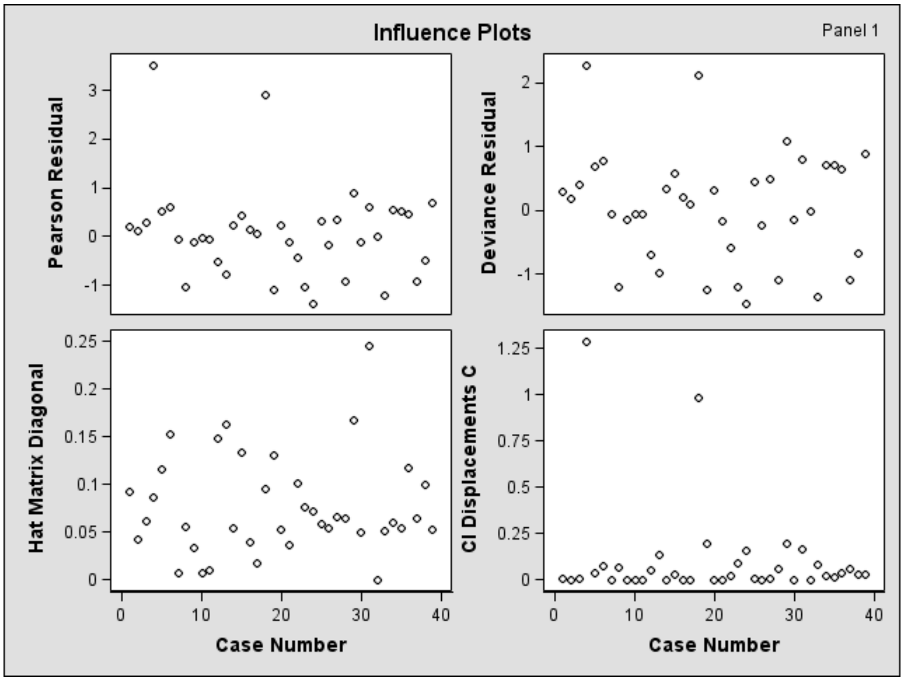
Output 42.6.2. Regression Diagnostics from the INFLUENCE Option (Experimental)

```
                           The LOGISTIC Procedure

                           Regression Diagnostics

            Covariates
                                                   Hat
  Case                 Log   Pearson  Deviance    Matrix  Intercept   LogRate
Number   LogRate    Volume  Residual  Residual  Diagonal     DfBeta    DfBeta

     1   -0.1924    1.3083    0.2205    0.3082    0.0927    -0.0165    0.0193
     2    0.0862    1.2528    0.1349    0.1899    0.0429    -0.0134    0.0151
     3    0.9163    0.2231    0.2923    0.4049    0.0612    -0.0492    0.0660
     4    0.4055   -0.2877    3.5181    2.2775    0.0867     1.0734   -0.9302
     5    1.1632   -0.2231    0.5287    0.7021    0.1158    -0.0832    0.1411
     6    1.2528   -0.3567    0.6090    0.7943    0.1524    -0.0922    0.1710
     7   -0.2877   -0.5108   -0.0328   -0.0464   0.00761   -0.00280   0.00274
     8    0.5306    0.0953   -1.0196   -1.1939    0.0559    -0.1444    0.0613
     9   -0.2877   -0.1054   -0.0938   -0.1323    0.0342    -0.0178    0.0173
    10   -0.7985   -0.1054   -0.0293   -0.0414   0.00721   -0.00245   0.00246
    11   -0.5621   -0.2231   -0.0370   -0.0523   0.00969   -0.00361   0.00358
    12    1.0116   -0.5978   -0.5073   -0.6768    0.1481    -0.1173    0.0647
    13    1.0986   -0.5108   -0.7751   -0.9700    0.1628    -0.0931  -0.00946
    14    0.8459    0.3365    0.2559    0.3562    0.0551    -0.0414    0.0538
    15    1.3218   -0.2877    0.4352    0.5890    0.1336    -0.0940    0.1408
    16    0.4947    0.8329    0.1576    0.2215    0.0402    -0.0198    0.0234
    17    0.4700    1.1632    0.0709    0.1001    0.0172   -0.00630   0.00701
    18    0.3471   -0.1625    2.9062    2.1192    0.0954     0.9595   -0.8279
    19    0.0583    0.5306   -1.0718   -1.2368    0.1315    -0.2591    0.2024
    20    0.5878    0.5878    0.2405    0.3353    0.0525    -0.0331    0.0421
    21    0.6931   -0.9163   -0.1076   -0.1517    0.0373    -0.0180    0.0158
    22    0.3075   -0.0513   -0.4193   -0.5691    0.1015    -0.1449    0.1237
    23    0.3001    0.3001   -1.0242   -1.1978    0.0761    -0.1961    0.1275
    24    0.3075    0.4055   -1.3684   -1.4527    0.0717    -0.1281    0.0410
    25    0.5766    0.4700    0.3347    0.4608    0.0587    -0.0403    0.0570
    26    0.4055   -0.5108   -0.1595   -0.2241    0.0548    -0.0366    0.0329
    27    0.4055    0.5878    0.3645    0.4995    0.0661    -0.0327    0.0496
    28    0.6419   -0.0513   -0.8989   -1.0883    0.0647    -0.1423    0.0617
    29   -0.0513    0.6419    0.8981    1.0876    0.1682     0.2367   -0.1950
    30   -0.9163    0.4700   -0.0992   -0.1400    0.0507    -0.0224    0.0227
    31   -0.2877    0.9933    0.6198    0.8064    0.2459     0.1165   -0.0996
    32   -3.5066    0.8544  -0.00073  -0.00103  0.000022    -3.22E-6  3.405E-6
    33    0.6043    0.0953   -1.2062   -1.3402    0.0510    -0.0882   -0.0137
    34    0.7885    0.0953    0.5447    0.7209    0.0601    -0.0425    0.0877
    35    0.6931    0.1823    0.5404    0.7159    0.0552    -0.0340    0.0755
    36    1.2030   -0.2231    0.4828    0.6473    0.1177    -0.0867    0.1381
    37    0.6419   -0.0513   -0.8989   -1.0883    0.0647    -0.1423    0.0617
    38    0.6419   -0.2877   -0.4874   -0.6529    0.1000    -0.1395    0.1032
    39    0.4855    0.2624    0.7053    0.8987    0.0531     0.0326    0.0190
```

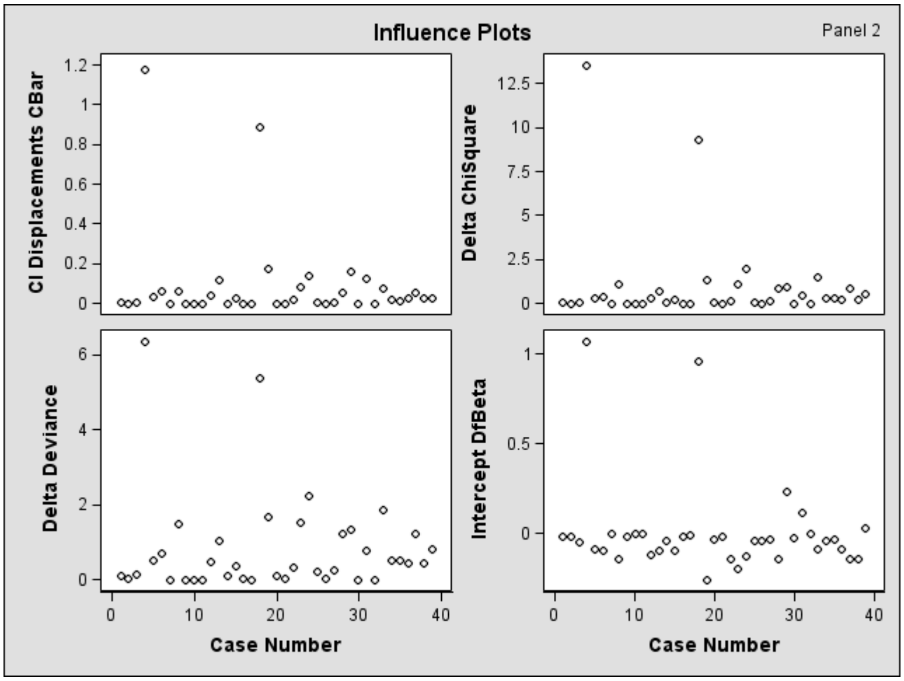
Output 42.6.2. (continued)

```
                              The LOGISTIC Procedure

                              Regression Diagnostics

                          Confidence        Confidence
                 Log        Interval          Interval
   Case       Volume    Displacement      Displacement        Delta         Delta
 Number       DfBeta               C              CBar     Deviance    Chi-Square

      1       0.0556         0.00548           0.00497       0.1000        0.0536
      2       0.0261        0.000853          0.000816       0.0369        0.0190
      3       0.0589         0.00593           0.00557       0.1695        0.0910
      4      -1.0180          1.2873            1.1756       6.3626       13.5523
      5       0.0583          0.0414            0.0366       0.5296        0.3161
      6       0.0381          0.0787            0.0667       0.6976        0.4376
      7      0.00265        8.321E-6          8.258E-6      0.00216       0.00109
      8       0.0570          0.0652            0.0616       1.4870        1.1011
      9       0.0153        0.000322          0.000311       0.0178       0.00911
     10      0.00211        6.256E-6          6.211E-6      0.00172      0.000862
     11      0.00319        0.000014          0.000013      0.00274       0.00138
     12       0.1651          0.0525            0.0447       0.5028        0.3021
     13       0.1775          0.1395            0.1168       1.0577        0.7175
     14       0.0527         0.00404           0.00382       0.1307        0.0693
     15       0.0643          0.0337            0.0292       0.3761        0.2186
     16       0.0307         0.00108           0.00104       0.0501        0.0259
     17      0.00914        0.000089          0.000088       0.0101       0.00511
     18      -0.8477          0.9845            0.8906       5.3817        9.3363
     19     -0.00488          0.2003            0.1740       1.7037        1.3227
     20       0.0518         0.00338           0.00320       0.1156        0.0610
     21       0.0208        0.000465          0.000448       0.0235        0.0120
     22       0.1179          0.0221            0.0199       0.3437        0.1956
     23       0.0357          0.0935            0.0864       1.5212        1.1355
     24      -0.1004          0.1558            0.1447       2.2550        2.0171
     25       0.0708         0.00741           0.00698       0.2193        0.1190
     26       0.0373         0.00156           0.00147       0.0517        0.0269
     27       0.0788          0.0101           0.00941       0.2589        0.1423
     28       0.1025          0.0597            0.0559       1.2404        0.8639
     29       0.0286          0.1961            0.1631       1.3460        0.9697
     30       0.0159        0.000554          0.000526       0.0201        0.0104
     31       0.1322          0.1661            0.1253       0.7755        0.5095
     32      2.48E-6        1.18E-11          1.18E-11     1.065E-6      5.324E-7
     33     -0.00216          0.0824            0.0782       1.8744        1.5331
     34       0.0671          0.0202            0.0190       0.5387        0.3157
     35       0.0711          0.0180            0.0170       0.5295        0.3091
     36       0.0631          0.0352            0.0311       0.4501        0.2641
     37       0.1025          0.0597            0.0559       1.2404        0.8639
     38       0.1397          0.0293            0.0264       0.4526        0.2639
     39       0.0489          0.0295            0.0279       0.8355        0.5254
```

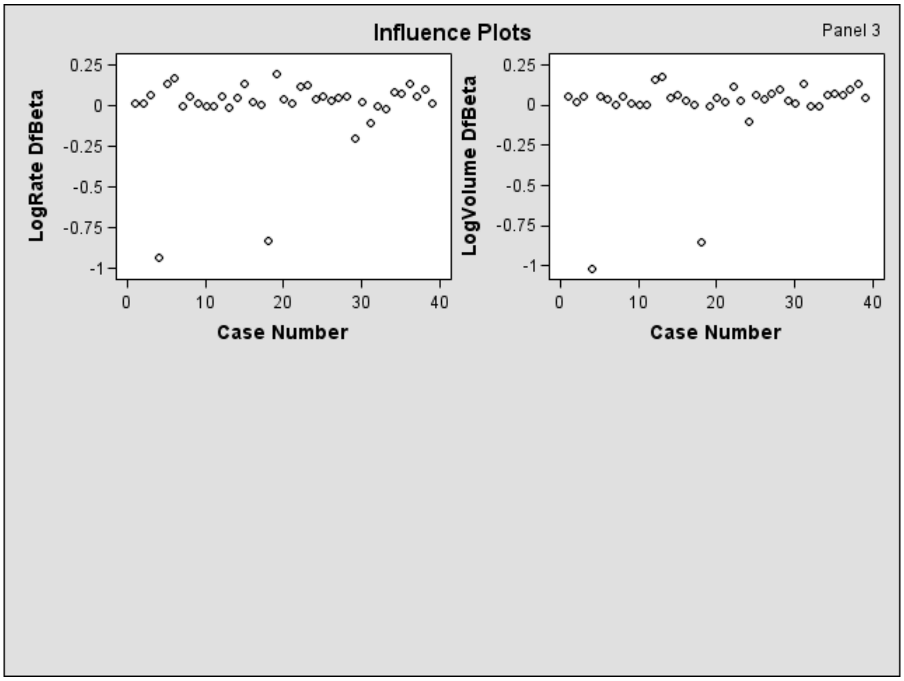
Output 42.6.3. Residuals, Hat Matrix, and CI Displacement C (Experimental)



Output 42.6.4. CI Displacement CBar, Change in Deviance and Pearson χ^2, and DFBETAS for the Intercept (Experimental)



Output 42.6.5. DFBETAS for LogRate and LogVolume (Experimental)



The index plots of the Pearson residuals and the deviance residuals (Output 42.6.3) indicate that case 4 and case 18 are poorly accounted for by the model. The index plot of the diagonal elements of the hat matrix (Output 42.6.3) suggests that case 31 is an extreme point in the design space. The index plots of DFBETAS (Output 42.6.4 and Output 42.6.5) indicate that case 4 and case 18 are causing instability in all three parameter estimates. The other four index plots in Output 42.6.3 and Output 42.6.4 also point to these two cases as having a large impact on the coefficients and goodness of fit.

Example 42.7. ROC Curve, Customized Odds Ratios, Goodness-of-Fit Statistics, R-Square, and Confidence Limits

This example plots an ROC curve, estimates a customized odds ratio, produces the traditional goodness-of-fit analysis, displays the generalized R^2 measures for the fitted model, calculates the normal confidence intervals for the regression parameters, and produces an experimental display of the probability function and prediction curves for the fitted model. The data consist of three variables: n (number of subjects in a sample), disease (number of diseased subjects in the sample), and age (age for the sample). A linear logistic regression model is used to study the effect of age on the probability of contracting the disease.

The SAS statements are as follows:

```
data Data1;
   input disease n age;
   datalines;
 0 14 25
 0 20 35
 0 19 45
 7 18 55
 6 12 65
17 17 75
;


ods html;
ods graphics on;

proc logistic data=Data1;
   model disease/n=age / scale=none
                         clparm=wald
                         clodds=pl
                         rsquare
                         outroc=roc1;
   units age=10;
run;

ods graphics off;
ods html close;
```

The option SCALE=NONE is specified to produce the deviance and Pearson goodness-of-fit analysis without adjusting for overdispersion. The RSQUARE option is specified to produce generalized R^2 measures of the fitted model. The CLPARM=WALD option is specified to produce the Wald confidence intervals for the regression parameters. The UNITS statement is specified to produce customized odds ratio estimates for a change of 10 years in the age variable, and the CLODDS=PL option is specified to produce profile likelihood confidence limits for the odds ratio. The OUTROC= option outputs the data for the ROC curve to the SAS data set, roc1.

Results are shown in Output 42.7.1 and Output 42.7.2.

Output 42.7.1. Deviance and Pearson Goodness-of-Fit Analysis

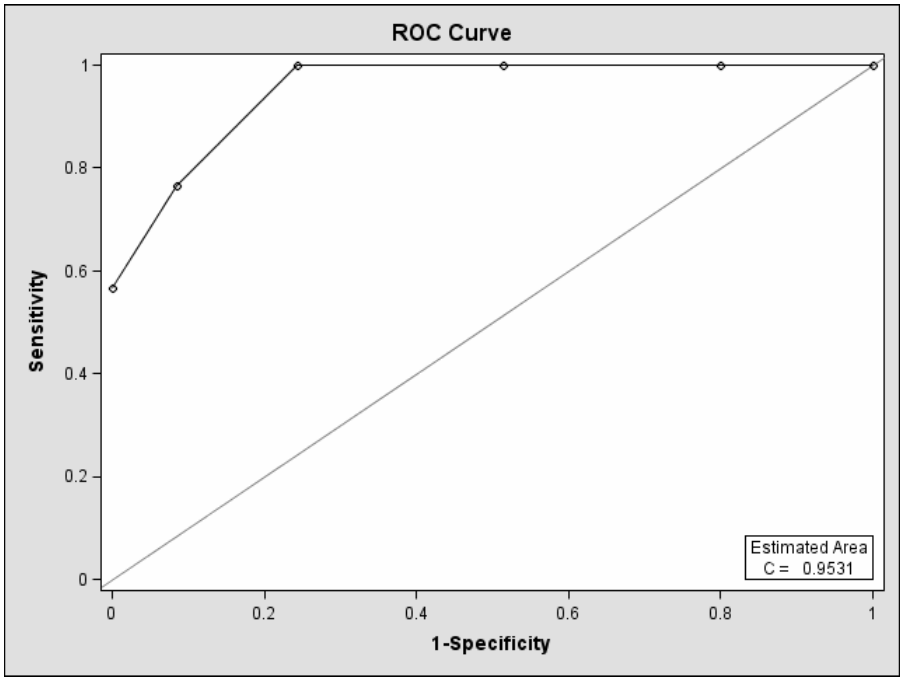
```
                    The LOGISTIC Procedure

       Deviance and Pearson Goodness-of-Fit Statistics

Criterion          Value       DF     Value/DF     Pr > ChiSq

Deviance          7.7756        4       1.9439         0.1002
Pearson           6.6020        4       1.6505         0.1585

           Number of events/trials observations: 6
```

Output 42.7.2. R-Square, Confidence Intervals, and Customized Odds Ratio

```
                         Model Fit Statistics

                                                 Intercept
                                   Intercept           and
                    Criterion           Only    Covariates

                    AIC              124.173        52.468
                    SC               126.778        57.678
                    -2 Log L         122.173        48.468


            R-Square    0.5215    Max-rescaled R-Square    0.7394


                   Testing Global Null Hypothesis: BETA=0

           Test                 Chi-Square       DF     Pr > ChiSq

           Likelihood Ratio        73.7048        1         <.0001
           Score                   55.3274        1         <.0001
           Wald                    23.3475        1         <.0001


                  Analysis of Maximum Likelihood Estimates

                                        Standard          Wald
     Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

     Intercept     1    -12.5016      2.5555       23.9317        <.0001
     age           1      0.2066      0.0428       23.3475        <.0001


        Association of Predicted Probabilities and Observed Responses

              Percent Concordant      92.6    Somers' D    0.906
              Percent Discordant       2.0    Gamma        0.958
              Percent Tied             5.4    Tau-a        0.384
              Pairs                   2100    c            0.953


                   Wald Confidence Interval for Parameters

               Parameter     Estimate     95% Confidence Limits

               Intercept     -12.5016     -17.5104      -7.4929
               age             0.2066       0.1228       0.2904


        Profile Likelihood Confidence Interval for Adjusted Odds Ratios

          Effect          Unit     Estimate     95% Confidence Limits

          age          10.0000        7.892        3.881       21.406
```

Since the experimental ODS GRAPHICS statement is specified, a graphical display of the ROC curve is produced as shown in Output 42.7.3. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the LOGISTIC procedure, see the "ODS Graphics" section on page 2388.

Output 42.7.3. Receiver Operating Characteristic Curve (Experimental)



Note that the area under the ROC curve is given by the statistic *c* in the "Association of Predicted Probabilities and Observed Responses" table. In this example, the area under the ROC curve is 0.953.

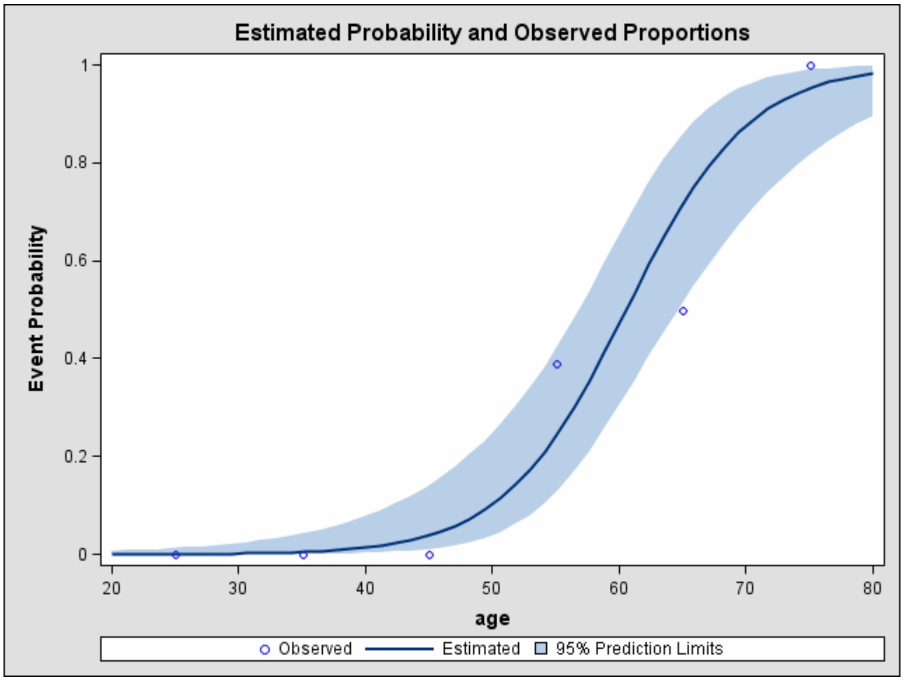
The ROC curve may also be displayed with the GPLOT procedure by using the following code.

```
symbol1 i=join v=none c=black;
proc gplot data=roc1;
   title 'ROC Curve';
   plot _sensit_*_1mspec_=1 / vaxis=0 to 1 by .1 cframe=white;
run;
```

Because there is only one continuous covariate, if the experimental ODS GRAPHICS statement and the experimental GRAPHICS option ESTPROB are specified, then a graphical display of the estimated probability curve with bounding 95% prediction limits is displayed as shown in Output 42.7.4.

```
ods html;
ods graphics on;

proc logistic data=Data1;
   model disease/n=age / scale=none
                         clparm=wald
                         clodds=pl
                         rsquare
                         outroc=roc1;
   units age=10;
   graphics estprob;
run;

ods graphics off;
ods html close;
```

Output 42.7.4. Estimated Probability and 95% Prediction Limits (Experimental)



Example 42.8. Goodness-of-Fit Tests and Subpopulations

A study is done to investigate the effects of two binary factors, A and B, on a binary response, Y. Subjects are randomly selected from subpopulations defined by the four possible combinations of levels of A and B. The number of subjects responding with each level of Y is recorded and entered into data set A.

```
data a;
   do A=0,1;
      do B=0,1;
         do Y=1,2;
            input F @@;
            output;
         end;
      end;
   end;
   datalines;
23 63 31 70 67 100 70 104
;
```

A full model is fit to examine the main effects of A and B as well as the interaction effect of A and B.

```
proc logistic data=a;
   freq F;
   model Y=A B A*B;
run;
```

Output 42.8.1. Full Model Fit

```
                 The LOGISTIC Procedure

                   Model Information

     Data Set                      WORK.A
     Response Variable             Y
     Number of Response Levels     2
     Frequency Variable            F
     Model                         binary logit
     Optimization Technique        Fisher’s scoring


        Number of Observations Read           8
        Number of Observations Used           8
        Sum of Frequencies Read             528
        Sum of Frequencies Used             528


                    Response Profile

           Ordered                      Total
             Value            Y     Frequency

                 1            1           191
                 2            2           337

              Probability modeled is Y=1.


                Model Convergence Status

     Convergence criterion (GCONV=1E-8) satisfied.


                  Model Fit Statistics

                                      Intercept
                       Intercept            and
         Criterion          Only     Covariates

         AIC             693.061        691.914
         SC              697.330        708.990
         -2 Log L        691.061        683.914


        Testing Global Null Hypothesis: BETA=0

Test                 Chi-Square       DF     Pr > ChiSq

Likelihood Ratio         7.1478        3         0.0673
Score                    6.9921        3         0.0721
Wald                     6.9118        3         0.0748
```

Output 42.8.1. (continued)

```
                   Analysis of Maximum Likelihood Estimates

                                         Standard          Wald
      Parameter     DF     Estimate        Error     Chi-Square    Pr > ChiSq

      Intercept      1      -1.0074       0.2436        17.1015        <.0001
      A              1       0.6069       0.2903         4.3714        0.0365
      B              1       0.1929       0.3254         0.3515        0.5533
      A*B            1      -0.1883       0.3933         0.2293        0.6321


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     42.2    Somers' D    0.118
               Percent Discordant     30.4    Gamma        0.162
               Percent Tied           27.3    Tau-a        0.054
               Pairs                 64367    c            0.559
```

Pearson and Deviance goodness-of-fit tests cannot be obtained for this model since a full model containing four parameters is fit, leaving no residual degrees of freedom. For a binary response model, the goodness-of-fit tests have $m-q$ degrees of freedom, where m is the number of subpopulations and q is the number of model parameters. In the preceding model, $m = q = 4$, resulting in zero degrees of freedom for the tests.

Results of the model fit are shown in Output 42.8.1. Notice that neither the A*B interaction nor the B main effect is significant. If a reduced model containing only the A effect is fit, two degrees of freedom become available for testing goodness of fit. Specifying the SCALE=NONE option requests the Pearson and deviance statistics. With *single-trial* syntax, the AGGREGATE= option is needed to define the subpopulations in the study. Specifying AGGREGATE=(A B) creates subpopulations of the four combinations of levels of A and B. Although the B effect is being dropped from the model, it is still needed to define the original subpopulations in the study. If AGGREGATE=(A) were specified, only two subpopulations would be created from the levels of A, resulting in $m = q = 2$ and zero degrees of freedom for the tests.

```
proc logistic data=a;
   freq F;
   model Y=A / scale=none aggregate=(A B);
run;
```

Output 42.8.2. Reduced Model Fit

```
                              The LOGISTIC Procedure

                                Model Information

                  Data Set                      WORK.A
                  Response Variable             Y
                  Number of Response Levels     2
                  Frequency Variable            F
                  Model                         binary logit
                  Optimization Technique        Fisher's scoring


                     Number of Observations Read           8
                     Number of Observations Used           8
                     Sum of Frequencies Read             528
                     Sum of Frequencies Used             528


                                 Response Profile

                        Ordered                      Total
                          Value             Y     Frequency

                              1             1           191
                              2             2           337

                           Probability modeled is Y=1.


                             Model Convergence Status

                  Convergence criterion (GCONV=1E-8) satisfied.


                 Deviance and Pearson Goodness-of-Fit Statistics

          Criterion          Value       DF     Value/DF     Pr > ChiSq

          Deviance          0.3541        2       0.1770         0.8377
          Pearson           0.3531        2       0.1765         0.8382

                           Number of unique profiles: 4


                               Model Fit Statistics

                                                      Intercept
                                        Intercept           and
                      Criterion              Only    Covariates

                      AIC                 693.061       688.268
                      SC                  697.330       696.806
                      -2 Log L            691.061       684.268


                     Testing Global Null Hypothesis: BETA=0

             Test                 Chi-Square       DF     Pr > ChiSq

             Likelihood Ratio         6.7937        1         0.0091
             Score                    6.6779        1         0.0098
             Wald                     6.6210        1         0.0101
```

Output 42.8.2. (continued)

```
                    Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF     Estimate      Error    Chi-Square    Pr > ChiSq

      Intercept     1      -0.9013     0.1614       31.2001        <.0001
      A             1       0.5032     0.1955        6.6210        0.0101


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     28.3    Somers' D    0.112
               Percent Discordant     17.1    Gamma        0.246
               Percent Tied           54.6    Tau-a        0.052
               Pairs                 64367    c            0.556
```

The goodness-of-fit tests (Output 42.8.2) show that dropping the B main effect and the A*B interaction simultaneously does not result in significant lack of fit of the model. The tests' large p-values indicate insufficient evidence for rejecting the null hypothesis that the model fits.

Example 42.9. Overdispersion

In a seed germination test, seeds of two cultivars were planted in pots of two soil conditions. The following SAS statements create the data set seeds, which contains the observed proportion of seeds that germinated for various combinations of cultivar and soil condition. Variable n represents the number of seeds planted in a pot, and variable r represents the number germinated. The indicator variables cult and soil represent the cultivar and soil condition, respectively.

```
data seeds;
   input pot n r cult soil;
   datalines;
 1 16     8      0       0
 2 51    26      0       0
 3 45    23      0       0
 4 39    10      0       0
 5 36     9      0       0
 6 81    23      1       0
 7 30    10      1       0
 8 39    17      1       0
 9 28     8      1       0
10 62    23      1       0
11 51    32      0       1
12 72    55      0       1
13 41    22      0       1
14 12     3      0       1
15 13    10      0       1
16 79    46      1       1
17 30    15      1       1
```

```
18 51    32      1       1
19 74    53      1       1
20 56    12      1       1
;
```

PROC LOGISTIC is used to fit a logit model to the data, with cult, soil, and cult × soil interaction as explanatory variables. The option SCALE=NONE is specified to display goodness-of-fit statistics.

```
proc logistic data=seeds;
   model r/n=cult soil cult*soil/scale=none;
   title 'Full Model With SCALE=NONE';
run;
```

Output 42.9.1. Results of the Model Fit for the Two-Way Layout

```
                            Full Model With SCALE=NONE

                             The LOGISTIC Procedure

                 Deviance and Pearson Goodness-of-Fit Statistics

        Criterion          Value       DF      Value/DF     Pr > ChiSq

        Deviance         68.3465       16        4.2717         <.0001
        Pearson          66.7617       16        4.1726         <.0001

                     Number of events/trials observations: 20


                              Model Fit Statistics

                                                 Intercept
                                   Intercept           and
                    Criterion           Only    Covariates

                    AIC             1256.852      1213.003
                    SC              1261.661      1232.240
                    -2 Log L        1254.852      1205.003


                     Testing Global Null Hypothesis: BETA=0

           Test                 Chi-Square       DF     Pr > ChiSq

           Likelihood Ratio        49.8488        3         <.0001
           Score                   49.1682        3         <.0001
           Wald                    47.7623        3         <.0001


                    Analysis of Maximum Likelihood Estimates

                                       Standard          Wald
     Parameter     DF     Estimate        Error    Chi-Square    Pr > ChiSq

     Intercept      1      -0.3788       0.1489        6.4730        0.0110
     cult           1      -0.2956       0.2020        2.1412        0.1434
     soil           1       0.9781       0.2128       21.1234        <.0001
     cult*soil      1      -0.1239       0.2790        0.1973        0.6569
```

Results of fitting the full factorial model are shown in Output 42.9.1. Both Pearson χ^2 and deviance are highly significant ($p < 0.0001$), suggesting that the model does not fit well. If the link function and the model specification are correct and if there are no outliers, then the lack of fit may be due to overdispersion. Without adjusting for the overdispersion, the standard errors are likely to be underestimated, causing the Wald tests to be too sensitive. In PROC LOGISTIC, there are three SCALE= options to accommodate overdispersion. With unequal sample sizes for the observations, SCALE=WILLIAMS is preferred. The Williams model estimates a scale parameter ϕ by equating the value of Pearson χ^2 for the full model to its approximate expected value. The full model considered here is the model with cultivar, soil condition, and their interaction. Using a full model reduces the risk of contaminating ϕ with lack of fit due to incorrect model specification.

```
proc logistic data=seeds;
   model r/n=cult soil cult*soil / scale=williams;
   title 'Full Model With SCALE=WILLIAMS';
run;
```

Output 42.9.2. Williams' Model for Overdispersion

```
                          Full Model With SCALE=WILLIAMS

                              The LOGISTIC Procedure

                                Model Information

          Data Set                       WORK.SEEDS
          Response Variable (Events)     r
          Response Variable (Trials)     n
          Weight Variable                1 / ( 1 + 0.075941 * (n - 1) )
          Model                          binary logit
          Optimization Technique         Fisher's scoring


                     Number of Observations Read          20
                     Number of Observations Used          20
                     Sum of Frequencies Read             906
                     Sum of Frequencies Used             906
                     Sum of Weights Read            198.3216
                     Sum of Weights Used            198.3216


                                 Response Profile

                Ordered     Binary          Total           Total
                  Value     Outcome     Frequency          Weight

                      1     Event             437        92.95346
                      2     Nonevent          469       105.36819


                            Model Convergence Status

                  Convergence criterion (GCONV=1E-8) satisfied.


                 Deviance and Pearson Goodness-of-Fit Statistics

          Criterion          Value       DF     Value/DF     Pr > ChiSq

          Deviance         16.4402       16       1.0275         0.4227
          Pearson          16.0000       16       1.0000         0.4530

                     Number of events/trials observations: 20

NOTE: Since the Williams method was used to accommodate overdispersion, the
      Pearson chi-squared statistic and the deviance can no longer be used to
      assess the goodness of fit of the model.


                              Model Fit Statistics

                                                 Intercept
                                  Intercept            and
                      Criterion          Only     Covariates

                      AIC             276.155        273.586
                      SC              280.964        292.822
                      -2 Log L        274.155        265.586
```

Output 42.9.2. (continued)

```
                     Full Model With SCALE=WILLIAMS

                Testing Global Null Hypothesis: BETA=0

        Test                 Chi-Square       DF     Pr > ChiSq

        Likelihood Ratio         8.5687        3         0.0356
        Score                    8.4856        3         0.0370
        Wald                     8.3069        3         0.0401


               Analysis of Maximum Likelihood Estimates

                                   Standard          Wald
  Parameter     DF     Estimate       Error    Chi-Square    Pr > ChiSq

  Intercept      1      -0.3926      0.2932        1.7932        0.1805
  cult           1      -0.2618      0.4160        0.3963        0.5290
  soil           1       0.8309      0.4223        3.8704        0.0491
  cult*soil      1      -0.0532      0.5835        0.0083        0.9274


      Association of Predicted Probabilities and Observed Responses

           Percent Concordant       50.6    Somers' D    0.258
           Percent Discordant       24.8    Gamma        0.343
           Percent Tied             24.6    Tau-a        0.129
           Pairs                  204953    c            0.629
```

Results using Williams' method are shown in Output 42.9.2. The estimate of ϕ is 0.075941 and is given in the formula for the Weight Variable at the beginning of the displayed output. Since neither cult nor cult $\times$ soil is statistically significant ($p = 0.5290$ and $p = 0.9274$, respectively), a reduced model that contains only the soil condition factor is fitted, with the observations weighted by $1/(1+0.075941(N-1))$. This can be done conveniently in PROC LOGISTIC by including the scale estimate in the SCALE=WILLIAMS option as follows:

```
proc logistic data=seeds;
   model r/n=soil / scale=williams(0.075941);
   title 'Reduced Model With SCALE=WILLIAMS(0.075941)';
run;
```

Output 42.9.3. Reduced Model with Overdispersion Controlled

```
                  Reduced Model With SCALE=WILLIAMS(0.075941)

                           The LOGISTIC Procedure

                   Analysis of Maximum Likelihood Estimates

                                        Standard          Wald
Parameter     DF     Estimate         Error    Chi-Square    Pr > ChiSq

Intercept      1      -0.5249        0.2076        6.3949        0.0114
soil           1       0.7910        0.2902        7.4284        0.0064
```

Results of the reduced model fit are shown in Output 42.9.3. Soil condition remains a significant factor ($p = 0.0064$) for the seed germination.

Example 42.10. Conditional Logistic Regression for Matched Pairs Data

In matched pairs, or *case-control*, studies, conditional logistic regression is used to investigate the relationship between an outcome of being an event (case) or a nonevent (control) and a set of prognostic factors.

The data in this example are a subset of the data from the Los Angeles Study of the Endometrial Cancer Data in Breslow and Day (1980). There are 63 matched pairs, each consisting of a case of endometrial cancer (Outcome=1) and a control (Outcome=0). The case and corresponding control have the same ID. Two prognostic factors are included: Gall (an indicator variable for gall bladder disease) and Hyper (an indicator variable for hypertension). The goal of the case-control analysis is to determine the relative risk for gall bladder disease, controlling for the effect of hypertension.

```
data Data1;
  do ID=1 to 63;
    do Outcome = 1 to 0 by -1;
      input Gall Hyper @@;
      output;
    end;
  end;
  datalines;
0 0  0 0    0 0  0 0    0 1  0 1    0 0  1 0    1 0  0 1
0 1  0 0    1 0  0 0    1 1  0 1    0 0  0 0    0 0  0 0
1 0  0 0    0 0  0 1    1 0  0 1    1 0  1 0    1 0  0 1
0 1  0 0    0 0  1 1    0 0  1 1    0 0  0 1    0 1  0 0
0 0  1 1    0 1  0 1    0 1  0 0    0 0  0 0    0 0  0 0
0 0  0 1    1 0  0 1    0 0  0 1    1 0  0 0    0 1  0 0
0 1  0 0    0 1  0 0    0 1  0 0    0 0  0 0    1 1  1 1
0 0  0 1    0 1  0 0    0 1  0 1    0 1  0 1    0 1  0 0
0 0  0 0    0 1  1 0    0 0  0 1    0 0  0 0    1 0  0 0
0 0  0 0    1 1  0 0    0 1  0 0    0 0  0 0    0 1  0 1
0 0  0 0    0 1  0 1    0 1  0 0    0 1  0 0    1 0  0 0
```

```
0 0  0 0   1 1  1 0   0 0  0 0   0 0  0 0   1 1  0 0
1 0  1 0   0 1  0 0   1 0  0 0
;
```

There are several ways to approach this problem with PROC LOGISTIC.

- Specify the STRATA statement to perform a conditional logistic regression.
- Specify EXACT and STRATA statements to perform an exact conditional logistic regression on the original data set, if you believe the data set is too small or too sparse for the usual asymptotics to hold.
- Transform each matched pair into a single observation then specify a PROC LOGISTIC statement on this transformed data without a STRATA statement; this also performs a conditional logistic regression and produces essentially the same results.
- Specify an EXACT statement on the transformed data.

SAS statements and selected results for these four approaches are given in the remainder of this example.

Conditional Analysis Using the STRATA Statement

In the following SAS statements, PROC LOGISTIC is invoked with the ID variable declared in the STRATA statement to obtain the conditional logistic model estimates. Two models are fitted. The first model contains Gall as the only predictor variable, and the second model contains both Gall and Hyper as predictor variables. Because the option CLODDS=Wald is specified, PROC LOGISTIC computes a 95% Wald confidence interval for the odds ratio for each predictor variable.

```
proc logistic data=Data1;
   strata ID;
   model outcome(event='1')=Gall / clodds=Wald;
run;


proc logistic data=Data1;
   strata ID;
   model outcome(event='1')=Gall Hyper /clodds=Wald;
run;
```

Results from the two conditional logistic analyses are shown in Output 42.10.1 and Output 42.10.2. Note that there is only one response level listed in the “Response Profile” tables, and there is no intercept term in the “Analysis of Maximum Likelihood Estimates” tables.

Output 42.10.1. Conditional Logistic Regression (Gall as Risk Factor)

```
                              The LOGISTIC Procedure

                               Conditional Analysis

                                Model Information

                Data Set                      WORK.DATA1
                Response Variable             Outcome
                Number of Response Levels     2
                Number of Strata              63
                Model                         binary logit
                Optimization Technique        Newton-Raphson ridge


                     Number of Observations Read          126
                     Number of Observations Used          126


                                 Response Profile

                        Ordered                     Total
                          Value      Outcome     Frequency

                              1            0            63
                              2            1            63

                        Probability modeled is Outcome=1.


                                  Strata Summary

                              Outcome
                  Response    -------    Number of
                   Pattern    0     1       Strata    Frequency

                         1    1     1           63          126
```

Output 42.10.1. (continued)

```
                              Conditional Analysis

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                     Without           With
                    Criterion     Covariates     Covariates

                    AIC               87.337         85.654
                    SC                87.337         88.490
                    -2 Log L          87.337         83.654


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         3.6830        1         0.0550
            Score                    3.5556        1         0.0593
            Wald                     3.2970        1         0.0694


                   Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

      Gall          1      0.9555      0.5262        3.2970        0.0694


               Wald Confidence Interval for Adjusted Odds Ratios

           Effect         Unit     Estimate     95% Confidence Limits

           Gall         1.0000        2.600        0.927        7.293
```

Output 42.10.2. Conditional Logistic Regression (Gall and Hyper as Risk Factors)

```
              The LOGISTIC Procedure

               Conditional Analysis

                Model Information

Data Set                      WORK.DATA1
Response Variable             Outcome
Number of Response Levels     2
Number of Strata              63
Model                         binary logit
Optimization Technique        Newton-Raphson ridge


     Number of Observations Read         126
     Number of Observations Used         126


                 Response Profile

        Ordered                      Total
          Value      Outcome     Frequency

              1            0            63
              2            1            63

        Probability modeled is Outcome=1.


                  Strata Summary

              Outcome
  Response    -------    Number of
   Pattern    0     1       Strata    Frequency

         1    1     1           63          126
```

Output 42.10.2. (continued)

```
                              Conditional Analysis

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                   Without           With
                    Criterion     Covariates     Covariates

                    AIC               87.337         86.788
                    SC                87.337         92.460
                    -2 Log L          87.337         82.788


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         4.5487        2         0.1029
            Score                    4.3620        2         0.1129
            Wald                     4.0060        2         0.1349


                   Analysis of Maximum Likelihood Estimates

                                       Standard          Wald
      Parameter     DF     Estimate       Error    Chi-Square     Pr > ChiSq

      Gall           1       0.9704      0.5307        3.3432         0.0675
      Hyper          1       0.3481      0.3770        0.8526         0.3558


               Wald Confidence Interval for Adjusted Odds Ratios

           Effect         Unit      Estimate     95% Confidence Limits

           Gall         1.0000         2.639        0.933        7.468
           Hyper        1.0000         1.416        0.677        2.965
```

In the first model, where Gall is the only predictor variable (Output 42.10.1), the odds ratio estimate for Gall is 2.60, which is marginally significant (p=0.0694) and which is an estimate of the relative risk for gall bladder disease. A 95% confidence interval for this relative risk is (0.927, 7.293).

In the second model, where both Gall and Hyper are present (Output 42.10.2), the odds ratio estimate for Gall is 2.639, which is an estimate of the relative risk for gall bladder disease adjusted for the effects of hypertension. A 95% confidence interval for this adjusted relative risk is (0.933, 7.468). Note that the adjusted values (accounting for hypertension) for gall bladder disease are not very different from the unadjusted values (ignoring hypertension). This is not surprising since the prognostic factor Hyper is highly statistically insignificant. The 95% Wald confidence interval for the odds ratio for Hyper is (0.677, 2.965), which contains unity with a p-value greater than 0.3.

Exact Analysis Using the STRATA Statement

When you believe there is not enough data or that the data are too sparse, you can perform a stratified exact conditional logistic regression. The following statements perform stratified exact conditional logistic regressions on the original data set by specifying both the STRATA and EXACT statements.

```
proc logistic data=Data1 exactonly;
   strata ID;
   model outcome(event='1')=Gall;
   exact Gall / estimate=both;
run;
```

```
proc logistic data=Data1 exactonly;
   strata ID;
   model outcome(event='1')=Gall Hyper;
   exact Gall Hyper / jointonly estimate=both;
run;
```

Output 42.10.3. Exact Conditional Logistic Regression (Gall as Risk Factor)

```
                              The LOGISTIC Procedure

                            Exact Conditional Analysis

                             Conditional Exact Tests

                                                   --- p-Value ---
                Effect   Test          Statistic    Exact      Mid

                Gall     Score            3.5556   0.0963   0.0799
                         Probability      0.0327   0.0963   0.0799


                            Exact Parameter Estimates

                                         95% Confidence
             Parameter    Estimate           Limits           p-Value

             Gall           0.9555     -0.1394      2.2316     0.0963


                                Exact Odds Ratios

                                        95% Confidence
              Parameter   Estimate          Limits          p-Value

              Gall           2.600      0.870      9.315     0.0963
```

Output 42.10.4. Exact Conditional Logistic Regression (Gall and Hyper as Risk Factors)

```
                          The LOGISTIC Procedure

                        Exact Conditional Analysis

                         Conditional Exact Tests

                                                   --- p-Value ---
          Effect    Test           Statistic     Exact       Mid

          Joint     Score             4.3620    0.1150    0.1134
                    Probability      0.00316    0.1150    0.1134


                        Exact Parameter Estimates

                                     95% Confidence
       Parameter    Estimate             Limits            p-Value

       Gall           0.9530     -0.1407      2.2292      0.0969
       Hyper          0.3425     -0.4486      1.1657      0.4622


                            Exact Odds Ratios

                                    95% Confidence
        Parameter   Estimate            Limits            p-Value

        Gall           2.593      0.869      9.293      0.0969
        Hyper          1.408      0.639      3.208      0.4622
```

Note that the score statistics in the "Conditional Exact Tests" tables in Output 42.10.3 and Output 42.10.4 are identical to the score statistics in the conditional analyses in Output 42.10.1 and Output 42.10.2, respectively. The exact odds ratio confidence intervals are much wider than their conditional analysis counterparts, but the parameter estimates are similar. The exact analyses confirm the marginal significance of Gall and the insignificance of Hyper as predictor variables.

Conditional Analysis Using Transformed Data

When each matched set consists of one event and one nonevent, the conditional likelihood is given by

$$\prod_i (1 + \exp(-\boldsymbol{\beta}'(\mathbf{x}_{i1} - \mathbf{x}_{i0}))^{-1}$$

where $\mathbf{x}_{i1}$ and $\mathbf{x}_{i0}$ are vectors representing the prognostic factors for the event and nonevent, respectively, of the ith matched set. This likelihood is identical to the likelihood of fitting a logistic regression model to a set of data with constant response, where the model contains no intercept term and has explanatory variables given by $\mathbf{d}_i = \mathbf{x}_{i1} - \mathbf{x}_{i0}$ (Breslow 1982).

To apply this method, each matched pair is transformed into a single observation, where the variables Gall and Hyper contain the differences between the corresponding values for the case and the control (case − control). The variable Outcome, which will be used as the response variable in the logistic regression model, is given a constant value of 0 (which is the Outcome value for the control, although any constant, numeric or character, will do).

```
data Data2;
   set Data1;
   drop id1 gall1 hyper1;
   retain id1 gall1 hyper1 0;
   if (ID = id1) then do;
      Gall=gall1-Gall; Hyper=hyper1-Hyper;
      output;
   end;
   else do;
      id1=ID; gall1=Gall; hyper1=Hyper;
   end;
run;
```

Note that there are 63 observations in the data set, one for each matched pair. The variable Outcome has a constant value of 0.

In the following SAS statements, PROC LOGISTIC is invoked with the NOINT option to obtain the conditional logistic model estimates. Because the option CLODDS=PL is specified, PROC LOGISTIC computes a 95% profile likelihood confidence interval for the odds ratio for each predictor variable; note that profile likelihood confidence intervals are not currently available when a STRATA statement is specified.

```
proc logistic data=Data2;
   model outcome=Gall / noint clodds=PL;
run;



proc logistic data=Data2;
   model outcome=Gall Hyper / noint clodds=PL;
run;
```

The results are not displayed here.

Exact Analysis Using Transformed Data

Sometimes the original data set in a matched-pairs study may be too large for the exact methods to handle. In such cases it may be possible to use the transformed data set. The following code performs exact conditional logistic regressions on the transformed data set. The results are not displayed here.

```
proc logistic data=Data2 exactonly;
   model outcome=Gall / noint;
   exact Gall / estimate=both;
run;
proc logistic data=Data2 exactonly;
   model outcome=Gall Hyper / noint;
   exact Gall Hyper / jointonly estimate=both;
run;
```

Example 42.11. Complementary Log-Log Model for Infection Rates

Antibodies produced in response to an infectious disease like malaria remain in the body after the individual has recovered from the disease. A serological test detects the presence or absence of such antibodies. An individual with such antibodies is termed seropositive. In areas where the disease is endemic, the inhabitants are at fairly constant risk of infection. The probability of an individual never having been infected in Y years is $\exp(-\mu Y)$, where μ is the mean number of infections per year (refer to the appendix of Draper, Voller, and Carpenter 1972). Rather than estimating the unknown μ, it is of interest to epidemiologists to estimate the probability of a person living in the area being infected in one year. This infection rate γ is given by

$$\gamma = 1 - e^{-\mu}$$

The following statements create the data set sero, which contains the results of a serological survey of malarial infection. Individuals of nine age groups (Group) were tested. Variable A represents the midpoint of the age range for each age group. Variable N represents the number of individuals tested in each age group, and variable R represents the number of individuals that are seropositive.

```
data sero;
   input Group A N R;
   X=log(A);
   label X='Log of Midpoint of Age Range';
   datalines;
1  1.5  123  8
2  4.0  132  6
3  7.5  182 18
4 12.5  140 14
5 17.5  138 20
6 25.0  161 39
7 35.0  133 19
8 47.0   92 25
9 60.0   74 44
;
```

For the ith group with age midpoint A_i, the probability of being seropositive is $p_i = 1 - \exp(-\mu A_i)$. It follows that

$$\log(-\log(1-p_i)) = \log(\mu) + \log(A_i)$$

By fitting a binomial model with a complementary log-log link function and by using X=log(A) as an offset term, you can estimate $\beta_0 = \log(\mu)$ as an intercept parameter. The following SAS statements invoke PROC LOGISTIC to compute the maximum likelihood estimate of β_0. The LINK=CLOGLOG option is specified to request the complementary log-log link function. Also specified is the CLPARM=PL option, which requests the profile likelihood confidence limits for β_0.

```
proc logistic data=sero;
   model R/N= / offset=X
                link=cloglog
                clparm=pl
                scale=none;
title 'Constant Risk of Infection';
run;
```

Output 42.11.1. Modeling Constant Risk of Infection

```
                           Constant Risk of Infection

                             The LOGISTIC Procedure

                               Model Information

Data Set                       WORK.SERO
Response Variable (Events)     R
Response Variable (Trials)     N
Offset Variable                X                    Log of Midpoint of Age Range
Model                          binary cloglog
Optimization Technique         Fisher's scoring


                     Number of Observations Read           9
                     Number of Observations Used           9
                     Sum of Frequencies Read            1175
                     Sum of Frequencies Used            1175


                                Response Profile

                        Ordered      Binary          Total
                          Value      Outcome      Frequency

                              1      Event              193
                              2      Nonevent           982


                    Intercept-Only Model Convergence Status

                Convergence criterion (GCONV=1E-8) satisfied.

                             -2 Log L = 967.1158
```

Output 42.11.1. (continued)

```
                 Deviance and Pearson Goodness-of-Fit Statistics

          Criterion           Value        DF     Value/DF     Pr > ChiSq

          Deviance          41.5032         8       5.1879         <.0001
          Pearson           50.6883         8       6.3360         <.0001

                     Number of events/trials observations: 9


                    Analysis of Maximum Likelihood Estimates

                                        Standard          Wald
       Parameter     DF     Estimate       Error    Chi-Square     Pr > ChiSq

       Intercept      1      -4.6605      0.0725     4133.5626         <.0001
       X              1       1.0000           0         .              .


                          Profile Likelihood Confidence
                             Interval for Parameters

                 Parameter     Estimate     95% Confidence Limits

                 Intercept      -4.6605      -4.8057      -4.5219
```

Results of fitting this constant risk model are shown in Output 42.11.1. The maximum likelihood estimate of $\beta_0 = \log(\mu)$ and its estimated standard error are $\widehat{\beta}_0 = -4.6605$ and $\widehat{\sigma}_{\widehat{\beta}_0} = 0.0725$, respectively. The infection rate is estimated as

$$\widehat{\gamma} = 1 - e^{-\widehat{\mu}} = 1 - e^{-e^{\widehat{\beta}_0}} = 1 - e^{-e^{-4.6605}} = 0.00942$$

The 95% confidence interval for γ, obtained by back-transforming the 95% confidence interval for β_0, is (0.0082, 0.0108); that is, there is a 95% chance that, in repeated sampling, the interval of 8 to 11 infections per thousand individuals contains the true infection rate.

The goodness of fit statistics for the constant risk model are statistically significant ($p < 0.0001$), indicating that the assumption of constant risk of infection is not correct. You can fit a more extensive model by allowing a separate risk of infection for each age group. Suppose μ_i is the mean number of infections per year for the ith age group. The probability of seropositive for the ith group with age midpoint A_i is $p_i = 1 - \exp(-\mu_i A_i)$, so that

$$\log(-\log(1 - p_i)) = \log(\mu_i) + \log(A_i)$$

In the following statements, a complementary log-log model is fit containing Group as an explanatory classification variable with the GLM coding (so that a dummy variable is created for each age group), no intercept term, and X=log(A) as an offset term. The ODS OUTPUT statement saves the estimates and their 95% profile likelihood confidence limits to ClparmPL data set. Note that $\log(\mu_i)$ is the regression parameter associated with Group$= i$.

```
proc logistic data=sero;
   ods output ClparmPL=ClparmPL;
   class Group / param=glm;
   model R/N=Group / noint
                     offset=X
                     link=cloglog
                     clparm=pl;
   title 'Infectious Rates and 95% Confidence Intervals';
run;
```

Output 42.11.2. Modeling Separate Risk of Infection

```
                  Infectious Rates and 95% Confidence Intervals

                              The LOGISTIC Procedure

                      Analysis of Maximum Likelihood Estimates

                                              Standard          Wald
        Parameter      DF    Estimate        Error    Chi-Square    Pr > ChiSq

        Group     1     1     -3.1048       0.3536       77.0877        <.0001
        Group     2     1     -4.4542       0.4083      119.0164        <.0001
        Group     3     1     -4.2769       0.2358      328.9593        <.0001
        Group     4     1     -4.7761       0.2674      319.0600        <.0001
        Group     5     1     -4.7165       0.2238      443.9920        <.0001
        Group     6     1     -4.5012       0.1606      785.1350        <.0001
        Group     7     1     -5.4252       0.2296      558.1114        <.0001
        Group     8     1     -4.9987       0.2008      619.4666        <.0001
        Group     9     1     -4.1965       0.1559      724.3157        <.0001
        X               1      1.0000            0             .             .


                         Profile Likelihood Confidence
                            Interval for Parameters

                 Parameter       Estimate     95% Confidence Limits

                 Group     1      -3.1048      -3.8880      -2.4833
                 Group     2      -4.4542      -5.3769      -3.7478
                 Group     3      -4.2769      -4.7775      -3.8477
                 Group     4      -4.7761      -5.3501      -4.2940
                 Group     5      -4.7165      -5.1896      -4.3075
                 Group     6      -4.5012      -4.8333      -4.2019
                 Group     7      -5.4252      -5.9116      -5.0063
                 Group     8      -4.9987      -5.4195      -4.6289
                 Group     9      -4.1965      -4.5164      -3.9037
```

Results of fitting the model with a separate risk of infection are shown in Output 42.11.2. For the first age group (Group=1), the point estimate of $\log(\mu_1)$ is -3.1048, which transforms into an infection rate of $1 - \exp(-\exp(-3.1048)) = 0.0438$. A 95% confidence interval for this infection rate is obtained by transforming the 95% confidence interval for $\log(\mu_1)$. For the first age group, the lower and upper confidence limits are $1-\exp(-\exp(-3.8880) = 0.0203$ and $1-\exp(-\exp(-2.4833)) = 0.0801$, respectively; that is, there is a 95% chance that, in repeated sampling, the interval of 20 to 80 infections per thousand individuals contains the true infection rate.

The following statements perform this transformation on the estimates and confidence limits saved in the ClparmPL data set; the resulting estimated infection rates in one year's time for each age group are displayed in Table 42.5. Note that the infection rate for the first age group is high compared to the other age groups.

```
data ClparmPL;
   set ClparmPL;
   Estimate=round( 1000*( 1-exp(-exp(Estimate)) ) );
   LowerCL =round( 1000*( 1-exp(-exp(LowerCL )) ) );
   UpperCL =round( 1000*( 1-exp(-exp(UpperCL )) ) );
run;
```

Table 42.5. Infection Rate in One Year

	Number Infected per 1,000 People		
Age	Point	95% Confidence Limits	
Group	Estimate	Lower	Upper
1	44	20	80
2	12	5	23
3	14	8	21
4	8	5	14
5	9	6	13
6	11	8	15
7	4	3	7
8	7	4	10
9	15	11	20

Example 42.12. Complementary Log-Log Model for Interval-Censored Survival Times

Often survival times are not observed more precisely than the interval (for instance, a day) within which the event occurred. Survival data of this form are known as grouped or interval-censored data. A discrete analogue of the continuous proportional hazards model (Prentice and Gloeckler 1978; Allison 1982) is used to investigate the relationship between these survival times and a set of explanatory variables.

Suppose T_i is the discrete survival time variable of the ith subject with covariates $\mathbf{x}_i$. The discrete-time hazard rate λ_{it} is defined as

$$\lambda_{it} = \Pr(T_i = t \mid T_i \geq t, \mathbf{x}_i), \quad t = 1, 2, \ldots$$

Using elementary properties of conditional probabilities, it can be shown that

$$\Pr(T_i = t) = \lambda_{it} \prod_{j=1}^{t-1}(1 - \lambda_{ij}) \quad \text{and} \quad \Pr(T_i > t) = \prod_{j=1}^{t}(1 - \lambda_{ij})$$

Suppose t_i is the observed survival time of the ith subject. Suppose $\delta_i = 1$ if $T_i = t_i$ is an event time and 0 otherwise. The likelihood for the grouped survival data is given by

$$\begin{aligned} L &= \prod_i [\Pr(T_i = t_i)]^{\delta_i} [\Pr(T_i > t_i)]^{1-\delta_i} \\ &= \prod_i \left(\frac{\lambda_{it_i}}{1 - \lambda_{it_i}} \right)^{\delta_i} \prod_{j=1}^{t_i} (1 - \lambda_{ij}) \\ &= \prod_i \prod_{j=1}^{t_i} \left(\frac{\lambda_{ij}}{1 - \lambda_{ij}} \right)^{y_{ij}} (1 - \lambda_{ij}) \end{aligned}$$

where $y_{ij} = 1$ if the ith subject experienced an event at time $T_i = j$ and 0 otherwise.

Note that the likelihood L for the grouped survival data is the same as the likelihood of a binary response model with event probabilities λ_{ij}. If the data are generated by a continuous-time proportional hazards model, Prentice and Gloeckler (1978) have shown that

$$\lambda_{ij} = 1 - \exp(-\exp(\alpha_j + \boldsymbol{\beta}'\mathbf{x}_i))$$

where the coefficient vector $\boldsymbol{\beta}$ is identical to that of the continuous-time proportional hazards model, and α_j is a constant related to the conditional survival probability in the interval defined by $T_i = j$ at $\mathbf{x}_i = \mathbf{0}$. The grouped data survival model is therefore equivalent to the binary response model with complementary log-log link function. To fit the grouped survival model using PROC LOGISTIC, you must treat each discrete time unit for each subject as a separate observation. For each of these observations, the response is dichotomous, corresponding to whether or not the subject died in the time unit.

Consider a study of the effect of insecticide on flour-beetles. Four different concentrations of an insecticide were sprayed on separate groups of flour-beetles. The numbers of male and female flour-beetles dying in successive intervals were saved in the data set beetles.

```
data beetles(keep=time sex conc freq);
   input time m20 f20 m32 f32 m50 f50 m80 f80;
   conc=.20;
   freq= m20; sex=1; output;
   freq= f20; sex=2; output;
   conc=.32;
   freq= m32; sex=1; output;
   freq= f32; sex=2; output;
   conc=.50;
   freq= m50; sex=1; output;
   freq= f50; sex=2; output;
   conc=.80;
   freq= m80; sex=1; output;
   freq= f80; sex=2; output;
   datalines;
 1   3   0  7  1  5  0  4  2
 2  11   2 10  5  8  4 10  7
 3  10   4 11 11 11  6  8 15
 4   7   8 16 10 15  6 14  9
 5   4   9  3  5  4  3  8  3
 6   3   3  2  1  2  1  2  4
 7   2   0  1  0  1  1  1  1
 8   1   0  0  1  1  4  0  1
 9   0   0  1  1  0  0  0  0
10   0   0  0  0  0  0  1  1
11   0   0  0  0  1  1  0  0
12   1   0  0  0  0  1  0  0
13   1   0  0  0  0  1  0  0
14 101 126 19 47  7 17  2  4
;
```

The data set beetles contains four variables: time, sex, conc, and freq. time represents the interval death time; for example, time=2 is the interval between day 1 and day 2. Insects surviving the duration (13 days) of the experiment are given a time value of 14. The variable sex represents the sex of the insects (1=male, 2=female), conc represents the concentration of the insecticide (mg/cm^2), and freq represents the frequency of the observations.

To use PROC LOGISTIC with the grouped survival data, you must expand the data so that each beetle has a separate record for each day of survival. A beetle that died in the third day (time=3) would contribute three observations to the analysis, one for each day it was alive at the beginning of the day. A beetle that survives the 13-day duration of the experiment (time=14) would contribute 13 observations.

A new data set days that contains the beetle-day observations is created from the data set beetles. In addition to the variables sex, conc and freq, the data set contains an outcome variable y and 13 indicator variables day1, day2, . . ., day13. y has a value of 1 if the observation corresponds to the day that the beetle died and has a value of 0 otherwise. An observation for the first day will have a value of 1 for day1 and a value of 0 for day2–day13; an observation for the second day will have a value of 1 for day2 and a value of 0 for day1 and day2–day13. For instance, Output 42.12.1 shows an observation in the beetles data set with time=3, and Output 42.12.2 shows the corresponding beetle-day observations in the data set days.

```
data days;
   retain day1-day13 0;
   array dd[13] day1-day13;
   set beetles;
   if time = 14 then do day=1 to 13;
      y=0; dd[day]=1;
      output;
      dd[day]=0;
   end;
   else do day=1 to time;
      if day=time then y=1;
      else y=0;
      dd[day]=1;
      output;
      dd[day]=0;
   end;
```

Output 42.12.1. An Observation with Time=3 in Data Set Beetles

```
Obs     time     conc     freq     sex

 17       3       0.2      10       1
```

Output 42.12.2. Corresponding Beetle-day Observations in Days

```
                                                    d  d  d  d
    t   c    f           d  d  d  d  d  d  d  d  d  a  a  a  a
O   i   o    r  s  d     a  a  a  a  a  a  a  a  a  y  y  y  y
b   m   n    e  e  a     y  y  y  y  y  y  y  y  y  1  1  1  1
s   e   c    q  x  y  y  1  2  3  4  5  6  7  8  9  0  1  2  3

25  3  0.2  10  1  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0
26  3  0.2  10  1  2  0  0  1  0  0  0  0  0  0  0  0  0  0  0
27  3  0.2  10  1  3  1  0  0  1  0  0  0  0  0  0  0  0  0  0
```

The following SAS statements invoke PROC LOGISTIC to fit a complementary log-log model for binary data with response variable Y and explanatory variables day1–day13, sex, and conc. Specifying the EVENT= option ensures that the event (y=1) probability is modeled. The coefficients of day1–day13 can be used to estimate the baseline survival function. The NOINT option is specified to prevent any redundancy in estimating the coefficients of day1–day13. The Newton-Raphson algorithm is used for the maximum likelihood estimation of the parameters.

```
proc logistic data=days outest=est1;
   model y(event='1')= day1-day13 sex conc
         / noint link=cloglog technique=newton;
   freq freq;
run;
```

Output 42.12.3. Parameter Estimates for the Grouped Proportional Hazards Model

```
                              The LOGISTIC Procedure

                    Analysis of Maximum Likelihood Estimates

                                        Standard          Wald
      Parameter     DF     Estimate        Error    Chi-Square    Pr > ChiSq

      day1           1      -3.9314       0.2934      179.5602        <.0001
      day2           1      -2.8751       0.2412      142.0596        <.0001
      day3           1      -2.3985       0.2299      108.8833        <.0001
      day4           1      -1.9953       0.2239       79.3960        <.0001
      day5           1      -2.4920       0.2515       98.1470        <.0001
      day6           1      -3.1060       0.3037      104.5799        <.0001
      day7           1      -3.9704       0.4230       88.1107        <.0001
      day8           1      -3.7917       0.4007       89.5233        <.0001
      day9           1      -5.1540       0.7316       49.6329        <.0001
      day10          1      -5.1350       0.7315       49.2805        <.0001
      day11          1      -5.1131       0.7313       48.8834        <.0001
      day12          1      -5.1029       0.7313       48.6920        <.0001
      day13          1      -5.0951       0.7313       48.5467        <.0001
      sex            1      -0.5651       0.1141       24.5477        <.0001
      conc           1       3.0918       0.2288      182.5665        <.0001
```

Results of the model fit are given in Output 42.12.3. Both sex and conc are statistically significant for the survival of beetles sprayed by the insecticide. Female beetles are more resilient to the chemical than male beetles, and increased concentration increases the effectiveness of the insecticide.

The coefficients of day1–day13 are the maximum likelihood estimates of $\alpha_1, \ldots, \alpha_{13}$, respectively. The baseline survivor function $S_0(t)$ is estimated by

$$\hat{S}_0(t) = \widehat{\Pr}(T > t) = \prod_{j \le t} \exp(-\exp(\hat{\alpha}_j))$$

and the survivor function for a given covariate pattern (sex=x_1 and conc=x_2) is estimated by

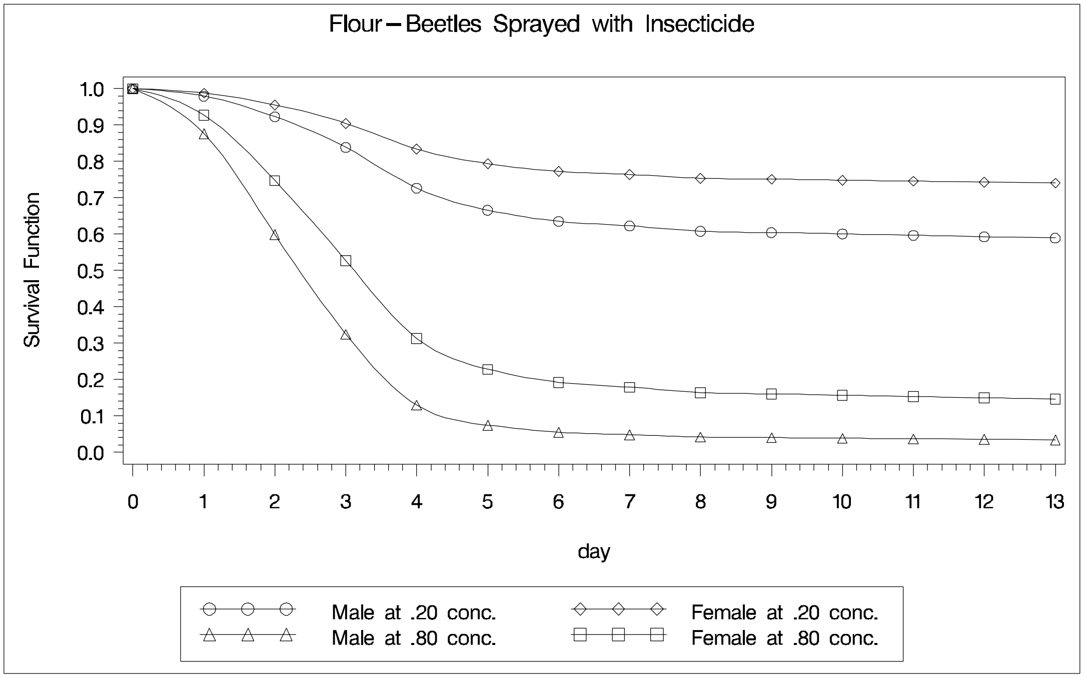
$$y\hat{S}(t) = [\hat{S}_0(t)]^{\exp(-0.5651x_1+3.0918x_2)}$$

The following statements compute the survivor curves for male and female flour-beetles exposed to the insecticide of concentrations 0.20 mg/cm^2 and 0.80 mg/cm^2. The GPLOT procedure in SAS/GRAPH software is used to plot the survival curves. Instead of plotting them as step functions, the SPLINE option is used to smooth the curves. These smoothed survival curves are displayed in Output 42.12.4.

```
legend1 label=none frame cframe=white cborder=black position=center
        value=(justify=center);
run;
axis1 label=(angle=90 'Survival Function');
proc gplot data=one;
   plot (s_m20 s_f20 s_m80 s_f80) * day
        / overlay legend=legend1 vaxis=axis1;
   symbol1 v=circle   i=spline c=black height=.8;
   symbol2 v=diamond  i=spline c=black height=.8;
   symbol3 v=triangle i=spline c=black height=.8;
   symbol4 v=square   i=spline c=black height=.8;
run;
```

The probability of survival is displayed on the vertical axis. Notice that most of the insecticide effect occurs by day 6 for both the high and low concentrations.

Output 42.12.4. Predicted Survival at Concentrations of 0.20 and 0.80 mg/cm^2



Example 42.13. Scoring Data Sets with the SCORE Statement

This example first illustrates the syntax used for scoring data sets, then uses a previously scored data set to score a new data set. A generalized logit model is fit to the remote-sensing data set used in Example 25.4 on page 1231 of Chapter 25, “The DISCRIM Procedure,” to illustrate discrimination and classification methods. The response variable is Crop and the prognostic factors are x1 through x4.

```
data Crops;
   length Crop $ 10;
   infile datalines truncover;
   input Crop $ @@;
   do i=1 to 3;
     input x1-x4 @@;
     if (x1 ^= .) then output;
   end;
   input;
   datalines;
Corn       16 27 31 33  15 23 30 30  16 27 27 26
Corn       18 20 25 23  15 15 31 32  15 32 32 15
Corn       12 15 16 73
Soybeans   20 23 23 25  24 24 25 32  21 25 23 24
Soybeans   27 45 24 12  12 13 15 42  22 32 31 43
Cotton     31 32 33 34  29 24 26 28  34 32 28 45
Cotton     26 25 23 24  53 48 75 26  34 35 25 78
Sugarbeets 22 23 25 42  25 25 24 26  34 25 16 52
Sugarbeets 54 23 21 54  25 43 32 15  26 54  2 54
Clover     12 45 32 54  24 58 25 34  87 54 61 21
Clover     51 31 31 16  96 48 54 62  31 31 11 11
Clover     56 13 13 71  32 13 27 32  36 26 54 32
Clover     53 08 06 54  32 32 62 16
;
```

You can specify a SCORE statement to score the Crops data using the fitted model. The data together with the predicted values are saved into the data set Score1.

```
proc logistic data=Crops;
   model Crop=x1-x4 / link=glogit;
   score out=Score1;
run;
```

The OUTMODEL= option saves the fitted model information in a data set. In the following statements, the model is again fit, the data and the predicted values are saved into the data set Score2, and the model information is saved in the permanent SAS data set sasuser.CropModel.

```
proc logistic data=Crops outmodel=sasuser.CropModel;
   model Crop=x1-x4 / link=glogit;
   score data=Crops out=Score2;
run;
```

To score data without refitting the model, specify the INMODEL= option to identify a previously saved SAS data set of model information. In the following statements, the model is read from the sasuser.CropModel data set, and the data and the predicted values are saved into the data set Score3.

```
proc logistic inmodel=sasuser.CropModel;
   score data=Crops out=Score3;
run;
```

To set prior probabilities on the responses, specify the PRIOR= option to identify a SAS data set containing the response levels and their priors. In the following statements, the Prior data set contains the values of the response variable (because this example uses single-trial MODEL syntax) and a _PRIOR_ variable containing values proportional to the default priors. The model is fit, then the data and the predicted values are saved into the data set Score4.

```
data Prior;
   input Crop $ 1-10 _PRIOR_;
   datalines;
Clover     11
Corn        7
Cotton      6
Soybeans    6
Sugarbeets  6
;


proc logistic inmodel=sasuser.CropModel;
   score data=Crops prior=prior out=Score4;
run;
```

The data sets Score1, Score2, Score3, and Score4 are identical.

The following statements display the results of scoring the Crops data set in Output 42.13.1.

```
proc freq data=Score1;
   table F_Crop*I_Crop / nocol nocum nopercent;
run;
```

Output 42.13.1. Classification of Data used for Scoring

```
                              The FREQ Procedure

                          Table of F_Crop by I_Crop

         F_Crop(From: Crop)     I_Crop(Into: Crop)

         Frequency  |
         Row Pct    |Clover  |Corn    |Cotton  |Soybeans|Sugarbee|  Total
                    |        |        |        |        |ts      |
         -----------+--------+--------+--------+--------+--------+
         Clover     |      6 |      0 |      2 |      2 |      1 |     11
                    |  54.55 |   0.00 |  18.18 |  18.18 |   9.09 |
         -----------+--------+--------+--------+--------+--------+
         Corn       |      0 |      7 |      0 |      0 |      0 |      7
                    |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |
         -----------+--------+--------+--------+--------+--------+
         Cotton     |      4 |      0 |      1 |      1 |      0 |      6
                    |  66.67 |   0.00 |  16.67 |  16.67 |   0.00 |
         -----------+--------+--------+--------+--------+--------+
         Soybeans   |      1 |      1 |      1 |      3 |      0 |      6
                    |  16.67 |  16.67 |  16.67 |  50.00 |   0.00 |
         -----------+--------+--------+--------+--------+--------+
         Sugarbeets |      2 |      0 |      0 |      2 |      2 |      6
                    |  33.33 |   0.00 |   0.00 |  33.33 |  33.33 |
         -----------+--------+--------+--------+--------+--------+
         Total            13        8        4        8        3       36
```

Now the previously fit data set sasuser.CropModel is used to score the new observations in the Test data set. The following statements save the results of scoring the test data in the ScoredTest data set and produces Output 42.13.2.

```
data Test;
   input Crop $ 1-10 x1-x4;
   datalines;
Corn       16 27 31 33
Soybeans   21 25 23 24
Cotton     29 24 26 28
Sugarbeets 54 23 21 54
Clover     32 32 62 16
;
```

```
proc logistic noprint inmodel=sasuser.CropModel;
   score data=Test out=ScoredTest;
proc print data=ScoredTest label noobs;
   var F_Crop I_Crop P_Clover P_Corn P_Cotton P_Soybeans P_Sugarbeets;
run;
```

Output 42.13.2. Classification of Test Data

```
                                    Predicted        Predicted
                    Into:          Probability:     Probability:
   From: Crop       Crop            Crop=Clover       Crop=Corn

   Corn             Corn             0.00342         0.90067
   Soybeans         Soybeans         0.04801         0.03157
   Cotton           Clover           0.43180         0.00015
   Sugarbeets       Clover           0.66681         0.00000
   Clover           Cotton           0.41301         0.13386

     Predicted        Predicted         Predicted
   Probability:     Probability:      Probability:
    Crop=Cotton     Crop=Soybeans    Crop=Sugarbeets

      0.00500          0.08675           0.00416
      0.02865          0.82933           0.06243
      0.21267          0.07623           0.27914
      0.17364          0.00000           0.15955
      0.43649          0.00033           0.01631
```

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons, Inc.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Agresti, A. (1992), “A Survey of Exact Inference for Contingency Tables,” *Statistical Science*, 7, 131–177

Aitchison, J. and Silvey, S.D. (1957), “The Generalization of Probit Analysis to the Case of Multiple Responses,” *Biometrika*, 44, 131–40.

Albert, A. and Anderson, J.A. (1984), “On the Existence of Maximum Likelihood Estimates in Logistic Regression Models,” *Biometrika*, 71, 1–10.

Allison, P.D. (1982), “Discrete-Time Methods for the Analysis of Event Histories,” in *Sociological Methods and Research*, 15, ed S. Leinhardt, San Francisco: Jossey-Bass, 61–98.

Allison, P.D. (1999), *Logistic Regression Using the SAS System: Theory and Application*, Cary, NC: SAS Institute Inc.

Ashford, J.R. (1959), “An Approach to the Analysis of Data for Semi-Quantal Responses in Biology Response,” *Biometrics*, 15, 573–81.

Bartolucci, A.A. and Fraser, M.D. (1977), “Comparative Step-Up and Composite Test for Selecting Prognostic Indicator Associated with Survival,” *Biometrical Journal*, 19, 437–448.

Breslow, N.E. (1982), “Covariance Adjustment of Relative-Risk Estimates in Matched Studies,” *Biometrics*, 38, 661–672.

Breslow, N.E. and Day W. (1980), *Statistical Methods in Cancer Research, Volume 1–The Analysis of Case-Control Studies*, Lyon: IARC Scientific Publication No. 32.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman and Hall.

Cook, R.D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman and Hall.

Cox, D.R. (1970), *Analysis of Binary Data*, New York: Chapman and Hall.

Cox, D.R. and Snell, E.J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman and Hall.

DeLong, E.R., DeLong, D.M., and Clarke-Pearson, D.L. (1988), “Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: a Nonparametric Approach,” *Biometrics*, 44, 837–845.

Draper, C.C., Voller, A., and Carpenter, R.G. (1972), “The Epidemiologic Interpretation of Serologic Data in Malaria,” *American Journal of Tropical Medicine and Hygiene*, 21, 696–703.

Finney, D.J. (1947), “The Estimation from Individual Records of the Relationship between Dose and Quantal Response,” *Biometrika*, 34, 320–334.

Fleiss, J.L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons, Inc.

Freeman, D.H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker, Inc.

Furnival, G.M. and Wilson, R.W. (1974), “Regressions by Leaps and Bounds,” *Technometrics*, 16, 499–511.

Gail, M.H., Lubin, J.H., and Rubinstein, L.V. (1981), “Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Death Times,” *Biometrika*, 68, 703–707.

Hanley, J.A. and McNeil, B.J. (1982), “The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve,” *Radiology*, 143 29–36.

Harrell, F.E. (1986), “The LOGIST Procedure,” *SUGI Supplemental Library Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Hirji, K.F. (1992), “Computing Exact Distributions for Polytomous Response Data,” *Journal of the American Statistical Association*, 87, 487–492.

Hirji, K.F., Mehta, C.R., and Patel, N.R. (1987), “Computing Distributions for Exact Logistic Regression,” *Journal of the American Statistical Association*, 82, 1110–1117.

Hirji, K.F., Tsiatis, A.A., and Mehta, C.R. (1989), “Median Unbiased Estimation for Binary Data,” *American Statistician*, 43, 7–11.

Hosmer, D.W, Jr. and Lemeshow, S. (2000), *Applied Logistic Regression*, Second Edition, New York: John Wiley & Sons, Inc.

Howard, S. in the discussion of Cox, D.R. (1972), "Regression Models and Life Tables," *Journal of the Royal Statistical Society, Series B*, 34, 187–220.

Lachin, J. M., (2000), *Biostatistical Methods: The Assessment of Relative Risks*, New York: John Wiley & Sons, Inc.

Lancaster, H. O., (1961), "Significance Tests in Discrete Distributions," *Journal of the American Statistical Association*, 56, 223–234.

LaMotte, L.R., (2002), Personal communication, June 2002 e-mail.

Lawless, J.F. and Singhal, K. (1978), "Efficient Screening of Nonnormal Regression Models," *Biometrics*, 34, 318–327.

Lee, E.T. (1974), "A Computer Program for Linear Logistic Regression Analysis," *Computer Programs in Biomedicine*, 80–92.

McCullagh, P. and Nelder, J.A. (1989), *Generalized Linear Models*, London: Chapman Hall.

McFadden, D. (1974), "Conditional Logit Analysis of Qualitative Choice Behaviour" in *Frontiers in Econometrics*, edited by P. Zarembka, New York: Academic Press.

Mehta, C.R. and Patel, N.R. (1995), "Exact Logistic Regression: Theory and Examples," *Statistics in Medicine*, 14, 2143–2160.

Mehta, C.R., Patel, N. and Senchaudhuri, P. (1992), "Exact Stratified Linear Rank Tests for Ordered Categorical and Binary Data," *Journal of Computational and Graphical Statistics*, 1, 21–40.

Mehta, C.R., Patel, N. and Senchaudhuri, P. (2000), "Efficient Monte Carlo Methods for Conditional Logistic Regression," *Journal of the American Statistical Association*, 95, 99–108.

Moolgavkar, S.H., Lustbader, E.D., and Venzon, D.J. (1985), "Assessing the Adequacy of the Logistic Regression Model for Matched Case-Control Studies," *Statistics in Medicine*, 4, 425–435.

Naessens, J.M., Offord, K.P., Scott, W.F., and Daood, S.L., (1986), "The MCSTRAT Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC., SAS Institute Inc. 307–328.

Nagelkerke, N.J.D. (1991), "A Note on a General Definition of the Coefficient of Determination," *Biometrika*, 78, 691–692.

Nelder, J.A. and Wedderburn, R.W.M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 761–768.

Pregibon, D. (1981), "Logistic Regression Diagnostics," *Annals of Statistics*, 9, 705–724.

Pregibon, D. (1984), "Data Analytic Methods for Matched Case-Control Studies," *Biometrics*, 40, 639–651.

Prentice, P.L. and Gloeckler, L.A. (1978), "Regression Analysis of Grouped Survival Data with Applications to Breast Cancer Data," *Biometrics*, 34, 57–67.

Press, S.J. and Wilson, S. (1978), "Choosing Between Logistic Regression and Discriminant Analysis," *Journal of the American Statistical Association*, 73, 699–705.

Santner, T.J. and Duffy, E.D. (1986), "A Note on A. Albert and J.A. Anderson's Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 73, 755–758.

SAS Institute Inc. (1995), *Logistic Regression Examples Using the SAS System*, Cary, NC: SAS Institute Inc.

Stokes, M.E., Davis, C.S., and Koch, G.G. (2000), *Categorical Data Analysis Using the SAS System, Second Edition*, Cary, NC: SAS Institute Inc.

Storer, B.E. and Crowley, J. (1985), "A Diagnostic for Cox Regression and General Conditional Likelihoods," *Journal of the American Statistical Association*, 80, 139–147.

Venzon, D.J. and Moolgavkar, S.H. (1988), "A Method for Computing Profile-Likelihood Based Confidence Intervals," *Applied Statistics*, 37, 87–94.

Vollset, S.E., Hirji, K.F., and Afifi, A.A. (1991), "Evaluation of Exact and Asymptotic Interval Estimators in Logistic Analysis of Matched Case-Control Studies," *Biometrics*, 47, 1311–1325.

Walker, S.H. and Duncan, D.B. (1967), "Estimation of the Probability of an Event as a Function of Several Independent Variables," *Biometrika*, 54, 167–179.

Williams, D.A. (1982), "Extra-Binomial Variation in Logistic Linear Models," *Applied Statistics*, 31, 144–148.

Chapter 43
The MDS Procedure

Chapter Contents

Chapter 43
The MDS Procedure

Overview

Multidimensional scaling (MDS) is a class of methods that estimates the coordinates of a set of objects in a space of specified dimensionality that come from data measuring the distances between pairs of objects. A variety of models can be used that include different ways of computing distances and various functions relating the distances to the actual data. The MDS procedure fits two- and three-way, metric and nonmetric multidimensional scaling models. PROC MDS shares many of the features of the ALSCAL procedure (Young, Lewyckyj, and Takane 1986; Young 1982), as well as some features of the MLSCALE procedure (Ramsay 1986). Both PROC ALSCAL and PROC MLSCALE are described in the *SUGI Supplemental Library User's Guide, Version 5 Edition*.

The data for the MDS procedure consist of one or more square symmetric or asymmetric matrices of similarities or dissimilarities between *objects* or *stimuli* (Kruskal and Wish 1978, pp. 7−11). Such data are also called *proximity* data. In psychometric applications, each matrix typically corresponds to a *subject*, and models that fit different parameters for each subject are called *individual difference* models.

Missing values are allowed. In particular, if the data are all missing except within some off-diagonal rectangle, the analysis is called *unfolding*. There are, however, many difficulties intrinsic to unfolding models (Heiser, 1981). PROC MDS does not perform external unfolding; for analyses requiring external unfolding, use the TRANSREG procedure instead.

The MDS procedure estimates the following parameters by nonlinear least squares:

configuration

the coordinates of each object in a Euclidean (Kruskal and Wish 1978, pp. 17−19) or weighted Euclidean space (Kruskal and Wish 1978, pp. 61−63) of one or more dimensions

dimension coefficients

for each data matrix, the coefficients that multiply each coordinate of the *common* or *group* weighted Euclidean space to yield the *individual* unweighted Euclidean space. These coefficients are the square roots of the *subject weights* (Kruskal and Wish 1978, pp. 61−63). A plot of the dimension coefficients is directly interpretable in that it shows how a unit square in the group space is transformed to a rectangle in each individual space. A plot of subject weights has no such simple interpretation. The weighted Euclidean model is related to the INDSCAL model (Carroll and Chang 1970).

transformation parameters intercept, slope, or exponent in a linear, affine, or power transformation relating the distances to the data (Kruskal and Wish 1978, pp. 19–22). For a nonmetric analysis, monotone transformations involving no explicit parameters are used (Kruskal and Wish 1978, pp. 22–25). For a discussion of metric versus nonmetric transformations, refer to Kruskal and Wish (1978, pp. 76–78).

Depending on the LEVEL= option, PROC MDS fits either a regression model of the form

$$fit(datum) = fit(trans(distance)) + error$$

or a measurement model of the form

$$fit(trans(datum)) = fit(distance) + error$$

where

fit is a predetermined power or logarithmic transformation specified by the FIT= option.

trans is an estimated (“optimal”) linear, affine, power, or monotone transformation specified by the LEVEL= option.

datum is a measure of the similarity or dissimilarity of two objects or stimuli.

distance is a distance computed from the estimated coordinates of the two objects and estimated dimension coefficients in a space of one or more dimensions. If there are no dimension coefficients (COEF=IDENTITY), this is an unweighted Euclidean distance. If dimension coefficients are used (COEF=DIAGONAL), this is a weighted Euclidean distance where the weights are the squares of the dimension coefficients; alternatively, you can multiply each dimension by its coefficient and compute an unweighted Euclidean distance.

error is an error term assumed to have an approximately normal distribution and to be independently and identically distributed for all data. Under these assumptions, least-squares estimation is statistically appropriate.

For an introduction to multidimensional scaling, refer to Kruskal and Wish (1978) and Arabie, Carroll, and DeSarbo (1987). A more advanced treatment is given by Young (1987). Many practical issues of data collection and analysis are discussed in Schiffman, Reynolds, and Young (1981). The fundamentals of psychological measurement, including both unidimensional and multidimensional scaling, are expounded by Torgerson (1958). Nonlinear least-squares estimation of PROC MDS models is discussed in Null and Sarle (1982).

Getting Started

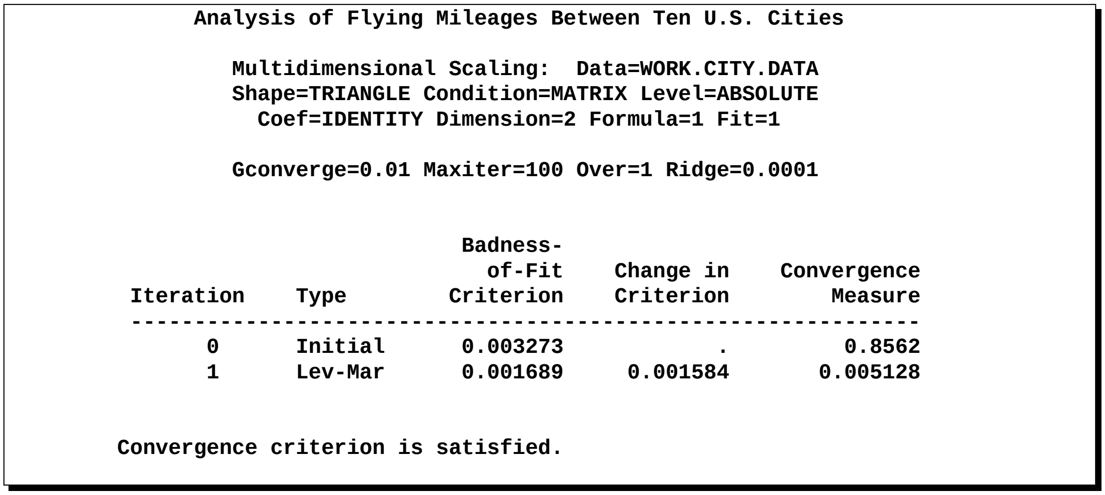
The simplest application of PROC MDS is to reconstruct a map from a table of distances between points on the map (Kruskal and Wish 1978, pp. 7–9). For example, the following DATA step reads a table of flying mileages between ten U.S. cities:

```
data   city;
   title 'Analysis of Flying Mileages Between Ten U.S. Cities';
   input  (atlanta chicago denver houston losangeles
           miami newyork sanfran seattle washdc) (5.)
           @56 city $15.;
   datalines;
    0                                                  Atlanta
  587    0                                             Chicago
 1212  920    0                                        Denver
  701  940  879    0                                   Houston
 1936 1745  831 1374    0                              Los Angeles
  604 1188 1726  968 2339    0                         Miami
  748  713 1631 1420 2451 1092    0                    New York
 2139 1858  949 1645  347 2594 2571    0               San Francisco
 2182 1737 1021 1891  959 2734 2408  678    0          Seattle
  543  597 1494 1220 2300  923  205 2442 2329    0     Washington D.C.
;
```

Since the flying mileages are very good approximations to Euclidean distance, no transformation is needed to convert distances from the model to data. The analysis can therefore be done at the absolute level of measurement, as displayed in the following PROC MDS step (LEVEL=ABSOLUTE). An output data set containing the estimated configuration (coordinates on the map) is created with the OUT= option and then used to display the map using the %PLOTIT macro. The ID statement copies the names of the cities to the OUT= data set so that they can be used on the plot. The following statements produce Figure 43.1:

```
proc mds data=city level=absolute out=out;
   id city;
run;
```

By default, PROC MDS displays only the iteration history. In this example, only one iteration is required. The badness-of-fit criterion 0.001689 indicates that the data fit the model extremely well.



```
               Analysis of Flying Mileages Between Ten U.S. Cities

                  Multidimensional Scaling:  Data=WORK.CITY.DATA
                  Shape=TRIANGLE Condition=MATRIX Level=ABSOLUTE
                    Coef=IDENTITY Dimension=2 Formula=1 Fit=1

                  Gconverge=0.01 Maxiter=100 Over=1 Ridge=0.0001


                                    Badness-
                                      of-Fit    Change in    Convergence
          Iteration     Type       Criterion    Criterion        Measure
          -------------------------------------------------------------
                  0     Initial     0.003273            .         0.8562
                  1     Lev-Mar     0.001689     0.001584       0.005128


         Convergence criterion is satisfied.
```

Figure 43.1. Iteration History from PROC MDS

In order for the plot to represent the geometry of the configuration correctly, the axes must be scaled so that a given physical distance on one axis represents the same number of units as the same physical distance on the other axis. You can use the %PLOTIT macro to obtain such a scaling by specifying the VTOH= option. The VTOH= option indicates the ratio of the vertical distance between lines to the horizontal distance between characters; obtaining a suitable value may require some experimentation. The %PLOTIT macro can create graphical scatter plots with iteratively derived optimal label placement. See Appendix B, "Using the %PLOTIT Macro," for more information on the %PLOTIT macro.

The following statements produce Figure 43.2:

```
%plotit(data=out, datatype=mds, labelvar=city,
        vtoh=1.75, labfont=swissb);
run;
```

While PROC MDS can recover the relative positions of the cities, it cannot determine absolute location or orientation. In this case, north is toward the bottom of the plot.

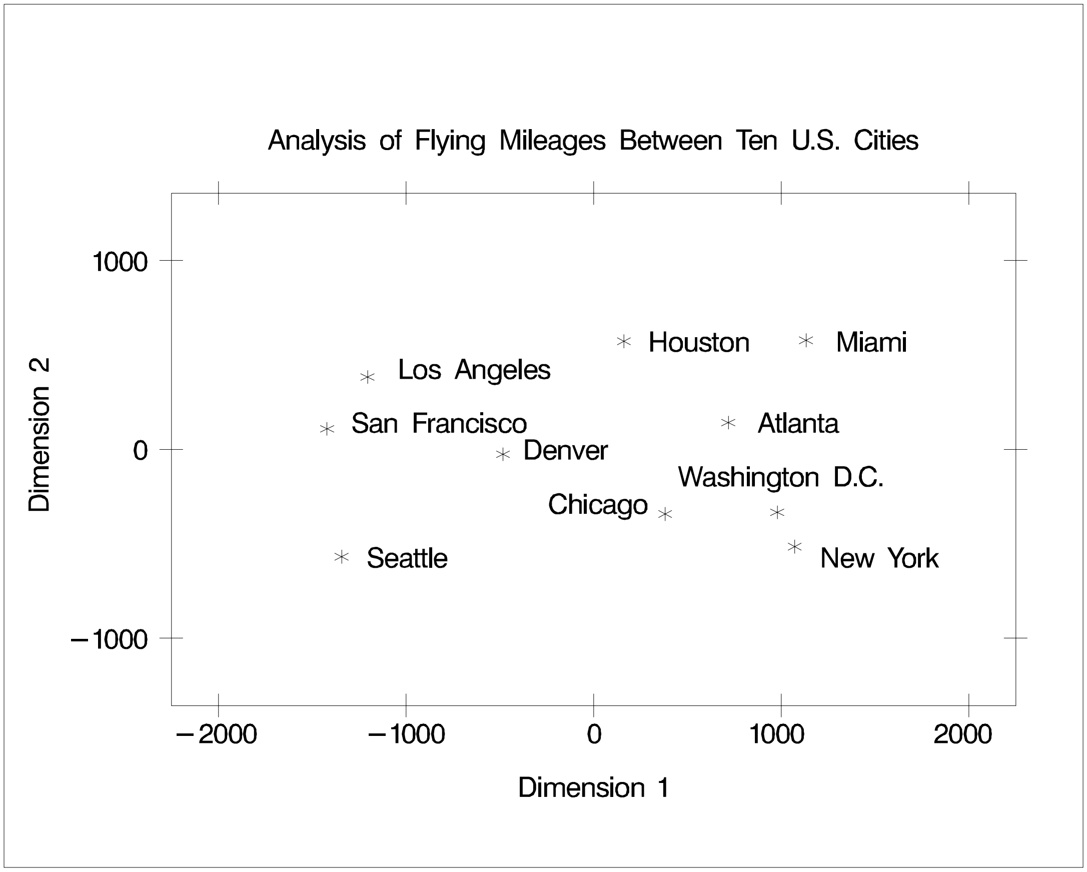


Figure 43.2. Plot of Estimated Configuration Using the %PLOTIT Macro

Syntax

You can specify the following statements with the MDS procedure:

PROC MDS < *options* > **;**
  VAR *variables* **;**
  INVAR *variables* **;**
  ID | **OBJECT** *variable* **;**
  MATRIX | **SUBJECT** *variable* **;**
  WEIGHT *variables* **;**
  BY *variables* **;**

The PROC MDS statement is required. All other statements are optional.

PROC MDS Statement

PROC MDS < *options* > **;**

By default, the only result produced by the MDS procedure is the iteration history. Hence, you should always specify one or more options for output data sets (OUT=, OUTFIT=, and OUTRES=) or displayed output (such as PFINAL). PROC MDS does not produce any plots; to produce plots, use the output data sets with PROC PLOT or PROC GPLOT.

The types of estimates written to the OUT= data set are determined by the OCONFIG, OCOEF, OTRANS, and OCRIT options. If you do not specify any of these four options, the estimates of all the parameters of the PROC MDS model and the value of the badness-of-fit criterion appear in the OUT= data set. If you specify one or more of these options, only the information requested by the specified options appear in the OUT= data set. Also, the OITER option causes these statistics to be written to the OUT= data set after initialization and on each iteration, as well as after the iterations have terminated.

Displayed output is controlled by the interaction of the PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options with the PININ, PINIT, PITER, and PFINAL options. The PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options specify *which* estimates and fit statistics are to be displayed. The PININ, PINIT, PITER, and PFINAL options specify *when* the estimates and fit statistics are to be displayed. If you specify at least one of the PCONFIG, PCOEF, PTRANS, PFIT and PFITROW options but none of the PININ, PINIT, PITER, and PFINAL options, the final results (PFINAL) are displayed. If you specify at least one of the PININ, PINIT, PITER, and PFINAL options but none of the PCONFIG, PCOEF, PTRANS, PFIT and PFITROW options, all estimates (PCONFIG, PCOEF, PTRANS) and the fit statistics for each matrix and for the entire sample (PFIT) are displayed. If you do not specify any of these nine options, no estimates or fit statistics are displayed (except the badness-of-fit criterion in the iteration history).

ALTERNATE | ALT=NONE | NO | N
ALTERNATE | ALT=MATRIX | MAT | M | SUBJECT | SUB | S
ALTERNATE | ALT=ROW | R <=*n*>

determines what form of alternating-least-squares algorithm is used. The default depends on the amount of memory available. The following ALTERNATE= options are listed in order of decreasing memory requirements:

ALT=NONE causes all parameters to be adjusted simultaneously on each iteration. This option is usually best for a small number of subjects and objects.

ALT=MATRIX adjusts all the parameters for the first subject, then all the parameters for the second subject, and so on, and finally adjusts all parameters that do not correspond to a subject, such as coordinates and unconditional transformations. This option usually works best for a large number of subjects with a small number of objects.

ALT=ROW — treats subject parameters the same way as the ALTERNATE=MATRIX option but also includes separate stages for unconditional parameters and for subsets of the objects. The ALT=ROW option usually works best for a large number of objects.

Specifying ALT=ROW=*n* divides the objects into subsets of *n* objects each, except possibly for one subset when *n* does not divide the number of objects evenly. If you omit =*n*, the number of objects in the subsets is determined from the amount of memory available. The smaller the value of *n*, the less memory is required.

When you specify the LEVEL=ORDINAL option, the monotone transformation is always computed in a separate stage and is listed as a separate iteration in the iteration history. In this case, estimation is done by iteratively reweighted least squares. The weights are recomputed according to the FORMULA= option on each monotone iteration; hence, it is possible for the badness-of-fit criterion to increase after a monotone iteration.

COEF=IDENTITY | IDEN | I
COEF=DIAGONAL | DIAG | D

specifies the type of matrix for the dimension coefficients.

COEF=IDENTITY — is the default, which yields Euclidean distances.

COEF=DIAGONAL — produces weighted Euclidean distances, in which each subject is allowed differential weights for the dimensions. The dimension coefficients that PROC MDS outputs are related to the square roots of what are called subject weights in PROC ALSCAL; the normalization in PROC MDS also differs from that in PROC ALSCAL. The weighted Euclidean model is related to the INDSCAL model (Carroll and Chang 1970).

CONDITION | COND=UN | U
CONDITION | COND=MATRIX | MAT | M | SUBJECT | SUB | S
CONDITION | COND=ROW | R

specifies the conditionality of the data (Young 1987, pp. 60−63). The default is CONDITION=MATRIX.

The data are divided into disjoint subsets called *partitions*. Within each partition, a separate transformation is applied, as specified by the LEVEL= option.

COND=UN — puts all the data into a single partition.

COND=MATRIX — makes each data matrix a partition.

COND=ROW — makes each row of each data matrix a partition.

The CONDITION= option also determines the default value for the SHAPE= option. If you specify the CONDITION=ROW option and omit the SHAPE= option, each data matrix is stored as a square and possibly asymmetric matrix. If you specify the CONDITION=UN or CONDITION=MATRIX option and omit the SHAPE= option, only one triangle is stored. See the SHAPE= option on page 2484 for details.

CONVERGE | CONV=*p*

sets both the gradient convergence criterion and the monotone convergence criterion to *p*, where $0 \leq p \leq 1$. The default is CONVERGE=.01; smaller values may greatly increase the number of iterations required. Values less than .0001 may be impossible to satisfy because of the limits of machine precision. See the GCONVERGE= (page 2480) and MCONVERGE= (page 2481) options.

CUTOFF=*n*

causes data less than *n* to be replaced by missing values. The default is CUTOFF=0.

DATA=*SAS-data-set*

specifies the SAS data set containing one or more square matrices to be analyzed. In typical psychometric data, each matrix contains judgments from one subject, so there is a one-to-one correspondence between data matrices and subjects.

The data matrices contain similarity or dissimilarity measurements to be modeled and, optionally, weights for these data. The data are generally assumed to be dissimilarities unless you use the SIMILAR option. However, if there are nonmissing diagonal values and these values are predominantly larger than the off-diagonal values, the data are assumed to be similarities and are treated as if the SIMILAR option is specified. The diagonal elements are not otherwise used in fitting the model.

Each matrix must have exactly the same number of observations as the number of variables specified by the VAR statement or determined by defaults. This number is the number of objects or stimuli.

The first observation and variable are assumed to contain data for the first object, the second observation and variable are assumed to contain data for the second object, and so on.

When there are two or more matrices, the observations in each matrix must correspond to the same objects in the same order as in the first matrix.

The matrices can be symmetric or asymmetric, as specified by the SHAPE= option.

DECIMALS | DEC=*n*

specifies how many decimal places to use when displaying the parameter estimates and fit statistics. The default is DECIMALS=2, which is generally reasonable except in conjunction with the LEVEL=ABSOLUTE option and very large or very small data.

DIMENSION | DIMENS | DIM=*n* $<$ **TO** *m* $<$ **BY=***i* $>>$

specifies the number of dimensions to use in the MDS model, where $1 \leq n, m <$ *number of objects*. The parameter *i* can be either positive or negative but not zero. If you specify different values for *n* and *m*, a separate model is fitted for each requested dimension. If you specify only DIMENSION=*n*, then only *n* dimensions are fitted. The default is DIMENSION=2 if there are three or more objects; otherwise, DIMENSION=1 is the only valid specification. The analyses for each number of dimensions are done independently. For information on choosing the dimensionality, refer to Kruskal and Wish (1978, pp. 48−60).

EPSILON | EPS=*n*

specifies a number *n*, $0 < n < 1$, that determines the amount added to squared distances computed from the model to avoid numerical problems such as division by 0. This amount is computed as ϵ equal to n times the mean squared distance in the initial configuration. The distance in the MDS model is thus computed as

$$distance = \sqrt{sqdist + \epsilon}$$

where *sqdist* is the squared Euclidean distance or the weighted squared Euclidean distance.

The default is EPSILON=1E−12, which is small enough to have no practical effect on the estimates unless the FIT= value is nonpositive and there are dissimilarities that are very close to 0. Hence, when the FIT= value is nonpositive, dissimilarities less than *n* times 100 times the maximum dissimilarity are disallowed.

FIT=DISTANCE | DIS | D
FIT=SQUARED | SQU | S
FIT=LOG | L
FIT=*n*

specifies a predetermined (not estimated) transformation to apply to both sides of the MDS model before the error term is added.

The default is FIT=DISTANCE or, equivalently, FIT=1, which fits data to distances.

The option FIT=SQUARED or FIT=2 fits squared data to squared distances. This gives greater importance to large data and distances and lesser importance to small data and distances in fitting the model.

The FIT=LOG or FIT=0 option fits log data to log distances. This gives lesser importance to large data and distances and greater importance to small data and distances in fitting the model.

In general, the FIT=*n* option fits *n*th-power data to *n*th-power distances. Values of *n* that are large in absolute value can cause floating-point overflows.

If the FIT= value is 0 or negative, the data must be strictly positive (see the EPSILON= option). Negative data may produce strange results with any value other than FIT=1.

FORMULA | FOR=*0* | OLS | O
FORMULA | FOR=*1* | USS | U
FORMULA | FOR=*2* | CSS | C

determines how the badness-of-fit criterion is standardized in correspondence with stress formulas 1 and 2 (Kruskal and Wish 1978, pp. 24−26). The default is FORMULA=1 unless you specify FIT=LOG, in which case the default is FORMULA=2. Data partitions are defined by the CONDITION= option.

FORMULA=0 fits a regression model by ordinary least squares (Null and Sarle 1982) without standardizing the partitions; this option cannot be used with the LEVEL=ORDINAL option. The badness-of-fit criterion is the square root of the error sum of squares.

FORMULA=1 standardizes each partition by the uncorrected sum of squares of the (possibly transformed) data; this option should not be used with the FIT=LOG option. With the FIT=DISTANCE and LEVEL=ORDINAL options, this is equivalent to Kruskal's stress formula 1 or an obvious generalization thereof. With the FIT=SQUARED and LEVEL=ORDINAL options, this is equivalent to Young's s-stress formula 1 or an obvious generalization thereof. The badness-of-fit criterion is analogous to $\sqrt{1 - R^2}$, where R is a multiple correlation about the origin.

FORMULA=2 standardizes each partition by the corrected sum of squares of the (possibly transformed) data; this option is the recommended method for unfolding. With the FIT=DISTANCE and LEVEL=ORDINAL options, this is equivalent to Kruskal's stress formula 2 or an obvious generalization thereof. With the FIT=SQUARED and LEVEL=ORDINAL options, this is equivalent to Young's s-stress formula 2 or an obvious generalization thereof. The badness-of-fit criterion is analogous to $\sqrt{1 - R^2}$, where R is a multiple correlation computed with a denominator corrected for the mean.

GCONVERGE | GCONV=*p*

sets the gradient convergence criterion to *p*, where $0 \leq p \leq 1$. The default is GCONVERGE=0.01; smaller values may greatly increase the number of iterations required. Values less than 0.0001 may be impossible to satisfy because of the limits of machine precision.

The gradient convergence measure is the multiple correlation of the Jacobian matrix with the residual vector, uncorrected for the mean. See the CONVERGE= (page 2478) and MCONVERGE= (page 2481) options.

INAV=DATA | D
INAV=SSCP | S

affects the computation of initial coordinates. The default is INAV=DATA.

INAV=DATA computes a weighted average of the data matrices. Its value is estimated only if an element is missing from every data matrix. The weighted average of the data matrices with missing values filled in is then converted to a scalar products matrix (or what would be a scalar products matrix if the fit were perfect), from which the initial coordinates are computed.

INAV=SSCP estimates missing values in each data matrix and converts each data matrix to a scalar products matrix. The initial coordinates are computed from the unweighted average of the scalar products matrices.

INITIAL | IN=*SAS-data-set*

specifies a SAS data set containing initial values for some or all of the parameters of the MDS model. If the INITIAL= option is omitted, the initial values are computed from the data.

LEVEL=ABSOLUTE | ABS | A
LEVEL=RATIO | RAT | R
LEVEL=INTERVAL | INT | I
LEVEL=LOGINTERVAL | LOG | L
LEVEL=ORDINAL | ORD | O

specifies the measurement level of the data and hence the type of estimated (optimal) transformations applied to the data or distances (Young 1987, pp. 57−60; Krantz et. al. 1971, pp. 9−12) within each partition as specified by the CONDITION= option. LEVEL=ORDINAL specifies a nonmetric analysis, while all other LEVEL= options specify metric analyses. The default is LEVEL=ORDINAL.

LEVEL=ABSOLUTE	allows no optimal transformations. Hence, the distinction between regression and measurement models is irrelevant.
LEVEL=RATIO	fits a regression model in which the distances are multiplied by a slope parameter in each partition (a linear transformation). In this case, the regression model is equivalent to the measurement model with the slope parameter reciprocated.
LEVEL=INTERVAL	fits a regression model in which the distances are multiplied by a slope parameter and added to an intercept parameter in each partition (an affine transformation). In this case, the regression and measurement models differ if there is more than one partition.
LEVEL=LOGINTERVAL	fits a regression model in which the distances are raised to a power and multiplied by a slope parameter in each partition (a power transformation).
LEVEL=ORDINAL	fits a measurement model in which a least-squares monotone increasing transformation is applied to the data in each partition. At the ordinal measurement level, the regression and measurement models differ.

MAXITER | ITER=*n*

specifies the maximum number of iterations, where $n \geq 0$. The default is MAXITER=100.

MCONVERGE | MCONV=*p*

sets the monotone convergence criterion to *p*, where $0 \leq p \leq 1$, for use with the LEVEL=ORDINAL option. The default is MCONVERGE=0.01; if you want greater precision, MCONVERGE=0.001 is usually reasonable, but smaller values may greatly increase the number of iterations required.

The monotone convergence criterion is the Euclidean norm of the change in the optimally scaled data divided by the Euclidean norm of the optimally scaled data, averaged across partitions defined by the CONDITION= option. See the CONVERGE= (page 2478) and GCONVERGE= (page 2480) options.

MINCRIT | CRITMIN=*n*

causes iteration to terminate when the badness-of-fit criterion is less than or equal to *n*, where $n \geq 0$. The default is MINCRIT=1E$-$6.

NEGATIVE

allows slopes or powers to be negative with the LEVEL=RATIO, INTERVAL, or LOGINTERVAL option.

NONORM

suppresses normalization of the initial and final estimates.

NOPHIST | NOPRINT | NOP

suppresses the output of the iteration history.

NOULB

causes missing data to be estimated during initialization by the average nonmissing value, where the average is computed according to the FIT= option. Otherwise, missing data are estimated by interpolating between the Rabinowitz (1976) upper and lower bounds.

OCOEF

writes the dimension coefficients to the OUT= data set. See the OUT= option for interactions with other options.

OCONFIG

writes the coordinates of the objects to the OUT= data set. See the OUT= option for interactions with other options.

OCRIT

writes the badness-of-fit criterion to the OUT= data set. See the OUT= option for interactions with other options.

OITER | OUTITER

writes current values to the output data sets after initialization and on every iteration. Otherwise, only the final values are written to any output data sets. See the OUT=, OUTFIT=, and OUTRES= options.

OTRANS

writes the transformation parameter estimates to the OUT= data set if any such estimates are computed. There are no transformation parameters with the LEVEL=ORDINAL option. See the OUT= option for interactions with other options.

OUT=*SAS-data-set*

creates a SAS data set containing, by default, the estimates of all the parameters of the PROC MDS model and the value of the badness-of-fit criterion. However, if you specify one or more of the OCONFIG, OCOEF, OTRANS, and OCRIT options, only the information requested by the specified options appears in the OUT= data set. See also the OITER option.

OUTFIT=*SAS-data-set*

creates a SAS data set containing goodness-of-fit and badness-of-fit measures for each partition as well as for the entire data set. See also the OITER option.

OUTRES=*SAS-data-set*

creates a SAS data set containing one observation for each nonmissing datum from the DATA= data set. Each observation contains the original datum, the estimated distance computed from the MDS model, transformed data and distances, and the residual. See also the OITER option.

OVER=*n*

specifies the maximum overrelaxation factor, where $n \geq 1$. Values between 1 and 2 are generally reasonable. The default is OVER=2 with the LEVEL=ORDINAL, ALTERNATE=MATRIX, or ALTERNATE=ROW option; otherwise, the default is OVER=1. Use this option only if you have convergence problems.

PCOEF

produces the estimated dimension coefficients.

PCONFIG

produces the estimated coordinates of the objects in the configuration.

PDATA

displays each data matrix.

PFINAL

displays final estimates.

PFIT

displays the badness-of-fit criterion and various types of correlations between the data and fitted values for each data matrix, as well as for the entire sample.

PFITROW

displays the badness-of-fit criterion and various types of correlations between the data and fitted values for each row as well as for each data matrix and for the entire sample. This option works only with the CONDITION=ROW option.

PINAVDATA

displays the sum of the weights and the weighted average of the data matrices computed during initialization with the INAV=DATA option.

PINEIGVAL

displays the eigenvalues computed during initialization.

PINEIGVEC

displays the eigenvectors computed during initialization.

PININ

displays values read from the INITIAL= data set. Since these values may be incomplete, the PFIT and PFITROW options do not apply.

PINIT

displays initial values.

PITER

displays estimates on each iteration.

PTRANS

displays the estimated transformation parameters if any are computed. There are no transformation parameters with the LEVEL=ORDINAL option.

RANDOM<**=***seed*>

causes initial coordinate values to be pseudorandom numbers. In one dimension, the pseudorandom numbers are uniformly distributed on an interval. In two or more dimensions, the pseudorandom numbers are uniformly distributed on the circumference of a circle or the surface of a (hyper)sphere.

RIDGE=*n*

specifies the initial ridge value, where $n \geq 0$. The default is RIDGE=1E−4.

If you get a floating-point overflow in the first few iterations, specify a larger value such as RIDGE=0.01 or RIDGE=1 or RIDGE=100.

If you know that the initial estimates are very good, using RIDGE=0 may speed convergence.

SHAPE=TRIANGULAR | TRIANGLE | TRI | T
SHAPE=SQUARE | SQU | S

determines whether the entire data matrix for each subject or only one triangle of the matrix is stored and analyzed. If you specify the CONDITION=ROW option, the default is SHAPE=SQUARE. Otherwise, the default is SHAPE=TRIANGLE.

SHAPE=SQUARE	causes the entire matrix to be stored and analyzed. The matrix can be asymmetric.
SHAPE=TRIANGLE	causes only one triangle to be stored. However, PROC MDS reads both upper and lower triangles to look for nonmissing values and to symmetrize the data if needed. If corresponding elements in the upper and lower triangles both contain nonmissing values, only the average of the two values is stored and analyzed (Kruskal and Wish 1978, p. 74). Also, if an OUTRES= data set is requested, only the average of the two corresponding elements is output.

SIMILAR | SIM<**=***max*>

causes the data to be treated as similarity measurements rather than dissimilarities. If =*max* is not specified, each datum is converted to a dissimilarity by subtracting it from the maximum value in the data set or BY group. If =*max* is specified, each datum is subtracted from the maximum of *max* and the data. The diagonal data are included in computing these maxima.

By default, the data are assumed to be dissimilarities unless there are nonmissing diagonal values and these values are predominantly larger than the off-diagonal values. In this case, the data are assumed to be similarities and are treated as if the SIMILAR option is specified.

SINGULAR=*p*

specifies the singularity criterion p, $0 \leq p \leq 1$. The default is SINGULAR=1E-8.

UNTIE

allows tied data to be assigned different optimally scaled values with the LEVEL=ORDINAL option. Otherwise, tied data are assigned equal optimally scaled values. The UNTIE option has no effect with values of the LEVEL= option other than LEVEL=ORDINAL.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MDS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MDS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

If the INITIAL= data set contains the BY variables, the BY groups must appear in the same order as in the DATA= data set. If the BY variables are not in the INITIAL= data set, the entire data set is used to provide initial values for each BY group in the DATA= data set.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

ID Statement

ID | **OBJECT** | **OBJ** *variable* **;**

The ID statement specifies a variable in the DATA= data set that contains descriptive labels for the objects. The labels are used in the output and are copied to the OUT= data set. If there is more than one data matrix, only the ID values from the observations containing the first data matrix are used.

The ID variable is not used to establish any correspondence between observations and variables.

If the ID statement is omitted, the variable labels or names are used as object labels.

INVAR Statement

INVAR *variables* **;**

The INVAR statement specifies the numeric variables in the INITIAL= data set that contain initial parameter estimates. The first variable corresponds to the first dimension, the second variable to the second dimension, and so on.

If the INVAR statement is omitted, the variables DIM1, . . . , DIM*m* are used, where *m* is the maximum number of dimensions.

MATRIX Statement

MATRIX | **MAT** | **SUBJECT** | **SUB** *variable* **;**

The MATRIX statement specifies a variable in the DATA= data set that contains descriptive labels for the data matrices or subjects. The labels are used in the output and are copied to the OUT= and OUTRES= data sets. Only the first observation from each data matrix is used to obtain the label for that matrix.

If the MATRIX statement is omitted, the matrices are labeled 1, 2, 3, and so on.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables in the DATA= data set that contain similarity or dissimilarity measurements on a set of objects or stimuli. Each variable corresponds to one object.

If the VAR statement is omitted, all numeric variables that are not specified in another statement are used.

To analyze a subset of the objects in a data set, you can specify the variable names corresponding to the columns in the subset, but you must also use a DATA step or a WHERE clause to specify the rows in the subset. PROC MDS expects to read one or more square matrices, and you must ensure that the rows in the data set correctly correspond to the columns in number and order.

WEIGHT Statement

WEIGHT *variables* **;**

The WEIGHT statement specifies numeric variables in the DATA= data set that contain weights for each similarity or dissimilarity measurement. These weights are used to compute weighted least-squares estimates. The number of WEIGHT variables must be the same as the number of VAR variables, and the variables in the WEIGHT statement must be in the same order as the corresponding variables in the VAR statement.

If the WEIGHT statement is omitted, all data within a partition are assigned equal weights.

Data with 0 or negative weights are ignored in fitting the model but are included in the OUTRES= data set and in monotone transformations.

Details

Formulas

The following notation is used:

A_p	intercept for partition *p*
B_p	slope for partition *p*
C_p	power for partition *p*
D_{rcs}	distance computed from the model between objects *r* and *c* for subject *s*
F_{rcs}	data weight for objects *r* and *c* for subject *s* obtained from the *c*th WEIGHT variable, or 1 if there is no WEIGHT statement
f	value of the FIT= option
N	number of objects
O_{rcs}	observed dissimilarity between objects *r* and *c* for subject *s*
P_{rcs}	partition index for objects *r* and *c* for subject *s*
Q_{rcs}	dissimilarity after applying any applicable estimated transformation for objects *r* and *c* for subject *s*
R_{rcs}	residual for objects *r* and *c* for subject *s*
S_p	standardization factor for partition *p*
$T_p(\cdot)$	estimated transformation for partition *p*
V_{sd}	coefficient for subject *s* on dimension *d*
X_{nd}	coordinate for object *n* on dimension *d*

Summations are taken over nonmissing values.

Distances are computed from the model as

$$\begin{aligned} D_{rcs} &= \sqrt{\sum_d (X_{rd} - X_{cd})^2} && \text{for COEF=IDENTITY: Euclidean distance} \\ &= \sqrt{\sum_d V_{sd}^2 (X_{rd} - X_{cd})^2} && \text{for COEF=DIAGONAL: weighted Euclidean distance} \end{aligned}$$

Partition indexes are

$$\begin{aligned} P_{rcs} &= 1 && \text{for CONDITION=UN} \\ &= s && \text{for CONDITION=MATRIX} \\ &= (s-1)N + r && \text{for CONDITION=ROW} \end{aligned}$$

The estimated transformation for each partition is

$$\begin{aligned} T_p(d) &= d && \text{for LEVEL=ABSOLUTE} \\ &= B_p d && \text{for LEVEL=RATIO} \\ &= A_p + B_p d && \text{for LEVEL=INTERVAL} \\ &= B_p d^{C_p} && \text{for LEVEL=LOGINTERVAL} \end{aligned}$$

For LEVEL=ORDINAL, $T_p(\cdot)$ is computed as a least-squares monotone transformation.

For LEVEL=ABSOLUTE, RATIO, or INTERVAL, the residuals are computed as

$$\begin{aligned} Q_{rcs} &= O_{rcs} \\ R_{rcs} &= Q_{rcs}^f - [T_{P_{rcs}}(D_{rcs})]^f \end{aligned}$$

For LEVEL=ORDINAL, the residuals are computed as

$$\begin{aligned} Q_{rcs} &= T_{P_{rcs}}(O_{rcs}) \\ R_{rcs} &= Q_{rcs}^f - D_{rcs}^f \end{aligned}$$

If f is 0, then natural logarithms are used in place of the fth powers.

For each partition, let

$$U_p = \frac{\displaystyle\sum_{r,c,s} F_{rcs}}{\displaystyle\sum_{r,c,s|P_{rcs}=p} F_{rcs}}$$

and

$$\overline{Q}_p = \frac{\displaystyle\sum_{r,c,s|P_{rcs}=p} Q_{rcs} F_{rcs}}{\displaystyle\sum_{r,c,s|P_{rcs}=p} F_{rcs}}$$

Then the standardization factor for each partition is

$$\begin{aligned} S_p &= 1 && \text{for FORMULA=0} \\ &= U_p \sum_{r,c,s|P_{rcs}=p} Q_{rcs}^2 F_{rcs} && \text{for FORMULA=1} \\ &= U_p \sum_{r,c,s|P_{rcs}=p} (Q_{rcs} - \overline{Q}_p)^2 F_{rcs} && \text{for FORMULA=2} \end{aligned}$$

The badness-of-fit criterion that the MDS procedure tries to minimize is

$$\sqrt{\sum_{r,c,s} \frac{R_{rcs}^2 F_{rcs}}{S_{P_{rcs}}}}$$

OUT= Data Set

The OUT= data set contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions
- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains. This variable contains a missing value for observations that pertain to the data set as a whole and not to a particular matrix, such as the coordinates (_TYPE_='CONFIG').
- _TYPE_, a character variable of length 10 identifying the type of information in the observation

  The values of _TYPE_ are as follows:

CONFIG	the estimated coordinates of the configuration of objects
DIAGCOEF	the estimated dimension coefficients for COEF=DIAGONAL
INTERCEPT	the estimated intercept parameters
SLOPE	the estimated slope parameters
POWER	the estimated power parameters
CRITERION	the badness-of-fit criterion

- _LABEL_ or the variable in the ID statement, containing the variable label or value of the ID variable of the object to which the observation pertains. This variable contains a missing value for observations that do not pertain to a particular object or dimension.
- _NAME_, a character variable of length 8 containing the variable name of the object or dimension to which the observation pertains. This variable contains a missing value for observations that do not pertain to a particular object or dimension.
- DIM1, . . . , DIM*m*, where *m* is the maximum number of dimensions

OUTFIT= Data Set

The OUTFIT= data set contains various measures of goodness and badness of fit. There is one observation for the entire sample plus one observation for each matrix. For the CONDITION=ROW option, there is also one observation for each row.

The OUTFIT= data set contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions
- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains
- _LABEL_ or the variable in the ID statement, containing the variable label or value of the ID variable of the object to which the observation pertains when CONDITION=ROW
- _NAME_, a character variable of length 8 containing the variable name of the object or dimension to which the observation pertains when CONDITION=ROW
- N, the number of nonmissing data
- WEIGHT, the weight of the partition
- CRITER, the badness-of-fit criterion
- DISCORR, the correlation between the transformed data and the distances for LEVEL=ORDINAL or the correlation between the data and the transformed distances otherwise
- UDISCORR, the correlation uncorrected for the mean between the transformed data and the distances for LEVEL=ORDINAL or the correlation between the data and the transformed distances otherwise
- FITCORR, the correlation between the fit-transformed data and the fit-transformed distances
- UFITCORR, the correlation uncorrected for the mean between the fit-transformed data and the fit-transformed distances

OUTRES= Data Set

The OUTRES= data set has one observation for each nonmissing datum. It contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions

- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains
- _ROW_, containing the variable label or value of the ID variable of the row to which the observation pertains
- _COL_, containing the variable label or value of the ID variable of the column to which the observation pertains
- DATA, the original datum
- TRANDATA, the optimally transformed datum when LEVEL=ORDINAL
- DISTANCE, the distance computed from the PROC MDS model
- TRANSDIST, the optimally transformed distance when the LEVEL= option is not ORDINAL or ABSOLUTE
- FITDATA, the datum further transformed according to the FIT= option
- FITDIST, the distance further transformed according to the FIT= option
- WEIGHT, the combined weight of the datum based on the WEIGHT variable(s), if any, and the standardization specified by the FORMULA= option
- RESIDUAL, FITDATA minus FITDIST

To cause a datum to appear in the OUTRES= data set, yet be ignored in fitting the model, give the datum a nonmissing value but a 0 weight (see "WEIGHT Statement").

INITIAL= Data Set

The INITIAL= data set has the same structure as the OUT= data set but is not required to have all of the variables or observations that appear in the OUT= data set. You can use an OUT= data set previously created by PROC MDS (without the OUTITER option) as an INITIAL= data set in a subsequent invocation of the procedure.

The only variables that are required are DIM1, . . . , DIM*m* (where *m* is the maximum number of dimensions) or equivalent variables specified in the INVAR statement. If these are the only variables, then all the observations are assumed to contain coordinates of the configuration; you cannot read dimension coefficients or transformation parameters.

To read initial values for the dimension coefficients or transformation parameters, the INITIAL= data set must contain the _TYPE_ variable and either the variable specified in the ID statement or, if no ID statement is used, the variable _NAME_. In addition, if there is more than one data matrix, either the variable specified in the MATRIX statement or, if no MATRIX statement is used, the variable _MATRIX_ or _MATNUM_ is required.

If the INITIAL= data set contains the variable _DIMENS_, initial values are obtained from observations with the corresponding number of dimensions. If there is no _DIMENS_ variable, the same observations are used for each number of dimensions analyzed.

If you want PROC MDS to read initial values from some but not all of the observations in the INITIAL= data set, use the WHERE= data set option to select the desired observations.

Missing Values

Missing data in the similarity or dissimilarity matrices are ignored in fitting the model and are omitted from the OUTRES= data set. Any matrix that is completely missing is omitted from the analysis.

Missing weights are treated as 0.

Missing values are also allowed in the INITIAL= data set, but a large number of missing values may yield a degenerate initial configuration.

Normalization of the Estimates

In multidimensional scaling models, the parameter estimates are not uniquely determined; the estimates can be transformed in various ways without changing their badness of fit. The initial and final estimates from PROC MDS are, therefore, normalized (unless you specify the NONORM option) to make it easier to compare results from different analyses.

The configuration always has a mean of 0 for each dimension.

With the COEF=IDENTITY option, the configuration is rotated to a principal-axis orientation. Unless you specify the LEVEL=ABSOLUTE option, the entire configuration is scaled so that the root-mean-square element is 1, and the transformations are adjusted to compensate.

With the COEF=DIAGONAL option, each dimension is scaled to a root-mean-square value of 1, and the dimension coefficients are adjusted to compensate. Unless you specify the LEVEL=ABSOLUTE option, the dimension coefficients are normalized as follows. If you specify the CONDITION=UN option, all of the dimension coefficients are scaled to a root-mean-square value of 1. For other values of the CONDITION= option, the dimension coefficients are scaled separately for each subject to a root-mean-square value of 1. In either case, the transformations are adjusted to compensate.

Each dimension is reflected to give a positive rank correlation with the order of the objects in the data set.

For the LEVEL=ORDINAL option, if the intercept, slope, or power parameters are fitted, the transformed data are normalized to eliminate these parameters if possible.

Comparison with the ALSCAL Procedure

The MDS procedure generally produces results similar to those from the ALSCAL procedure (Young, Lewyckyj, and Takane 1986; Young 1982) if you use the following options in PROC MDS:

- FIT=SQUARED
- FORMULA=1 except for unfolding data, which require FORMULA=2
- PFINAL to get output similar to that from PROC ALSCAL

Unlike PROC ALSCAL, PROC MDS produces no plots, so you must use output data sets and PROC PLOT or PROC GPLOT.

The MDS and ALSCAL procedures may sometimes produce different results for the following reasons:

- With the LEVEL=INTERVAL option, PROC MDS fits a regression model while PROC ALSCAL fits a measurement model. These models are not equivalent if there is more than one partition, although the differences in the parameter estimates are usually minor.
- PROC MDS and PROC ALSCAL use different algorithms for initialization and optimization. Hence, different local optima may be found by PROC MDS and PROC ALSCAL for some data sets with poor fit. Using the INAV=SSCP option causes the initial estimates from PROC MDS to be more like those from PROC ALSCAL.
- The default convergence criteria in PROC MDS are more strict than those in PROC ALSCAL. The convergence measure in PROC ALSCAL may cause PROC ALSCAL to stop iterating because progress is slow rather than because a local optimum has been reached. Even if you run PROC ALSCAL with a very small convergence criterion and a very large iteration limit, PROC ALSCAL may never achieve the same degree of precision as PROC MDS. For most applications, this problem is of no practical consequence since two- or three-digit precision is sufficient. If the model does not fit well, obtaining higher precision may require hundreds of iterations.

PROC MDS accepts some PROC ALSCAL options as synonyms for the preceding options, as displayed in Table 43.1.

Table 43.1. PROC MDS Options Compared to PROC ALSCAL Options

PROC ALSCAL Option	Accepted by PROC MDS?	Related PROC MDS Option or Comments
CONDITION=	Yes	
CONVERGE=	Yes	Convergence measures are not comparable
CUTOFF=	Yes	
DATA=	Yes	

Table 43.1. (continued)

PROC ALSCAL Option	Accepted by PROC MDS?	Related PROC MDS Option or Comments
DEGREE=	No	
DIMENS=	Yes	
DIRECTIONS=	No	
HEADER	Yes	Default in PROC MDS
IN=	Yes	
ITER=	Yes	MAXITER=
LEVEL=	Yes	LEVEL=NOMINAL is not supported
MAXDIM=*m*	Yes	DIMENSION=*n* TO *m*
MINDIM=*n*	Yes	DIMENSION=*n* TO *m*
MINSTRESS=	Yes	MINCRIT=

MODEL=EUCLID
MODEL=INDSCAL
MODEL=GEMSCAL
MODEL=ASYMSCAL
MODEL=ASYMINDS
	Yes
	Yes
	No
	No
	No
		COEF=IDENTITY
		COEF=DIAGONAL

NEGATIVE	(Yes)	In PROC MDS, the NEGATIVE option affects slopes and powers, not subject weights.
NOULB	Yes	
OUT=	Yes	Some differences in contents
PLOT	No	
PLOTALL	No	
PRINT	No	
READV, etc.	No	Use WHERE data set option
READFIXV, etc.	No	
ROWS=	No	

SHAPE=SYMMETRI
SHAPE=ASYMMETR
SHAPE=RECTANGU
	Yes
	Yes
	No
		SHAPE=TRIANGLE
		SHAPE=SQUARE
		Use SHAPE=TRIANGLE with extra missing values to fill out the matrix.

SIMILAR	Yes	
TIESTORE=	Yes	Ignored by PROC MDS
UNTIE	Yes	

Comparison with the MLSCALE Procedure

Running the MDS procedure with the options

```
   proc mds fit=log level=loginterval ... ;
```

generally produces results similar to using the MLSCALE procedure (Ramsay 1986) with the options

```
   proc mlscale stvarnce=constant suvarnce=constant ... ;
```

Alternatively, using the FIT=DISTANCE option in the PROC MDS statement produces results similar to specifying the NORMAL option in the PROC MLSCALE statement.

The MDS procedure uses the least-squares method of estimation. The least-squares method is equivalent to the maximum-likelihood method if the error terms are assumed to be independent and identically distributed normal random variables. Unlike PROC MLSCALE, PROC MDS does not provide any options for unequal error variances.

PROC MDS accepts some PROC MLSCALE options as synonyms for the options described previously, as displayed in Table 43.2.

Table 43.2. PROC MDS Options Compared to PROC MLSCALE Options

PROC MLSCALE Option	Accepted by PROC MDS?	Related PROC MDS Option or Comments
SQUARE	Yes	SHAPE=SQUARE
INPUT=MATRIX	No	Default
INPUT=VECTOR	No	
STLABEL=	No	ID statement
STLBDS	No	
SULABEL=	No	MATRIX statement
SULBDS	No	
CONFIG	No	
CONFDS=	No	IN= data set
NEQU=	No	
CONSDS=	No	
METVAL	No	
METVDS	No	IN=
SEWGTS	No	
SEWGDS=	No	
SPLVAL	No	
SLPVDS=	No	
DIMENS=	Yes	
METRIC=IDENTITY	Yes	COEF=IDENTITY
METRIC=DIAGONAL	Yes	COEF=DIAGONAL
METRIC=FULL	No	
TRANSFRM=SCALE	Yes	LEVEL=RATIO
TRANSFRM=POWER	Yes	LEVEL=LOGINTERVAL
TRANSFRM=SPLINE	No	
STVARNCE=	No	
SUVARNCE=	No	
NORMAL	No	Default (FIT=DISTANCE)
ITMAX=	Yes	MAXITER=
ITXMAX=	No	
ITWMAX=	No	

Table 43.2. (continued)

PROC MLSCALE Option	Accepted by PROC MDS?	Related PROC MDS Option or Comments
ITAMAX=	No	
ITPMAX=	No	
CONV=	(Yes)	Meaning is different
FACTOR=	No	
HISTORY	No	PITER
ASYMP	No	
OUTCON	No	OUT=
OUTDIS	No	
OUTMET	No	OUT=
OUTSPL	No	
OUTRES	(Yes)	OUTRES= data set

Displayed Output

Unless you specify the NOPHIST option, PROC MDS displays the iteration history containing

- Iteration number
- Type of iteration:

Initial	initial configuration
Monotone	monotone transformation
Gau-New	Gauss-Newton step
Lev-Mar	Levenberg-Marquardt step

- Badness-of-Fit Criterion
- Change in Criterion
- Convergence Measures:

Monotone   the Euclidean norm of the change in the optimally scaled data divided by the Euclidean norm of the optimally scaled data, averaged across partitions

Gradient   the multiple correlation of the Jacobian matrix with the residual vector, uncorrected for the mean

Depending on what options are specified, PROC MDS may also display the following tables:

- Data Matrix and possibly Weight Matrix for each subject
- Eigenvalues from the computation of the initial coordinates
- Sum of Data Weights and Pooled Data Matrix computed during initialization with INAV=DATA

- Configuration, the estimated coordinates of the objects
- Dimension Coefficients
- A table of transformation parameters, including one or more of the following:

  Intercept
  Slope
  Power

- A table of fit statistics for each matrix and possibly each row, including

  Number of Nonmissing Data

  Weight of the matrix or row, allowing for both observation weights and standardization factors

  Badness-of-Fit Criterion

  Distance Correlation computed between the distances and data with optimal transformation

  Uncorrected Distance Correlation not corrected for the mean

  Fit Correlation computed after applying the FIT= transformation to both distances and data

  Uncorrected Fit Correlation not corrected for the mean

ODS Table Names

PROC MDS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 43.3. ODS Tables Produced in PROC MDS

ODS Table Name	Description	Option
ConvergenceStatus	Convergence status	default
DimensionCoef	Dimension coefficients	PCOEF w/COEF= not IDENTITY
FitMeasures	Measures of fit	PFIT
IterHistory	Iteration history	default
PConfig	Estimated coordinates of the objects in the configuration	PCONFIG
PData	Data matrices	PDATA
PInAvData	Initial sum of weights and weighted average of data matrices with INAV=DATA	PINAVDATA
PInEigval	Initial eigenvalues	PINEIGVAL
PInEigvec	Initial eigenvectors	PINEIGVEC
PInWeight	Initialization weights	PINWEIGHT
Transformations	Transformation parameters	PTRANS w/LEVEL=RATIO, INTERVAL, LOGINTERVAL

Example

Example 43.1. Jacobowitz Body Parts Data from Children and Adults

Jacobowitz (1975) collected conditional rank-order data regarding perceived similarity of parts of the body from children of ages 6, 8, and 10 years and from college sophomores. The following analysis includes data from 15 children (6-year-olds) and 15 sophomores. The method of data collection and some results of an analysis are also described by Young (1987, pp. 4−10). A portion of the data is included below. See “Mds Documentation Examples” in the SAS/STAT Sample Program Library for the complete data set.

```
data body;
   title  ’Jacobowitz Body Parts Data from 6 Yr Olds and
           Adults’;
   title2 ’First 15 subjects (obs 1-225) are children’;
   title3 ’Second 15 subjects (obs 226-450) are adults’;
   input  (cheek face mouth head ear body arm elbow hand
       palm finger leg knee foot toe) (2.);
   if _n_ <= 225 then subject=’C’; else subject=’A’;
   datalines;
 0 2 1 3 410 5 9 6 7 811121314
 2 012 113 3 81011 9 7 4 5 614
 3 2 0 1 4 9 511 6 7 810131214
 2 1 3 0 4 9 5 611 7 810121314
10 111 2 0 6 3 4 51213 7 814 9
1412 9 613 0 8 7 51011 1 4 2 3
1214111013 5 0 4 1 3 2 6 9 7 8
 5 714 8 6 9 1 0 2 3 410111213
1311121014 9 3 4 0 1 2 6 5 7 8
 8 6 7 9 4 5 310 1 0 212111314
14 513 6 912 3 4 1 2 0 7 81011
14121311 9 7 4 6 5 310 0 8 1 2
1211141013 4 5 8 6 7 9 1 0 2 3
1214101311 9 4 5 8 6 7 2 3 0 1
13 8 91114 3 6 5 71012 2 4 1 0
 0 4 2 311 91412 1 713 8 6 510
 7 011 9 1 2 8 3131410 612 4 5
1011 0 3 7 1 813 212 6 914 5 4
 4 811 0 1 2 3 9 71310 51412 6
14 61110 0 1 2 4 9 8 5 713 312
11 61412 3 013 2 1 9 5 410 7 8
1412 8 3 1 7 013 2 6 5 911 410
 610 914 3 412 0 813 5 711 1 2
131412 1 4 6 2 7 010 9 311 8 5
 6 9 711 3 5 112 2 013 810 414
10 6 9 81113 712 2 1 014 5 3 4
 6 9 710 4 21214 1 813 0 3 511
13 2 8 3 1 5 9141211 7 6 010 4
1411 8 413 212 9 5 310 6 7 0 1
14 3 4 713 6 2 8 91110 112 5 0
```

```
   ... 405 lines omitted ...

 0 1 2 4 3141210 6 5 81311 7 9
 2 0 3 1 414 810 5 612 911 713
 2 1 0 4 31412 9 511 71310 6 8
 2 1 4 0 314 810 61112 7 9 513
 1 3 2 4 01412 9 8 7 5131011 6
131011 114 0 3 7 512 8 2 6 4 9
13 911101214 0 1 3 6 5 2 4 7 8
1012 813 914 1 0 611 4 3 2 5 7
12 911101314 5 4 0 1 3 8 6 2 7
 91211131014 5 4 1 0 2 6 7 3 8
10121113 914 6 5 1 3 0 8 7 4 2
121011 91314 2 5 6 8 7 0 1 3 4
1112 9131014 3 2 7 8 4 1 0 5 6
121011 91314 5 8 1 6 7 3 4 0 2
10121113 914 8 7 4 6 2 3 5 1 0
;
```

The data are analyzed as row conditional (CONDITION=ROW) at the ordinal level of measurement (LEVEL=ORDINAL) using the weighted Euclidean model (COEF=DIAGONAL) in three dimensions (DIMENSION=3). The final estimates are displayed (PFINAL). The estimates (OUT=OUT) and fitted values (OUTRES=RES) are saved in output data sets. The following statements produce Output 43.1.1:

```
options ps=60;
proc mds data=body
   condition=row
   level=ordinal
   coef=diagonal
   dimension=3
   pfinal
   out=out
   outres=res
   ;
subject subject;
title5 'Nonmetric Weighted MDS';
run;
```

Output 43.1.1. Iteration History and Final Estimates for Body Parts Data

```
             Jacobowitz Body Parts Data from 6 Yr Olds and Adults
                  First 15 subjects (obs 1-225) are children
                 Second 15 subjects (obs 226-450) are adults

                            Nonmetric Weighted MDS

                Multidimensional Scaling:  Data=WORK.BODY.DATA
                   Shape=SQUARE Condition=ROW Level=ORDINAL
                  Coef=DIAGONAL Dimension=3 Formula=1 Fit=1

Mconverge=0.01 Gconverge=0.01 Maxiter=100 Over=2 Ridge=0.0001 Alternate=MATRIX


                            Badness-                  Convergence Measures
                              of-Fit    Change in    ----------------------
  Iteration    Type        Criterion    Criterion     Monotone     Gradient
  -------------------------------------------------------------------------
        0      Initial        0.4091            .            .            .
        1      Monotone       0.2053       0.2038       0.3012       0.3190
        2      Gau-New        0.1937       0.0116            .            .
        3      Monotone       0.1862     0.007533       0.0410       0.2314
        4      Gau-New        0.1847     0.001496            .            .
        5      Monotone       0.1779     0.006754       0.0372       0.1516
        6      Gau-New        0.1773     0.000654            .            .
        7      Monotone       0.1758     0.001488       0.0187       0.0975
        8      Gau-New        0.1755     0.000253            .            .
        9      Monotone       0.1751     0.000478       0.0104       0.0767
       10      Gau-New        0.1750     0.000113            .            .
       11      Monotone       0.1748     0.000199     0.006467       0.0582
       12      Gau-New        0.1747    0.0000592            .       0.0384
       13      Gau-New        0.1747    0.0000184            .     0.009952


       Convergence criteria are satisfied.


                                Configuration

                                Dim1        Dim2        Dim3
                  ------------------------------------------
                  cheek         1.46        0.77       -0.30
                  face          1.40       -0.56        0.43
                  mouth         1.25       -0.97       -0.51
                  head          1.31        0.39        0.97
                  ear           1.23        0.03       -1.39
                  body          0.24       -0.18        2.24
                  arm          -0.70        1.23        1.08
                  elbow        -0.73        0.40        0.29
                  hand         -0.71        1.50       -0.26
                  palm         -0.24        1.59       -0.96
                  finger       -0.74        0.57       -1.35
                  leg          -0.69       -1.44        0.88
                  knee         -1.44       -0.64        0.42
                  foot         -0.78       -1.51       -0.36
                  toe          -0.87       -1.20       -1.19
```

```
               Jacobowitz Body Parts Data from 6 Yr Olds and Adults
                    First 15 subjects (obs 1-225) are children
                   Second 15 subjects (obs 226-450) are adults

                             Nonmetric Weighted MDS

                  Multidimensional Scaling:  Data=WORK.BODY.DATA
                     Shape=SQUARE Condition=ROW Level=ORDINAL
                    Coef=DIAGONAL Dimension=3 Formula=1 Fit=1

                             Dimension Coefficients

                   subject             1            2            3
                   -------------------------------------------------
                   C                1.18         1.02         0.76
                   C                0.94         1.05         1.01
                   C                0.94         1.01         1.05
                   C                1.08         1.10         0.79
                   C                0.98         1.05         0.97
                   C                1.28         0.85         0.79
                   C                0.90         1.03         1.06
                   C                0.95         1.04         1.01
                   C                1.18         1.06         0.69
                   C                0.95         0.99         1.05
                   C                1.07         1.13         0.76
                   C                0.98         0.94         1.08
                   C                1.34         0.82         0.73
                   C                0.95         0.99         1.05
                   C                1.03         1.09         0.86
                   A                1.30         0.82         0.80
                   A                1.05         0.64         1.22
                   A                1.28         0.85         0.81
                   A                0.95         0.88         1.15
                   A                1.34         0.75         0.80
                   A                1.27         0.81         0.85
                   A                1.50         0.37         0.78
                   A                1.38         0.82         0.64
                   A                1.44         0.56         0.78
                   A                1.53         0.73         0.36
                   A                1.18         0.95         0.84
                   A                0.85         0.86         1.24
                   A                1.29         0.62         0.97
                   A                1.20         0.58         1.10
                   A                1.25         0.61         1.04
               Jacobowitz Body Parts Data from 6 Yr Olds and Adults
                    First 15 subjects (obs 1-225) are children
                   Second 15 subjects (obs 226-450) are adults

                              Nonmetric Weighted MDS

                  Multidimensional Scaling:  Data=WORK.BODY.DATA
                     Shape=SQUARE Condition=ROW Level=ORDINAL
                    Coef=DIAGONAL Dimension=3 Formula=1 Fit=1

             Number of             Badness-of-                  Uncorrected
            Nonmissing                     Fit     Distance        Distance
subject           Data   Weight      Criterion  Correlation     Correlation
-----------------------------------------------------------------------------
C                  210     0.03           0.16         0.85            0.99
C                  210     0.03           0.25         0.51            0.97
C                  210     0.03           0.23         0.58            0.97
C                  210     0.03           0.16         0.85            0.99
C                  210     0.03           0.21         0.69            0.98
C                  210     0.03           0.15         0.89            0.99
C                  210     0.03           0.26         0.40            0.96
C                  210     0.03           0.25         0.47            0.97
C                  210     0.03           0.13         0.91            0.99
C                  210     0.03           0.24         0.55            0.97
C                  210     0.03           0.15         0.87            0.99
C                  210     0.03           0.23         0.59            0.97
C                  210     0.03           0.15         0.90            0.99
C                  210     0.03           0.21         0.69            0.98
C                  210     0.03           0.19         0.75            0.98
A                  210     0.03           0.12         0.94            0.99
A                  210     0.03           0.17         0.84            0.98
A                  210     0.03           0.12         0.93            0.99
A                  210     0.03           0.21         0.68            0.98
A                  210     0.03           0.14         0.91            0.99
A                  210     0.03           0.16         0.87            0.99
A                  210     0.03           0.09         0.98            1.00
A                  210     0.03           0.11         0.95            0.99
A                  210     0.03           0.11         0.96            0.99
A                  210     0.03           0.13         0.95            0.99
A                  210     0.03           0.13         0.90            0.99
A                  210     0.03           0.21         0.72            0.98
A                  210     0.03           0.12         0.93            0.99
A                  210     0.03           0.10         0.95            0.99
A                  210     0.03           0.11         0.94            0.99

- All -           6300     1.00           0.17         0.84            0.98
```

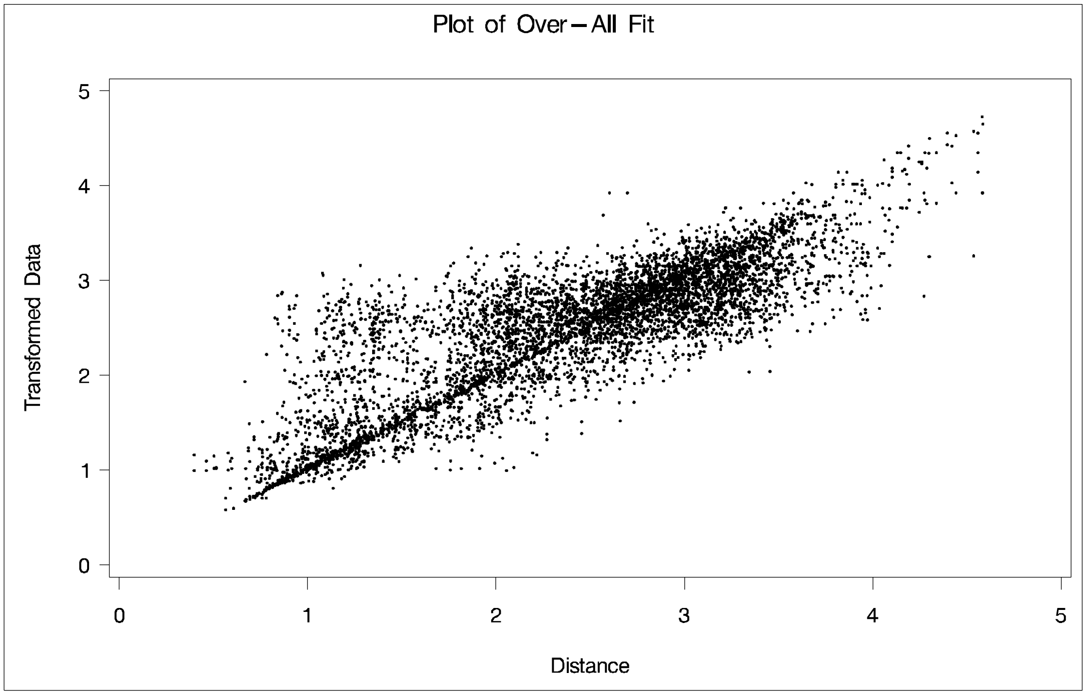
The OUTRES= data set is used to produce a plot showing the overall fit of the model, with the transformed data on the vertical axis and the distances from the model on the horizontal axis. If the model fits perfectly, all points lie on a diagonal line from lower left to upper right. The vertical departure of each point from this diagonal line represents the residual of the corresponding observation. The HAXIS and VAXIS options in the PLOT statement specify that the horizontal and vertical axes use the definitions AXIS2 and AXIS1, respectively. The identical ORDER= options in each of these AXIS statements make physical distances on each axis comparable.

The following statements produce Output 43.1.2:

```
title1 ’Plot of Over-All Fit’;
axis1 label=(angle=90 rotate=0) minor=none
```

```
      order=(0 to 5 by 1);
axis2 minor=none  order=(0 to 5 by 1);

proc gplot data=res;
   plot fitdata*fitdist/vaxis=axis1 haxis=axis2
                          frame cframe=ligr;
run;
```

Output 43.1.2. Plot of Over-All Fit for Body Parts Data



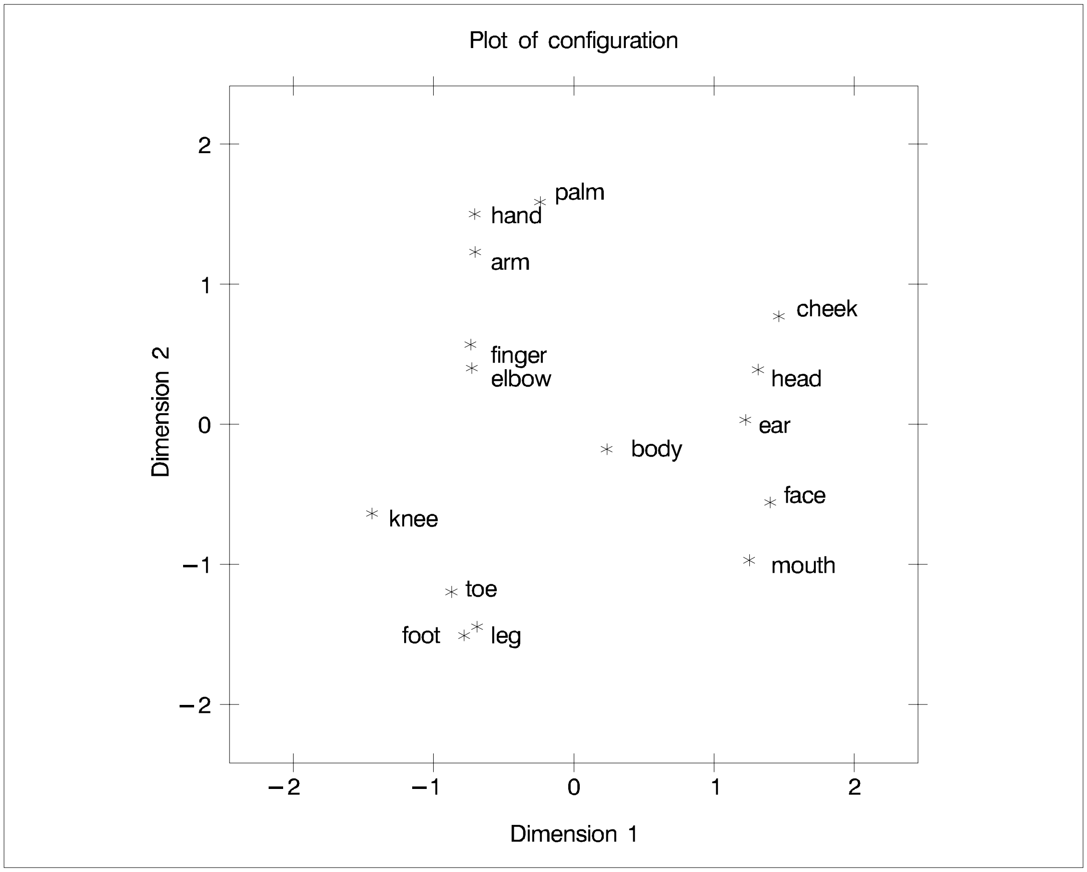
The OUT= data set is used to plot the configuration and dimension coefficients using the %PLOTIT macro. Again it is necessary to use the VTOH= option to make the axes commensurable. The configuration is plotted by selecting observations having _TYPE_='CONFIG' with a WHERE statement and by using the _NAME_ variable to identify each body part on the plot.

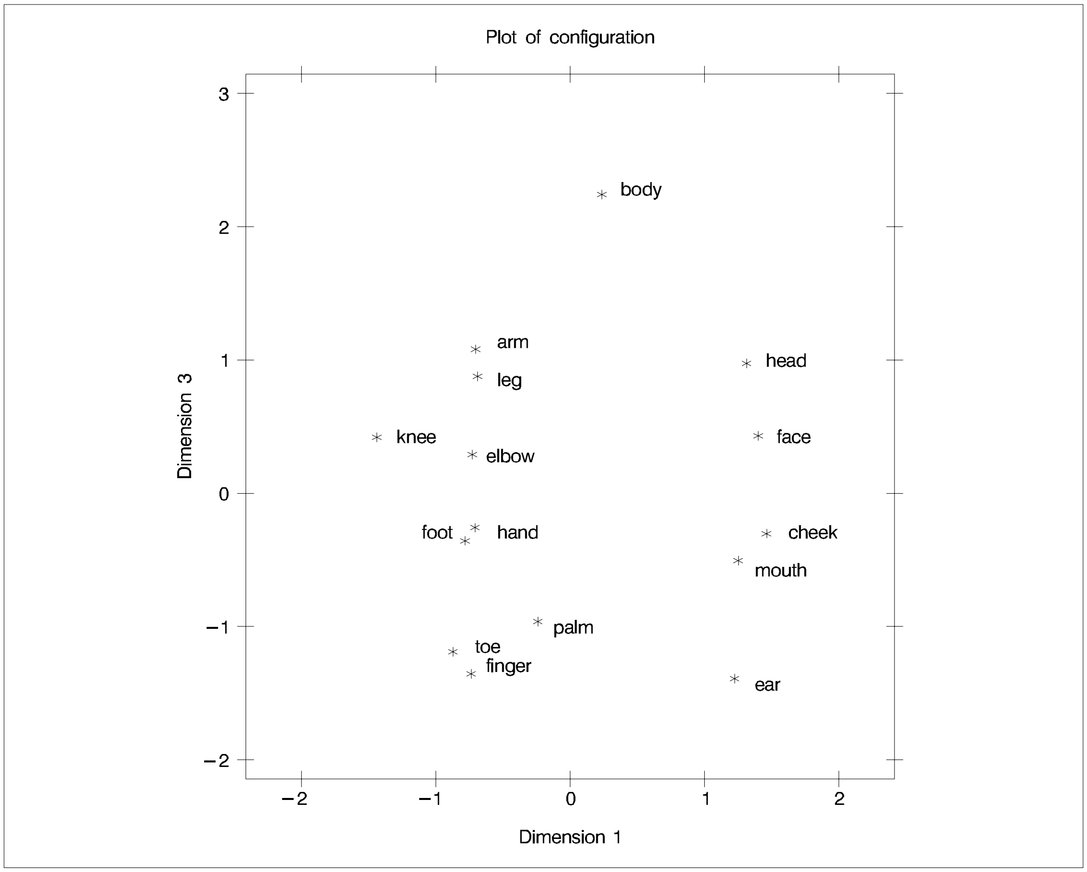
The dimension coefficients are plotted by selecting observations having _TYPE_='DIAGCOEF' and by using the Subject variable to distinguish children from adults on the plot. The following statements produce Output 43.1.3:

```
title1 'Plot of configuration';
%plotit(data=out(where=(_type_='CONFIG')), datatype=mds,
        labelvar=_name_, vtoh=1.75);
%plotit(data=out(where=(_type_='CONFIG')), datatype=mds,
        plotvars=dim3 dim1, labelvar=_name_, vtoh=1.75);
run;

title1 'Plot of Dimension Coefficients for Each Subject';
%plotit(data=out(where=(_type_='DIAGCOEF')), symtype=vector,
        symbols='',datatype=mds, place=0, labelvar=subject,
        vechead=, ls=100, plotopts=hzero vzero);

%plotit(data=out(where=(_type_='DIAGCOEF')),
        plotvars=dim3 dim1, symtype=vector, symbols='',
        datatype=mds, place=0, labelvar=subject, ls=100,
        vechead=, plotopts=hzero vzero);
run;
```

The configuration displayed in Output 43.1.3 has a tripodal shape with Body at the apex. The three legs of the tripod can be distinguished in the plot of dimension 2 by dimension 1, which shows three distinct clusters with Body in the center. Dimension 1 separates head parts from arm and leg parts. Dimension 2 separates arm parts from leg parts. The plot of dimension 3 by dimension 1 shows the tripod from the side. Dimension 3 distinguishes the more inclusive body parts (at the top) from the less inclusive body parts (at the bottom).

Output 43.1.3. Plot of Configuration for Body Parts Data

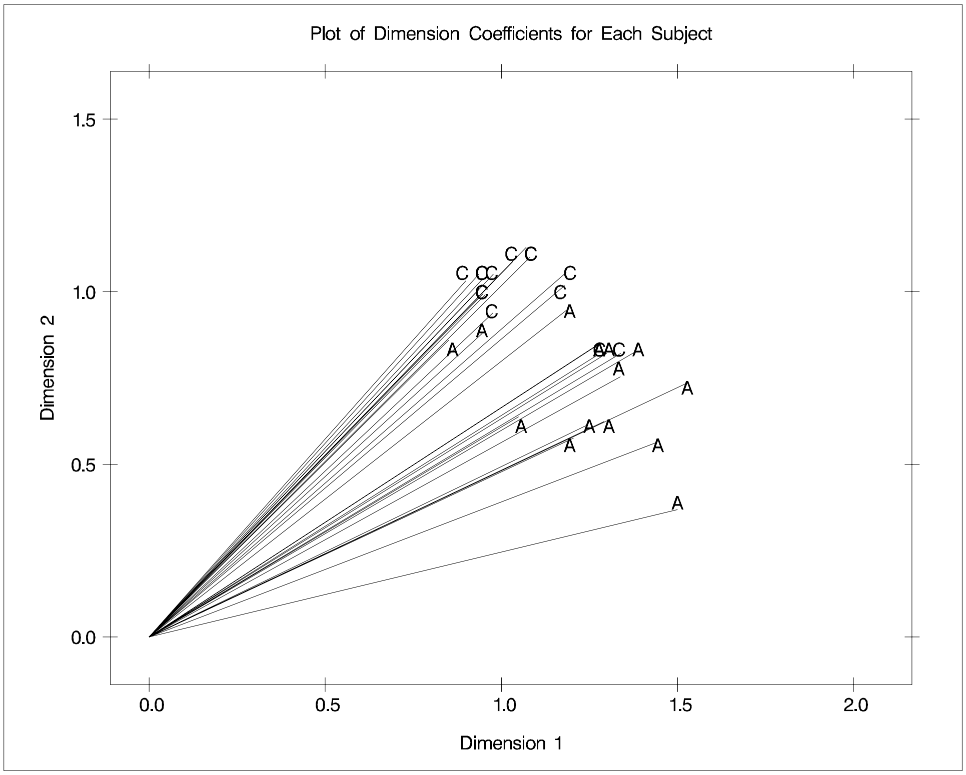


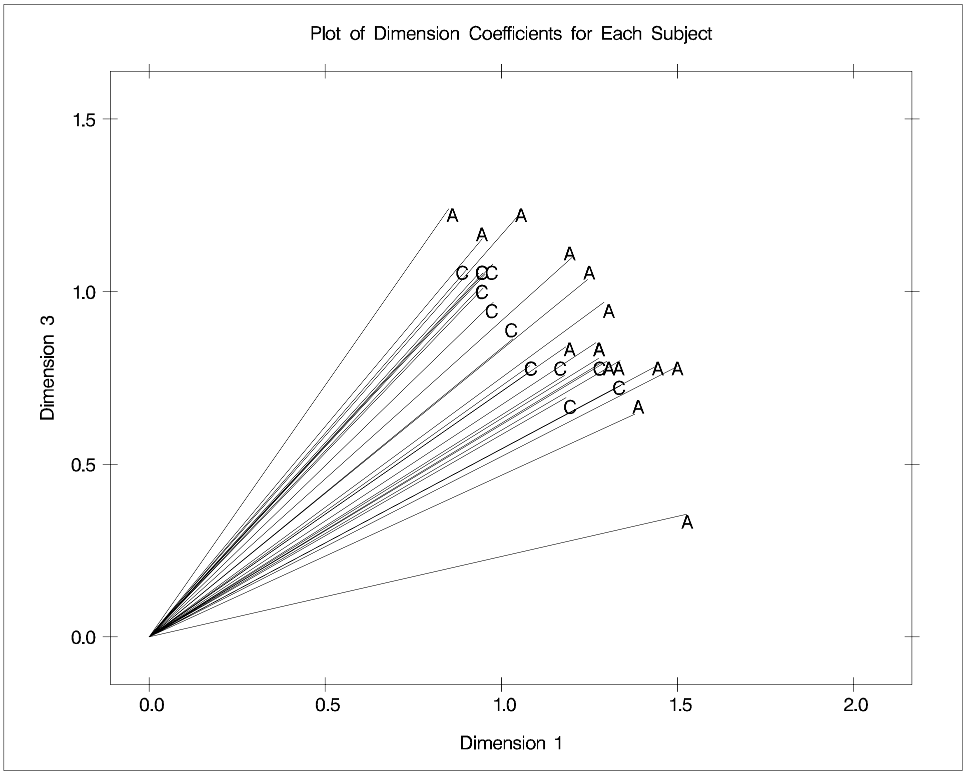


The plots of dimension coefficients in Output 43.1.4 show that children differ from adults primarily in the emphasis given to dimension 2. Children give about the same weight (approximately 1) to each dimension. Adults are much more variable than children, but all have coefficients less than 1.0 for dimension 2, with an average of

about 0.7. Referring back to the configuration plot, you can see that adults consider arm parts to be more similar to leg parts than do children. Many adults also give a high weight to dimension 1, indicating that they consider head parts to be more dissimilar from arm and leg parts than do children. Dimension 3 shows considerable variability for both children and adults.

Output 43.1.4. Plot of Dimension Coefficients for Body Parts Data





References

Arabie, P., Carroll, J.D., and DeSarbo, W.S. (1987), *Three-Way Scaling and Clustering*, Sage University Paper series on Quantitative Applications in the Social Sciences, 07-065, Beverly Hills and London: Sage Publications.

Carroll, J.D. and Chang, J.J. (1970), "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of the 'Eckart-Young' Decomposition," *Psychometrika*, 35, 283–319.

Davison, M.L. (1983), *Multidimensional Scaling*, New York: John Wiley & Sons, Inc.

Heiser, W.J. (1981), *Unfolding Analysis of Proximity Data*, Leiden: Department of Datatheory, University of Leiden.

Jacobowitz, D. (1975), *The Acquisition of Semantic Structures*, Doctoral dissertation, University of North Carolina at Chapel Hill.

Krantz, D.H., Luce, R.D., Suppes, P., and Tversky, A. (1971), *Foundations of Measurement*, New York: Academic Press.

Kruskal, J.B. and Wish, M. (1978), *Multidimensional Scaling*, Sage University Paper series on Quantitative Applications in the Social Sciences, 07–011, Beverly Hills and London: Sage Publications.

Kuhfeld, W.F. (1994), "Graphical Scatter Plots of Labeled Points," *Observations*, Fourth Quarter, 1994, Cary, NC: SAS Institute Inc.

Null, C.H. and Sarle, W.S. (1982), "Multidimensional Scaling by Least Squares," in *Proceedings of the Seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Rabinowitz, G. (1976), "A Procedure for Ordering Object Pairs Consistent with the Multidimensional Unfolding Model," *Psychometrika*, 41, 349–373.

Ramsay, J.O. (1986), "The MLSCALE Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Schiffman, S.S., Reynolds, M.L., and Young, F.W. (1981), *Introduction to Multidimensional Scaling*, New York: Academic Press.

Torgerson, W.S. (1958), *Theory and Methods of Scaling*, New York: John Wiley & Sons, Inc.

Young, F.W. (1982), "Enhancements in ALSCAL-82," in *Proceedings of the Seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Young, F.W. (1987), *Multidimensional Scaling: History, Theory, and Applications*, ed. R.M. Hamer, Hillsdale, NJ: Lawrence Erlbaum Associates.

Young, F.W., Lewyckyj, R., and Takane, Y. (1986), "The ALSCAL Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Chapter 44

The MI Procedure

Chapter Contents

Chapter 44
The MI Procedure

Overview

The MI procedure performs multiple imputation of missing data. Missing values are an issue in a substantial number of statistical analyses. Most SAS statistical procedures exclude observations with any missing variable values from the analysis. These observations are called incomplete cases. While analyzing only complete cases has its simplicity, the information contained in the incomplete cases is lost. This approach also ignores possible systematic differences between the complete cases and the incomplete cases, and the resulting inference may not be applicable to the population of all cases, especially with a small number of complete cases.

Some SAS procedures use all the available cases in an analysis, that is, cases with useful information. For example, the CORR procedure estimates a variable mean by using all cases with nonmissing values for this variable, ignoring the possible missing values in other variables. PROC CORR also estimates a correlation by using all cases with nonmissing values for this pair of variables. This makes better use of the available data than use only the complete cases, but the resulting correlation matrix may not be positive definite.

Another strategy for handling missing data is single imputation, which substitutes a value for each missing value. Standard statistical procedures for complete data analysis can then be used with the filled-in data set. For example, each missing value can be imputed with the variable mean of the complete cases, or it can be imputed with the mean conditional on observed values of other variables. This approach treats missing values as if they were known in the complete-data analysis. However, single imputation does not reflect the uncertainty about the predictions of the unknown missing values, and the resulting estimated variances of the parameter estimates will be biased toward zero (Rubin 1987, p. 13).

Instead of filling in a single value for each missing value, multiple imputation (Rubin 1976; 1987) replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute. The multiply imputed data sets are then analyzed by using standard procedures for complete data and combining the results from these analyses. No matter which complete-data analysis is used, the process of combining results from different data sets is essentially the same.

Multiple imputation does not attempt to estimate each missing value through simulated values. Instead, it draws a random sample of the missing values from its distribution. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values; for example, confidence intervals with the correct probability coverage.

Multiple imputation inference involves three distinct phases:

1. The missing data are filled in *m* times to generate *m* complete data sets.
2. The *m* complete data sets are analyzed using standard statistical analyses.
3. The results from the *m* complete data sets are combined to produce inferential results.

The MI procedure creates multiply imputed data sets for incomplete multivariate data. It uses methods that incorporate appropriate variability across the *m* imputations. The method of choice depends on the patterns of missingness.

For data sets with monotone missing patterns, either a parametric method that assumes multivariate normality or a nonparametric method is appropriate. Parametric methods available include the regression method (Rubin 1987, pp. 166–167) and the predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996). The nonparametric method is the propensity score method (Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995).

For data sets with arbitrary missing patterns, a Markov Chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality is used to impute all missing values or just enough missing values to make the imputed data sets have monotone missing patterns.

Once the *m* complete data sets are analyzed using standard SAS procedures, the MIANALYZE procedure can be used to generate valid statistical inferences about these parameters by combining results from the *m* analyses.

Often, as few as three to five imputations are adequate in multiple imputation (Rubin 1996, p. 480). The relative efficiency of the small m imputation estimator is high for cases with little missing information (Rubin 1987, p. 114). Also see the "Multiple Imputation Efficiency" section on page 2562.

Multiple imputation inference assumes that the model (variables) you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models may not be the same. The consequences for different scenarios (Schafer 1997, pp. 139–143) are discussed in the "Imputer's Model Versus Analyst's Model" section on page 2563.

In SAS 9, an experimental CLASS statement has been added to specify classification variables, which can be used either as covariates for imputed variables or as imputed variables for data sets with monotone missing patterns. The CLASS statement must be used in conjunction with the MONOTONE statement.

Experimental graphics using ODS are now available with the MI procedure. For more information, see the "ODS Graphics" section on page 2567.

Getting Started

Consider the following Fitness data set that has been altered to contain an arbitrary pattern of missingness:

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
data FitMiss;
   input Oxygen RunTime RunPulse @@;
   datalines;
44.609  11.37  178     45.313  10.07  185
54.297   8.65  156     59.571    .      .
49.874   9.22    .     44.811  11.63  176
  .     11.95  176          .  10.85    .
39.442  13.08  174     60.055   8.63  170
50.541    .      .     37.388  14.03  186
44.754  11.12  176     47.273    .      .
51.855  10.33  166     49.156   8.95  180
40.836  10.95  168     46.672  10.00    .
46.774  10.25    .     50.388  10.08  168
39.407  12.63  174     46.080  11.17  156
45.441   9.63  164       .      8.92    .
45.118  11.08    .     39.203  12.88  168
45.790  10.47  186     50.545   9.93  148
48.673   9.40  186     47.920  11.50  170
47.467  10.50  170
;
```

Suppose that the data are multivariate normally distributed and the missing data are missing at random (MAR). That is, the probability that an observation is missing can depend on the observed variable values of the individual, but not on the missing variable values of the individual. See the "Statistical Assumptions for Multiple Imputation" section on page 2537 for a detailed description of the MAR assumption.

The following statements invoke the MI procedure and impute missing values for the FitMiss data set.

```
proc mi data=FitMiss seed=501213 mu0=50 10 180 out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

```
                              The MI Procedure

                             Model Information

           Data Set                             WORK.FITMISS
           Method                               MCMC
           Multiple Imputation Chain            Single Chain
           Initial Estimates for MCMC           EM Posterior Mode
           Start                                Starting Value
           Prior                                Jeffreys
           Number of Imputations                5
           Number of Burn-in Iterations         200
           Number of Iterations                 100
           Seed for random number generator     501213
```

Figure 44.1. Model Information

The “Model Information” table displayed in Figure 44.1 describes the method used in the multiple imputation process. By default, the procedure uses the Markov Chain Monte Carlo (MCMC) method with a single chain to create five imputations. The posterior mode, the highest observed-data posterior density, with a noninformative prior, is computed from the EM algorithm and is used as the starting value for the chain.

The MI procedure takes 200 burn-in iterations before the first imputation and 100 iterations between imputations. In a Markov chain, the information in the current iteration has influence on the state of the next iteration. The burn-in iterations are iterations in the beginning of each chain that are used both to eliminate the series of dependence on the starting value of the chain and to achieve the stationary distribution. The between-imputation iterations in a single chain are used to eliminate the series of dependence between the two imputations.

```
                              The MI Procedure

                            Missing Data Patterns

                                Run      Run
          Group     Oxygen     Time     Pulse         Freq      Percent

              1     X          X        X               21        67.74
              2     X          X        .                4        12.90
              3     X          .        .                3         9.68
              4     .          X        X                1         3.23
              5     .          X        .                2         6.45

                           Missing Data Patterns

                    -----------------Group Means----------------
           Group           Oxygen         RunTime         RunPulse

               1        46.353810       10.809524       171.666667
               2        47.109500       10.137500                .
               3        52.461667               .                .
               4                .       11.950000       176.000000
               5                .        9.885000                .
```

Figure 44.2. Missing Data Patterns

The “Missing Data Patterns” table displayed in Figure 44.2 lists distinct missing data patterns with corresponding frequencies and percents. Here, an “X” means that the variable is observed in the corresponding group and a “.” means that the variable is missing. The table also displays group-specific variable means. The MI procedure sorts the data into groups based on whether an individual’s value is observed or missing for each variable to be analyzed. For a detailed description of missing data patterns, see the “Missing Data Patterns” section on page 2538.

```
                              The MI Procedure

                  Multiple Imputation Variance Information

                   -----------------Variance-----------------
       Variable         Between          Within           Total       DF

       Oxygen          0.056930        0.954041        1.022356   25.549
       RunTime         0.000811        0.064496        0.065469   27.721
       RunPulse        0.922032        3.269089        4.375528   15.753

                   Multiple Imputation Variance Information

                          Relative       Fraction
                          Increase        Missing        Relative
            Variable   in Variance    Information      Efficiency

            Oxygen        0.071606       0.068898        0.986408
            RunTime       0.015084       0.014968        0.997015
            RunPulse      0.338455       0.275664        0.947748
```

Figure 44.3. Variance Information

After the completion of *m* imputations, the “Multiple Imputation Variance Information” table shown in Figure 44.3 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency (in units of variance) for each variable are also displayed. A detailed description of these statistics is provided in the “Combining Inferences from Multiply Imputed Data Sets” section on page 2561.

The following “Multiple Imputation Parameter Estimates” table shown in Figure 44.4 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the *t* distribution. The table also displays a 95% confidence interval for the mean and a *t* statistic with the associated *p*-value for the hypothesis that the population mean is equal to the value specified with the MU0= option. A detailed description of these statistics is provided in the “Combining Inferences from Multiply Imputed Data Sets” section on page 2561.

```
                                The MI Procedure

                    Multiple Imputation Parameter Estimates

  Variable            Mean     Std Error    95% Confidence Limits        DF

  Oxygen         47.094040      1.011116      45.0139      49.1742   25.549
  RunTime        10.572073      0.255870      10.0477      11.0964   27.721
  RunPulse      171.787793      2.091776     167.3478     176.2278   15.753

                    Multiple Imputation Parameter Estimates

                                                          t for H0:
Variable         Minimum        Maximum            Mu0     Mean=Mu0   Pr > |t|

Oxygen         46.783898      47.395550      50.000000        -2.87     0.0081
RunTime        10.526392      10.599616      10.000000         2.24     0.0336
RunPulse      170.774818     173.122002     180.000000        -3.93     0.0012
```

Figure 44.4. Parameter Estimates

In addition to the output tables, the procedure also creates a data set with imputed values. The imputed data sets are stored in the outmi data set, with the index variable _Imputation_ indicating the imputation numbers. The data set can now be analyzed using standard statistical procedures with _Imputation_ as a BY variable.

The following statements list the first ten observations of data set outmi.

```
proc print data=outmi (obs=10);
  title ’First 10 Observations of the Imputed Data Set’;
run;
```

```
                 First 10 Observations of the Imputed Data Set

                                                                Run
             Obs    _Imputation_      Oxygen     RunTime       Pulse

               1          1          44.6090     11.3700     178.000
               2          1          45.3130     10.0700     185.000
               3          1          54.2970      8.6500     156.000
               4          1          59.5710      8.0747     155.925
               5          1          49.8740      9.2200     176.837
               6          1          44.8110     11.6300     176.000
               7          1          42.8857     11.9500     176.000
               8          1          46.9992     10.8500     173.099
               9          1          39.4420     13.0800     174.000
              10          1          60.0550      8.6300     170.000
```

Figure 44.5. Imputed Data Set

The table displayed in Figure 44.5 shows that the precision of the imputed values differs from the precision of the observed values. You can use the ROUND= option to make the imputed values consistent with the observed values.

Syntax

The following statements are available in PROC MI.

PROC MI < *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  EM < *options* > ;
  FREQ *variable* ;
  MCMC < *options* > ;
  MONOTONE < *options* > ;
  TRANSFORM *transform (variables* < */ options* >*)*
    < . . . *transform (variables* < */ options* >*)* > ;
  VAR *variables* ;

The BY statement specifies groups in which separate multiple imputation analyses are performed.

The CLASS statement lists the classification variables in the VAR statement. Classification variables can be either character or numeric.

The EM statement uses the EM algorithm to compute the maximum likelihood estimate (MLE) of the data with missing values, assuming a multivariate normal distribution for the data.

The FREQ statement specifies the variable that represents the frequency of occurrence for other values in the observation.

The MCMC statement uses a Markov chain Monte Carlo method to impute values for a data set with an arbitrary missing pattern, assuming a multivariate normal distribution for the data.

The MONOTONE statement specifies monotone methods to impute continuous and CLASS variables for a data set with a monotone missing pattern. Note that you can use either an MCMC statement or a MONOTONE statement, but not both. When neither of these two statements is specified, the MCMC method with its default options is used.

The TRANSFORM statement lists the variables to be transformed before the imputation process. The imputed values of these transformed variables will be reverse-transformed to the original forms before the imputation.

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not listed in other statements are used.

The PROC MI statement is the only required statement for the MI procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MI statement. The remaining statements are in alphabetical order.

PROC MI Statement

PROC MI $<$ *options* $>$ **;**

The following table summarizes the options available in the PROC MI statement.

Table 44.1. Summary of PROC MI Options

Tasks	Options
Specify data sets	
input data set	DATA=
output data set with imputed values	OUT=
Specify imputation details	
number of imputations	NIMPUTE=
seed to begin random number generator	SEED=
units to round imputed variable values	ROUND=
maximum values for imputed variable values	MAXIMUM=
minimum values for imputed variable values	MINIMUM=
maximum number of iterations to impute values in the specified range	MINMAXITER=
singularity tolerance	SINGULAR=
Specify statistical analysis	
level for the confidence interval, $(1 - \alpha)$	ALPHA=
means under the null hypothesis	MU0=
Control printed output	
suppress all displayed output	NOPRINT
displays univariate statistics and correlations	SIMPLE

The following options can be used in the PROC MI statement (in alphabetical order):

ALPHA=α

specifies that confidence limits be constructed for the mean estimates with confidence level $100(1-\alpha)\%$, where $0 < \alpha < 1$. The default is ALPHA=0.05.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC MI. By default, the procedure uses the most recently created SAS data set.

MAXIMUM=*numbers*

specifies maximum values for imputed variables. When an intended imputed value is greater than the maximum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the maximum for the corresponding variable

The MAXIMUM= option is related to the MINIMUM= and ROUND= options, which are used to make the imputed values more consistent with the observed variable values. These options are applicable only if you use the MCMC method or the monotone regression method.

When specifying a maximum for the first variable only, you must also specify a missing value after the maximum. Otherwise, the maximum is used for all variables. For example, the MAXIMUM= 100 . option sets a maximum of 100 for the first analysis variable only and no maximum for the remaining variables. The MAXIMUM= . 100 option sets a maximum of 100 for the second analysis variable only and no maximum for the other variables.

MINIMUM=*numbers*

specifies the minimum values for imputed variables. When an intended imputed value is less than the minimum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the minimum for the corresponding variable

MINMAXITER=*number*

specifies the maximum number of iterations for imputed values to be in the specified range when the option MINIMUM or MAXIMUM is also specified. The default is MINMAXITER=100.

MU0=*numbers*
THETA0=*numbers*

specifies the parameter values $\boldsymbol{\mu}_0$ under the null hypothesis $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ for the population means corresponding to the analysis variables. Each hypothesis is tested with a t test. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default is MU0=0.

If a variable is transformed as specified in a TRANSFORM statement, then the same transformation for that variable is also applied to its corresponding specified MU0= value in the t test. If the parameter values $\boldsymbol{\mu}_0$ for a transformed variable is not specified, then a value of zero is used for the resulting $\boldsymbol{\mu}_0$ after transformation.

NIMPUTE=*number*

specifies the number of imputations. The default is NIMPUTE=5. You can specify NIMPUTE=0 to skip the imputation. In this case, only tables of model information, missing data patterns, descriptive statistics (SIMPLE option), and MLE from the EM algorithm (EM statement) are displayed.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*

creates an output SAS data set containing imputation results. The data set includes an index variable, _Imputation_, to identify the imputation number. For each imputation, the data set contains all variables in the input data set with missing values being replaced by the imputed values. See the "Output Data Sets" section on page 2559 for a description of this data set.

ROUND=*numbers*

specifies the units to round variables in the imputation. If only one number is specified, that number is used for all continuous variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. When the CLASS variables are listed in the VAR statement, their corresponding roundoff units are not used. The default number is a missing value, which indicates no rounding for imputed variables.

When specifying a roundoff unit for the first variable only, you must also specify a missing value after the roundoff unit. Otherwise, the roundoff unit is used for all variables. For example, the option "ROUND= 10 ." sets a roundoff unit of 10 for the first analysis variable only and no rounding for the remaining variables. The option "ROUND= . 10" sets a roundoff unit of 10 for the second analysis variable only and no rounding for other variables.

The ROUND= option sets the precision of imputed values. For example, with a roundoff unit of 0.001, each value is rounded to the nearest multiple of 0.001. That is, each value has three significant digits after the decimal point. See Example 44.3 for an illustration of this option.

SEED=*number*

specifies a positive integer to start the pseudo-random number generator. The default is a value generated from reading the time of day from the computer's clock. However, in order to duplicate the results under identical situations, you must use the same value of the seed explicitly in subsequent runs of the MI procedure.

The seed information is displayed in the “Model Information” table so that the results can be reproduced by specifying this seed with the SEED= option. You need to specify the same seed number in the future to reproduce the results.

SIMPLE

displays simple descriptive univariate statistics and pairwise correlations from available cases. For a detailed description of these statistics, see the “Descriptive Statistics” section on page 2535.

SINGULAR=*p*

specifies the criterion for determining the singularity of a covariance matrix based on standardized variables, where $0 < p < 1$. The default is SINGULAR=1E$-$8.

Suppose that $\mathbf{S}$ is a covariance matrix and v is the number of variables in $\mathbf{S}$. Based on the spectral decomposition $\mathbf{S} = \mathbf{\Gamma\Lambda\Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_j, $j = 1, \ldots, v$, where $\lambda_i \geq \lambda_j$ when $i < j$, and $\mathbf{\Gamma}$ is a matrix with the corresponding orthonormal eigenvectors of $\mathbf{S}$ as columns, $\mathbf{S}$ is considered singular when an eigenvalue λ_j is less than $p\bar{\lambda}$, where the average $\bar{\lambda} = \sum_{k=1}^{v} \lambda_k / v$.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MI to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MI procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement (Experimental)

CLASS *variables* **;**

The CLASS statement specifies the classification variables in the VAR statement. Classification variables can be either character or numeric. The CLASS statement must be used in conjunction with the MONOTONE statement.

Class levels are determined from the formatted values of the CLASS variables. Refer to the chapter titled "The FORMAT Procedure" in the *SAS Procedures Guide*.

EM Statement

EM $<$ *options* $>$ **;**

The expectation-maximization (EM) algorithm is a technique for maximum likelihood estimation in parametric models for incomplete data. The EM statement uses the EM algorithm to compute the MLE for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the means and covariance matrix, of a multivariate normal distribution from the input data set with missing values. Either the means and covariances from complete cases or the means and standard deviations from available cases can be used as the initial estimates for the EM algorithm. You can also specify the correlations for the estimates from available cases.

You can also use the EM statement with the NIMPUTE=0 option in the PROC statement to compute the EM estimates without multiple imputation, as shown in Example 44.1 in the "Examples" section on page 2568.

The following seven options are available with the EM statement.

CONVERGE=*p*
XCONV=*p*

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than p for each parameter. That is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E-4.

INITIAL=CC | AC | AC(R=*r*)

sets the initial estimates for the EM algorithm. The INITIAL=CC option uses the means and covariances from complete cases, the INITIAL=AC option uses the means and standard deviations from available cases and the correlations are set to zero, and the INITIAL=AC(R= r) option uses the means and standard deviations from available cases with correlation r, where $-1/(p-1) < r < 1$ and p is the number of variables to be analyzed. The default is INITIAL=AC.

ITPRINT

prints the iteration history in the EM algorithm.

MAXITER=*number*

specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

OUT=*SAS-data-set*

creates an output SAS data set containing results from the EM algorithm. The data set contains all variables in the input data set with missing values being replaced by the expected values from the EM algorithm. See the "Output Data Sets" section on page 2559 for a description of this data set.

OUTEM=*SAS-data-set*

creates an output SAS data set of TYPE=COV containing the MLE of the parameter vector $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. These estimates are computed with the EM algorithm. See the "Output Data Sets" section on page 2559 for a description of this output data set.

OUTITER < **(** *options* **)** > **=***SAS-data-set*

creates an output SAS data set of TYPE=COV containing parameters for each iteration. The data set includes a variable named _Iteration_ to identify the iteration number. The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options to output the mean and covariance parameters. When no options are specified, the output data set contains the mean parameters for each iteration. See the "Output Data Sets" section on page 2559 for a description of this data set.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for other values in the observation, specify the variable name in a FREQ statement. PROC MI then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC MI calculates significance probabilities.

MCMC Statement

MCMC < *options* > **;**

The MCMC statement specifies the details of the MCMC method for imputation.

The following table summarizes the options available for the MCMC statement.

Table 44.2. Summary of Options in MCMC

Tasks	Options
Specify data sets	
input parameter estimates for imputations	INEST=
output parameter estimates used in imputations	OUTEST=
output parameter estimates used in iterations	OUTITER=
Specify imputation details	
monotone/full imputation	IMPUTE=
single/multiple chain	CHAIN=
number of burn-in iterations for each chain	NBITER=
number of iterations between imputations in a chain	NITER=
initial parameter estimates for MCMC	INITIAL=
prior parameter information	PRIOR=
starting parameters	START=
Specify output graphics	
displays time-series plots	TIMEPLOT=
displays autocorrelation plots	ACFPLOT=
graphics catalog name for saving graphics output	GOUT=
Control printed output	
displays worst linear function	WLF
displays initial parameter values for MCMC	DISPLAYINIT

The following options are available for the MCMC statement (in alphabetical order):

ACFPLOT < *(options < / display-options >)* >

displays the autocorrelation function plots of parameters from iterations.

The available options are:

COV < *(< variables > < variable1*variable2 > < . . . variable1*variable2 >)* >

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < *(variables)* >

displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF

displays the plot for the worst linear function.

When the ACFPLOT is specified without the preceding options, the procedure displays plots of means for all variables that are used.

The display-options provide additional information for the autocorrelation function plots. The available display-options are:

CCONF=*color*
specifies the color of the displayed confidence limits. The default is CCONF=BLACK.

CFRAME=*color*
specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CNEEDLES=*color*
specifies the color of the vertical line segments (needles) that connect autocorrelations to the reference line. The default is CNEEDLES=BLACK.

CREF=*color*
specifies the color of the displayed reference line. The default is CREF=BLACK.

CSYMBOL=*color*
specifies the color of the displayed data points. The default is CSYMBOL=BLACK.

HSYMBOL=*number*
specifies the height for data points in percentage screen units. The default is HSYMBOL=1.

LCONF=*linetype*
specifies the line type for the displayed confidence limits. The default is LCONF=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used to compute the autocorrelations. It's generally used for the variances of variables. When a parameter has values less than or equal to zero, the corresponding plot is not created.

LREF=*linetype*
specifies the line type for the displayed reference line. The default is LREF=3, a dashed line.

NLAG=*number*
specifies the maximum lag of the series. The default is NLAG=20. The autocorrelations at each lag are displayed in the graph.

SYMBOL=*value*
specifies the symbol for data points in percentage screen units. The default is SYMBOL=STAR.

TITLE=*'string'*
specifies the title to be displayed in the autocorrelation function plots. The default is TITLE='Autocorrelation Plot'.

WCONF=*number*

specifies the width for the displayed confidence limits in percentage screen units. If you specify the WCONF=0 option, the confidence limits are not displayed. The default is WCONF=1.

WNEEDLES=*number*

specifies the width for the displayed needles that connect autocorrelations to the reference line in percentage screen units. If you specify the WNEEDLES=0 option, the needles are not displayed. The default is WNEEDLES=1.

WREF=*number*

specifies the width for the displayed reference line in percentage screen units. If you specify the WREF=0 option, the reference line is not displayed. The default is WREF=1.

For example, the statement

```
acfplot( mean( y1) cov(y1) /log);
```

requests autocorrelation function plots for the means and variances of the variable `y1`, respectively. Logarithmic transformations of both the means and variances are used in the plots. For a detailed description of the autocorrelation function plot, see the "Autocorrelation Function Plot" section on page 2557; refer also to Schafer (1997, pp. 120-126) and the *SAS/ETS User's Guide*.

CHAIN=SINGLE | MULTIPLE

specifies whether a single chain is used for all imputations or a separate chain is used for each imputation. The default is CHAIN=SINGLE.

DISPLAYINIT

displays initial parameter values in the MCMC process for each imputation.

GOUT=*graphics-catalog*

specifies the graphics catalog for saving graphics output from PROC MI. The default is WORK.GSEG. For more information, refer to the chapter "The GREPLAY Procedure" in *SAS/GRAPH Software: Reference*.

IMPUTE=FULL | MONOTONE

specifies whether a full-data imputation is used for all missing values or a monotone-data imputation is used for a subset of missing values to make the imputed data sets have a monotone missing pattern. The default is IMPUTE=FULL. When IMPUTE=MONOTONE is specified, the order in the VAR statement is used to complete the monotone pattern.

INEST=*SAS-data-set*

names a SAS data set of TYPE=EST containing parameter estimates for imputations. These estimates are used to impute values for observations in the DATA= data set. A detailed description of the data set is provided in the "Input Data Sets" section on page 2558.

INITIAL=EM $<$ (*options*) $>$
INITIAL=INPUT=*SAS-data-set*

specifies the initial mean and covariance estimates for the MCMC process. The default is INITIAL=EM.

You can specify INITIAL=INPUT=*SAS-data-set* to read the initial estimates of the mean and covariance matrix for each imputation from a SAS data set. See the "Input Data Sets" section on page 2558 for a description of this data set.

With INITIAL=EM, PROC MI derives parameter estimates for a posterior mode, the highest observed-data posterior density, from the EM algorithm. The MLE from EM is used to start the EM algorithm for the posterior mode, and the resulting EM estimates are used to begin the MCMC process. The prior information specified in the PRIOR= option is also used in the process to compute the posterior mode.

The following four options are available with INITIAL=EM.

BOOTSTRAP $<$ **=***number* $>$

requests bootstrap resampling, which uses a simple random sample with replacement from the input data set for the initial estimate. You can explicitly specify the number of observations in the random sample. Alternatively, you can implicitly specify the number of observations in the random sample by specifying the proportion $p, 0 < p <= 1$, to request $[np]$ observations in the random sample, where *n* is the number of observations in the data set and $[np]$ is the integer part of np. This produces an overdispersed initial estimate that provides different starting values for the MCMC process. If you specify the BOOTSTRAP option without the number, *p*=0.75 is used by default.

CONVERGE=*p*

XCONV=*p*

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than p for each parameter. That is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E-4.

ITPRINT

prints the iteration history in the EM algorithm for the posterior mode.

MAXITER=*number*

specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

NBITER=*number*

specifies the number of burn-in iterations before the first imputation in each chain. The default is NBITER=200.

NITER=*number*

specifies the number of iterations between imputations in a single chain. The default is NITER=100.

OUTEST=*SAS-data-set*

creates an output SAS data set of TYPE=EST. The data set contains parameter estimates used in each imputation. The data set also includes a variable named _Imputation_ to identify the imputation number. See the "Output Data Sets" section on page 2559 for a description of this data set.

OUTITER < **(** *options* **)** > **=***SAS-data-set*

creates an output SAS data set of TYPE=COV containing parameters used in the imputation step for each iteration. The data set includes variables named _Imputation_ and _Iteration_ to identify the imputation number and iteration number.

The parameters in the output data set depend on the options specified. You can specify options MEAN, STD, COV, LR, LR_POST, and WLF to output parameters of means, standard deviations, covariances, -2 log LR statistic, -2 log LR statistic of the posterior mode, and the worst linear function. When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. See the "Output Data Sets" section on page 2559 for a description of this data set.

PRIOR=*name*

specifies the prior information for the means and covariances. Valid values for *name* are as follows:

JEFFREYS	specifies a noninformative prior.
RIDGE=*number*	specifies a ridge prior.
INPUT=*SAS-data-set*	specifies a data set containing prior information.

For a detailed description of the prior information, see the "Bayesian Estimation of the Mean Vector and Covariance Matrix" section on page 2549 and the "Posterior Step" section on page 2550. If you do not specify the PRIOR= option, the default is PRIOR=JEFFREYS.

The PRIOR=INPUT= option specifies a TYPE=COV data set from which the prior information of the mean vector and the covariance matrix is read. See the "Input Data Sets" section on page 2558 for a description of this data set.

START=VALUE | DIST

specifies that the initial parameter estimates are used as either the starting value (START=VALUE) or as the starting distribution (START=DIST) in the first imputation step of each chain. If the IMPUTE=MONOTONE option is specified, then START=VALUE is used in the procedure. The default is START=VALUE.

TIMEPLOT < *(options < / display-options >)* >

displays the time-series plots of parameters from iterations.

The available options are:

COV < *(< variables > < variable1*variable2 > < . . . variable1*variable2 >)* >

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < *(variables)* >

displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF

displays the plot for the worst linear function.

When the TIMEPLOT is specified without the preceding options, the procedure displays plots of means for all variables are used.

The display-options provide additional information for the time-series plots. The available display-options are:

CCONNECT=*color*

specifies the color for the line segments that connect data points in the time-series plots. The default is CCONNECT=BLACK.

CFRAME=*color*

specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CSYMBOL=*color*

specifies the color of the data points to be displayed in the time-series plots. The default is CSYMBOL=BLACK.

HSYMBOL=*number*

specifies the height for data points in percentage screen units. The default is HSYMBOL=1.

LCONNECT=*linetype*

specifies the line type for the line segments that connect data points in the time-series plots. The default is LCONNECT=1, a solid line.

LOG

requests that the logarithmic transformations of parameters be used. It's generally used for the variances of variables. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot.

SYMBOL=*value*

specifies the symbol for data points in percentage screen units. The default is SYMBOL=PLUS.

TITLE=*'string'*

specifies the title to be displayed in the time-series plots. The default is TITLE='Time-series Plot for Iterations'.

WCONNECT=*number*

specifies the width for the line segments that connect data points in the time-series plots in percentage screen units. If you specify the WCONNECT=0 option, the data points are not connected. The default is WCONNECT=1.

For a detailed description of the time-series plot, see the "Time-Series Plot" section on page 2556 and Schafer (1997, pp. 120–126).

WLF

displays the worst linear function of parameters. This scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is "worst" in the sense that its values from iterations converge most slowly among parameters. For a detailed description of this statistic, see the "Worst Linear Function of Parameters" section on page 2556.

MONOTONE Statement

MONOTONE < *method* < (< *imputed* < = *effects* > > < / *options* >) > >

< ... *method* < (< *imputed* < = *effects* > > < / *options* >) > > **;**

The MONOTONE statement specifies imputation methods for data sets with monotone missingness. You must also specify a VAR statement and the data set must have a monotone missing pattern with variables ordered in the VAR list. When both MONOTONE and MCMC statements are specified, the MONOTONE statement is not used.

For each method, you can specify the imputed variables and optionally, a set of the effects to impute these variables. Each effect is a variable or a combination of variables preceding the imputed variable in the VAR statement. The syntax for specification of effects is the same as for the GLM procedure. See See Chapter 32, "The GLM Procedure," for more information.

One general form of an effect involving several variables is

X1 ∗ *X2* ∗ *A* ∗ *B* ∗ *C* (*D E*)

where A, B, C, D, and E are class variables and X1 and X2 are continuous variables.

If no covariates are specified, then all preceding variables are used as the covariates. That is, each preceding continuous variable is used as a regressor effect, and each preceding class variable is used as a main effect. For the discriminant function method, only the continuous variables can be used as covariate effects.

When a method for continuous variables is specified without imputed variables, the method is used for all continuous variables in the VAR statement that are not specified in other methods. Similarly, when a method for class variables is specified without imputed variables, the method is used for all class variables in the VAR statement that are not specified in other methods.

When a MONOTONE statement is used without specifying any methods, the regression method is used for all continuous variables and the discriminant function method is used for all class variables. The preceding variables of each imputed variable in the VAR statement are used as the covariates.

With a MONOTONE statement, the variables are imputed sequentially in the order given by the VAR statement. For a continuous variable, you can use a regression method, a regression predicted mean matching method, or a propensity score method to impute missing values.

For a nominal class variable, you can use a discriminant function method to impute missing values without using the ordering of the class levels. For a ordinal class variable, you can use a logistic regression method to impute missing values using the ordering of the class levels. For a binary class variable, either a discriminant function method or a logistic regression method can be used.

Note that except for the regression method, all other methods impute values from the observed observation values. You can specify the following methods in a MONOTONE statement.

DISCRIM $<$ **(** *imputed* $<$ **=** *effects* $>$ $<$ / *options* $>$ **)** $>$

specifies the discriminant function method of class variables. Only the continuous variables are allowed as covariate effects. The available options are DETAILS, PCOV=, and PRIOR=. The DETAILS option displays the group means and pooled covariance matrix used in each imputation. The PCOV= option specifies the pooled covariance used in the discriminant method. Valid values for the PCOV= option are:

FIXED uses the observed-data pooled covariance matrix for each imputation.

POSTERIOR draws a pooled covariance matrix from its posterior distribution.

The default is PCOV=POSTERIOR. See the "Discriminant Function Method for Monotone Missing Data" section on page 2544 for a detailed description of the method.

The PRIOR= option specifies the prior probabilities of group membership. Valid values for the PRIOR= option are:

EQUAL set the prior probabilities equal for all groups.

PROPORTIONAL set the prior probabilities proportion to the group sample sizes.

JEFFREYS $<$ =*c* $>$ specifies a noninformative prior, $0 < c < 1$. If the number c is not specified, JEFFREYS=0.5.

RIDGE $<$ =*d* $>$ specifies a ridge prior, $d > 0$. If the number d is not specified, RIDGE=0.25.

The default is PRIOR=JEFFREYS. See the "Discriminant Function Method for Monotone Missing Data" section on page 2544 for a detailed description of the method.

LOGISTIC $<$ **(** *imputed* $<$ **=** *effects* $>$ $<$ / *options* $>$ **)** $>$

specifies the logistic regression method of class variables. The available options are DETAILS, ORDER=, and DESCENDING. The DETAILS option displays the regression coefficients in the logistic regression model used in each imputation.

When the imputed variable has more than two response levels, the ordinal logistic regression method is used. The ORDER= option specifies the sorting order for the levels of the response variable. Valid values for the ORDER= option are:

DATA	sorts by the order of appearance in the input data set
FORMATTED	sorts by their external formatted values
FREQ	sorts by the descending frequency counts
INTERNAL	sorts by the unformatted values

By default, ORDER=FORMATTED.

The option DESCENDING reverses the sorting order for the levels of the response variables.

See the "Logistic Regression Method for Monotone Missing Data" section on page 2546 for a detailed description of the method.

REG | REGRESSION $<$ **(** *imputed* $<$ **=** *effects* $>$ $<$ / **DETAILS** $>$ **)** $>$

specifies the regression method of continuous variables. The DETAILS option displays the regression coefficients in the regression model used in each imputation.

With a regression method, the MAXIMUM=, MINIMUM=, and ROUND= options can be used to make the imputed values more consistent with the observed variable values.

See the "Regression Method for Monotone Missing Data" section on page 2541 for a detailed description of the method.

REGPMM $<$ **(** *imputed* $<$ **=** *effects* $>$ $<$ / *options* $>$ **)** $>$
REGPREDMEANMATCH $<$ **(** *imputed* $<$ **=** *effects* $>$ $<$ / *options* $>$ **)** $>$

specifies the predictive mean matching method for continuous variables. This method is similar to the regression method except that it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

The available options are DETAILS and K=. The DETAILS option displays the regression coefficients in the regression model used in each imputation. The K= option specifies the number of closest observations to be used in the selection. The default is K=5.

Note that an optimal K= value is currently not available in the literature on multiple imputation. The default K=5 is experimental and may change in future releases.

See the "Predictive Mean Matching Method for Monotone Missing Data" section on page 2542 for a detailed description of the method.

PROPENSITY < **(** *imputed* < **=** *effects* > < / *options* > **)** >

specifies the propensity scores method of variables. Each variable is either a class variable or a continuous variable. The available options are DETAILS and NGROUPS=. The DETAILS option displays the regression coefficients in the logistic regression model for propensity scores. The NGROUPS= option specifies the number of groups created based on propensity scores. The default is NGROUPS=5.

See the "Propensity Score Method for Monotone Missing Data" section on page 2543 for a detailed description of the method.

With a MONOTONE statement, the missing values of a variable are imputed when the variable is either explicitly specified in the method or implicitly specified when a method is specified without imputed variables. These variables are imputed sequentially in the order specified in the VAR statement. For example, the following MI procedure

```
proc mi;
   class c1;
   var y1 y2 c1 y3;
   monotone reg(y3= y1 y2 c1) logistic(c1= y1 y2 y1*y2);
run;
```

uses the logistic regression method to impute variable $c1$ from effects $y1$, $y2$, and $y1 * y2$ first, then uses the regression method to impute variable $y3$ from effects $y1$, $y2$, and $c1$. The variables $y1$ and $y2$ are not imputed since $y1$ is the leading variable in the VAR statement and $y2$ is not specified as an imputed variable in the MONOTONE statement.

TRANSFORM Statement

TRANSFORM *transform (variables < / options >)*

< . . . *transform (variables < / options >)* > **;**

The TRANSFORM statement lists the transformations and their associated variables to be transformed. The options are transformation options that provide additional information for the transformation.

The MI procedure assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used. When some variables in a data set are clearly non-normal, it is useful to transform these variables to conform to the multivariate normality assumption. With a TRANSFORM statement, variables are transformed before the imputation process and these transformed variable values are displayed in all of the results. When you specify an OUT= option, the variable values are back-transformed to create the imputed data set.

The following transformations can be used in the TRANSFORM statement.

BOXCOX

specifies the Box-Cox transformation of variables. The variable Y is transformed to $\frac{(\mathrm{Y}+c)^{\lambda}-1}{\lambda}$, where c is a constant such that each value of $\mathrm{Y}+c$ must be positive and the constant $\lambda > 0$.

EXP

specifies the exponential transformation of variables. The variable Y is transformed to $\mathrm{e}^{(\mathrm{Y}+c)}$, where c is a constant.

LOG

specifies the logarithmic transformation of variables. The variable Y is transformed to $\log(\mathrm{Y}+c)$, where c is a constant such that each value of $\mathrm{Y}+c$ must be positive.

LOGIT

specifies the logit transformation of variables. The variable Y is transformed to $\log(\frac{\mathrm{Y}/c}{1-\mathrm{Y}/c})$, where the constant $c > 0$ and the values of Y/c must be between 0 and 1.

POWER

specifies the power transformation of variables. The variable Y is transformed to $(\mathrm{Y}+c)^{\lambda}$, where c is a constant such that each value of $\mathrm{Y}+c$ must be positive and the constant $\lambda \neq 0$.

The following options provide the constant c and λ values in the transformations.

C=*number*

specifies the c value in the transformation. The default is $c = 1$ for logit transformation and $c = 0$ for other transformations.

LAMBDA=*number*

specifies the λ value in the power and Box-Cox transformations. You must specify the λ value for these two transformations.

For example, the statement

```
transform log(y1) power(y2/c=1 lambda=.5);
```

requests that variables $\log(y1)$, a logarithmic transformation for the variable y1, and $\sqrt{y2+1}$, a power transformation for the variable y2, be used in the imputation.

If the MU0= option is used to specify a parameter value $\boldsymbol{\mu}_0$ for a transformed variable, the same transformation for the variable is also applied to its corresponding MU0= value in the t test. Otherwise, $\boldsymbol{\mu}_0 = 0$ is used for the transformed variable. See Example 44.10 for a usage of the TRANSFORM statement.

VAR Statement

VAR *variables* **;**

The VAR statement lists the variables to be analyzed. The variables can be either character or numeric. If you omit the VAR statement, all continuous variables not mentioned in other statements are used. The VAR statement is required if you specify a MONOTONE statement, an IMPUTE=MONOTONE option in the MCMC statement, or more than one number in the MU0=, MAXIMUM=, MINIMUM=, or ROUND= option.

The character variables are allowed only when they are specified as CLASS variables and the MONOTONE statement is also specified.

Details

Descriptive Statistics

Suppose $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n)\prime$ is the $(n{\times}p)$ matrix of complete data, which may not be fully observed, n_0 is the number of observations fully observed, and n_j is the number of observations with observed values for variable Y_j.

With complete cases, the sample mean vector is

$$\overline{\mathbf{y}} = \frac{1}{n_0} \sum \mathbf{y}_i$$

and the CSSCP matrix is

$$\sum (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})\prime$$

where each summation is over the fully observed observations.

The sample covariance matrix is

$$\mathbf{S} = \frac{1}{n_0 - 1} \sum (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})\prime$$

and is an unbiased estimate of the covariance matrix.

The correlation matrix $\mathbf{R}$ containing the Pearson product-moment correlations of the variables is derived by scaling the corresponding covariance matrix:

$$\mathbf{R} = \mathbf{D}^{-1} \mathbf{S}\, \mathbf{D}^{-1}$$

where $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the square roots of the diagonal elements of $\mathbf{S}$.

With available cases, the corrected sum of squares for variable Y_j is

$$\sum (y_{ji} - \overline{y}_j)^2$$

where $\overline{y}_j = \frac{1}{n_j} \sum y_{ji}$ is the sample mean and each summation is over observations with observed values for variable Y_j.

The variance is

$$s^2_{jj} = \frac{1}{n_j - 1} \sum (y_{ji} - \overline{y}_j)^2$$

The correlations for available cases contain pairwise correlations for each pair of variables. Each correlation is computed from all observations that have nonmissing values for the corresponding pair of variables.

EM Algorithm for Data with Missing Values

The EM algorithm (Dempster, Laird, and Rubin 1977) is a technique that finds maximum likelihood estimates in parametric models for incomplete data. The books by Little and Rubin (1987), Schafer (1997), and McLachlan and Krishnan (1997) provide detailed description and applications of the EM algorithm.

The EM algorithm is an iterative procedure that finds the MLE of the parameter vector by repeating the following steps:

1. The expectation E-step:
Given a set of parameter estimates, such as a mean vector and covariance matrix for a multivariate normal distribution, the E-step calculates the conditional expectation of the complete-data log likelihood given the observed data and the parameter estimates.

2. The maximization M-step:
Given a complete-data log likelihood, the M-step finds the parameter estimates to maximize the complete-data log likelihood from the E-step.

The two steps are iterated until the iterations converge.

In the EM process, the observed-data log likelihood is non-decreasing at each iteration. For multivariate normal data, suppose there are G groups with distinct missing patterns. Then the observed-data log likelihood being maximized can be expressed as

$$\log L(\boldsymbol{\theta}|Y_{obs}) = \sum_{g=1}^{G} \log L_g(\boldsymbol{\theta}|Y_{obs})$$

where $\log L_g(\boldsymbol{\theta}|Y_{obs})$ is the observed-data log likelihood from the g_{th} group, and

$$\log L_g(\boldsymbol{\theta}|Y_{obs}) = -\frac{n_g}{2} \log |\Sigma_g| - \frac{1}{2} \sum_{ig} (\mathbf{y}_{ig} - \boldsymbol{\mu}_g)' {\Sigma_g}^{-1} (\mathbf{y}_{ig} - \boldsymbol{\mu}_g)$$

where n_g is the number of observations in the g_{th} group, the summation is over observations in the g_{th} group, $\mathbf{y}_{ig}$ is a vector of observed values corresponding to observed variables, $\boldsymbol{\mu}_g$ is the corresponding mean vector, and $\boldsymbol{\Sigma}_g$ is the associated covariance matrix.

A sample covariance matrix is computed at each step of the EM algorithm. If the covariance matrix is singular, the linearly dependent variables for the observed data are excluded from the likelihood function. That is, for each observation with linear dependency among its observed variables, the dependent variables are excluded from the likelihood function. Note that this may result in an unexpected change in the likelihood between iterations prior to the final convergence.

Refer to Schafer (1997, pp. 163–181) for a detailed description of the EM algorithm for multivariate normal data.

PROC MI uses the means and standard deviations from available cases as the initial estimates for the EM algorithm. The correlations are set to zero. It provides a good starting value with positive definite covariance matrix. For a discussion of suggested starting values for the algorithm, see Schafer (1997, p. 169).

You can specify the convergence criterion with the CONVERGE= option in the EM statement. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. You can also specify the maximum number of iterations used in the EM algorithm with the MAXITER= option.

The MI procedure displays tables of the initial parameter estimates used to begin the EM process and the MLE parameter estimates derived from EM. You can also display the EM iteration history with the ITPRINT option. PROC MI lists the iteration number, the likelihood -2 Log L, and the parameter values $\boldsymbol{\mu}$ at each iteration. You can also save the MLE derived from the EM algorithm in a SAS data set specified with the OUTEM= option.

Statistical Assumptions for Multiple Imputation

The MI procedure assumes that the data are from a continuous multivariate distribution and contain missing values that can occur for any of the variables. It also assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used.

Suppose $\mathbf{Y}$ is the $n \times p$ matrix of complete data, which is not fully observed, and denote the observed part of $\mathbf{Y}$ by $\mathbf{Y}_{obs}$ and the missing part by $\mathbf{Y}_{mis}$. The SAS MI and MIANALYZE procedures assume that the missing data are missing at random (MAR), that is, the probability that an observation is missing can depend on $\mathbf{Y}_{obs}$, but not on $\mathbf{Y}_{mis}$ (Rubin 1976; 1987, p. 53).

To be more precise, suppose that $\mathbf{R}$ is the $n \times p$ matrix of response indicators whose elements are zero or one depending on whether the corresponding elements of Y are missing or observed. Then the MAR assumption is that the distribution of $\mathbf{R}$ can depend on Y_{obs} but not on Y_{mis}.

$$\mathrm{pr}(\mathbf{R}|Y_{obs}, Y_{mis}) = \mathrm{pr}(\mathbf{R}|Y_{obs})$$

For example, consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 that has missing values. MAR assumes that the probability that Y_3 is missing for an individual can be related to the individual's values of variables Y_1 and Y_2, but not to its value of Y_3. On the other hand, if a complete case and an incomplete case for Y_3 with exactly the same values for variables Y_1 and Y_2 have systematically different values, then there exists a response bias for Y_3, and MAR is violated.

The MAR assumption is not the same as missing completely at random (MCAR), which is a special case of MAR. Under the MCAR assumption, the missing data values are a simple random sample of all data values; the missingness does not depend on the values of any variables in the data set.

Although the MAR assumption cannot be verified with the data and it can be questionable in some situations, the assumption becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; van Buuren, Boshuizen, and Knook, 1999, p. 687).

Furthermore, the MI and MIANALYZE procedures assume that the parameters $\boldsymbol{\theta}$ of the data model and the parameters ϕ of the model for the missing data indicators are distinct. That is, knowing the values of $\boldsymbol{\theta}$ does not provide any additional information about ϕ, and vice versa. If both the MAR and distinctness assumptions are satisfied, the missing-data mechanism is said to be ignorable (Rubin 1987, pp. 50–54; Schafer 1997, pp. 10–11) .

Missing Data Patterns

The MI procedure sorts the data into groups based on whether an individual's value is observed or missing for each variable to be analyzed. Note that the input data set does not need to be sorted in any order.

For example, with variables Y_1, Y_2, and Y_3 (in that order) in a data set, up to eight groups of observations can be formed from the data set. The following figure displays the eight groups of observations and an unique missing pattern for each group:

```
                    Missing Data Patterns

                   Group    Y1    Y2    Y3

                     1      X     X     X
                     2      X     X     .
                     3      X     .     X
                     4      X     .     .
                     5      .     X     X
                     6      .     X     .
                     7      .     .     X
                     8      .     .     .
```

Figure 44.6. Missing Data Patterns

Here, an "X" means that the variable is observed in the corresponding group and a "." means that the variable is missing.

The variable order is used to derive the order of the groups from the data set, and thus, determines the order of missing values in the data to be imputed. If you specify a different order of variables in the VAR statement, then the results are different even if the other specifications remain the same.

A data set with variables Y_1, Y_2, ..., Y_p (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables Y_k, $k > j$, are missing for that individual. Alternatively, when a variable Y_j is observed for a particular individual, it is assumed that all previous variables Y_k, $k < j$, are also observed for that individual.

For example, the following figure displays a data set of three variables with a monotone missing pattern.

```
                    Monotone Missing Data Patterns

                       Group     Y1     Y2     Y3

                         1       X      X      X
                         2       X      X      .
                         3       X      .      .
```

Figure 44.7. Monotone Missing Patterns

The following figure displays a data set of three variables with a non-monotone missing pattern.

```
                  Non-monotone Missing Data Patterns

                       Group     Y1     Y2     Y3

                         1       X      X      X
                         2       X      .      X
                         3       .      X      .
                         4       .      .      X
```

Figure 44.8. Non-monotone Missing Patterns

A data set with an *arbitrary missing pattern* is a data set with either a monotone missing pattern or a non-monotone missing pattern.

Imputation Methods

This section describes the methods for multiple imputation that are available in the MI procedure. The method of choice depends on the pattern of missingness in the data and the type of the imputed variable, as summarized in the following table:

Table 44.3. Imputation Methods in PROC MI

Pattern of Missingness	**Type of Imputed Variable**	**Recommended Methods**
Monotone	Continuous	• Regression • Predicted Mean Matching • Propensity Score
Monotone	Classification (Ordinal)	• Logistic Regression
Monotone	Classification (Nominal)	• Discriminant Function Method
Arbitrary	Continuous	• MCMC Full-Data Imputation • MCMC Monotone-Data Imputation

To impute missing values for a continuous variable in data sets with monotone missing patterns, you should use either a parametric method that assumes multivariate normality or a nonparametric method that uses propensity scores (Rubin 1987, p. 124, 158; Lavori, Dawson, and Shera 1995). Parametric methods available include the regression method (Rubin 1987, pp. 166–167) and the predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996).

To impute missing values for a CLASS variable in data sets with monotone missing patterns, you should use the logistic regression method or the discriminant function method. Use the logistic regression method when the CLASS variable has a binary or ordinal response, and the discriminant function method when the CLASS variable has a binary or nominal response.

For continuous variables in data sets with arbitrary missing patterns, you can use the Markov Chain Monte Carlo (MCMC) method (Schafer 1997) to impute either all the missing values or just enough missing values to make the imputed data sets have monotone missing patterns.

With a monotone missing data pattern, you have greater flexibility in your choice of imputation models. In addition to the MCMC method, you can implement other methods, such as the regression method, that do not use Markov chains. You can also specify a different set of covariates for each imputed variable.

With an arbitrary missing data pattern, you can often use the MCMC method, which creates multiple imputations by drawing simulations from a Bayesian predictive distribution for normal data. Another way to handle a data set with an arbitrary missing data pattern is to use the MCMC approach to impute just enough values to make the missing data pattern monotone. Then, you can use a more flexible imputation method. This approach is described in the “Producing Monotone Missingness with the MCMC Method” section on page 2552.

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from multivariate normality if the amount of missing information is not large, because the imputation model is effectively applied not to the entire data set but only to its missing part (Schafer 1997, pp. 147–148).

You can also use a TRANSFORM statement to transform variables to conform to the multivariate normality assumption. Variables are transformed before the imputation process and then are reverse-transformed to create the imputed data set.

Li (1988) presented a theoretical argument for convergence of the MCMC method in the continuous case and used it to create imputations for incomplete multivariate continuous data. In practice, however, it is not easy to check the convergence of a Markov chain, especially for a large number of parameters. PROC MI generates statistics and plots which you can use to check for convergence of the MCMC process. The details are described in the “Checking Convergence in MCMC” section on page 2555.

Regression Method for Monotone Missing Data

The regression method is the default imputation method for continuous variables in a data set with a monotone missing pattern.

In the regression method, a regression model is fitted for a continuous variable with the covariates constructed from a set of effects. Based on the fitted regression model, a new regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 166–167). That is, for a continuous variable Y_j with missing values, a model

$$Y_j = \beta_0 + \beta_1 \, X_1 + \beta_2 \, X_2 + \ldots + \beta_k \, X_k$$

is fitted using observations with observed values for the variable Y_j and its covariates X_1, X_2, ..., X_k.

The fitted model includes the regression parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix derived from the intercept and covariates X_1, X_2, ..., X_k.

The following steps are used to generate imputed values for each imputation:

1. New parameters $\beta_* = (\beta_{*0}, \beta_{*1}, ..., \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

$$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - k - 1)/g$$

where g is a $\chi^2_{n_j-k-1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

$$\beta_* = \hat{\beta} + \sigma_{*j} \mathbf{V}_{hj}' \mathbf{Z}$$

where $\mathbf{V}_{hj}'$ is the upper triangular matrix in the Cholesky decomposition,
$\mathbf{V}_j = \mathbf{V}_{hj}' \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k+1$ independent random normal variates.

2. The missing values are then replaced by

$$\beta_{*0} + \beta_{*1} \, x_1 + \beta_{*2} \, x_2 + \ldots + \beta_{*(k)} \, x_k + z_i \, \sigma_{*j}$$

where $x_1, x_2, ..., x_k$ are the values of the covariates and z_i is a simulated normal deviate.

Predictive Mean Matching Method for Monotone Missing Data

The predictive mean matching method is also an imputation method available for continuous variables. It is similar to the regression method except that for each missing value, it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

Following the description of the model in the "Regression Method for Monotone Missing Data" section on page 2541, the following steps are used to generate imputed values:

1. New parameters $\beta_* = (\beta_{*0}, \beta_{*1}, ..., \beta_{*(k)})$ and σ^2_{*j} are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$, σ^2_j, and $\mathbf{V}_j$. The variance is drawn as

$$\sigma^2_{*j} = \hat{\sigma}^2_j (n_j - k - 1)/g$$

where g is a $\chi^2_{n_j-k-1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

$$\beta_* = \hat{\beta} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj} \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k+1$ independent random normal variates.

2. For each missing value, a predicted value

$$y_{i*} = \beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(k)}\, x_k$$

is computed with the covariate values $x_1, x_2, ..., x_k$.

3. A set of k_0 observations whose corresponding predicted values are closest to y_{i*} is generated. You can specify k_0 with the K= option.

4. The missing value is then replaced by a value drawn randomly from these k_0 observed values.

The predictive mean matching method requires the number of closest observations to be specified. A smaller k_0 tends to increase the correlation among the multiple imputations for the missing observation and results in a higher variability of point estimators in repeated sampling. On the other hand, a larger k_0 tends to lessen the effect from the imputation model and results in biased estimators (Schenker and Taylor 1996, p. 430). An optimal k_0 is currently not available in the literature on multiple imputation. The default is K=5. This default value is experimental and may change in future releases.

The predictive mean matching method ensures that imputed values are plausible and may be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Note that in SAS 9.0, the predictive mean matching method replaces each missing value by the observed value closest to its predicted value. This may result in a higher variability of point estimators in repeated sampling (Schenker and Taylor 1996, p. 430).

Propensity Score Method for Monotone Missing Data

The propensity score method is another imputation method available for continuous variables when the data set has a monotone missing pattern.

A propensity score is generally defined as the conditional probability of assignment to a particular treatment given a vector of observed covariates (Rosenbaum and Rubin 1983). In the propensity score method, for a variable with missing values, a propensity score is generated for each observation to estimate the probability that the observation is missing. The observations are then grouped based on these propensity scores, and an approximate Bayesian bootstrap imputation (Rubin 1987, p. 124) is applied to each group (Lavori, Dawson, and Shera 1995).

The propensity score method uses the following steps to impute values for variable Y_j with missing values:

1. Create an indicator variable R_j with the value 0 for observations with missing Y_j and 1 otherwise.

2. Fit a logistic regression model

$$\text{logit}(p_j) = \beta_0 + \beta_1 \, X_1 + \beta_2 \, X_2 + \ldots + \beta_k \, X_k$$

where $X_1, X_2, ..., X_k$ are covariates for Y_j, $p_j = Pr(R_j = 0 | X_1, X_2, ..., X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.

3. Create a propensity score for each observation to estimate the probability that it is missing.

4. Divide the observations into a fixed number of groups (typically assumed to be five) based on these propensity scores.

5. Apply an approximate Bayesian bootstrap imputation to each group. In group k, suppose that Y_{obs} denotes the n_1 observations with nonmissing Y_j values and Y_{mis} denotes the n_0 observations with missing Y_j. The approximate Bayesian bootstrap imputation first draws n_1 observations randomly with replacement from Y_{obs} to create a new data set Y^*_{obs}. This is a nonparametric analogue of drawing parameters from the posterior predictive distribution of the parameters. The process then draws the n_0 values for Y_{mis} randomly with replacement from Y^*_{obs}.

Steps 1 through 5 are repeated sequentially for each variable with missing values.

Note that the propensity score method was originally designed for a randomized experiment with repeated measures on the response variables. The goal was to impute the missing values on the response variables. The method uses only the covariate information that is associated with whether the imputed variable values are missing. It does not use correlations among variables. It is effective for inferences about the distributions of individual imputed variables, such as an univariate analysis, but it is not appropriate for analyses involving relationship among variables, such as a regression analysis (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

Discriminant Function Method for Monotone Missing Data

The discriminant function method is the default imputation method for CLASS variables in a data set with a monotone missing pattern.

For a nominal class variable Y_j with responses 1, ..., g, and a set of effects from its preceding variables, if the covariates X_1, X_2, ..., X_k associated with these effects within each group is approximately multivariate normal and the within-group covariance matrices are approximately equal, the discriminant function method (Brand 1999, pp. 95–96) can be used to impute missing values for the variable Y_j.

Denote the group-specific means for covariates X_1, X_2, ..., X_k by

$$\overline{\mathbf{X}}_t = (\overline{X}_{t1}, \overline{X}_{t2}, \ldots, \overline{X}_{tk}),\ t = 1, 2, ..., g$$

then the pooled covariance matrix is computed as

$$\mathbf{S} = \frac{1}{n-g} \sum_{t=1}^{g} (n_t - 1)\mathbf{S}_t$$

where $\mathbf{S}_t$ is the within-group covariance matrix, n_t is the group-specific sample size, and $n = \sum_{t=1}^{g} n_t$ is the total sample size.

In each imputation, new parameters of the group-specific means ($\mathbf{m}_{*t}$), pooled covariance matrix ($\mathbf{S}_*$), and prior probabilities of group membership (q_{*t}) can be drawn from their corresponding posterior distributions (Schafer 1997, p. 356).

Pooled Covariance Matrix and Group-specific Means

For each imputation, the MI procedure uses either the fixed observed pooled covariance matrix (PCOV=FIXED) or a drawn pooled covariance matrix (PCOV=POSTERIOR) from its posterior distribution with a noninformative prior. That is,

$$\mathbf{\Sigma}|\mathbf{X} \quad \sim \quad W^{-1}\left(n-g,\, (n-g)\mathbf{S} \right)$$

where W^{-1} is an inverted Wishart distribution.

The group-specific means are then drawn from their posterior distributions with a noninformative prior

$$\boldsymbol{\mu}_t | (\mathbf{\Sigma}, \overline{\mathbf{X}}_t) \quad \sim \quad N\left(\overline{\mathbf{X}}_t,\ \frac{1}{n_t}\mathbf{\Sigma} \right)$$

See the “Bayesian Estimation of the Mean Vector and Covariance Matrix” section on page 2549 for a complete description of the inverted Wishart distribution and posterior distributions using a noninformative prior.

Prior Probabilities of Group Membership

The prior probabilities are computed through the drawing of new group sample sizes. When the total sample size n is considered fixed, the group sample sizes $(n_1, n_2, ..., n_g)$ has a multinomial distribution. A new multinomial parameters (group sample sizes) can be drawn from its posterior distribution using a Dirichlet prior with parameters $(\alpha_1, \alpha_2, ..., \alpha_g)$.

After the new sample sizes are drawn from the posterior distribution of $(n_1, n_2, ..., n_g)$, the prior probabilities q_{*t} are computed proportionally to the drawn sample sizes.

Refer to Schafer (1997, pp. 247–255) for a complete description of the Dirichlet prior.

Imputation Steps

The discriminant function method uses the following steps in each imputation to impute values for a nominal class variable Y_j with g responses:

1. Draw a pooled covariance matrix $\mathbf{S}_*$ from its posterior distribution if the PCOV=POSTERIOR option is used.

2. For each group, draw group means $\mathbf{m}_{*t}$ from the observed group mean $\overline{\mathbf{X}}_t$ and either the observed pooled covariance matrix (PCOV=FIXED) or the drawn pooled covariance matrix $\mathbf{S}_*$ (PCOV=POSTERIOR).

3. For each group, compute or draw q_{*t}, prior probabilities of group membership, based on the PRIOR= option:

- PRIOR=EQUAL, $q_{*t} = 1/g$, prior probabilities of group membership are all equal.
- PRIOR=PROPORTIONAL, $q_{*t} = n_t/n$, prior probabilities are proportional to their group sample sizes.
- PRIOR=JEFFREYS=c, a noninformative Dirichlet prior with $\alpha_t = c$ is used.
- PRIOR=RIDGE=d, a ridge prior is used with $\alpha_t = d * n_t/n$ for $d \geq 1$ and $\alpha_t = d * n_t$ for $d < 1$.

4. With the group means $\mathbf{m}_{*t}$, the pooled covariance matrix $\mathbf{S}_*$, and the prior probabilities of group membership q_{*t}, the discriminant function method derives linear discriminant function and computes the posterior probabilities of an observation belonging to each group

$$p_t(\mathbf{x}) = \frac{\exp(-0.5 D_t^2(\mathbf{x}))}{\sum_{u=1}^{g} \exp(-0.5 D_u^2(\mathbf{x}))}$$

where $D_t^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_{*t})'\mathbf{S}_*^{-1}(\mathbf{x} - \mathbf{m}_{*t}) - 2\log(q_{*t})$ is the generalized squared distance from $\mathbf{x}$ to group t.

5. Draw a random uniform variate u, between 0 and 1, for each observation with missing group value. With the posterior probabilities, $p_1(\mathbf{x}) + p_2(\mathbf{x}) + ..., + p_g(\mathbf{x}) = 1$, the discriminant function method imputes $Y_j = 1$ if the value of u is less than $p_1(\mathbf{x})$, $Y_j = 2$ if the value is greater than or equal to $p_1(\mathbf{x})$ but less than $p_1(\mathbf{x}) + p_2(\mathbf{x})$, and so on.

Logistic Regression Method for Monotone Missing Data

The logistic regression method is another imputation method available for CLASS variables in a data set with a monotone missing pattern.

In the logistic regression method, a logistic regression model is fitted for a class variable with a set of covariates constructed from the effects. For a binary class variable, based on the fitted regression model, a new logistic regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 169–170).

For a binary variable Y_j with responses 1 and 2, a logistic regression model is fitted using observations with observed values for the imputed variable Y_j and its covariates X_1, X_2, ..., X_k.

$$\text{logit}(p_j) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

where $X_1, X_2, ..., X_k$ are covariates for Y_j, $p_j = \Pr(R_j = 1 | X_1, X_2, ..., X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.

The fitted model includes the regression parameter estimates $\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1, ..., \hat{\beta}_k)$ and the associated covariance matrix $\mathbf{V}_j$.

The following steps are used to generate imputed values for a binary variable Y_j with responses 1 and 2:

1. New parameters $\beta_* = (\beta_{*0}, \beta_{*1}, ..., \beta_{*(k)})$ are drawn from the posterior predictive distribution of the parameters.

$$\beta_* = \hat{\beta} + \mathbf{V}'_{hj}\mathbf{Z}$$

where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition,
$\mathbf{V}_j = \mathbf{V}'_{hj}\mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k+1$ independent random normal variates.

2. For an observation with missing Y_j and covariates $x_1, x_2, ..., x_k$, compute the expected probability that $Y_j = 1$.

$$p_j = \frac{\exp(\mu_j)}{1 + \exp(\mu_j)}$$

where $\mu_j = \beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(k)}\, x_k$.

3. Draw a random uniform variate, u, between 0 and 1. If the value of u is less than p_j, impute $Y_j = 1$, otherwise impute $Y_j = 2$.

The preceding logistic regression method can be extended to include the ordinal class variables with more than two levels of responses. The options ORDER= and DESCENDING can be used to specify the sorting order for the levels of the imputed variables.

MCMC Method for Arbitrary Missing Data

The Markov Chain Monte Carlo (MCMC) method originated in physics as a tool for exploring equilibrium distributions of interacting molecules. In statistical applications, it is used to generate pseudo-random draws from multidimensional and otherwise intractable probability distributions via Markov chains. A Markov chain is a sequence of random variables in which the distribution of each element depends only on the value of the previous one.

In MCMC simulation, one constructs a Markov chain long enough for the distribution of the elements to stabilize to a stationary distribution, which is the distribution of interest. By repeatedly simulating steps of the chain, the method simulates draws from the distribution of interest. Refer to Schafer (1997) for a detailed discussion of this method.

In Bayesian inference, information about unknown parameters is expressed in the form of a posterior probability distribution. This posterior distribution is computed using Bayes' theorem

$$p(\boldsymbol{\theta}|y) = \frac{p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

MCMC has been applied as a method for exploring posterior distributions in Bayesian inference. That is, through MCMC, one can simulate the entire joint posterior distribution of the unknown quantities and obtain simulation-based estimates of posterior parameters that are of interest.

In many incomplete data problems, the observed-data posterior $p(\boldsymbol{\theta}|Y_{obs})$ is intractable and cannot easily be simulated. However, when Y_{obs} is augmented by an estimated/simulated value of the missing data Y_{mis}, the complete-data posterior $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis})$ is much easier to simulate. Assuming that the data are from a multivariate normal distribution, data augmentation can be applied to Bayesian inference with missing data by repeating the following steps:

1. **The imputation I-step:**
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. That is, if you denote the variables with missing values for observation *i* by $Y_{i(mis)}$ and the variables with observed values by $Y_{i(obs)}$, then the I-step draws values for $Y_{i(mis)}$ from a conditional distribution for $Y_{i(mis)}$ given $Y_{i(obs)}$.

2. **The posterior P-step:**
Given a complete sample, the P-step simulates the posterior population mean vector and covariance matrix. These new estimates are then used in the next I-step. Without prior information about the parameters, a noninformative prior is used. You can also use other informative priors. For example, a prior information about the covariance matrix can be helpful to stabilize the inference about the mean vector for a near singular covariance matrix.

The two steps are iterated long enough for the results to be reliable for a multiply imputed data set (Schafer 1997, p. 72). That is, with a current parameter estimate $\boldsymbol{\theta}^{(t)}$ at the $t\mathrm{th}$ iteration, the I-step draws $Y_{mis}^{(t+1)}$ from $p(Y_{mis}|Y_{obs}, \boldsymbol{\theta}^{(t)})$ and the P-step draws $\boldsymbol{\theta}^{(t+1)}$ from $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis}^{(t+1)})$.

This creates a Markov chain

$(Y_{mis}^{(1)}, \boldsymbol{\theta}^{(1)})$, $(Y_{mis}^{(2)}, \boldsymbol{\theta}^{(2)})$, ... ,

which converges in distribution to $p(Y_{mis}, \boldsymbol{\theta}|Y_{obs})$. Assuming the iterates converge to a stationary distribution, the goal is to simulate an approximately independent draw of the missing values from this distribution.

To validate the imputation results, you should repeat the process with different random number generators and starting values based on different initial parameter estimates.

The next three sections provide details for the imputation step, Bayesian estimation of the mean vector and covariance matrix, and the posterior step.

Imputation Step

In each iteration, starting with a given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the imputation step draws values for the missing data from the conditional distribution Y_{mis} given Y_{obs}.

Suppose $\boldsymbol{\mu} = [\boldsymbol{\mu}_1', \boldsymbol{\mu}_2']'$ is the partitioned mean vector of two sets of variables, Y_{obs} and Y_{mis}, where $\boldsymbol{\mu}_1$ is the mean vector for variables Y_{obs} and $\boldsymbol{\mu}_2$ is the mean vector for variables Y_{mis}.

Also suppose

$$\boldsymbol{\Sigma} = \left[\begin{array}{cc} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}' & \boldsymbol{\Sigma}_{22} \end{array} \right]$$

is the partitioned covariance matrix for these variables, where $\boldsymbol{\Sigma}_{11}$ is the covariance matrix for variables Y_{obs}, $\boldsymbol{\Sigma}_{22}$ is the covariance matrix for variables Y_{mis}, and $\boldsymbol{\Sigma}_{12}$ is the covariance matrix between variables Y_{obs} and variables Y_{mis}.

By using the sweep operator (Goodnight 1979) on the pivots of the $\mathbf{\Sigma}_{11}$ submatrix, the matrix becomes

$$\begin{bmatrix} \mathbf{\Sigma}_{11}^{-1} & \mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12} \\ -\mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1} & \mathbf{\Sigma}_{22.1} \end{bmatrix}$$

where $\mathbf{\Sigma}_{22.1} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$ can be used to compute the conditional covariance matrix of $\mathbf{Y}_{mis}$ after controlling for $\mathbf{Y}_{obs}$.

For an observation with the preceding missing pattern, the conditional distribution of Y_{mis} given $Y_{obs} = \mathbf{y}_1$ is a multivariate normal distribution with the mean vector

$$\boldsymbol{\mu}_{2.1} = \boldsymbol{\mu}_2 + \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1)$$

and the conditional covariance matrix

$$\mathbf{\Sigma}_{22.1} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$$

Bayesian Estimation of the Mean Vector and Covariance Matrix

Suppose that $\mathbf{Y} = (\mathbf{y}_1', \mathbf{y}_2', ..., \mathbf{y}_n')'$ is an $(n{\times}p)$ matrix made up of n $(p{\times}1)$ independent vectors $\mathbf{y}_i$, each of which has a multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Lambda}$. Then the SSCP matrix

$$\mathbf{A} = \mathbf{Y}'\mathbf{Y} = \sum_i \mathbf{y}_i\mathbf{y}_i'$$

has a Wishart distribution $W(n, \mathbf{\Lambda})$.

When each observation $\mathbf{y}_i$ is distributed with a multivariate normal distribution with an unknown mean $\boldsymbol{\mu}$, then the CSSCP matrix

$$\mathbf{A} = \sum_i (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})'$$

has a Wishart distribution $W(n-1, \mathbf{\Lambda})$.

If $\mathbf{A}$ has a Wishart distribution $W(n, \mathbf{\Lambda})$, then $\mathbf{B} = \mathbf{A}^{-1}$ has an inverted Wishart distribution $W^{-1}(n, \mathbf{\Psi})$, where n is the degrees of freedom and $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ is the precision matrix (Anderson 1984).

Note that, instead of using the parameter $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ for the inverted Wishart distribution, Schafer (1997) uses the parameter $\mathbf{\Lambda}$.

Suppose that each observation in the data matrix $\mathbf{Y}$ has a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. Then with a prior inverted Wishart distribution for $\mathbf{\Sigma}$ and a prior normal distribution for $\boldsymbol{\mu}$

$$\begin{aligned} \mathbf{\Sigma} \quad &\sim \quad W^{-1}(m, \mathbf{\Psi}) \\ \boldsymbol{\mu}|\mathbf{\Sigma} \quad &\sim \quad N\left(\boldsymbol{\mu}_0, \ \frac{1}{\tau}\mathbf{\Sigma}\right) \end{aligned}$$

where $\tau > 0$ is a fixed number.

The posterior distribution (Anderson 1984, p. 270; Schafer 1997, p. 152) is

$$\begin{aligned}\boldsymbol{\Sigma}|\mathbf{Y} &\sim W^{-1}\left(n+m,\ (n-1)\mathbf{S}+\boldsymbol{\Psi}+\frac{n\tau}{n+\tau}(\overline{\mathbf{y}}-\boldsymbol{\mu}_0)(\overline{\mathbf{y}}-\boldsymbol{\mu}_0)' \right) \\ \boldsymbol{\mu}|(\boldsymbol{\Sigma},\mathbf{Y}) &\sim N\left(\frac{1}{n+\tau}(n\overline{\mathbf{y}}+\tau\boldsymbol{\mu}_0),\ \frac{1}{n+\tau}\boldsymbol{\Sigma} \right)\end{aligned}$$

where $(n-1)\mathbf{S}$ is the CSSCP matrix.

Posterior Step

In each iteration, the posterior step simulates the posterior population mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ from prior information for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and the complete sample estimates.

You can specify the prior parameter information using one of the following methods:

- PRIOR=JEFFREYS, which uses a noninformative prior.
- PRIOR=INPUT=, which provides a prior information for $\boldsymbol{\Sigma}$ in the data set. Optionally, it also provides a prior information for $\boldsymbol{\mu}$ in the data set.
- PRIOR=RIDGE=, which uses a ridge prior.

The next four subsections provide details of the posterior step for different prior distributions.

1. A Noninformative Prior

Without prior information about the mean and covariance estimates, a noninformative prior can be used by specifying the PRIOR=JEFFREYS option. The posterior distributions (Schafer 1997, p. 154) are

$$\begin{aligned}\boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n-1,\ (n-1)\mathbf{S} \right) \\ \boldsymbol{\mu}^{(t+1)}|(\boldsymbol{\Sigma}^{(t+1)},\mathbf{Y}) &\sim N\left(\overline{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)} \right)\end{aligned}$$

2. An Informative Prior for μ and Σ

When prior information is available for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, you can provide it with a SAS data set that you specify with the PRIOR=INPUT= option.

$$\begin{aligned}\boldsymbol{\Sigma} &\sim W^{-1}\left(d^*,\ d^*\mathbf{S}^* \right) \\ \boldsymbol{\mu}|\boldsymbol{\Sigma} &\sim N\left(\boldsymbol{\mu}_0,\ \frac{1}{n_0}\boldsymbol{\Sigma} \right)\end{aligned}$$

To obtain the prior distribution for $\boldsymbol{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads $n^* = d^* + 1$ from observations with _TYPE_='N'.

To obtain the prior distribution for $\boldsymbol{\mu}$, PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN', and it reads n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

The resulting posterior distribution, as described in the "Bayesian Estimation of the Mean Vector and Covariance Matrix" section on page 2549, is given by

$$\begin{aligned}
\boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} \quad &\sim \quad W^{-1}\left(\, n+d^*,\ (n-1)\mathbf{S}+d^*\mathbf{S}^*+\mathbf{S}_m\right) \\
\boldsymbol{\mu}^{(t+1)} \mid \left(\boldsymbol{\Sigma}^{(t+1)},\mathbf{Y}\right) \quad &\sim \quad N\left(\frac{1}{n+n_0}\,(n\overline{\mathbf{y}}+n_0\boldsymbol{\mu}_0),\ \frac{1}{n+n_0}\boldsymbol{\Sigma}^{(t+1)}\right)
\end{aligned}$$

where

$$\mathbf{S}_m = \frac{nn_0}{n+n_0}(\overline{\mathbf{y}}-\boldsymbol{\mu}_0)(\overline{\mathbf{y}}-\boldsymbol{\mu}_0)'$$

3. An Informative Prior for Σ

When the sample covariance matrix $\mathbf{S}$ is singular or near singular, prior information about $\boldsymbol{\Sigma}$ can also be used without prior information about $\boldsymbol{\mu}$ to stabilize the inference about $\boldsymbol{\mu}$. You can provide it with a SAS data set that you specify with the PRIOR=INPUT= option.

To obtain the prior distribution for $\boldsymbol{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads n^* from observations with _TYPE_='N'.

The resulting posterior distribution for $(\boldsymbol{\mu},\boldsymbol{\Sigma})$ (Schafer 1997, p. 156) is

$$\begin{aligned}
\boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} \quad &\sim \quad W^{-1}\left(\, n+d^*,\ (n-1)\mathbf{S}+d^*\mathbf{S}^*\right) \\
\boldsymbol{\mu}^{(t+1)} \mid \left(\boldsymbol{\Sigma}^{(t+1)},\mathbf{Y}\right) \quad &\sim \quad N\left(\overline{\mathbf{y}},\ \frac{1}{n}\,\boldsymbol{\Sigma}^{(t+1)}\right)
\end{aligned}$$

Note that if the PRIOR=INPUT= data set also contains observations with _TYPE_='MEAN', then a complete informative prior for both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ will be used.

4. A Ridge Prior

A special case of the preceding adjustment is a ridge prior with $\mathbf{S}^*$ = Diag $(\mathbf{S})$ (Schafer 1997, p. 156). That is, $\mathbf{S}^*$ is a diagonal matrix with diagonal elements equal to the corresponding elements in $\mathbf{S}$.

You can request a ridge prior by using the PRIOR=RIDGE= option. You can explicitly specify the number $d^* \geq 1$ in the PRIOR=RIDGE=d^* option. Or you can implicitly specify the number by specifying the proportion p in the PRIOR=RIDGE=p option to request $d^* = (n-1)p$.

The posterior is then given by

$$\begin{aligned}\mathbf{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n+d^{*},\ (n-1)\mathbf{S}+d^{*}\mathrm{Diag}(\mathbf{S})\right)\\ \boldsymbol{\mu}^{(t+1)}\left|\left(\mathbf{\Sigma}^{(t+1)},\mathbf{Y}\right)\right. &\sim N\left(\overline{\mathbf{y}},\ \frac{1}{n}\,\mathbf{\Sigma}^{(t+1)}\right)\end{aligned}$$

Producing Monotone Missingness with the MCMC Method

The monotone data MCMC method was first proposed by Li (1988), and Liu (1993) described the algorithm. The method is useful especially when a data set is close to having a monotone missing pattern. In this case, the method only needs to impute a few missing values to the data set to have a monotone missing pattern in the imputed data set. Compared to a full data imputation that imputes all missing values, the monotone data MCMC method imputes fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227).

You can request the monotone MCMC method by specifying the option IMPUTE=MONOTONE in the MCMC statement. The “Missing Data Patterns” table now denotes the variables with missing values by “.” or “O”. The value “.” means that the variable is missing and will be imputed and the value “O” means that the variable is missing and will not be imputed. The tables of “Multiple Imputation Variance Information” and “Multiple Imputation Parameter Estimates” are not created.

You must specify the variables in the VAR statement. The variable order in the list determines the monotone missing pattern in the imputed data set. With a different order in the VAR list, the results will be different because the monotone missing pattern to be constructed will be different.

Assuming that the data are from a multivariate normal distribution, then similar to the MCMC method, the monotone MCMC method repeats the following steps:

1. **The imputation I-step:**
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. Only a subset of missing values are simulated to achieve a monotone pattern of missingness.

2. **The posterior P-step:**
Given a new sample with a monotone pattern of missingness, the P-step simulates the posterior population mean vector and covariance matrix with a noninformative Jeffreys prior. These new estimates are then used in the next I-step.

Imputation Step

The I-step is almost identical to the I-step described in the “MCMC Method for Arbitrary Missing Data” section on page 2547 except that only a subset of missing values need to be simulated. To state this precisely, denote the variables with observed values for observation *i* by $Y_{i(obs)}$ and the variables with missing values by $Y_{i(mis)} = (Y_{i(m1)}, Y_{i(m2)})$, where $Y_{i(m1)}$ is a subset of the the missing variables that will result a monotone missingness when their values are imputed. Then the I-step draws values for $Y_{i(m1)}$ from a conditional distribution for $Y_{i(m1)}$ given $Y_{i(obs)}$.

Posterior Step

The P-step is different from the P-step described in the "MCMC Method for Arbitrary Missing Data" section on page 2547. Instead of simulating the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters from the full imputed data set, this P-step simulates the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters through simulated regression coefficients from regression models based on the imputed data set with a monotone pattern of missingness. The step is similar to the process described in the "Regression Method for Monotone Missing Data" section on page 2541.

That is, for the variable Y_j, a model

$$Y_j = \beta_0 + \beta_1 \, Y_1 + \beta_2 \, Y_2 + \ldots + \beta_{j-1} \, Y_{j-1}$$

is fitted using n_j nonmissing observations for variable Y_j in the imputed data sets.

The fitted model consists of the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix from the intercept and variables $Y_1, Y_2, ..., Y_{j-1}$.

For each imputation, new parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(j-1)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

$$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - j)/g$$

where g is a $\chi^2_{n_j - p + j - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

$$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition
$\mathbf{V}_j = \mathbf{V}'_{hj} \mathbf{V}_{hj}$ and $\mathbf{Z}$ is a vector of j independent random normal variates.

These simulated values of $\boldsymbol{\beta}_*$ and σ_{*j}^2 are then used to re-create the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For a detailed description of how to produce monotone-missingness with the MCMC method for a multivariate normal data, refer to Schafer (1997, pp. 226–235).

MCMC Method Specifications

With MCMC, you can impute either all missing values (IMPUTE=FULL) or just enough missing values to make the imputed data set have a monotone missing pattern (IMPUTE=MONOTONE). In the process, either a single chain for all imputations (CHAIN=SINGLE) or a separate chain for each imputation (CHAIN=MULTIPLE) is used. The single chain may be somewhat more precise for estimating a single quantity such as posterior mean (Schafer 1997, p. 138). Refer to Schafer (1997, pp. 137–138) for a discussion of single versus multiple chains.

You can specify the number of initial burn-in iterations before the first imputation with the NBITER= option. This number is also used for subsequent chains for multiple chains. For a single chain, you can also specify the number of iterations between imputations with the NITER= option.

You can explicitly specify initial parameter values for the MCMC process with the INITIAL=INPUT= data set option. Alternatively, you can use the EM algorithm to derive a set of initial parameter values for MCMC with the option INITIAL=EM. These estimates are used as either the starting value (START=VALUE) or as the starting distribution (START=DIST) for the MCMC process. For multiple chains, these estimates are used again as either the starting value (START=VALUE) or as the starting distribution (START=DIST) for the subsequent chains.

You can specify the prior parameter information in the PRIOR= option. You can use a noninformative prior (PRIOR=JEFFREYS), a ridge prior (PRIOR=RIDGE), or an informative prior specified in a data set (PRIOR=INPUT).

The parameter estimates used to generate imputed values in each imputation can be saved in a data set with the OUTEST= option. Later, this data set can be read with the INEST= option to provide the reference distribution for imputing missing values for a new data set.

By default, the MCMC method uses a single chain to produce five imputations. It completes 200 burn-in iterations before the first imputation and 100 iterations between imputations. The posterior mode computed from the EM algorithm with a noninformative prior is used as the starting values for the MCMC process.

INITIAL=EM Specifications

The EM algorithm is used to find the maximum likelihood estimates for incomplete data in the EM statement. You can also use the EM algorithm to find a posterior mode, the parameter estimates that maximize the observed-data posterior density. The resulting posterior mode provides a good starting value for the MCMC process.

With INITIAL=EM, PROC MI uses the MLE of the parameter vector as the initial estimates in the EM algorithm for the posterior mode. You can use the ITPRINT option in INITIAL=EM to display the iteration history for the EM algorithm.

You can use the CONVERGE= option to specify the convergence criterion in deriving the EM posterior mode. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. By default, CONVERGE=1E-4.

You can also use the MAXITER= option to specify the maximum number of iterations in the EM algorithm. By default, MAXITER=200.

With the BOOTSTRAP option, you can use overdispersed starting values for the MCMC process. In this case, PROC MI applies the EM algorithm to a bootstrap sample, a simple random sample with replacement from the input data set, to derive the initial estimates for each chain (Schafer 1997, p. 128).

Checking Convergence in MCMC

The theoretical convergence of the MCMC process has been explored under various conditions, as described in Schafer (1997, p. 70). However, in practice, verification of convergence is not a simple matter.

The parameters used in the imputation step for each iteration can be saved in an output data set with the OUTITER= option. These include the means, standard deviations, covariances, the worst linear function, and observed-data LR statistics. You can then monitor the convergence in a single chain by displaying time-series plots and autocorrelations for those parameter values (Schafer 1997, p. 120). The time-series and autocorrelation function plots for parameters such as variable means, covariances, and the worst linear function can be displayed by specifying the TIMEPLOT and ACFPLOT option.

You can apply EM to a bootstrap sample to obtain overdispersed starting values for multiple chains (Gelman and Rubin 1992). This provides a conservative estimate of the number of iterations needed before each imputation.

The next four subsections describe useful statistics and plots that can be used to check the convergence of the MCMC process.

LR Statistics

You can save the observed-data likelihood ratio (LR) statistic in each iteration with the LR option in the OUTITER= data set. The statistic is based on the observed-data likelihood with parameter values used in the iteration and the observed-data maximum likelihood derived from the EM algorithm.

In each iteration, the LR statistic is given by

$$-2\log\left(\frac{f(\hat{\boldsymbol{\theta}}_i)}{f(\hat{\boldsymbol{\theta}})}\right)$$

where $f(\hat{\boldsymbol{\theta}})$ is the observed-data maximum likelihood derived from the EM algorithm and $f(\hat{\boldsymbol{\theta}}_i)$ is the observed-data likelihood for $\hat{\boldsymbol{\theta}}_i$ used in the iteration.

Similarly, you can also save the observed-data LR posterior mode statistic for each iteration with the LR_POST option. This statistic is based on the observed-data posterior density with parameter values used in each iteration and the observed-data posterior mode derived from the EM algorithm for posterior mode.

For large samples, these LR statistics tends to be approximately χ^2 distributed with degrees of freedom equal to the dimension of $\boldsymbol{\theta}$ (Schafer 1997, p. 131). For example, with a large number of iterations, if the values of the LR statistic do not behave like a random sample from the described χ^2 distribution, then there is evidence that the MCMC process has not converged.

Worst Linear Function of Parameters

The worst linear function (WLF) of parameters (Schafer 1997, pp. 129-131) is a scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that is "worst" in the sense that its function values converge most slowly among parameters in the MCMC process. The convergence of this function is evidence that other parameters are likely to converge as well.

For linear functions of parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a worst linear function of $\boldsymbol{\theta}$ has the highest asymptotic rate of missing information. The function can be derived from the iterative values of $\boldsymbol{\theta}$ near the posterior mode in the EM algorithm. That is, an estimated worst linear function of $\boldsymbol{\theta}$ is

$$w(\boldsymbol{\theta}) = \mathbf{v}' \, (\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

where $\hat{\boldsymbol{\theta}}$ is the posterior mode and the coefficients $\mathbf{v} = \hat{\boldsymbol{\theta}}_{(-1)} - \hat{\boldsymbol{\theta}}$ is the difference between the estimated value of $\boldsymbol{\theta}$ one step prior to convergence and the converged value $\hat{\boldsymbol{\theta}}$.

You can display the coefficients of the worst linear function, $\mathbf{v}$, by specifying the WLF option in the MCMC statement. You can save the function value from each iteration in an OUTITER= data set by specifying the WLF option in the OUTITER option. You can also display the worst linear function values from iterations in an autocorrelation plot or a time-series plot by specifying WLF as an ACFPLOT or TIMEPLOT option, respectively.

Note that when the observed-data posterior is nearly normal, the WLF is one of the slowest functions to approach stationarity. When the posterior is not close to normal, other functions may take much longer than the WLF to converge, as described in Schafer (1997, p.130).

Time-Series Plot

A time-series plot for a parameter ξ is a scatter plot of successive parameter estimates ξ_i against the iteration number i. The plot provides a simple way to examine the convergence behavior of the estimation algorithm for ξ. Long-term trends in the plot indicate that successive iterations are highly correlated and that the series of iterations has not converged.

You can display time-series plots for the worst linear function, the variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for positive parameters in the plots with the LOG option. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot.

By default, the MI procedure uses solid line segments to connect data points in a time-series plot. You can use the CCONNECT=, LCONNECT=, and WCONNECT= options to change the color, line type, and width of the line segments. When WCONNECT=0 is specified, the data points are not connected, and the procedure uses the plus sign (+) as the plot symbol to display the points with a height of one (percentage screen unit) in a time-series plot You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol.

By default, the plot title "Time-Series Plot" is displayed in a time-series plot. You can request another title by using the TITLE= option in the TIMEPLOT option. When another title is also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. Refer to the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference* for an illustration of title options. See Example 44.8 for a usage of the time-series plot.

Autocorrelation Function Plot

To examine relationships of successive parameter estimates ξ, the autocorrelation function (ACF) can be used. For a stationary series, $\xi_i, i \geq 1$, in time series data, the autocorrelation function at lag k is

$$\rho_k = \frac{\mathrm{Cov}(\xi_i, \xi_{i+k})}{\mathrm{Var}(\xi_i)}$$

The sample k^{th} order autocorrelation is computed as

$$r_k = \frac{\sum_{i=1}^{n-k}(\xi_i - \overline{\xi})(\xi_{i+k} - \overline{\xi})}{\sum_{i=1}^{n}(\xi_i - \overline{\xi})^2}$$

You can display autocorrelation function plots for the worst linear function, the variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for parameters in the plots with the LOG option. When a parameter has values less than or equal to zero, the corresponding plot is not created.

You specify the maximum number of lags of the series with the NLAG= option. The autocorrelations at each lag less than or equal to the specified lag are displayed in the graph. In addition, the plot also displays approximate 95% confidence limits for the autocorrelations. At lag k, the confidence limits indicate a set of approximate 95% critical values for testing the hypothesis $\rho_j = 0, j \geq k$.

By default, the MI procedure uses the star sign (*) as the plot symbol to display the points with a height of one (percentage screen unit) in the plot, a solid line to display the reference line of zero autocorrelation, vertical line segments to connect autocorrelations to the reference line, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations.

You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol, and the CNEEDLES= and WNEEDLES= options to change the color and width of the needles. You can also use the LREF=, CREF=, and WREF= options to change the line type, color, and width of the reference line. Similarly, you can use the LCONF=, CCONF=, and WCONF= options to change the line type, color, and width of the confidence limits.

By default, the plot title “Autocorrelation Plot” is displayed in a autocorrelation function plot. You can request another title by using the TITLE= option in ACFPLOT. When another title is also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. Refer to the chapter “The SAS/GRAPH Statements” in *SAS/GRAPH Software: Reference* for a description of title options. See Example 44.8 for an illustration of the autocorrelation function plot.

Input Data Sets

You can specify the input data set with missing values with the DATA= option in the PROC MI statement. When an MCMC method is used, you can specify the data set containing the reference distribution information for imputation with the INEST= option, the data set containing initial parameter estimates for the MCMC process with the INITIAL=INPUT= option, and the data set containing information for the prior distribution with the PRIOR=INPUT= option in the MCMC statement.

DATA=*SAS-data-set*

The input DATA= data set is an ordinary SAS data set containing multivariate data with missing values.

INEST=*SAS-data-set*

The input INEST= data set is a TYPE=EST data set and contains a variable _Imputation_ to identify the imputation number. For each imputation, PROC MI reads the point estimate from the observations with _TYPE_=‘PARM’ or _TYPE_=‘PARMS’ and the associated covariances from the observations with _TYPE_=‘COV’ or _TYPE_=‘COVB’. These estimates are used as the reference distribution to impute values for observations in the DATA= data set. When the input INEST= data set also contains observations with _TYPE_=‘SEED’, PROC MI reads the seed information for the random number generator from these observations. Otherwise, the SEED= option provides the seed information.

INITIAL=INPUT=*SAS-data-set*

The input INITIAL=INPUT= data set is a TYPE=COV or CORR data set and provides initial parameter estimates for the MCMC process. The covariances derived from the TYPE=COV/CORR data set are divided by the number of observations to get the correct covariance matrix for the point estimate (sample mean).

If TYPE=COV, PROC MI reads the number of observations from the observations with _TYPE_=‘N’, the point estimate from the observations with _TYPE_=‘MEAN’, and the covariances from the observations with _TYPE_=‘COV’.

If TYPE=CORR, PROC MI reads the number of observations from the observations with _TYPE_=‘N’, the point estimate from the observations with

TYPE='MEAN', the correlations from the observations with _TYPE_='CORR', and the standard deviations from the observations with _TYPE_='STD'.

PRIOR=INPUT=*SAS-data-set*

The input PRIOR=INPUT= data set is a TYPE=COV data set that provides information for the prior distribution. You can use the data set to specify a prior distribution for $\mathbf{\Sigma}$ of the form

$$\mathbf{\Sigma} \sim W^{-1}\left(d^{*},\, d^{*}\mathbf{S}^{*} \right)$$

where $d^{*} = n^{*} - 1$ is the degrees of freedom. PROC MI reads the matrix $\mathbf{S}^{*}$ from observations with _TYPE_='COV' and n^{*} from observations with _TYPE_='N'.

You can also use this data set to specify a prior distribution for $\boldsymbol{\mu}$ of the form

$$\boldsymbol{\mu} \sim N\left(\boldsymbol{\mu}_0,\, \frac{1}{n_0}\mathbf{\Sigma} \right)$$

PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN' and n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

Output Data Sets

You can specify the output data set of imputed values with the OUT= option in the PROC MI statement. When an EM statement is used, you can specify the data set containing the original data set with missing values being replaced by the expected values from the EM algorithm with the OUT= option in the EM statement. You can also specify the data set containing MLE computed with the EM algorithm with the OUTEM= option.

When an MCMC method is used, you can specify the data set containing parameter estimates used in each imputation with the OUTEST= option and the data set containing parameters used in the imputation step for each iteration with the OUTITER option in the MCMC statement.

OUT=*SAS-data-set* **in the PROC MI statement**

The OUT= data set contains all the variables in the original data set and a new variable named _Imputation_ that identifies the imputation. For each imputation, the data set contains all variables in the input DATA= data set with missing values being replaced by imputed values. Note that when the NIMPUTE=1 option is specified, the variable _Imputation_ is not created.

OUT=*SAS-data-set* **in an EM statement**

The OUT= data set contains the original data set with missing values being replaced by expected values from the EM algorithm.

OUTEM=*SAS-data-set*

The OUTEM= data set is a TYPE=COV data set and contains the MLE computed with the EM algorithm. The observations with _TYPE_='MEAN' contain the estimated mean and the observations with _TYPE_='COV' contain the estimated covariances.

OUTEST=*SAS-data-set*

The OUTEST= data set is a TYPE=EST data set and contains parameter estimates used in each imputation in the MCMC method. It also includes an index variable named _Imputation_, which identifies the imputation.

The observations with _TYPE_='SEED' contain the seed information for the random number generator. The observations with _TYPE_='PARM' or _TYPE_='PARMS' contain the point estimate and the observations with _TYPE_='COV' or _TYPE_='COVB' contain the associated covariances. These estimates are used as the parameters of the reference distribution to impute values for observations in the DATA= dataset.

Note that these estimates are the values used in the I-step before each imputation. These are not the parameter values simulated from the P-step in the same iteration. See Example 44.9 for a usage of this option.

OUTITER < **(** *options* **)** > **=***SAS-data-set* **in an EM statement**

The OUTITER= data set in an EM statement is a TYPE=COV data set and contains parameters for each iteration. It also includes a variable _Iteration_ that provides the iteration number.

The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options for OUTITER. With the MEAN option, the output data set contains the mean parameters in observations with the variable _TYPE_='MEAN'. Similarly, with the MEAN option, the output data set contains the covariance parameters in observations with the variable _TYPE_='COV'. When no options are specified, the output data set contains the mean parameters for each iteration.

OUTITER < **(** *options* **)** > **=***SAS-data-set* **in an MCMC statement**

The OUTITER= data set in an MCMC statement is a TYPE=COV data set and contains parameters used in the imputation step for each iteration. It also includes variables named _Imputation_ and _Iteration_, which provide the imputation number and iteration number.

The parameters in the output data set depend on the options specified. The following table summarizes the options available for OUTITER and the corresponding values for the output variable _TYPE_.

Table 44.4. Summary of Options for OUTITER in an MCMC statement

Options	**Output Parameters**	_TYPE_
MEAN	mean parameters	MEAN
STD	standard deviations	STD
COV	covariances	COV
LR	-2 log LR statistic	LOG_LR

Table 44.4. (continued)

Options	**Output Parameters**	_TYPE_
LR_POST	-2 log LR statistic of the posterior mode	LOG_POST
WLF	worst linear function	WLF

When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. For a detailed description of the worst linear function and LR statistics, see the "Checking Convergence in MCMC" section on page 2555.

Combining Inferences from Multiply Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose $\hat{Q}_i$ and $\hat{U}_i$ are the point and variance estimates from the ith imputed data set, i=1, 2, ..., m. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m} \sum_{i=1}^{m} \hat{Q}_i$$

Suppose $\overline{U}$ is the within-imputation variance, which is the average of the m complete-data estimates:

$$\overline{U} = \frac{1}{m} \sum_{i=1}^{m} \hat{U}_i$$

and B is the between-imputation variance

$$B = \frac{1}{m-1} \sum_{i=1}^{m} (\hat{Q}_i - \overline{Q})^2$$

Then the variance estimate associated with $\overline{Q}$ is the total variance (Rubin 1987)

$$T = \overline{U} + (1 + \frac{1}{m})B$$

The statistic $(Q - \overline{Q})T^{-(1/2)}$ is approximately distributed as t with v_m degrees of freedom (Rubin 1987), where

$$v_m = (m-1) \left[1 + \frac{\overline{W}}{(1+m^{-1})B} \right]^2$$

The degrees of freedom v_m depends on m and the ratio

$$r = \frac{(1+m^{-1})B}{\overline{U}}$$

The ratio r is called the relative increase in variance due to nonresponse (Rubin 1987). When there is no missing information about Q, the values of r and B are both zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution of $(Q - \overline{Q})T^{-(1/2)}$ will be approximately normal.

Another useful statistic is the fraction of missing information about Q:

$$\hat{\lambda} = \frac{r + 2/(v_m + 3)}{r + 1}$$

Both statistics r and λ are helpful diagnostics for assessing how the missing data contribute to the uncertainty about Q.

When the complete-data degrees of freedom v_0 is small, and there is only a modest proportion of missing data, the computed degrees of freedom, v_m, can be much larger than v_0, which is inappropriate. For example, with $m = 5$ and $r = 10\%$, the computed degrees of freedom $v_m = 484$, which is inappropriate for data sets with complete-data degrees of freedom less than 484.

Barnard and Rubin (1999) recommend the use of an adjusted degrees of freedom

$$v_m^* = \left[\frac{1}{v_m} + \frac{1}{\hat{v}_{obs}} \right]^{-1}$$

where $\hat{v}_{obs} = (1 - \gamma)\, v_0(v_0 + 1)/(v_0 + 3)$ and $\gamma = (1 + m^{-1})B/T$.

Note that the MI procedure uses the adjusted degrees of freedom, v_m^*, for inference.

Multiple Imputation Efficiency

The relative efficiency (RE) of using the finite m imputation estimator, rather than using an infinite number for the fully efficient imputation, in units of variance, is approximately a function of m and λ (Rubin 1987, p. 114).

$$RE = \left(1 + \frac{\lambda}{m} \right)^{-1}$$

The following table shows relative efficiencies with different values of m and λ.

Table 44.5. Relative Efficiency

	λ				
m	10%	20%	30%	50%	70%
3	0.9677	0.9375	0.9091	0.8571	0.8108
5	0.9804	0.9615	0.9434	0.9091	0.8772
10	0.9901	0.9804	0.9709	0.9524	0.9346
20	0.9950	0.9901	0.9852	0.9756	0.9662

The table shows that for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of m imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Imputer's Model Versus Analyst's Model

Multiple imputation inference assumes that the model you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models may not be the same (Schafer 1997, p. 139).

Schafer (1997, pp. 139–143) provides comprehensive coverage of this topic, and the following example is based on his work.

Consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 with missing values. An imputer creates multiple imputations with the model $Y_3 = Y_1\ Y_2$. However, the analyst can later use the simpler model $Y_3 = Y_1$. In this case, the analyst assumes more than the imputer. That is, the analyst assumes there is no relationship between variables Y_3 and Y_2.

The effect of the discrepancy between the models depends on whether the analyst's additional assumption is true. If the assumption is true, the imputer's model still applies. The inferences derived from multiple imputations will still be valid, although they may be somewhat conservative because they reflect the additional uncertainty of estimating the relationship between Y_3 and Y_2.

On the other hand, suppose that the analyst models $Y_3 = Y_1$, and there is a relationship between variables Y_3 and Y_2. Then the model $Y_3 = Y_1$ will be biased and is inappropriate. Appropriate results can be generated only from appropriate analyst models.

Another type of discrepancy occurs when the imputer assumes more than the analyst. For example, suppose that an imputer creates multiple imputations with the model $Y_3 = Y_1$, but the analyst later fits a model $Y_3 = Y_1\ Y_2$. When the assumption is true, the imputer's model is a correct model and the inferences still hold.

On the other hand, suppose there is a relationship between Y_3 and Y_2. Imputations created under the incorrect assumption that there is no relationship between Y_3 and Y_2 will make the analyst's estimate of the relationship biased toward zero. Multiple imputations created under an incorrect model can lead to incorrect conclusions.

Thus, generally you should include as many variables as you can when doing multiple imputation. The precision you lose with included unimportant predictors is usually a relatively small price to pay for the general validity of analyses of the resultant multiply imputed data set (Rubin 1996). But at the same time, you need to keep the model building and fitting feasible (Barnard and Meng, 1999, pp. 19–20).

To produce high-quality imputations for a particular variable, the imputation model should also include variables that are potentially related to the imputed variable and variables that are potentially related to the missingness of the imputed variable (Schafer 1997, p. 143).

Similar suggestions were also given by van Buuren, Boshuizen, and Knook (1999, p. 687). They recommended the imputation model includes three sets of covariates: variables in the analyst's model, variables associated with the missingness of the imputed variable, and variables correlated with the imputed variable. They also recommended the removal of the covariates not in the analyst's model if they have too many missing values for observations with missing imputed variable.

Note that it is good practice to include a description of the imputer's model with the multiply imputed data set (Rubin 1996, p.479). That way, the analysts will have information about the variables involved in the imputation and which relationships among the variables have been implicitly set to zero.

Parameter Simulation Versus Multiple Imputation

As an alternative to multiple imputation, parameter simulation can also be used to analyze the data for many incomplete-data problems. Although the MI procedure does not offer parameter simulation, the trade-offs between the two methods (Schafer 1997, pp. 89–90, 135–136) are examined in this section.

The parameter simulation method simulates random values of parameters from the observed-data posterior distribution and makes simple inferences about these parameters (Schafer 1997, p. 89). When a set of well-defined population parameters $\boldsymbol{\theta}$ are of interest, parameter simulation can be used to directly examine and summarize simulated values of $\boldsymbol{\theta}$. This usually requires a large number of iterations, and involves calculating appropriate summaries of the resulting dependent sample of the iterates of the $\boldsymbol{\theta}$. If only a small set of parameters are involved, parameter simulation is suitable (Schafer 1997).

Multiple imputation only requires a small number of imputations. Generating and storing a few imputations can be more efficient than generating and storing a large number of iterations for parameter simulation.

When fractions of missing information are low, methods that average over simulated values of the missing data, as in multiple imputation, can be much more efficient than methods that average over simulated values of $\boldsymbol{\theta}$ as in parameter simulation (Schafer 1997).

Summary of Issues in Multiple Imputation

This section summarizes issues which are encountered in applications of the MI procedure.

The MAR Assumption

The missing at random (MAR) assumption is needed for the imputation methods in the MI Procedure. Although this assumption cannot be verified with the data, it becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; van Buuren, Boshuizen, and Knook, 1999, p. 687).

Number of Imputations

Based on the theory of multiple imputation, only a small number of imputations are needed for a data set with little missing information (Rubin 1987, p. 114). The number of imputations can be informally verified by replicating sets of m imputations and checking whether the estimates are stable (Horton and Lipsitz 2001, p. 246).

Imputation Model

Generally you should include as many variables as you can in the imputation model (Rubin 1996), At the same time, however, it is important to keep the number of variables in control, as discussed by Barnard and Meng (1999, pp. 19–20). For the imputation of a particular variable, the model should include variables in the complete data model, variables that are correlated with the imputed variable, and variables that are associated with the missingness of the imputed variable (Schafer 1997, p. 143; van Buuren, Boshuizen, and Knook 1999, p. 687).

Multivariate Normality Assumption

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from the multivariate normality if the amount of missing information is not large (Schafer 1997, pp. 147–148).

You can use variable transformations to make the normality assumption more tenable. Variables are transformed before the imputation process and then back-transformed to create imputed values.

Monotone Regression Method

With the multivariate normality assumption, either the regression method or the predictive mean matching method can be used to impute continuous variables in data sets with monotone missing patterns.

The predictive mean matching method ensures that imputed values are plausible and may be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Monotone Propensity Score Method

The propensity score method can also be used to impute continuous variables in data sets with monotone missing patterns.

The propensity score method does not use correlations among variables and is not appropriate for analyses involving relationship among variables, such as a regression analysis (Schafer 1999, p.11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

MCMC Monotone-Data Imputation

The MCMC Method is used to impute continuous variables in data sets with arbitrary missing patterns, assuming a multivariate normal distribution for the data. It can also be used to impute just enough missing values to make the imputed data sets have a monotone missing pattern. Then, a more flexible monotone imputation method can be used for the remaining missing values.

Checking Convergence in MCMC

In an MCMC process, parameters are drawn after the MCMC is run long enough to converge to its stationary distribution. In practice, however, it is not simple to verify the convergence of the process, especially for a large number of parameters.

You can check for convergence by examining the observed-data likelihood ratio statistic and worst linear function of the parameters in each iteration. You can also check for convergence by examining a plot of autocorrelation function, as well as a time-series plot of parameters (Schafer 1997, p. 120).

EM Estimates

The EM algorithm can be used to compute the MLE of the mean vector and covariance matrix of the data with missing values, assuming a multivariate normal distribution for the data. However, the covariance matrix associated with the estimate of the mean vector cannot be derived from the EM algorithm.

In the MI procedure, you can use the EM algorithm to compute the posterior mode, which provides a good starting value for the MCMC process (Schafer 1997, p. 169).

ODS Table Names

PROC MI assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 44.6. ODS Tables Produced in PROC MI

ODS Table Name	Description	Statement	Option
Corr	Pairwise correlations		SIMPLE
EMEstimates	EM (MLE) estimates	EM	
EMInitEstimates	EM initial estimates	EM	
EMIterHistory	EM (MLE) iteration history	EM	ITPRINT
EMPostEstimates	EM (Posterior mode) estimates	MCMC	INITIAL=EM
EMPostIterHistory	EM (Posterior mode) iteration history	MCMC	INITIAL=EM (ITPRINT)
EMWLF	Worst linear function	MCMC	WLF
MCMCInitEstimates	MCMC initial estimates	MCMC	DISPLAYINIT
MissPattern	Missing data patterns		
ModelInfo	Model information		
MonoDiscrim	Discriminant model group means	MONOTONE	DISCRIM (/DETAILS)
MonoLogistic	Logistic model	MONOTONE	LOGISTIC (/DETAILS)
MonoModel	Multiple monotone models	MONOTONE	
MonoPropensity	Propensity score model logistic function	MONOTONE	PROPENSITY (/DETAILS)

Table 44.6. (continued)

ODS Table Name	Description	Statement	Option
MonoReg	Regression model	MONOTONE	REG (/DETAILS)
MonoRegPMM	Predicted mean matching model	MONOTONE	REGPMM (/DETAILS)
ParameterEstimates	Parameter estimates		
Transform	Variable transformations	TRANSFORM	
Univariate	Univariate statistics		SIMPLE
VarianceInfo	Between, within, and total variances		

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the MI procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the following options in the MCMC statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

ACFPLOT < *(options < / display-options >)* >
displays plots of the autocorrelation function of parameters from iterations.

For a detailed description of the ACFPLOT option, see the “Autocorrelation Function Plot” section on page 2557. Note that for the display-options, only the LOG, NLAG=, and TITLE= options are applicable.

TIMEPLOT < *(options < / display-options >)* >
displays time-series plots of parameters from iterations.

For a detailed description of the TIMEPLOT option, see the “Time-Series Plot” section on page 2556. Note that for the display-options, only the LOG, WCONNECT=, and TITLE= options are applicable. If you specify the WCONNECT=0 option, a scatter plot is created. Otherwise, a line plot is created.

ODS Graph Names

PROC MI assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 44.7.

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 44.7. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 44.7. ODS Graphics Produced by PROC MI

ODS Graph Name	Plot Description	Statement	Option
ACFPlot	ACF plot	MCMC	ACFPLOT
TimeScatterPlot	Time-series scatter plot	MCMC	TIMEPLOT(WCONNECT=0)
TimeSeriesPlot	Time-series plot	MCMC	TIMEPLOT

Examples

The Fish data described in the STEPDISC procedure are measurements of 159 fish of seven species caught in Finland’s lake Laengelmavesi. For each fish, the length, height, and width are measured. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail (Length1), from the nose to the notch of its tail (Length2), and from the nose to the end of its tail (Length3). See Chapter 67, “The STEPDISC Procedure,” for more information.

The Fish1 data set is constructed from the Fish data set and contains only one species of the fish and the three length measurements. Some values have been set to missing and the resulting data set has a monotone missing pattern in variables Length1 Length2, and Length3. The Fish1 data set is used in Example 44.2 with the propensity score method and in Example 44.3 with the regression method.

The Fish2 data set is also constructed from the Fish data set and contains two species of fish. Some values have been set to missing and the resulting data set has a monotone missing pattern in variables Length3, Height, Width, and Species. The Fish2 data set is used in Example 44.4 with the logistic regression method and in Example 44.5 with the discriminant function method. Note that some values of the variable Species have also been altered in the data set.

The FitMiss data set created in the “Getting Started” section is used in other examples. The following statements create the Fish1 data set.

```
/*----------- Fishes of Species Bream ----------*/
 data Fish1;
    title ’Fish Measurement Data’;
    input Length1 Length2 Length3 @@;
    datalines;
 23.2 25.4 30.0    24.0 26.3 31.2    23.9 26.5 31.1
 26.3 29.0 33.5    26.5 29.0   .     26.8 29.7 34.7
 26.8   .    .     27.6 30.0 35.0    27.6 30.0 35.1
 28.5 30.7 36.2    28.4 31.0 36.2    28.7   .    .
 29.1 31.5   .     29.5 32.0 37.3    29.4 32.0 37.2
 29.4 32.0 37.2    30.4 33.0 38.3    30.4 33.0 38.5
 30.9 33.5 38.6    31.0 33.5 38.7    31.3 34.0 39.5
 31.4 34.0 39.2    31.5 34.5   .     31.8 35.0 40.6
 31.9 35.0 40.5    31.8 35.0 40.9    32.0 35.0 40.6
 32.7 36.0 41.5    32.8 36.0 41.6    33.5 37.0 42.6
 35.0 38.5 44.1    35.0 38.5 44.0    36.2 39.5 45.3
 37.4 41.0 45.9    38.0 41.0 46.5
 ;
```

The Fish2 data set contains two of the seven species in the Fish data set. For each of the two species (Bream and Parkki), the length from the nose of the fish to the end of its tail, the height, and the width of each fish are measured. The height and width are recorded as percentages of the length variable.

The following statements create the Fish2 data set.

```
/*-------- Fishes of Species Bream and Parkki Pike --------*/
 data Fish2 (drop=HtPct WidthPct);
 title ’Fish Measurement Data’;
 input Species $ Length3 HtPct WidthPct @@;
 Height= HtPct*Length3/100;
 Width= WidthPct*Length3/100;
 datalines;
 Gp1  30.0 38.4 13.4   Gp1  31.2 40.0 13.8   Gp1  31.1 39.8 15.1
   .  33.5 38.0   .      .  34.0 36.6 15.1   Gp1  34.7 39.2 14.2
 Gp1  34.5 41.1 15.3   Gp1  35.0 36.2 13.4   Gp1  35.1 39.9 13.8
   .  36.2 39.3 13.7   Gp1  36.2 39.4 14.1     .  36.2 39.7 13.3
 Gp1  36.4 37.8 12.0     .  37.3 37.3 13.6   Gp1  37.2 40.2 13.9
 Gp1  37.2 41.5 15.0   Gp1  38.3 38.8 13.8   Gp1  38.5 38.8 13.5
 Gp1  38.6 40.5 13.3   Gp1  38.7 37.4 14.8   Gp1  39.5 38.3 14.1
 Gp1  39.2 40.8 13.7     .  39.7 39.1   .    Gp1  40.6 38.1 15.1
 Gp1  40.5 40.1 13.8   Gp1  40.9 40.0 14.8   Gp1  40.6 40.3 15.0
 Gp1  41.5 39.8 14.1   Gp2  41.6 40.6 14.9   Gp1  42.6 44.5 15.5
 Gp1  44.1 40.9 14.3   Gp1  44.0 41.1 14.3   Gp1  45.3 41.4 14.9
 Gp1  45.9 40.6 14.7   Gp1  46.5 37.9 13.7
 Gp2  16.2 25.6 14.0   Gp2  20.3 26.1 13.9   Gp2  21.2 26.3 13.7
 Gp2  22.2 25.3 14.3   Gp2  22.2 28.0 16.1   Gp2  22.8 28.4 14.7
 Gp2  23.1 26.7 14.7     .  23.7 25.8 13.9   Gp2  24.7 23.5 15.2
 Gp1  24.3 27.3 14.6   Gp2  25.3 27.8 15.1   Gp2  25.0 26.2 13.3
 Gp2  25.0 25.6 15.2   Gp2  27.2 27.7 14.1   Gp2  26.7 25.9 13.6
   .  26.8 27.6 15.4   Gp2  27.9 25.4 14.0   Gp2  29.2 30.4 15.4
 Gp2  30.6 28.0 15.6   Gp2  35.0 27.1 15.3
 ;
```

Example 44.1. EM Algorithm for MLE

This example uses the EM algorithm to compute the maximum likelihood estimates for parameters of a multivariate normal distribution using data with missing values. The following statements invoke the MI procedure and request the EM algorithm to compute the MLE for (μ, Σ) of a multivariate normal distribution from the input data set FitMiss.

```
proc mi data=FitMiss seed=1518971 simple nimpute=0;
   em itprint outem=outem;
   var Oxygen RunTime RunPulse;
run;
```

Note that when you specify the NIMPUTE=0 option, the missing values will not be imputed. The procedure generates the following output:

Output 44.1.1. Model Information

```
                              Fish Measurement Data

                                 The MI Procedure

                                Model Information

           Data Set                             WORK.FITMISS
           Method                               MCMC
           Multiple Imputation Chain            Single Chain
           Initial Estimates for MCMC           EM Posterior Mode
           Start                                Starting Value
           Prior                                Jeffreys
           Number of Imputations                0
           Number of Burn-in Iterations         200
           Number of Iterations                 100
           Seed for random number generator     1518971
```

The "Model Information" table shown in Output 44.1.1 describes the method and options used in the procedure if a positive number is specified in the NIMPUTE= option.

The "Missing Data Patterns" table shown in Output 44.1.2 lists distinct missing data patterns with corresponding frequencies and percents. Here, a value of "X" means that the variable is observed in the corresponding group and a value of "." means that the variable is missing. The table also displays group-specific variable means.

Output 44.1.2. Missing Data Patterns

```
                                 The MI Procedure

                               Missing Data Patterns

                                   Run      Run
             Group     Oxygen      Time     Pulse        Freq      Percent

                 1     X           X        X              21        67.74
                 2     X           X        .               4        12.90
                 3     X           .        .               3         9.68
                 4     .           X        X               1         3.23
                 5     .           X        .               2         6.45

                              Missing Data Patterns

                     -----------------Group Means----------------
              Group           Oxygen          RunTime        RunPulse

                  1        46.353810        10.809524      171.666667
                  2        47.109500        10.137500               .
                  3        52.461667                .               .
                  4                .        11.950000      176.000000
                  5                .         9.885000               .
```

With the SIMPLE option, the procedure displays simple descriptive univariate statistics for available cases in the "Univariate Statistics" table shown in Output 44.1.3 and correlations from pairwise available cases in the “Pairwise Correlations” table shown in Output 44.1.4.

Output 44.1.3. Univariate Statistics

```
                              The MI Procedure

                           Univariate Statistics

Variable           N          Mean       Std Dev       Minimum       Maximum

Oxygen            28      47.11618       5.41305      37.38800      60.05500
RunTime           28      10.68821       1.37988       8.63000      14.03000
RunPulse          22     171.86364      10.14324     148.00000     186.00000

                           Univariate Statistics

                                  ---Missing Values--
                      Variable        Count    Percent

                      Oxygen              3       9.68
                      RunTime             3       9.68
                      RunPulse            9      29.03
```

Output 44.1.4. Pairwise Correlations

```
                              The MI Procedure

                           Pairwise Correlations

                            Oxygen           RunTime          RunPulse

       Oxygen          1.000000000      -0.849118562      -0.343961742
       RunTime        -0.849118562       1.000000000       0.247258191
       RunPulse       -0.343961742       0.247258191       1.000000000
```

With the EM statement, the procedure displays the initial parameter estimates for EM in the “Initial Parameter Estimates for EM” table shown in Output 44.1.5.

Output 44.1.5. Initial Parameter Estimates for EM

```
                                 The MI Procedure

                        Initial Parameter Estimates for EM

        _TYPE_    _NAME_            Oxygen         RunTime        RunPulse

        MEAN                     47.116179       10.688214      171.863636
        COV       Oxygen         29.301078               0               0
        COV       RunTime                0        1.904067               0
        COV       RunPulse               0               0      102.885281
```

With the ITPRINT option in the EM statement, the “EM (MLE) Iteration History” table shown in Output 44.1.6 displays the iteration history for the EM algorithm.

Output 44.1.6. EM (MLE) Iteration History

```
                                 The MI Procedure

                            EM (MLE) Iteration History

    _Iteration_        -2 Log L         Oxygen        RunTime        RunPulse

              0      289.544782      47.116179      10.688214      171.863636
              1      263.549489      47.116179      10.688214      171.863636
              2      255.851312      47.139089      10.603506      171.538203
              3      254.616428      47.122353      10.571685      171.426790
              4      254.494971      47.111080      10.560585      171.398296
              5      254.483973      47.106523      10.556768      171.389208
              6      254.482920      47.104899      10.555485      171.385257
              7      254.482813      47.104348      10.555062      171.383345
              8      254.482801      47.104165      10.554923      171.382424
              9      254.482800      47.104105      10.554878      171.381992
             10      254.482800      47.104086      10.554864      171.381796
             11      254.482800      47.104079      10.554859      171.381708
             12      254.482800      47.104077      10.554858      171.381669
```

The “EM (MLE) Parameter Estimates” table shown in Output 44.1.7 displays the maximum likelihood estimates for μ and Σ of a multivariate normal distribution from the data set FitMiss.

Output 44.1.7. EM (MLE) Parameter Estimates

```
                                The MI Procedure

                          EM (MLE) Parameter Estimates

        _TYPE_     _NAME_            Oxygen          RunTime         RunPulse

        MEAN                      47.104077       10.554858       171.381669
        COV        Oxygen         27.797931       -6.457975       -18.031298
        COV        RunTime        -6.457975        2.015514         3.516287
        COV        RunPulse      -18.031298        3.516287        97.766857
```

You can also output the EM (MLE) parameter estimates into an output data set with the OUTEM= option. The following statements list the observations in the output data set outem.

```
proc print data=outem;
   title 'EM Estimates';
run;
```

Output 44.1.8. EM Estimates

```
                                  EM Estimates

          Obs     _TYPE_      _NAME_       Oxygen       RunTime     RunPulse

           1       MEAN                   47.1041       10.5549      171.382
           2       COV       Oxygen       27.7979       -6.4580      -18.031
           3       COV       RunTime      -6.4580        2.0155        3.516
           4       COV       RunPulse    -18.0313        3.5163       97.767
```

The output data set outem shown in Output 44.1.8 is a TYPE=COV data set. The observation with _TYPE_=‘MEAN’ contains the MLE for the parameter μ and the observations with _TYPE_=‘COV’ contain the MLE for the parameter Σ of a multivariate normal distribution from the data set FitMiss.

Example 44.2. Propensity Score Method

This example uses the propensity score method to impute missing values for variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the propensity score method. The resulting data set is named outex2.

```
proc mi data=Fish1 seed=899603 out=outex2;
   monotone propensity;
   var Length1 Length2 Length3;
run;
```

Note that the VAR statement is required and the data set must have a monotone missing pattern with variables as ordered in the VAR statement. The procedure generates the following output:

Output 44.2.1. Model Information

```
                                 The MI Procedure

                                Model Information

         Data Set                                   WORK.FISH1
         Method                                     Monotone
         Number of Imputations                      5
         Seed for random number generator           899603
```

The "Model Information" table shown in Output 44.2.1 describes the method and options used in the multiple imputation process. By default, five imputations are created for the missing data.

When monotone methods are used in the imputation, MONOTONE is displayed as the method. The "Monotone Model Specification" table shown in Output 44.2.2 displays the detailed model specification. By default, the observations are sorted into five groups based on their propensity scores.

Output 44.2.2. Monotone Model Specification

```
                                 The MI Procedure

                           Monotone Model Specification

                                                   Imputed
                  Method                           Variables

                  Propensity( Groups= 5)           Length2 Length3
```

Without covariates specified for imputed variables Length2 and Length3, the variable Length1 is used as the covariate for Length2, and variables Length1 and Length2 are used as covariates for Length3.

Output 44.2.3. Missing Data Patterns

```
                              The MI Procedure

                            Missing Data Patterns

          Group    Length1    Length2    Length3         Freq     Percent

              1    X          X          X                 30       85.71
              2    X          X          .                  3        8.57
              3    X          .          .                  2        5.71

                           Missing Data Patterns

                   -----------------Group Means----------------
              Group          Length1         Length2         Length3

                  1        30.603333       33.436667       38.720000
                  2        29.033333       31.666667               .
                  3        27.750000               .               .
```

The “Missing Data Patterns” table shown in Output 44.2.3 lists distinct missing data patterns with corresponding frequencies and percents. Here, values of “X” and “.” indicate that the variable is observed or missing in the corresponding group. The table confirms a monotone missing pattern for these three variables.

For the imputation process, first, missing values of Length2 in Group 3 are imputed using observed values of Length1. Then the missing values of Length3 in Group 2 are imputed using observed values of Length1 and Length2. And finally, the missing values of Length3 in Group 3 are imputed using observed values of Length1 and imputed values of Length2.

After the completion of *m* imputations, the “Multiple Imputation Variance Information” table shown in Output 44.2.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. A detailed description of these statistics is provided in the “Combining Inferences from Multiply Imputed Data Sets” section on page 2561.

Output 44.2.4. Variance Information

```
                               The MI Procedure

                  Multiple Imputation Variance Information

                   -----------------Variance-----------------
       Variable         Between         Within          Total       DF

       Length2         0.001500       0.465422       0.467223   32.034
       Length3         0.049725       0.547434       0.607104   27.103

                   Multiple Imputation Variance Information

                            Relative       Fraction
                            Increase        Missing       Relative
            Variable     in Variance    Information     Efficiency

            Length2         0.003869       0.003861       0.999228
            Length3         0.108999       0.102610       0.979891
```

The “Multiple Imputation Parameter Estimates” table shown in Output 44.2.5 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the *t*-distributions. For each variable, the table also displays a 95% mean confidence interval and a *t*-statistic with the associated *p*-value for the hypothesis that the population mean is equal to the value specified in the MU0= option, which is zero by default.

Output 44.2.5. Parameter Estimates

```
                               The MI Procedure

                   Multiple Imputation Parameter Estimates

  Variable            Mean      Std Error    95% Confidence Limits        DF

  Length2        33.006857       0.683537     31.61460     34.39912   32.034
  Length3        38.361714       0.779169     36.76328     39.96015   27.103

                   Multiple Imputation Parameter Estimates

                                                         t for H0:
Variable         Minimum        Maximum            Mu0    Mean=Mu0   Pr > |t|

Length2        32.957143      33.060000              0       48.29     <.0001
Length3        38.080000      38.545714              0       49.23     <.0001
```

The following statements list the first ten observations of the data set outex2, as shown in Output 44.2.6. The missing values are imputed from observed values with similar propensity scores.

```
proc print data=outex2(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 44.2.6. Imputed Data Set

```
                First 10 Observations of the Imputed Data Set

             Obs    _Imputation_    Length1    Length2    Length3

               1          1           23.2       25.4       30.0
               2          1           24.0       26.3       31.2
               3          1           23.9       26.5       31.1
               4          1           26.3       29.0       33.5
               5          1           26.5       29.0       38.6
               6          1           26.8       29.7       34.7
               7          1           26.8       29.0       35.0
               8          1           27.6       30.0       35.0
               9          1           27.6       30.0       35.1
              10          1           28.5       30.7       36.2
```

Example 44.3. Regression Method

This example uses the regression method to impute missing values for all variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the regression method for variable Length2 and the predictive mean matching method for variable Length3. The resulting data set is named outex3.

```
proc mi data=Fish1 round=.1  mu0= 0 35 45
        seed=13951639 out=outex3;
   monotone reg(Length2/ details)
            regpmm(Length3= Length1 Length2 Length1*Length2/ details);
   var Length1 Length2 Length3;
run;
```

The ROUND= option is used to round the imputed values to the same precision as observed values. The values specified with the ROUND= option are matched with the variables Length1, Length2, and Length3 in the order listed in the VAR statement. The MU0= option requests *t* tests for the hypotheses that the population means corresponding to the variables in the VAR statement are Length2=35 and Length3=45.

Note that an optimal K= value is currently not available for the REGPMM option in the literature on multiple imputation. The default K=5 is experimental and may change in future releases.

The “Missing Data Patterns” table lists distinct missing data patterns with corresponding frequencies and percents. It is identical to the table displayed in Output 44.2.3 in the previous example.

The “Monotone Model Specification” table shown in Output 44.3.1 displays the model specification.

Output 44.3.1. Monotone Model Specification

```
                            Fish Measurement Data

                               The MI Procedure

                        Monotone Model Specification

                                               Imputed
                     Method                    Variables

                     Regression                Length2
                     Regression-PMM( K= 5)     Length3
```

With the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 44.3.2 and Output 44.3.3.

Output 44.3.2. Regression Model

```
                               The MI Procedure

                    Regression Models for Monotone Method

Imputed                             ----------------Imputation----------------
Variable   Effect        Obs-Data              1              2              3

Length2    Intercept     -0.04249      -0.049184      -0.055470      -0.051346
Length2    Length1        0.98587       1.001934       0.995275       0.992294

                    Regression Models for Monotone Method

             Imputed                ---------Imputation---------
             Variable   Effect                 4               5

             Length2    Intercept      -0.064193       -0.030719
             Length2    Length1         0.983122        0.995883
```

Output 44.3.3. Regression Predicted Mean Matching Model

```
                                 The MI Procedure

         Regression Models for Monotone Predicted Mean Matching Method

Imputed                                ---------------Imputation--------------
Variable  Effect              Obs Data            1             2             3

Length3   Intercept           -0.01304     0.004134     -0.011417     -0.034177
Length3   Length1             -0.01332     0.025320     -0.037494      0.308765
Length3   Length2              0.98918     0.955510      1.025741      0.673374
Length3   Length1*Length2     -0.02521    -0.034964     -0.022017     -0.017919

         Regression Models for Monotone Predicted Mean Matching Method

            Imputed                    ---------Imputation---------
            Variable  Effect                      4               5

            Length3   Intercept           -0.010532        0.004685
            Length3   Length1              0.156606       -0.147118
            Length3   Length2              0.828384        1.146440
            Length3   Length1*Length2     -0.029335       -0.034671
```

After the completion of five imputations by default, the "Multiple Imputation Variance Information" table shown in Output 44.3.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the "Combining Inferences from Multiply Imputed Data Sets" section on page 2561.

Output 44.3.4. Variance Information

```
                                 The MI Procedure

                     Multiple Imputation Variance Information

                      -----------------Variance-----------------
        Variable         Between          Within           Total       DF

        Length2         0.000133        0.439512        0.439672    32.15
        Length3         0.000386        0.486913        0.487376   32.131

                      Multiple Imputation Variance Information

                             Relative       Fraction
                             Increase        Missing        Relative
             Variable     in Variance    Information      Efficiency

             Length2         0.000363       0.000363        0.999927
             Length3         0.000952       0.000951        0.999810
```

The “Multiple Imputation Parameter Estimates” table shown in Output 44.3.5 displays a 95% mean confidence interval and a *t*-statistic with its associated *p*-value for each of the hypotheses requested with the MU0= option.

Output 44.3.5. Parameter Estimates

```
                                The MI Procedure

                   Multiple Imputation Parameter Estimates

  Variable             Mean       Std Error     95% Confidence Limits        DF

  Length2         33.104571        0.663078      31.75417      34.45497    32.15
  Length3         38.424571        0.698123      37.00277      39.84637   32.131

                   Multiple Imputation Parameter Estimates

                                                                t for H0:
Variable          Minimum         Maximum              Mu0     Mean=Mu0   Pr > |t|

Length2         33.088571       33.117143        35.000000        -2.86     0.0074
Length3         38.397143       38.445714        45.000000        -9.42     <.0001
```

The following statements list the first ten observations of the data set outex3 in Output 44.3.6. Note that the imputed values of Length2 are rounded to the same precision as the observed values.

```
proc print data=outex3(obs=10);
   title ’First 10 Observations of the Imputed Data Set’;
run;
```

Output 44.3.6. Imputed Data Set

```
                 First 10 Observations of the Imputed Data Set

               Obs    _Imputation_    Length1    Length2    Length3

                 1          1           23.2       25.4       30.0
                 2          1           24.0       26.3       31.2
                 3          1           23.9       26.5       31.1
                 4          1           26.3       29.0       33.5
                 5          1           26.5       29.0       34.7
                 6          1           26.8       29.7       34.7
                 7          1           26.8       28.8       34.7
                 8          1           27.6       30.0       35.0
                 9          1           27.6       30.0       35.1
                10          1           28.5       30.7       36.2
```

Example 44.4. Logistic Regression Method for CLASS Variables

This example uses logistic regression method to impute values for a binary variable in a data set with a monotone missing pattern.

The logistic regression method is used for the binary and ordinal CLASS variables. Since the variable Species is not an ordinal variable, only the first two species are used.

```
proc mi data=Fish2 seed=1305417 out=outex4;
   class Species;
   monotone logistic( Species= Height Width Height*Width/ details);
   var Height Width Species;
run;
```

The “Model Information” table shown in Output 44.4.1 describes the method and options used in the multiple imputation process.

Output 44.4.1. Model Information

```
                                The MI Procedure

                               Model Information

        Data Set                                  WORK.FISH2
        Method                                    Monotone
        Number of Imputations                     5
        Seed for random number generator          1305417
```

The “Monotone Model Specification” table shown in Output 44.4.2 describes methods and imputed variables in the imputation model. The procedure uses the logistic regression method to impute variable Species in the model. Missing values in other variables are not imputed.

Output 44.4.2. Monotone Model Specification

```
                                The MI Procedure

                          Monotone Model Specification

                                                  Imputed
                           Method                 Variables

                           Logistic Regression    Species
```

The “Missing Data Patterns” table shown in Output 44.4.3 lists distinct missing data patterns with corresponding frequencies and percents. The table confirms a monotone missing pattern for these variables.

Output 44.4.3. Missing Data Patterns

```
                                The MI Procedure

                             Missing Data Patterns

                                                       --------Group Means-------
Group  Height  Width  Species      Freq   Percent         Height           Width

    1  X       X      X              47     85.45      12.097645        4.808204
    2  X       X      .               6     10.91      11.411050        4.567050
    3  X       .      .               2      3.64      14.126350               .
```

With the DETAILS option, parameters estimated from the observed data and the parameters used in each imputation are displayed in the “Logistic Models for Monotone Method” table in Output 44.4.4.

Output 44.4.4. Logistic Regression Model

```
                                The MI Procedure

                       Logistic Models for Monotone Method

 Imputed                             ---------------Imputation---------------
 Variable  Effect          Obs-Data             1              2              3

 Species   Intercept        2.65234      1.794014       5.392323       5.859932
 Species   Height           7.73757      3.727095      11.790557      12.200408
 Species   Width           -5.25709     -1.209209      -8.492849      -8.696497
 Species   Height*Width    -1.12990     -1.593964      -1.989302      -3.087310

                       Logistic Models for Monotone Method

             Imputed                 ---------Imputation---------
             Variable  Effect                   4                5

             Species   Intercept        -0.649860         6.393629
             Species   Height            2.449332        13.644077
             Species   Width             0.629963       -10.767135
             Species   Height*Width      0.979165        -2.389491
```

The following statements list the first ten observations of the data set outex4 in Output 44.4.5.

```
proc print data=outex4(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 44.4.5. Imputed Data Set

```
                  First 10 Observations of the Imputed Data Set

          Obs    _Imputation_    Species    Length3     Height     Width

            1          1           Gp1        30.0     11.5200    4.0200
            2          1           Gp1        31.2     12.4800    4.3056
            3          1           Gp1        31.1     12.3778    4.6961
            4          1                      33.5     12.7300     .
            5          1           Gp1        34.0     12.4440    5.1340
            6          1           Gp1        34.7     13.6024    4.9274
            7          1           Gp1        34.5     14.1795    5.2785
            8          1           Gp1        35.0     12.6700    4.6900
            9          1           Gp1        35.1     14.0049    4.8438
           10          1           Gp1        36.2     14.2266    4.9594
```

Note that a missing value of the variable Species is not imputed if the corresponding covariates are missing and not imputed, as shown by observation 4 in the table.

Example 44.5. Discriminant Function Method for CLASS Variables

This example uses discriminant monotone methods to impute values of a CLASS variable from the observed observation values in a data set with a monotone missing pattern.

The following statements impute the continuous variables Height and Width with the regression method and the CLASS variable Species with the discriminant function method.

```
proc mi data=Fish2 seed=7545417 nimpute=3 out=outex5;
   class Species;
   monotone reg( Height Width)
            discrim( Species= Length3 Height Width/ details);
   var Length3 Height Width Species;
run;
```

The “Model Information” table shown in Output 44.5.1 describes the method and options used in the multiple imputation process.

Output 44.5.1. Model Information

```
                               The MI Procedure

                              Model Information

           Data Set                              WORK.FISH2
           Method                                Monotone
           Number of Imputations                 3
           Seed for random number generator      7545417
```

The “Monotone Model Specification” table shown in Output 44.5.2 describes methods and imputed variables in the imputation model. The procedure uses the regression method to impute variables Height and Width, and uses the logistic regression method to impute variable Species in the model.

Output 44.5.2. Monotone Model Specification

```
                                The MI Procedure

                         Monotone Model Specification

                                              Imputed
                     Method                   Variables

                     Regression               Height Width
                     Discriminant Function    Species
```

The “Missing Data Patterns” table shown in Output 44.5.3 lists distinct missing data patterns with corresponding frequencies and percents. The table confirms a monotone missing pattern for these variables.

Output 44.5.3. Missing Data Patterns

```
                                The MI Procedure

                              Missing Data Patterns

     Group    Length3    Height    Width    Species         Freq     Percent

         1    X          X         X        X                 47       85.45
         2    X          X         X        .                  6       10.91
         3    X          X         .        .                  2        3.64

                             Missing Data Patterns

                      -----------------Group Means----------------
             Group         Length3          Height           Width

                 1       33.497872       12.097645        4.808204
                 2       32.366667       11.411050        4.567050
                 3       36.600000       14.126350               .
```

With the DETAILS option, parameters estimated from the observed data and parameters used in each imputation are displayed in Output 44.5.4.

Output 44.5.4. Discriminant Model

```
                                 The MI Procedure

                    Group Means for Monotone Discriminant Method

                                         ----------------Imputation---------------
Species     Variable     Obs-Data                 1               2               3

Gp1         Length3       0.61625          0.707861        0.662448        0.505410
Gp1         Height        0.67244          0.750984        0.732151        0.594226
Gp1         Width         0.57896          0.643334        0.665698        0.515014
Gp2         Length3      -0.98925         -0.776131       -0.987989       -0.887032
Gp2         Height       -1.08272         -0.934081       -1.081832       -1.004799
Gp2         Width        -0.86963         -0.680065       -0.811745       -0.722943
```

The following statements list the first ten observations of the data set outex5 in Output 44.5.5 Note that all missing values of variables Width and Species are imputed.

```
proc print data=outex5(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 44.5.5. Imputed Data Set

```
                  First 10 Observations of the Imputed Data Set

       Obs    _Imputation_    Species    Length3     Height      Width

         1          1           Gp1        30.0     11.5200    4.02000
         2          1           Gp1        31.2     12.4800    4.30560
         3          1           Gp1        31.1     12.3778    4.69610
         4          1           Gp1        33.5     12.7300    4.67966
         5          1           Gp2        34.0     12.4440    5.13400
         6          1           Gp1        34.7     13.6024    4.92740
         7          1           Gp1        34.5     14.1795    5.27850
         8          1           Gp1        35.0     12.6700    4.69000
         9          1           Gp1        35.1     14.0049    4.84380
        10          1           Gp1        36.2     14.2266    4.95940
```

Example 44.6. MCMC Method

This example uses the MCMC method to impute missing values for a data set with an arbitrary missing pattern. The following statements invoke the MI procedure and specify the MCMC method with six imputations.

```
proc mi data=FitMiss seed=21355417 nimpute=6 mu0=50 10 180 ;
   mcmc chain=multiple displayinit initial=em(itprint);
   var Oxygen RunTime RunPulse;
run;
```

Output 44.6.1. Model Information

```
                              The MI Procedure

                             Model Information

           Data Set                              WORK.FITMISS
           Method                                MCMC
           Multiple Imputation Chain             Multiple Chains
           Initial Estimates for MCMC            EM Posterior Mode
           Start                                 Starting Value
           Prior                                 Jeffreys
           Number of Imputations                 6
           Number of Burn-in Iterations          200
           Seed for random number generator      21355417
```

The “Model Information” table shown in Output 44.6.1 describes the method used in the multiple imputation process. With CHAIN=MULTIPLE, the procedure uses multiple chains and completes the default 200 burn-in iterations before each imputation. The 200 burn-in iterations are used to make the iterations converge to the stationary distribution before the imputation.

By default, the procedure uses a noninformative Jeffreys prior to derive the posterior mode from the EM algorithm as the starting values for the MCMC process.

The “Missing Data Patterns” table shown in Output 44.6.2 lists distinct missing data patterns with corresponding statistics.

Output 44.6.2. Missing Data Patterns

```
                              The MI Procedure

                            Missing Data Patterns

                                 Run     Run
             Group     Oxygen    Time    Pulse        Freq      Percent

                 1     X         X       X              21        67.74
                 2     X         X       .               4        12.90
                 3     X         .       .               3         9.68
                 4     .         X       X               1         3.23
                 5     .         X       .               2         6.45

                            Missing Data Patterns

                      -----------------Group Means----------------
              Group           Oxygen          RunTime         RunPulse

                  1        46.353810        10.809524       171.666667
                  2        47.109500        10.137500                .
                  3        52.461667                .                .
                  4                .        11.950000       176.000000
                  5                .         9.885000                .
```

With the ITPRINT option in INITIAL=EM, the procedure displays the “EM (Posterior Mode) Iteration History” table in Output 44.6.3.

Output 44.6.3. EM (Posterior Mode) Iteration History

```
                                The MI Procedure

                      EM (Posterior Mode) Iteration History

_Iteration_        -2 Log L   -2 Log Posterior         Oxygen         RunTime

          0      254.482800         282.909549      47.104077       10.554858
          1      255.081168         282.051584      47.104077       10.554857
          2      255.271408         282.017488      47.104077       10.554857
          3      255.318622         282.015372      47.104002       10.554523
          4      255.330259         282.015232      47.103861       10.554388
          5      255.333161         282.015222      47.103797       10.554341
          6      255.333896         282.015222      47.103774       10.554325
          7      255.334085         282.015222      47.103766       10.554320

                      EM (Posterior Mode) Iteration History

                           _Iteration_        RunPulse

                                     0      171.381669
                                     1      171.381652
                                     2      171.381644
                                     3      171.381842
                                     4      171.382053
                                     5      171.382150
                                     6      171.382185
                                     7      171.382196
```

With the DISPLAYINIT option in the MCMC statement, the “Initial Parameter Estimates for MCMC” table shown in Output 44.6.4 displays the starting mean and covariance estimates used in MCMC. The same starting estimates are used for the MCMC process for multiple chains because the EM algorithm is applied to the same data set in each chain. You can explicitly specify different initial estimates for different imputations, or you can use the bootstrap to generate different parameter estimates from the EM algorithm for the MCMC process.

Output 44.6.4. Initial Parameter Estimates

```
                                The MI Procedure

                       Initial Parameter Estimates for MCMC

      _TYPE_     _NAME_             Oxygen          RunTime         RunPulse

      MEAN                       47.103766        10.554320       171.382196
      COV        Oxygen          24.549967        -5.726112       -15.926036
      COV        RunTime         -5.726112         1.781407         3.124798
      COV        RunPulse       -15.926036         3.124798        83.164045
```

Output 44.6.5 and Output 44.6.6 display variance information and parameter estimates from the multiple imputation.

Output 44.6.5. Variance Information

```
                                The MI Procedure

                   Multiple Imputation Variance Information

                    -----------------Variance-----------------
         Variable          Between          Within           Total        DF

         Oxygen           0.051560        0.928170        0.988323    25.958
         RunTime          0.003979        0.070057        0.074699    25.902
         RunPulse         4.118578        4.260631        9.065638    7.5938

                    Multiple Imputation Variance Information

                               Relative        Fraction
                               Increase         Missing        Relative
              Variable      in Variance     Information      Efficiency

              Oxygen           0.064809        0.062253        0.989731
              RunTime          0.066262        0.063589        0.989513
              RunPulse         1.127769        0.575218        0.912517
```

Output 44.6.6. Parameter Estimates

```
                                The MI Procedure

                    Multiple Imputation Parameter Estimates

    Variable            Mean      Std Error    95% Confidence Limits         DF

    Oxygen         47.164819       0.994145      45.1212      49.2085    25.958
    RunTime        10.549936       0.273312       9.9880      11.1118    25.902
    RunPulse      170.969836       3.010920     163.9615     177.9782    7.5938

                    Multiple Imputation Parameter Estimates

                                                                 t for H0:
  Variable          Minimum         Maximum              Mu0       Mean=Mu0   Pr > |t|

  Oxygen          46.858020       47.363540        50.000000          -2.85     0.0084
  RunTime         10.476886       10.659412        10.000000           2.01     0.0547
  RunPulse       168.252615      172.894991       180.000000          -3.00     0.0182
```

Example 44.7. Producing Monotone Missingness with MCMC

This example uses the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern based on the order of variables in the VAR statement.

The following statements invoke the MI procedure and specify the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern. You must specify a VAR statement to provide the order of variables for the imputed data to achieve a monotone missing pattern.

```
proc mi data=FitMiss seed=17655417 out=outex7;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

Output 44.7.1. Model Information

```
                              The MI Procedure

                             Model Information

         Data Set                                WORK.FITMISS
         Method                                  Monotone-data MCMC
         Multiple Imputation Chain               Single Chain
         Initial Estimates for MCMC              EM Posterior Mode
         Start                                   Starting Value
         Prior                                   Jeffreys
         Number of Imputations                   5
         Number of Burn-in Iterations            200
         Number of Iterations                    100
         Seed for random number generator        17655417
```

The “Model Information” table shown in Output 44.7.1 describes the method used in the multiple imputation process.

The “Missing Data Patterns” table shown in Output 44.7.2 lists distinct missing data patterns with corresponding statistics. Here, an “X” means that the variable is observed in the corresponding group, a “.” means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an “O” means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 44.7.2. Missing Data Pattern

```
                                 The MI Procedure

                               Missing Data Patterns

                                Run      Run
            Group     Oxygen    Time     Pulse        Freq      Percent

                1     X         X        X              21        67.74
                2     X         X        O               4        12.90
                3     X         O        O               3         9.68
                4     .         X        X               1         3.23
                5     .         X        O               2         6.45

                              Missing Data Patterns

                      -----------------Group Means----------------
             Group           Oxygen          RunTime         RunPulse

                 1        46.353810        10.809524       171.666667
                 2        47.109500        10.137500                .
                 3        52.461667                .                .
                 4                .        11.950000       176.000000
                 5                .         9.885000                .
```

As shown in the table, the MI procedure only needs to impute three missing values from Group 4 and Group 5 to achieve a monotone missing pattern for the imputed data set.

When using the MCMC method to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements are used just to show the monotone missingness of the output data set outex7.

```
proc mi data=outex7  nimpute=0;
   var Oxygen RunTime RunPulse;
run;
```

Output 44.7.3. Monotone Missing Data Pattern

```
                              The MI Procedure

                            Missing Data Patterns

                                 Run     Run
          Group    Oxygen        Time    Pulse        Freq     Percent

              1    X             X       X             110       70.97
              2    X             X       .              30       19.35
              3    X             .       .              15        9.68

                           Missing Data Patterns

                    -----------------Group Means----------------
           Group           Oxygen          RunTime          RunPulse

               1        46.152428        10.861364        171.863636
               2        47.796038        10.053333                 .
               3        52.461667                .                 .
```

The “Missing Data Patterns” table shown in Output 44.7.3 displays a monotone missing data pattern.

The following statements impute one value for each missing value in the monotone missingness data set outex7.

```
proc mi data=outex7 nimpute=1 seed=51343672 out=outds;
   monotone method=reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the MIANALYZE procedure. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

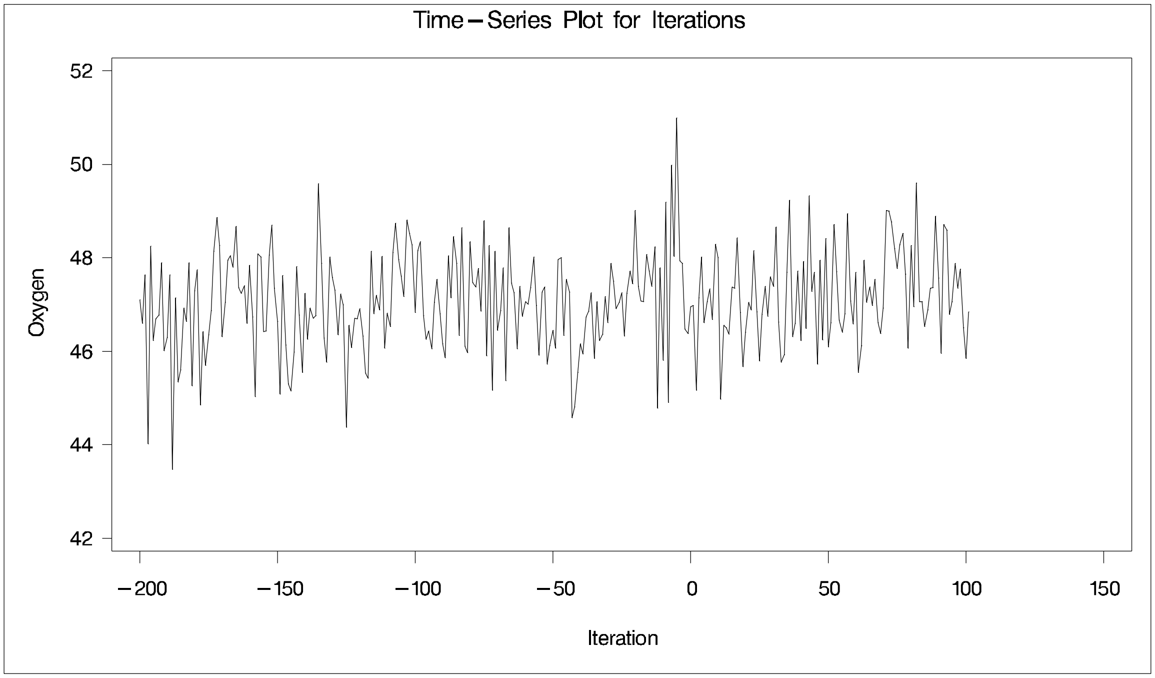
Example 44.8. Checking Convergence in MCMC

This example uses the MCMC method with a single chain. It also displays time-series and autocorrelation plots to check convergence for the single chain.

The following statements use the MCMC method to create an iteration plot for the successive estimates of the mean of Oxygen. Note that iterations during the burn-in period are indicated with negative iteration numbers. These statements also create an autocorrelation function plot for the variable Oxygen.

```
proc mi data=FitMiss seed=42037921 noprint nimpute=2;
   mcmc timeplot(mean(Oxygen)) acfplot(mean(Oxygen));
   var Oxygen RunTime RunPulse;
run;
```

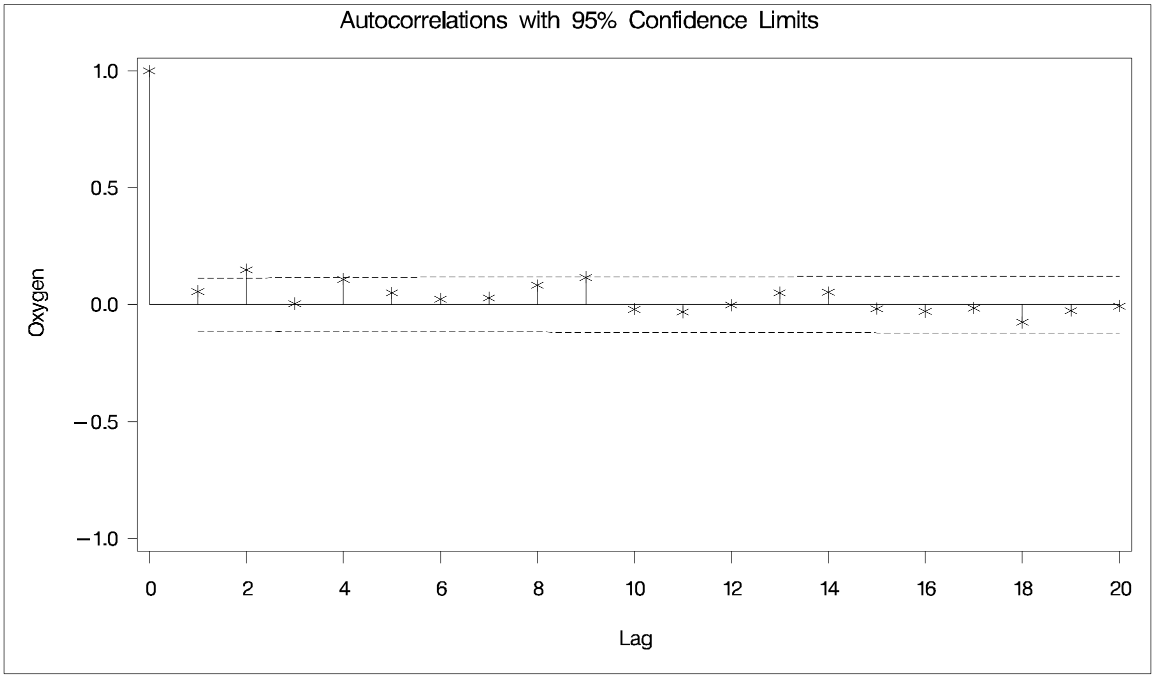
Output 44.8.1. Time-Series Plot for Oxygen



With the TIMEPLOT(MEAN(Oxygen)) option, the procedure displays a time-series plot for the mean of Oxygen in Output 44.8.1.

By default, the MI procedure displays solid line segments that connect data points in the time-series plot. The plot shows no apparent trends for the variable Oxygen.

Output 44.8.2. Autocorrelation Function Plot for Oxygen



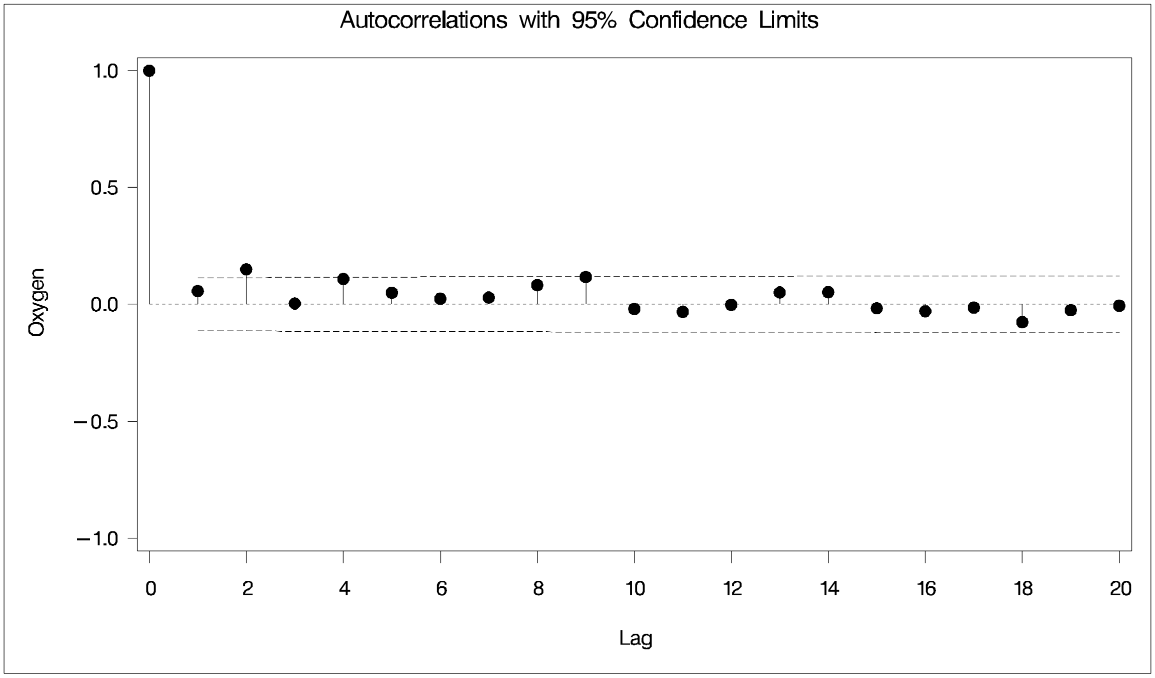
With the ACFPLOT(MEAN(oxygen)) option, the procedure displays an autocorrelation plot for the mean of Oxygen in Output 44.8.2.

By default, the MI procedure uses the star sign (*) as the plot symbol to display the points in the plot, a solid line to display the reference line of zero autocorrelation, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations. The autocorrelation function plot shows no significant positive or negative autocorrelation.

The following statements use display options to modify the autocorrelation function plot for Oxygen in Output 44.8.3.

```
proc mi data=FitMiss seed=42037921 noprint nimpute=2;
   mcmc acfplot(mean(Oxygen) / symbol=dot lref=2);
   var Oxygen RunTime RunPulse;
run;
```

Output 44.8.3. Autocorrelation Function Plot for Oxygen



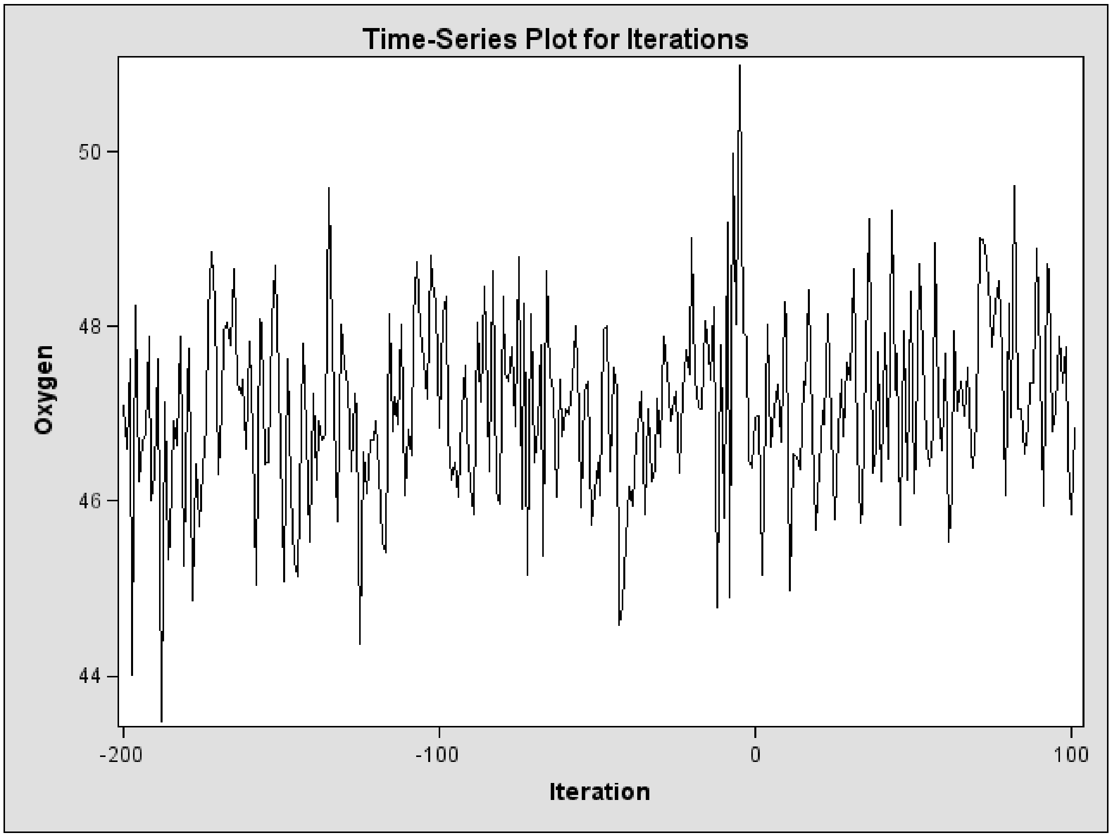
You can also create plots for the worst linear function, the means of other variables, the variances of variables, and covariances between variables. Alternatively, you can use the OUTITER option to save statistics such as the means, standard deviations, covariances, $-2 \log \text{LR}$ statistic, $-2 \log \text{LR}$ statistic of the posterior mode, and worst linear function from each iteration in an output data set. Then you can do a more in-depth time-series analysis of the iterations with other procedures, such as PROC AUTOREG and PROC ARIMA in the *SAS/ETS User's Guide*.

With the experimental ODS GRAPHICS statement specified in the following statements

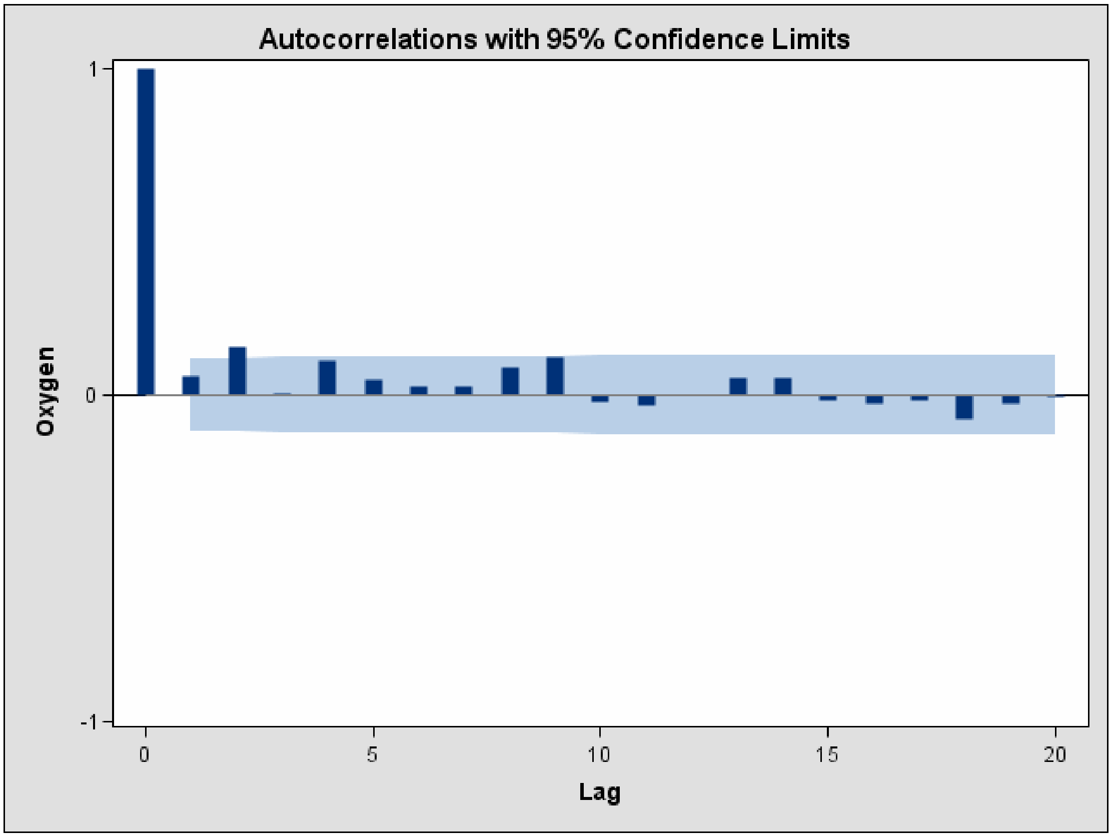
```
ods html;
ods graphics on;

proc mi data=FitMiss seed=42037921 noprint nimpute=2;
   mcmc timeplot(mean(Oxygen)) acfplot(mean(Oxygen));
   var Oxygen RunTime RunPulse;
run;

ods graphics off;
ods html close;
```

the MI procedure produces the experimental graphs, as shown in Output 44.8.4 and Output 44.8.5.

Output 44.8.4. Time-Series Plot for Oxygen (Experimental)



Output 44.8.5. Autocorrelation Function Plot for Oxygen (Experimental)



For general information about ODS graphics see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the MI procedure, see the "ODS Graphics" section on page 2567.

Example 44.9. Saving and Using Parameters for MCMC

This example uses the MCMC method with multiple chains as specified in Example 44.6. It saves the parameter values used for each imputation in an output data set of type EST called miest. This output data set can then be used to impute missing values in other similar input data sets. The following statements invoke the MI procedure and specify the MCMC method with multiple chains to create three imputations.

```
proc mi data=FitMiss seed=21355417 nimpute=6 mu0=50 10 180 ;
   mcmc chain=multiple initial=em outest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The following statements list the parameters used for the imputations in Output 44.9.1. Note that the data set includes observations with _TYPE_='SEED' containing the seed to start the next random number generator.

```
proc print data=miest(obs=15);
   title 'Parameters for the Imputations';
run;
```

Output 44.9.1. OUTEST Data Set

```
                        Parameters for the Imputations

Obs _Imputation_ _TYPE_  _NAME_          Oxygen        RunTime       RunPulse

  1       1       SEED             825240167.00   825240167.00   825240167.00
  2       1       PARM                    46.77          10.47         169.41
  3       1       COV   Oxygen            30.59          -8.32         -50.99
  4       1       COV   RunTime           -8.32           2.90          17.03
  5       1       COV   RunPulse         -50.99          17.03         200.09
  6       2       SEED            1895925872.00  1895925872.00  1895925872.00
  7       2       PARM                    47.41          10.37         173.34
  8       2       COV   Oxygen            22.35          -4.44         -21.18
  9       2       COV   RunTime           -4.44           1.76           1.25
 10       2       COV   RunPulse         -21.18           1.25         125.67
 11       3       SEED             137653011.00   137653011.00   137653011.00
 12       3       PARM                    48.21          10.36         170.52
 13       3       COV   Oxygen            23.59          -5.25         -19.76
 14       3       COV   RunTime           -5.25           1.66           5.00
 15       3       COV   RunPulse         -19.76           5.00         110.99
```

The following statements invoke the MI procedure and use the INEST= option in the MCMC statement.

```
proc mi data=FitMiss;
   mcmc inest=miest;
   var Oxygen RunTime RunPulse;
run;
```

Output 44.9.2. Model Information

```
                               The MI Procedure

                              Model Information

         Data Set                             WORK.FITMISS
         Method                               MCMC
         INEST Data Set                       WORK.MIEST
         Number of Imputations                6
```

The "Model Information" table shown in Output 44.9.2 describes the method used in the multiple imputation process. The remaining tables for the example are identical to the tables in Output 44.6.2, Output 44.6.4, Output 44.6.5, and Output 44.6.6 in Example 44.6.

Example 44.10. Transforming to Normality

This example applies the MCMC method to the FitMiss data set in which the variable Oxygen is transformed. Assume that Oxygen is skewed and can be transformed to normality with a logarithmic transformation. The following statements invoke the MI procedure and specify the transformation. The TRANSFORM statement specifies the log transformation for Oxygen. Note that the values displayed for Oxygen in all of the results correspond to transformed values.

```
proc mi data=FitMiss seed=32937921 mu0=50 10 180 out=outex10;
   transform log(Oxygen);
   mcmc chain=multiple displayinit;
   var Oxygen RunTime RunPulse;
run;
```

The “Missing Data Patterns” table shown in Output 44.10.1 lists distinct missing data patterns with corresponding statistics for the FitMiss data. Note that the values of Oxygen shown in the tables are transformed values.

Output 44.10.1. Missing Data Pattern

```
                    The MI Procedure

                  Missing Data Patterns

                   Run     Run
Group    Oxygen    Time    Pulse        Freq     Percent

    1    X         X       X              21       67.74
    2    X         X       .               4       12.90
    3    X         .       .               3        9.68
    4    .         X       X               1        3.23
    5    .         X       .               2        6.45

              Transformed Variables: Oxygen

                 Missing Data Patterns

          -----------------Group Means----------------
 Group           Oxygen         RunTime        RunPulse

     1         3.829760       10.809524      171.666667
     2         3.851813       10.137500               .
     3         3.955298               .               .
     4                .       11.950000      176.000000
     5                .        9.885000               .

             Transformed Variables: Oxygen
```

The “Variable Transformations” table shown in Output 44.10.2 lists the variables that have been transformed.

Output 44.10.2. Variable Transformations

```
                                  The MI Procedure

                              Variable Transformations

                              Variable     _Transform_

                              Oxygen       LOG
```

The “Initial Parameter Estimates for MCMC” table shown in Output 44.10.3 displays the starting mean and covariance estimates used in the MCMC process.

Output 44.10.3. Initial Parameter Estimates

```
                                  The MI Procedure

                        Initial Parameter Estimates for MCMC

        _TYPE_     _NAME_              Oxygen         RunTime        RunPulse

        MEAN                         3.846122       10.557605      171.382949
        COV        Oxygen            0.010827       -0.120891       -0.328772
        COV        RunTime          -0.120891        1.744580        3.011180
        COV        RunPulse         -0.328772        3.011180       82.747609

                           Transformed Variables: Oxygen
```

Output 44.10.4 displays variance information from the multiple imputation.

Output 44.10.4. Variance Information

```
                                The MI Procedure

                    Multiple Imputation Variance Information

                     -----------------Variance-----------------
         Variable         Between         Within          Total       DF

       * Oxygen       0.000016175       0.000401       0.000420   26.499
         RunTime         0.001762       0.065421       0.067536   27.118
         RunPulse        0.205979       3.116830       3.364004   25.222

                            * Transformed Variables

                    Multiple Imputation Variance Information

                              Relative       Fraction
                              Increase        Missing       Relative
              Variable     in Variance    Information     Efficiency

            * Oxygen          0.048454       0.047232       0.990642
              RunTime         0.032318       0.031780       0.993684
              RunPulse        0.079303       0.075967       0.985034

                             * Transformed Variables
```

Output 44.10.5 displays parameter estimates from the multiple imputation. Note that the parameter value of μ_0 has also been transformed using the logarithmic transformation.

Output 44.10.5. Parameter Estimates

```
                                The MI Procedure

                     Multiple Imputation Parameter Estimates

    Variable            Mean      Std Error    95% Confidence Limits        DF

  * Oxygen          3.845175       0.020494       3.8031       3.8873   26.499
    RunTime        10.560131       0.259876      10.0270      11.0932   27.118
    RunPulse      171.802181       1.834122     168.0264     175.5779   25.222

                             * Transformed Variables

                    Multiple Imputation Parameter Estimates

                                                           t for H0:
  Variable          Minimum        Maximum            Mu0    Mean=Mu0   Pr > |t|

* Oxygen          3.838599       3.848456       3.912023       -3.26     0.0030
  RunTime        10.493031      10.600498      10.000000        2.16     0.0402
  RunPulse      171.251777     172.498626     180.000000       -4.47     0.0001

                            * Transformed Variables
```

The following statements list the first ten observations of the data set outmi in Output 44.10.6. Note that the values for Oxygen are in the original scale.

```
proc print data=outex10(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 44.10.6. Imputed Data Set in Original Scale

```
                 First 10 Observations of the Imputed Data Set

                                                                Run
              Obs    _Imputation_     Oxygen    RunTime      Pulse

                1          1         44.6090    11.3700    178.000
                2          1         45.3130    10.0700    185.000
                3          1         54.2970     8.6500    156.000
                4          1         59.5710     7.1440    167.012
                5          1         49.8740     9.2200    170.092
                6          1         44.8110    11.6300    176.000
                7          1         38.5834    11.9500    176.000
                8          1         43.7376    10.8500    158.851
                9          1         39.4420    13.0800    174.000
               10          1         60.0550     8.6300    170.000
```

Note that the preceding results can also be produced from the following statements without using a TRANSFORM statement. A transformed value of log(50)=3.91202 is used in the MU0= option.

```
data temp;
   set FitMiss;
   LogOxygen= log(Oxygen);
run;

proc mi data=temp seed=14337921 mu0=3.91202 10 180 out=outtemp;
   mcmc chain=multiple displayinit;
   var LogOxygen RunTime RunPulse;
run;

data outex10;
   set outtemp;
   Oxygen= exp(LogOxygen);
run;
```

Example 44.11. Multistage Imputation

This example uses two separate imputation procedures to complete the imputation process. The first MI procedure uses the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern. The second MI procedure uses a MONOTONE statement to impute missing values for data sets with monotone missing patterns.

The following statements are identical to Example 44.7. The statements invoke the MI procedure and specify the the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern.

```
proc mi data=FitMiss seed=17655417 out=outex11;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The “Missing Data Patterns” table shown in Output 44.11.1 lists distinct missing data patterns with corresponding statistics. Here, an “X” means that the variable is observed in the corresponding group, a “.” means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an “O” means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 44.11.1. Missing Data Pattern

```
                                The MI Procedure

                              Missing Data Patterns

                                Run      Run
             Group     Oxygen    Time     Pulse        Freq      Percent

                 1     X         X        X              21        67.74
                 2     X         X        O               4        12.90
                 3     X         O        O               3         9.68
                 4     .         X        X               1         3.23
                 5     .         X        O               2         6.45

                             Missing Data Patterns

                       -----------------Group Means----------------
              Group           Oxygen          RunTime         RunPulse

                  1        46.353810        10.809524       171.666667
                  2        47.109500        10.137500                .
                  3        52.461667                .                .
                  4                .        11.950000       176.000000
                  5                .         9.885000                .
```

As shown in the table, the MI procedure only needs to impute three missing values from Group 4 and Group 5 to achieve a monotone missing pattern for the imputed data set. When the MCMC method is used to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements impute one value for each missing value in the monotone missingness data set outex11.

```
proc mi data=outex11
        nimpute=1 seed=51343672
        out=outex11a;
   monotone reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the procedure MIANALYZE. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

The “Model Information” table displayed in Output 44.11.2 shows that a monotone method is used to generate imputed values in the first BY group.

Output 44.11.2. Model Information

```
----------------------------- Imputation Number=1 ------------------------------

                                The MI Procedure

                               Model Information

          Data Set                             WORK.OUTEX11
          Method                               Monotone
          Number of Imputations                1
          Seed for random number generator     51343672
```

The “Monotone Model Specification” table shown in Output 44.11.3 describes methods and imputed variables in the imputation model. The procedure uses the regression method to impute variables RunTime and RunPulse in the model.

Output 44.11.3. Monotone Model Specification

```
----------------------------- Imputation Number=1 ------------------------------

                                The MI Procedure

                          Monotone Model Specification

                                         Imputed
                          Method         Variables

                          Regression     RunTime RunPulse
```

The “Missing Data Patterns” table shown in Output 44.11.4 lists distinct missing data patterns with corresponding statistics. It shows a monotone missing pattern for the imputed data set.

Output 44.11.4. Missing Data Pattern

```
----------------------------- Imputation Number=1 ------------------------------

                                The MI Procedure

                              Missing Data Patterns

                               Run     Run
            Group    Oxygen    Time    Pulse        Freq     Percent

                1    X         X       X              22       70.97
                2    X         X       .               6       19.35
                3    X         .       .               3        9.68

                             Missing Data Patterns

                      -----------------Group Means----------------
             Group          Oxygen         RunTime        RunPulse

                 1       46.057479       10.861364      171.863636
                 2       46.745227       10.053333               .
                 3       52.461667               .               .
```

The following statements list the first ten observations of the data set outex11a in Output 44.11.5.

```
proc print data=outex11a(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 44.11.5. Imputed Data Set

```
                 First 10 Observations of the Imputed Data Set

                                                             Run
              Obs    _Imputation_     Oxygen    RunTime     Pulse

                1          1         44.6090    11.3700    178.000
                2          1         45.3130    10.0700    185.000
                3          1         54.2970     8.6500    156.000
                4          1         59.5710     7.1569    169.914
                5          1         49.8740     9.2200    159.315
                6          1         44.8110    11.6300    176.000
                7          1         39.8345    11.9500    176.000
                8          1         45.3196    10.8500    151.252
                9          1         39.4420    13.0800    174.000
               10          1         60.0550     8.6300    170.000
```

This example presents an alternative to the full-data MCMC imputation. When imputation of only a few missing values are needed to achieve a monotone missing pattern for the imputed data set. The example uses a monotone MCMC method that impute fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227). The example also demonstrates how to combine the monotone MCMC method with a method for monotone missing data, which does not rely on iterations of steps.

References

In addition to the research literature listed below, the development of the MI and MIANALYZE procedures has benefited significantly from the advice and expertise of

- Paul Allison, University of Pennsylvania
- John Barnard Jr., Cleveland Clinic Foundation
- Nick Horton, Boston University
- Donald B. Rubin, Harvard University
- Joseph L. Schafer, Penn State University

We gratefully acknowledge the time and effort that these researchers have contributed.

Anderson, T.W. (1984), *An Introduction to Multivariate Statistical Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Allison, P.D. (2000), “Multiple Imputation for Missing Data: A Cautionary Tale,” *Sociological Methods and Research*, 28, 301–309.

Allison, P.D. (2001), “Missing Data,” Thousand Oaks, CA: Sage Publications.

Barnard, J., and Meng, X.L. (1999), “Applications of Multiple Imputation in Medical Studies: From AIDS to NHANES,” *Statistical Methods in Medical Research*, 8, 17–36.

Barnard, J. and Rubin, D.B. (1999), “Small-Sample Degrees of Freedom with Multiple Imputation,” *Biometrika*, 86, 948–955.

Brand, J.P.L. (1999), “Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets,” Ph.D. Thesis, Erasmus University Rotterdam.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977), “Maximum Likelihood from Incomplete Data via the EM Algorithm,” *Journal of the Royal Statistical Society*, Ser. B., 39, 1–38.

Gadbury, G.L., Coffey, C.S., and Allison, D.B. (2003), “Modern Statistical Methods for Handling Missing Repeated Measurements in Obesity Trial Data: Beyond LOCF,” *Obesity Reviews*, 4, 175–184.

Gelman, A. and Rubin, D.B. (1992), “Inference from Iterative Simulation Using Multiple Sequences,” *Statistical Science*, 7, 457–472.

Goodnight, J.H. (1979), “A Tutorial on the Sweep Operator,” *American Statistician*, 33, 149–158.

Heitjan, F. and Little, R.J.A. (1991), “Multiple Imputation for the Fatal Accident Reporting System,” *Applied Statistics*, 40, 13–29.

Horton, N.J. and Lipsitz, S.R. (2001), “Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables,” *Journal of the American Statistical Association*, 55, 244–254.

Lavori, P.W., Dawson, R., and Shera, D. (1995), “A Multiple Imputation Strategy for Clinical Trials with Truncation of Patient Data,” *Statistics in Medicine*, 14, 1913–1925.

Li, K.H. (1988), “Imputation Using Markov Chains,” *Journal of Statistical Computation and Simulation*, 30, 57–79.

Li, K.H., Raghunathan, T.E., and Rubin, D.B. (1991), “Large-Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics and an F Reference Distribution,” *Journal of the American Statistical Association*, 86, 1065–1073.

Little, R.J.A. and Rubin, D.B. (2002), *Statistical Analysis with Missing Data*, Second Edition, New York: John Wiley & Sons, Inc.

Liu, C. (1993), “Bartlett’s Decomposition of the Posterior Distribution of the Covariance for Normal Monotone Ignorable Missing Data,” *Journal of Multivariate Analysis*, 46, 198–206.

McLachlan, G.J. and Krishnan, T. (1997), *The EM Algorithm and Extensions*, New York: John Wiley & Sons, Inc.

Rosenbaum, P.R. and Rubin, D.B. (1983), “The Central Role of the Propensity Score in Observational Studies for Causal Effects,” *Biometrika*, 70, 41–55.

Rubin, D.B. (1976), “Inference and Missing Data,” *Biometrika*, 63, 581–592.

Rubin, D.B. (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: John Wiley & Sons, Inc.

Rubin, D.B. (1996), “Multiple Imputation After 18+ Years,” *Journal of the American Statistical Association*, 91, 473–489.

Schafer, J.L. (1997), *Analysis of Incomplete Multivariate Data*, New York: Chapman and Hall.

Schafer, J.L. (1999), “Multiple Imputation: A Primer,” *Statistical Methods in Medical Research*, 8, 3–15.

Schenker, N. and Taylor, J.M.G. (1996), “Partially Parametric Techniques for Multiple Imputation,” *Computational Statistics and Data Analysis*, 22, 425–446.

Tanner, M.A. and Wong, W.H. (1987), “The Calculation of Posterior Distributions by Data Augmentation,” *Journal of the American Statistical Association*, 82, 528–540.

van Buuren, S., Boshuizen, H.C., and Knook, D.L. (1999), “Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis,” *Statistics in Medicine*, 18, 681–694.

Chapter 45
The MIANALYZE Procedure

Chapter Contents

Chapter 45
The MIANALYZE Procedure

Overview

The MIANALYZE procedure combines the results of the analyses of imputations and generates valid statistical inferences. Multiple imputation provides a useful strategy for analyzing data sets with missing values. Instead of filling in a single value for each missing value, Rubin's (1976; 1987) multiple imputation strategy replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute. You can implement the strategy with two SAS procedures: PROC MI, which generates imputed data sets, and PROC MIANALYZE, which combines the results of analyses carried out on the data sets.

The analyses of imputations are obtained by using standard SAS procedures (such as PROC REG) for complete data. No matter which complete-data analysis is used, the process of combining results from different imputed data sets is essentially the same and results in valid statistical inferences that properly reflect the uncertainty due to missing values.

The MIANALYZE procedure reads parameter estimates and associated standard errors or covariance matrix that are computed by the standard statistical procedure for each imputed data set. The MIANALYZE procedure then derives valid univariate inference for these parameters. With an additional assumption about the population between and within imputation covariance matrices, multivariate inference based on Wald tests can also be derived.

For some parameters of interest, you can use TEST statements to test linear hypotheses about the parameters. For others, it is not straightforward to compute estimates and associated covariance matrices with standard statistical SAS procedures. Examples include correlation coefficients between two variables and ratios of variable means. These special cases are described in the "Examples of the Complete-Data Inferences" section on page 2628.

In SAS 9, the VAR statement has been replaced by the MODELEFFECTS statement to accommodate various effects to be analyzed. A STDERR statement has been added to specify the standard errors when both parameter estimates and associated standard errors are stored as variables in the same data set.

Also, an experimental CLASS statement has been added to specify classification variables in the MODELEFFECTS statement. The CLASS statement must be used in conjunction with the MODELEFFECTS statement.

Getting Started

The Fitness data set has been altered to contain an arbitrary missing pattern:

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
data FitMiss;
   input Oxygen RunTime RunPulse @@;
   datalines;
44.609  11.37  178     45.313  10.07  185
54.297   8.65  156     59.571    .      .
49.874   9.22    .     44.811  11.63  176
  .     11.95  176          .  10.85    .
39.442  13.08  174     60.055   8.63  170
50.541    .      .     37.388  14.03  186
44.754  11.12  176     47.273    .      .
51.855  10.33  166     49.156   8.95  180
40.836  10.95  168     46.672  10.00    .
46.774  10.25    .     50.388  10.08  168
39.407  12.63  174     46.080  11.17  156
45.441   9.63  164       .      8.92    .
45.118  11.08    .     39.203  12.88  168
45.790  10.47  186     50.545   9.93  148
48.673   9.40  186     47.920  11.50  170
47.467  10.50  170
;
```

Suppose that the data are multivariate normally distributed and that the missing data are missing at random (see the "Statistical Assumptions for Multiple Imputation" section in "The MI Procedure" chapter for a description of these assumptions). The following statements use the MI procedure to impute missing values for the FitMiss data set.

```
proc mi data=FitMiss noprint out=outmi seed=3237851;
   var Oxygen RunTime RunPulse;
run;
```

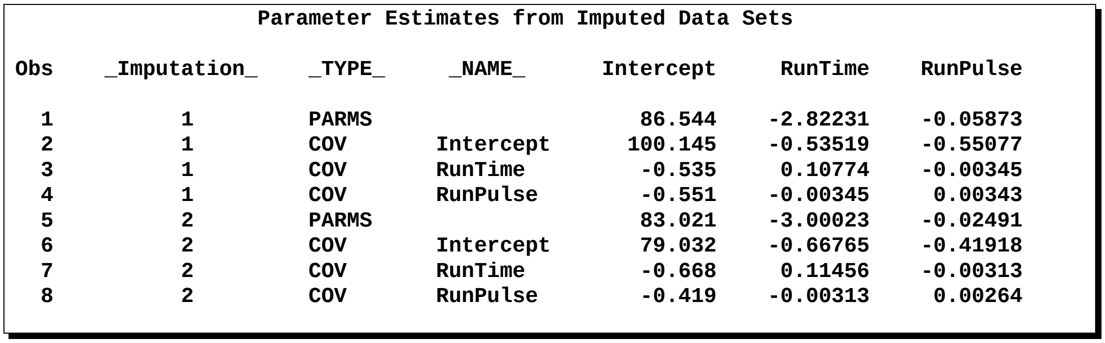
The MI procedure creates imputed data sets, which are stored in the outmi data set. A variable named _Imputation_ indicates the imputation numbers. Based on m imputations, m different sets of the point and variance estimates for a parameter can be computed. In this example, $m = 5$ is the default.

The following statements generate regression coefficients for each of the five imputed data sets:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display parameter estimates and covariance matrices from the first two imputed data sets in Figure 45.1.

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'Parameter Estimates from Imputed Data Sets';
run;
```

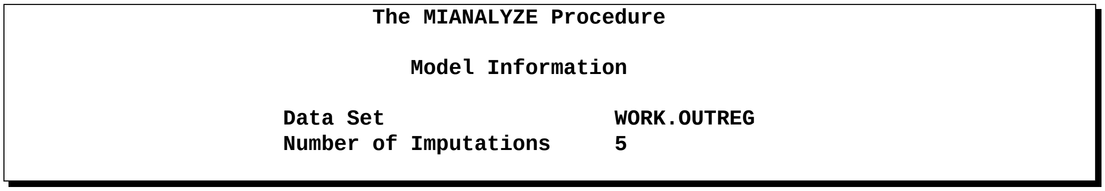


```
                    Parameter Estimates from Imputed Data Sets

Obs    _Imputation_    _TYPE_     _NAME_       Intercept      RunTime     RunPulse

  1          1         PARMS                      86.544     -2.82231     -0.05873
  2          1         COV       Intercept       100.145     -0.53519     -0.55077
  3          1         COV       RunTime          -0.535      0.10774     -0.00345
  4          1         COV       RunPulse         -0.551     -0.00345      0.00343
  5          2         PARMS                      83.021     -3.00023     -0.02491
  6          2         COV       Intercept        79.032     -0.66765     -0.41918
  7          2         COV       RunTime          -0.668      0.11456     -0.00313
  8          2         COV       RunPulse         -0.419     -0.00313      0.00264
```

Figure 45.1. Parameter Estimates

The following statements combine the five sets of regression coefficients:

```
proc mianalyze data=outreg;
   modeleffects Intercept RunTime RunPulse;
run;
```



```
                              The MIANALYZE Procedure

                                 Model Information

                         Data Set                    WORK.OUTREG
                         Number of Imputations       5
```

Figure 45.2. Model Information Table

The "Model Information" table shown in Figure 45.2 lists the input data set(s) and the number of imputations.

```
                            The MIANALYZE Procedure

                    Multiple Imputation Variance Information

                     -----------------Variance-----------------
         Parameter         Between         Within          Total       DF

         Intercept       45.529229      76.543614     131.178689   23.059
         RunTime          0.019390       0.106220       0.129487   123.88
         RunPulse         0.001007       0.002537       0.003746   38.419

                   Multiple Imputation Variance Information

                              Relative       Fraction
                              Increase        Missing       Relative
             Parameter     in Variance    Information     Efficiency

             Intercept        0.713777       0.461277       0.915537
             RunTime          0.219051       0.192620       0.962905
             RunPulse         0.476384       0.355376       0.933641
```

Figure 45.3. Variance Information Table

The "Multiple Imputation Variance Information" table shown in Figure 45.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences. It also displays the degrees of freedom for the total variance, the relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each parameter estimate.

```
                            The MIANALYZE Procedure

                    Multiple Imputation Parameter Estimates

   Parameter        Estimate      Std Error    95% Confidence Limits        DF

   Intercept       90.837440      11.453327     67.14779     114.5271   23.059
   RunTime         -3.032870       0.359844     -3.74511      -2.3206   123.88
   RunPulse        -0.068578       0.061204     -0.19243       0.0553   38.419

                    Multiple Imputation Parameter Estimates

                    Parameter         Minimum        Maximum

                    Intercept       83.020730     100.839807
                    RunTime         -3.204426      -2.822311
                    RunPulse        -0.112840      -0.024910

                    Multiple Imputation Parameter Estimates

                                                  t for H0:
             Parameter          Theta0   Parameter=Theta0   Pr > |t|

             Intercept               0               7.93     <.0001
             RunTime                 0              -8.43     <.0001
             RunPulse                0              -1.12     0.2695
```

Figure 45.4. Multiple Imputation Parameter Estimates

The “Multiple Imputation Parameter Estimates” table shown in Figure 45.4 displays a combined estimate and standard error for each regression coefficient (parameter). Inferences are based on *t* distributions. The table displays a 95% confidence interval and a *t*-test with the associated *p*-value for the hypothesis that the parameter is equal to the value specified with the THETA0= option (in this case, zero by default). The minimum and maximum parameter estimates from the imputed data sets are also displayed.

Syntax

The following statements are available in PROC MIANALYZE.

PROC MIANALYZE < *options* > **;**
  BY *variables* **;**
  CLASS *variables* **;**
  MODELEFFECTS *effects* **;**
  < *label:* > **TEST** *equation1* < , . . . , < *equationk* >>< */options* >**;**
  STDERR *variables* **;**

The BY statement specifies groups in which separate analyses are performed.

The CLASS statement lists the classification variables in the EFFECT statement. Classification variables can be either character or numeric.

The required MODELEFFECTS statement lists the effects to be analyzed. The variables in the statement that are not specified in a CLASS statement are assumed to be continuous.

The STDERR statement lists the standard errors associated with the effects in the MODELEFFECTS statement when both parameter estimates and standard errors are saved as variable in the same DATA= data set. The STDERR statement can be used only when each effect in the MODELEFFECTS statement is a continuous variable by itself.

The TEST statement tests linear hypotheses about the parameters. An F statistic is used to test jointly the null hypothesis ($H_0\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$) specified in a single TEST statement. Several TEST statements can be used.

The PROC MIANALYZE and MODELEFFECTS statements are required for the MI procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MIANALYZE statement. The remaining statements are in alphabetical order.

PROC MIANALYZE Statement

PROC MIANALYZE < *options* > **;**

The following table summarizes the options in the PROC MIANALYZE statement.

Table 45.1. Summary of PROC MIANALYZE Options

Tasks	Options
Specify input data sets	
COV, CORR, or EST type data set	DATA=
parameter estimates and standard errors	DATA=
parameter estimates	PARMS=
parameter information	PARMINFO=
covariance matrices	COVB=
$(X'X)^{-1}$ matrices	XPXI=
Specify statistical analysis	
parameters under the null hypothesis	THETA0=
level for the confidence interval	ALPHA=
complete-data degrees of freedom	EDF=
Display Printed Output	
within-imputation covariance matrix	WCOV
between-imputation covariance matrix	BCOV
total covariance matrix	TCOV
multivariate inferences	MULT

The following are explanations of the options that can be used in the PROC MIANALYZE statement (in alphabetical order):

ALPHA=*p*

specifies that confidence limits are to be constructed for the parameter estimates with confidence level $100(1-p)\%$, where $0 < p < 1$. The default is p=0.05.

BCOV

displays the between-imputation covariance matrix.

COVB <**(EFFECTVAR=STACKING | ROWCOL)**> **=***SAS-data-set*

names an input SAS data set that contains covariance matrices of the parameter estimates from imputed data sets. If you provide a COVB= data set, you must also provide a PARMS= data set.

The EFFECTVAR= option identifies the variables for parameters displayed in the covariance matrix and is used only when the PARMINFO= option is not specified. The default is EFFECTVAR= STACKING.

See the “Input Data Sets” section on page 2620 for a detailed description of the COVB= option.

DATA=*SAS-data-set*

names an input SAS data set.

If the input DATA= data set is not a specially structured SAS data set, the data set contains both the parameter estimates and associated standard errors. The parameter estimates are specified in the MODELEFFECTS statement and the standard errors are specified in the STDERR statement.

If the data set is a specially structured input SAS data set, it must have a TYPE of EST, COV, or CORR that contains estimates from imputed data sets:

- TYPE=EST, the data set contains the parameter estimates and associated covariance matrices.
- TYPE=COV, the data set contains the sample means, sample sizes, and covariance matrices. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.
- TYPE=CORR, the data set contains the sample means, sample sizes, standard errors, and correlation matrices. The covariance matrices are computed from the correlation matrices and associated standard errors. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. See the "Input Data Sets" section on page 2620 for a detailed description of the input data sets.

EDF=*number*

specifies the complete-data degrees of freedom for the parameter estimates. This is used to compute an adjusted degrees of freedom for each parameter estimate. By default, EDF=∞ and the degrees of freedom for each parameter estimate is not adjusted.

MULT
MULTIVARIATE

requests multivariate inference for the parameters. It is based on Wald tests and is a generalization of the univariate inference. See the "Multivariate Inferences" section on page 2626 for a detailed description of the multivariate inference.

PARMINFO=*SAS-data-set*

names an input SAS data set that contains parameter information associated with variables PRM1, PRM2,..., and so on. These variables are used as variables for parameters in a COVB= data set. See the "Input Data Sets" section on page 2620 for a detailed description of the PARMINFO= option.

PARMS <**(CLASSVAR=** *ctype***)**> **=***SAS-data-set*

names an input SAS data set that contains parameter estimates computed from imputed data sets. When a COVB= data set is not specified, the input PARMS= data set also contains standard errors associated with these parameter estimates. If multivariate inference is requested, you must also provide a COVB= or XPXI= data set.

When the effects contain CLASS variables, the option CLASSVAR= *ctype* can be used to identify the associated CLASS variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL. The default is CLASSVAR= FULL. See the "Input Data Sets" section on page 2620 for a detailed description of the PARMS= option.

TCOV

displays the total covariance matrix derived by assuming that the population between-imputation and within-imputation covariance matrices are proportional to each other.

THETA0=*numbers*
MU0=*numbers*

specifies the parameter values $\boldsymbol{\theta}_0$ under the null hypothesis $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ in the *t* tests for location for the effects. If only one number $\boldsymbol{\theta}_0$ is specified, that number is used for all effects. If more than one number is specified, the specified numbers correspond to effects in the MODELEFFECTS statement in the order in which they appear in the statement. When an effect contains CLASS variables, the corresponding value is not used and the test is not performed.

WCOV

displays the within-imputation covariance matrices.

XPXI=*SAS-data-set*

names an input SAS data set that contains the $(X'X)^{-1}$ matrices associated with the parameter estimates computed from imputed data sets. If you provide an XPXI= data set, you must also provide a PARMS= data set. In this case, PROC MIANALYZE reads the standard errors of the estimates from the PARMS= data. The standard errors and $(X'X)^{-1}$ matrices are used to derive the covariance matrices.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MIANALYZE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MI procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement (Experimental)

CLASS *variables* **;**

The CLASS statement specifies the classification variables in the MODELEFFECTS statement. Classification variables can be either character or numeric.

Classification levels are determined from the formatted values of the CLASS variables. Refer to the chapter titled “The FORMAT Procedure” in the *SAS Procedures Guide*.

MODELEFFECTS Statement

MODELEFFECTS *effects* **;**

The MODELEFFECTS statement lists the effects in the data set to be analyzed. Each effect is a variable or a combination of variables, and is specified with a special notation using variable names and operators.

Each variable is either a classification (or CLASS) variable or a continuous variable. If a variable is not declared in the CLASS statement, it is assumed to be continuous. Crossing and nesting operators can be used in an effect to create crossed and nested effects.

One general form of an effect involving several variables is

$$X1 * X2 * A * B * C\ (\ D\ E\ )$$

where A, B, C, D, and E are CLASS variables and X1 and X2 are continuous variables.

When the input DATA= data set is not a specially structured SAS data set, you must also specify standard errors of the parameter estimates in a STDERR statement.

STDERR Statement

STDERR *variables* **;**

The STDERR statement lists standard errors associated with effects in the MODELEFFECTS statement, when the input DATA= data set contains both parameter estimates and standard errors as variables in the data set.

With the STDERR statement, only continuous effects are allowed in the MODELEFFECTS statement. The specified standard errors are corresponding to parameter estimates in the order in which they appear in the MODELEFFECTS statement.

For example, you can use the following MODELEFFECTS and STDERR statements to identify both the parameter estimates and associated standard errors in a SAS data set.

```
proc mianalyze;
   var y1-y3;
   stderr sy1-sy3;
run;
```

TEST Statement

< *label:* > **TEST** *equation1* < , . . . , < *equationk* >>< */options* > **;**

The TEST statement tests linear hypotheses about the parameters $\boldsymbol{\beta}$. An F test is used to test jointly the null hypotheses (H_0: $\mathbf{L}\boldsymbol{\beta} = \mathbf{c}$) specified in a single TEST statement.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\mathbf{c}$ vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output. You can submit multiple TEST statements. When a label is not included in a TEST statement, a label of "Test j" is used for the jth TEST statement.

The form of an *equation* is as follows:

term < ±*term* . . . > < = ±*term* < ±*term* . . . >>

where *term* is a parameter of the model, or a constant, or a constant times a parameter. When no equal sign appears, the expression is set to 0. Only parameters for regressor effects (continuous variables by themselves) are allowed.

For each TEST statement, PROC MIANALYZE displays a "Test Specification" table of the $\mathbf{L}$ matrix and the $\mathbf{c}$ vector. The procedure also displays a "Multiple Imputation Variance Information" table of the between-imputation, within-imputation, and total variances for combining complete-data inferences, and a "Multiple Imputation Parameter Estimates" table of a combined estimate and standard error for each linear component. The linear components are labeled TestPrm1, TestPrm2, ... in the tables.

The following code illustrates possible uses of the TEST statement:

```
proc mianalyze;
   var intercept a1 a2 a3;
   test1: test intercept + a2 = 0;
   test2: test intercept + a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

The first and second TEST statements are equivalent and correspond to the following specification

```
                            The MIANALYZE Procedure
                                  Test: test1

                              Test Specification

           -----------------------L Matrix-----------------------
Parameter     intercept            a1            a2            a3             C

TestPrm1       1.000000             0      1.000000             0             0
```

Figure 45.5. Test Specification for test1 and test2

The third and fourth TEST statements are also equivalent and correspond to the specification

```
                            The MIANALYZE Procedure
                                  Test: test3

                              Test Specification

           -----------------------L Matrix-----------------------
Parameter     intercept            a1            a2            a3             C

TestPrm1              0      1.000000     -1.000000             0             0
TestPrm2              0             0      1.000000     -1.000000             0
```

Figure 45.6. Test Specification for test3 and test4

The ALPHA= and EDF options specified in the PROC MIANALYZE statement are also applied to the TEST statement. You can specify the following options in the TEST statement after a slash(/).

BCOV
displays the between-imputation covariance matrix.

MULT
displays the multivariate inference for parameters.

TCOV
displays the total covariance matrix.

WCOV
displays the within-imputation covariance matrix.

For more information, see the "Testing Linear Hypotheses about the Parameters" section on page 2628.

Details

Input Data Sets

You specify input data sets based on the type of inference you requested. For univariate inference, you can use one of the following options:

- a DATA= data set, which provides both parameter estimates and the associated standard errors.
- a DATA= type EST, COV, or CORR data set, which provides both parameter estimates and the associated standard errors either explicitly (type CORR) or through the covariance matrix (type EST, COV).
- PARMS= data set, which provides both parameter estimates and the associated standard errors.

For multivariate inference, which includes the testing of linear hypotheses about parameters, you can use one of the following option combinations:

- a DATA= type EST, COV, or CORR data set, which provides parameter estimates and the associated covariance matrix either explicitly (type EST, COV) or through the correlation matrix and standard errors (type CORR) in a single data set.
- PARMS= and COVB= data sets, which provide parameter estimates in a PARMS= data set and the associated covariance matrix in a COVB= data set.
- PARMS=, COVB=, and PARMINFO= data sets, which provide parameter estimates in a PARMS= data set, the associated covariance matrix in a COVB= data set with variables named PRM1, PRM2, ..., and the effects associated with these variables in a PARMINFO= data set.
- PARMS= and XPXI= data sets, which provide parameter estimates and the associated standard errors in a PARMS= data set and the associated $(X'X)^{-1}$ matrix in an XPXI= data set.

The appropriate combination depends on the type of inference and the SAS procedure you used to create the data sets. For instance, if you used PROC REG to create an OUTEST= data set containing the parameter estimates and covariance matrix, you would use the DATA= option to read the OUTEST= data set.

When the input DATA= data set is not a specially structured SAS data set, each observation corresponds to an imputation and contains both parameter estimates and associated standard errors. For others, each input data set must contains the variable _Imputation_ to identify the imputation by number.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. Note that with a DATA= data set, each effect represents a continuous variable, only regressor effects (continuous variables by themselves) are allowed in the MODELEFFECTS statement.

DATA= SAS data set

The DATA= data set provides both parameter estimates and the associated standard errors computed from imputed data sets. Such data sets are typically created with an OUTPUT statement using procedures such as PROC MEANS and PROC UNIVARIATE.

The MIANALYZE procedure reads parameter estimates from observations with variables in the MODELEFFECTS statement, and standard errors for parameter estimates from observations with variables in the STDERR statement. The order of the variables for standard errors must match the order of the variables for parameter estimates.

DATA= type EST, COV, or CORR SAS data set

The specially structured DATA= data set provides both parameter estimates and the associated covariance matrix computed from imputed data sets. Such data sets are created by procedures such as PROC CORR (type COV, CORR) and PROC REG (type EST).

With TYPE=EST, the MIANALYZE procedure reads parameter estimates from observations with _TYPE_=‘PARM’, _TYPE_=‘PARMS’, _TYPE_=‘OLS’, or _TYPE_=‘FINAL’, and covariance matrices for parameter estimates from observations with _TYPE_=‘COV’ or _TYPE_=‘COVB’.

With TYPE=COV, the procedure reads sample means from observations with _TYPE_=‘MEAN’, sample size n from observations with _TYPE_=‘N’, and covariance matrices for variables from observations with _TYPE_=‘COV’.

With TYPE=CORR, the procedure reads sample means from observations with _TYPE_=‘MEAN’, sample size n from observations with _TYPE_=‘N’, correlation matrices for variables from observations with _TYPE_=‘CORR’, and standard errors for variables from observations with _TYPE_=‘STD’. The standard errors and correlation matrix are used to generate a covariance matrix for the variables.

Note that with TYPE=COV or CORR, each covariance matrix for the variables is divided by n to create the covariance matrix for the sample means.

PARMS <(CLASSVAR= ctype)> = data set

The PARMS= data set contains parameter estimates and associated standard errors computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement using procedures such as PROC GENMOD, PROC GLM, PROC LOGISTIC, and PROC MIXED.

The MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

When the effects contain CLASS variables, the option CLASSVAR= *ctype* can be used to identify associated CLASS variables when reading the CLASS levels from observations. The available types are FULL, LEVEL, and CLASSVAL. The default is CLASSVAR= FULL.

With CLASSVAR=FULL, the data set contains the CLASS variables explicitly. PROC MIANALYZE reads the CLASS levels from observations with their corresponding CLASS variables. PROC MIXED generates this type of tables.

With CLASSVAR=LEVEL, PROC MIANALYZE reads the classification levels for the effect from observations with variables Level1, Level2, and so on, where the variable Level1 contains the classification level for the first CLASS variable in the effect, the variable Level2 contains the classification level for the second CLASS variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC GENMOD generates this type of tables.

For example, with the following statements,

```
proc mianalyze parms(classvar=Level)= dataparm;
   class c1 c2 c3;
   modeleffects c2 c3(c2 c1);
run;
```

the variable Level1 has the classification level of the variable c2 for the effect c2. For the effect c3(c2 c1), the variable Level1 has the classification level of the variable c3, Level2 has the level of c1, and Level3 has the level of c2.

Similarly, with CLASSVAR=CLASSVAL, PROC MIANALYZE reads the classification levels for the effect from observations with variables ClassVal0, ClassVal1, and so on, where the variable ClassVal0 contains the classification level for the first CLASS variable in the effect, the variable ClassVal1 contains the classification level for the second CLASS variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC LOGISTIC generates this type of tables.

PARMS <(CLASSVAR= *ctype*)>= and COVB= data sets

The PARMS= data set contains parameter estimates and the COVB= data set contains associated covariance matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement using procedures such as PROC LOGISTIC, PROC MIXED, and PROC REG.

With a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

When the effects contain CLASS variables, the option CLASSVAR= *ctype* can be used to identify the associated CLASS variables when reading the CLASS levels from observations. The available types are FULL, LEVEL, and CLASSVAL, and are described in the "PARMS <(CLASSVAR= *ctype*)> = data set" section on page 2621. The default is CLASSVAR= FULL.

The option EFFECTVAR=*etype* identifies the variables for parameters displayed in the covariance matrix. The available types are STACKING and ROWCOL. The default is EFFECTVAR=STACKING.

With EFFECTVAR=STACKING, each parameter is displayed by stacking variables in the effect. Begin with the variables in the crossed list, followed by the continuous list, then followed by the nested list. Each CLASS variable is displayed with its CLASS level attached. PROC LOGISTIC generates this type of tables.

When each effect is a continuous variable by itself, each stacked parameter name reduces to the effect name. PROC REG generates this type of tables.

With EFFECTVAR=STACKING, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the stacked variables in a COVB= data set.

With EFFECTVAR=ROWCOL, parameters are displayed by the variables Col1, Col2, ... The parameter associated with the variable Col1 is identified by the observation with value 1 for the variable Row. The parameter associated with the variable Col2 is identified by the observation with value 2 for the variable Row. PROC MIXED generates this type of table.

With EFFECTVAR=ROWCOL, the MIANALYZE procedure reads the parameter indices from observations with the variable Row, the effect names from observations with the variable Parameter, Effect, Variable, Parm, or RowName, and covariance matrices from observations with the variables Col1, Col2, ... in a COVB= data set.

When the effects contain CLASS variables, the data set contains the CLASS variables explicitly and the MIANALYZE procedure also reads the CLASS levels from their corresponding CLASS variables.

PARMS <(CLASSVAR= *ctype*)> =, PARMINFO=, and COVB= data sets

The input PARMS= data set contains parameter estimates and the input COVB= data set contains associated covariance matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement using procedure such as PROC GENMOD.

With a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

When the effects contain CLASS variables, the option CLASSVAR= *ctype* can be used to identify the associated CLASS variables when reading the CLASS levels from observations. The available types are FULL, LEVEL, and CLASSVAL, and are described in the "PARMS <(CLASSVAR= *ctype*)> = data set" section on page 2621. The default is CLASSVAR= FULL.

With a COVB= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName and covariance matrices from observations with variables Prm1, Prm2, and so on.

The parameters associated with the variables Prm1, Prm2, ... are identified in the PARMINFO= data set. PROC MIANALYZE reads the parameter names from observations with the variable Parameter and the corresponding effect from observations with the variable Effect. When the effects contain CLASS variables, the data set contains the CLASS variables explicitly and the MIANALYZE procedure also reads the CLASS levels from observations with their corresponding CLASS variables.

PARMS= and XPXI= data sets

The input PARMS= data set contains parameter estimates and the input XPXI= data set contains associated $(X'X)^{-1}$ matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement using a procedure such as PROC GLM.

With a PARMS= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

With a XPXI= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter and $(X'X)^{-1}$ matrices from observations with the parameter variables in the data set.

Note that this combination can only be used when each effect is a continuous variable by itself.

Combining Inferences from Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose that $\hat{Q}_i$ and $\hat{U}_i$ are the point and variance estimates from the ith imputed data set, i=1, 2, ..., m. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m} \sum_{i=1}^{m} \hat{Q}_i$$

Suppose that $\overline{W}$ is the within-imputation variance, which is the average of the m complete-data estimates:

$$\overline{W} = \frac{1}{m} \sum_{i=1}^{m} \hat{W}_i$$

and B be the between-imputation variance

$$B = \frac{1}{m-1} \sum_{i=1}^{m} (\hat{Q}_i - \overline{Q})^2$$

Then the variance estimate associated with $\overline{Q}$ is the total variance (Rubin 1987)

$$T = \overline{W} + (1 + \frac{1}{m})B$$

The statistic $(Q - \overline{Q})T^{-(1/2)}$ is approximately distributed as t with v_m degrees of freedom (Rubin 1987), where

$$v_m = (m-1)\left[1 + \frac{\overline{W}}{(1+m^{-1})B}\right]^2$$

The degrees of freedom v_m depends on m and the ratio

$$r = \frac{(1+m^{-1})B}{\overline{W}}$$

The ratio r is called the relative increase in variance due to nonresponse (Rubin 1987). When there is no missing information about Q, the values of r and B are both zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution of $(Q - \overline{Q})T^{-(1/2)}$ will be approximately normal.

Another useful statistic is the fraction of missing information about Q:

$$\hat{\lambda} = \frac{r + 2/(v_m + 3)}{r + 1}$$

Both statistics r and λ are helpful diagnostics for assessing how the missing data contribute to the uncertainty about Q.

When the complete-data degrees of freedom v_0 is small, and there is only a modest proportion of missing data, the computed degrees of freedom, v_m, can be much larger than v_0, which is inappropriate. For example, with $m = 5$ and $r = 10\%$, the computed degrees of freedom $v_m = 484$, which is inappropriate for data sets with complete-data degrees of freedom less than 484.

Barnard and Rubin (1999) recommend the use of an adjusted degrees of freedom

$$v_m^* = \left[\frac{1}{v_m} + \frac{1}{\hat{v}_{obs}}\right]^{-1}$$

where $\hat{v}_{obs} = (1-\gamma)\, v_0(v_0+1)/(v_0+3)$ and $\gamma = (1+m^{-1})B/T$.

If you specify the complete-data degrees of freedom v_0 with the EDF= option, the MIANALYZE procedure uses the adjusted degrees of freedom, v_m^*, for inference. Otherwise, the degrees of freedom v_m is used.

Multiple Imputation Efficiency

The relative efficiency (RE) of using the finite m imputation estimator, rather than using an infinite number for the fully efficient imputation, in units of variance, is approximately a function of m and λ (Rubin 1987, p. 114).

$$RE = (1 + \frac{\lambda}{m})^{-1}$$

The following table shows relative efficiencies with different values of m and λ.

Table 45.2. Relative Efficiency

	λ				
m	10%	20%	30%	50%	70%
3	0.9677	0.9375	0.9091	0.8571	0.8108
5	0.9804	0.9615	0.9434	0.9091	0.8772
10	0.9901	0.9804	0.9709	0.9524	0.9346
20	0.9950	0.9901	0.9852	0.9756	0.9662

The table shows that for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of m imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Multivariate Inferences

Multivariate inference based on Wald tests can be done with m imputed data sets. The approach is a generalization of the approach taken in the univariate case (Rubin 1987, p. 137; Schafer 1997, p. 113). Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{U}}_i$ are the point and covariance matrix estimates for a p-dimensional parameter $\mathbf{Q}$ (such as a multivariate mean) from the ith imputed data set, i=1, 2, ..., m. Then the combined point estimate for $\mathbf{Q}$ from the multiple imputation is the average of the m complete-data estimates:

$$\overline{\mathbf{Q}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{Q}}_i$$

Suppose that $\overline{\mathbf{U}}$ is the within-imputation covariance matrix, which is the average of the m complete-data estimates

$$\overline{\mathbf{W}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{W}}_i$$

and suppose that $\mathbf{B}$ is the between-imputation covariance matrix

$$\mathbf{B} = \frac{1}{m-1} \sum_{i=1}^{m} (\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})(\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})'$$

Then the covariance matrix associated with $\overline{\mathbf{Q}}$ is the total covariance matrix

$$\mathbf{T}_0 = \overline{\mathbf{W}} + (1 + \frac{1}{m})\mathbf{B}$$

The natural multivariate extension of the t statistic used in the univariate case is the F statistic

$$F_0 = (\mathbf{Q} - \overline{\mathbf{Q}})'\mathbf{T}_0^{-1}(\mathbf{Q} - \overline{\mathbf{Q}})$$

with degrees of freedom p and

$$v = (m-1)(1+1/r)^2$$

where

$$r = (1 + \frac{1}{m})trace(\mathbf{B}\overline{\mathbf{W}}^{-1})/p$$

is an average relative increase in variance due to nonresponse (Rubin 1987, p. 137; Schafer 1997, p. 114).

However, the reference distribution of the statistic F_0 is not easily derived. Especially for small m, the between-imputation covariance matrix $\mathbf{B}$ is unstable and does not have full rank for $m \leq p$ (Schafer 1997, p. 113).

One solution is to make an additional assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other (Schafer 1997, p. 113). This assumption implies that the fractions of missing information for all components of $\mathbf{Q}$ are equal. Under this assumption, a more stable estimate of the total covariance matrix is

$$\mathbf{T} = (1+r)\overline{\mathbf{W}}$$

With the total covariance matrix $\mathbf{T}$, the F statistic (Rubin 1987, p. 137)

$$F = (\mathbf{Q} - \overline{\mathbf{Q}})'\mathbf{T}^{-1}(\mathbf{Q} - \overline{\mathbf{Q}})/p$$

has an F distribution with degrees of freedom p and v_1, where

$$v_1 = \frac{1}{2}(p+1)(m-1)(1+\frac{1}{r})^2$$

For $t = p(m-1) \leq 4$, PROC MIANALYZE uses the degrees of freedom v_1 in the analysis. For $t = p(m-1) > 4$, PROC MIANALYZE uses v_2, a better approximation of the degrees of freedom given by Li, Raghunathan, and Rubin (1991).

$$v_2 = 4 + (t-4)\left[1 + \frac{1}{r}(1 - \frac{2}{t})\right]^2$$

Testing Linear Hypotheses about the Parameters

Linear hypotheses for parameters $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0 \colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\mathbf{c}$ is a vector of constants.

Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{U}}_i$ are the point and covariance matrix estimates for a p-dimensional parameter $\mathbf{Q}$ from the $i\text{th}$ imputed data set, i=1, 2, ..., m. Then for a given matrix $\mathbf{L}$, the point and covariance matrix estimates for the linear functions $\mathbf{LQ}$ in the $i\text{th}$ imputed data set are

$$\mathbf{L}\hat{\mathbf{Q}}_i$$

$$\mathbf{L}\hat{\mathbf{U}}_i\mathbf{L}'$$

The inferences described in the "Combining Inferences from Imputed Data Sets" section on page 2624 and the "Multivariate Inferences" section on page 2626 are applied to these linear estimates for testing the null hypothesis $H_0 \colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$.

For each TEST statement, the "Test Specification" table displays the $\mathbf{L}$ matrix and the $\mathbf{c}$ vector, the "Multiple Imputation Variance Information" table displays the between-imputation, within-imputation, and total variances for combining complete-data inferences, the "Multiple Imputation Parameter Estimates" table displays a combined estimate and standard error for each linear component.

With the WCOV and BCOV options in the TEST statement, the procedure displays the within-imputation and between-imputation covariance matrices, respectively.

With the TCOV option, the procedure displays the total covariance matrix derived under the assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other.

With the MULT option in the TEST statement, the "Multiple Imputation Multivariate Inference" table displays an F test for the null hypothesis $\mathbf{L}\boldsymbol{\beta} = \mathbf{c}$ of the linear components.

Examples of the Complete-Data Inferences

For a given parameter of interest, it is not always possible to compute the estimate and associated covariance matrix directly from a SAS procedure. This section describes examples of parameters with their estimates and associated covariance matrices, which provide the input to the MIANALYZE procedure. Some are straightforward, and others require special techniques.

Means

For a population mean vector $\boldsymbol{\mu}$, the usual estimate is the sample mean vector

$$\overline{\mathbf{y}} = \frac{1}{n} \sum \mathbf{y}_i$$

A variance estimate for $\overline{\mathbf{y}}$ is $\frac{1}{n}\mathbf{S}$, where $\mathbf{S}$ is the sample covariance matrix

$$\mathbf{S} = \frac{1}{n-1} \sum (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})'$$

These statistics can be computed from a procedure such as CORR. This approach is illustrated in Example 45.2.

Regression Coefficients

Many SAS procedures are available for regression analysis. Among them, PROC REG provides the most general analysis capabilities, and others like PROC LOGISTIC and PROC MIXED provide more specialized analyses.

Some regression procedures, such as REG and LOGISTIC, create an EST type data set that contains both the parameter estimates for the regression coefficients and their associated covariance matrix. You can read an EST type data set in the MIANALYZE procedure with the DATA= option. This approach is illustrated in Example 45.3.

Other procedures, such as GLM, MIXED, and GENMOD, do not generate EST type data sets for regression coefficients. For PROC MIXED and PROC GENMOD, you can use ODS OUTPUT statement to save parameter estimates in a data set and the associated covariance matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and COVB= options, respectively. This approach is illustrated in Example 45.4 for PROC MIXED and in Example 45.5 for PROC GENMOD.

PROC GLM does not display tables for covariance matrices. However, you can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set and the associated $(X'X)^{-1}$ matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and XPXI= options, respectively. This approach is illustrated in Example 45.6.

For univariate inference, only parameter estimates and associated standard errors are needed. You can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set. These data set is then read in the MIANALYZE procedure with the PARMS= option. This approach is illustrated in Example 45.4.

Correlation Coefficients

For the population correlation coefficient ρ, a point estimate is the sample correlation coefficient r. However, for nonzero ρ, the distribution of r is skewed.

The distribution of r can be normalized through Fisher's z transformation

$$z(r) = \frac{1}{2} \log \left(\frac{1+r}{1-r} \right)$$

$z(r)$ is approximately normally distributed with mean $z(\rho)$ and variance $1/(n-3)$.

With a point estimate $\hat{z}$ and an approximate 95% confidence interval (z_1, z_2) for $z(\rho)$, a point estimate $\hat{r}$ and a 95% confidence interval (r_1, r_2) for ρ can be obtained by applying the inverse transformation

$$r = tanh(z) = \frac{e^{2z}-1}{e^{2z}+1}$$

to $z = \hat{z}, z_1$, and z_2.

This approach is illustrated in Example 45.10.

Ratios of Variable Means

For the ratio μ_1/μ_2 of means for variables Y_1 and Y_2, the point estimate is $\overline{y}_1/\overline{y}_2$, the ratio of the sample means. The Taylor expansion and delta method can be applied to the function y_1/y_2 to obtain the variance estimate (Schafer 1997, p. 196)

$$\frac{1}{n} \left[\left(\frac{\overline{y}_1}{\overline{y}_2^2} \right)^2 s_{22} - 2 \left(\frac{\overline{y}_1}{y_2^2} \right) \left(\frac{1}{\overline{y}_2} \right) s_{12} + \left(\frac{1}{\overline{y}_2} \right)^2 s_{11} \right]$$

where s_{11} and s_{22} are the sample variances of Y_1 and Y_2, respectively, and s_{12} is the sample covariance between Y_1 and Y_2.

A ratio of sample means will be approximately unbiased and normally distributed if the coefficient of variation of the denominator (the standard error for the mean divided by the estimated mean) is 10% or less (Cochran 1977, p. 166; Schafer 1997, p. 196).

ODS Table Names

PROC MIANALYZE assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 45.3. ODS Tables Produced in PROC MIANALYZE

ODS Table Name	Description	Statement	Option
BCov	Between-imputation covariance matrix		BCOV
ModelInfo	Model information		
MultStat	Multivariate inference		MULT
ParameterEstimates	Parameter estimates		
TCov	Total covariance matrix		TCOV
TestBCov	Between-imputation covariance matrix for $\mathbf{L}\beta$	TEST	BCOV
TestMultStat	Multivariate inference for $\mathbf{L}\beta$	TEST	MULT
TestParameterEstimates	Parameter estimates for $\mathbf{L}\beta$	TEST	
TestSpec	Test specification, $\mathbf{L}$ and $\mathbf{c}$	TEST	
TestTCov	Total covariance matrix for $\mathbf{L}\beta$	TEST	TCOV
TestVarianceInfo	Variance information for $\mathbf{L}\beta$	TEST	
TestWCov	Within-imputation covariance matrix for $\mathbf{L}\beta$	TEST	WCOV
VarianceInfo	Variance information		
WCov	Within-imputation covariance matrix		WCOV

Examples

The following statements generate five imputed data sets to be used in this section. The data set FitMiss was created in the section "Getting Started" on page 2610. See "The MI Procedure" chapter for details concerning the MI procedure.

```
proc mi data=FitMiss seed=3237851 noprint out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

The Fish data described in the STEPDISC procedure are measurements of 159 fish of seven species caught in Finland's lake Laengelmavesi. For each fish, the length, height, and width are measured. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail (Length1), from the nose to the notch of its tail (Length2), and from the nose to the end of its tail (Length3). See Chapter 67, "The STEPDISC Procedure," for more information.

The Fish2 data set is constructed from the Fish data set and contains two species of fish. Some values have been set to missing and the resulting data set has a monotone missing pattern in variables Length3, Height, Width, and Species. Note that some values of the variable Species have also been altered in the data set.

The following statements create the Fish2 data set. It contains the first two species of fish in the Fish data set.

```
/*-------- Fishes of Species Bream and Parkki Pike --------*/
 data Fish2 (drop=HtPct WidthPct);
 title 'Fish Measurement Data';
 input Species $ Length3 HtPct WidthPct @@;
 Height= HtPct*Length3/100;
 Width= WidthPct*Length3/100;
 datalines;
 Gp1  30.0 38.4 13.4   Gp1  31.2 40.0 13.8   Gp1  31.1 39.8 15.1
   .  33.5 38.0   .      .  34.0 36.6 15.1   Gp1  34.7 39.2 14.2
 Gp1  34.5 41.1 15.3   Gp1  35.0 36.2 13.4   Gp1  35.1 39.9 13.8
   .  36.2 39.3 13.7   Gp1  36.2 39.4 14.1     .  36.2 39.7 13.3
 Gp1  36.4 37.8 12.0     .  37.3 37.3 13.6   Gp1  37.2 40.2 13.9
 Gp1  37.2 41.5 15.0   Gp1  38.3 38.8 13.8   Gp1  38.5 38.8 13.5
 Gp1  38.6 40.5 13.3   Gp1  38.7 37.4 14.8   Gp1  39.5 38.3 14.1
 Gp1  39.2 40.8 13.7     .  39.7 39.1   .    Gp1  40.6 38.1 15.1
 Gp1  40.5 40.1 13.8   Gp1  40.9 40.0 14.8   Gp1  40.6 40.3 15.0
 Gp1  41.5 39.8 14.1   Gp2  41.6 40.6 14.9   Gp1  42.6 44.5 15.5
 Gp1  44.1 40.9 14.3   Gp1  44.0 41.1 14.3   Gp1  45.3 41.4 14.9
 Gp1  45.9 40.6 14.7   Gp1  46.5 37.9 13.7
 Gp2  16.2 25.6 14.0   Gp2  20.3 26.1 13.9   Gp2  21.2 26.3 13.7
 Gp2  22.2 25.3 14.3   Gp2  22.2 28.0 16.1   Gp2  22.8 28.4 14.7
 Gp2  23.1 26.7 14.7     .  23.7 25.8 13.9   Gp2  24.7 23.5 15.2
 Gp1  24.3 27.3 14.6   Gp2  25.3 27.8 15.1   Gp2  25.0 26.2 13.3
 Gp2  25.0 25.6 15.2   Gp2  27.2 27.7 14.1   Gp2  26.7 25.9 13.6
   .  26.8 27.6 15.4   Gp2  27.9 25.4 14.0   Gp2  29.2 30.4 15.4
 Gp2  30.6 28.0 15.6   Gp2  35.0 27.1 15.3
 ;
```

The following statements generate five imputed data sets to be used in this section. The regression method is used to impute missing values in the variable Width and the discriminant function method is used to impute the variable Species.

```
proc mi data=Fish2 seed=1305417 out=outfish;
   class Species;
   monotone reg (Width)
            discrim( Species= Length3 Height Width);
   var Length3 Height Width Species;
run;
```

Examples 1-6 use different input option combinations to combine parameter estimates computed from different procedures, Examples 7-8 combine parameter estimates with CLASS variables, Example 9 shows the use of a TEST statement, and Example 10 combines statistics that are not directly derived from procedures.

Example 45.1. Reading Means and Standard Errors from Variables in a DATA= Data Set

This example creates an ordinary SAS data set that contains sample means and standard errors computed from imputed data sets. These estimates are then combined to generate valid univariate inferences about the population means.

The following statements use the UNIVARIATE procedure to generate sample means and standard errors for the variables in each imputed data set.

```
proc univariate data=outmi noprint;
   var Oxygen RunTime RunPulse;
   output out=outuni mean=Oxygen RunTime RunPulse
                     stderr=SOxygen SRunTime SRunPulse;
   by _Imputation_;
run;
```

The following statements display the output data set from PROC UNIVARIATE in Output 45.1.1:

```
proc print data=outuni;
   title 'UNIVARIATE Means and Standard Errors';
run;
```

Output 45.1.1. UNIVARIATE Output Data Set

```
                  UNIVARIATE Means and Standard Errors

                                                Run                  SRun      SRun
Obs    _Imputation_     Oxygen    RunTime     Pulse     SOxygen      Time     Pulse

 1            1        47.0120    10.4441    171.216    0.95984    0.28520   1.59910
 2            2        47.2407    10.5040    171.244    0.93540    0.26661   1.75638
 3            3        47.4995    10.5922    171.909    1.00766    0.26302   1.85795
 4            4        47.1485    10.5279    171.146    0.95439    0.26405   1.75011
 5            5        47.0042    10.4913    172.072    0.96528    0.27275   1.84807
```

The following statements combine the means and standard errors from imputed data sets, The EDF= option requests that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outuni edf=30;
   modeleffects Oxygen RunTime RunPulse;
   stderr SOxygen SRunTime SRunPulse;
run;
```

Output 45.1.2. Multiple Imputation Variance Information

```
                    The MIANALYZE Procedure

                       Model Information

             Data Set                  WORK.OUTUNI
             Number of Imputations     5


            Multiple Imputation Variance Information

             -----------------Variance-----------------
Parameter         Between         Within          Total       DF

Oxygen           0.041478       0.930853       0.980626   26.298
RunTime          0.002948       0.073142       0.076679   26.503
RunPulse         0.191086       3.114442       3.343744   25.463

           Multiple Imputation Variance Information

                     Relative       Fraction
                     Increase        Missing       Relative
    Parameter     in Variance    Information     Efficiency

    Oxygen           0.053471       0.051977       0.989712
    RunTime          0.048365       0.047147       0.990659
    RunPulse         0.073626       0.070759       0.986046
```

The “Model Information” table shown in Output 45.1.2 lists the input data set(s) and the number of imputations.

The “Multiple Imputation Variance Information” table shown in Output 45.1.2 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each imputed variable are also displayed. A detailed description of these statistics is provided in the “Combining Inferences from Imputed Data Sets” section on page 2624 and the “Multiple Imputation Efficiency” section on page 2626.

Output 45.1.3. Multiple Imputation Parameter Estimates

```
                           The MIANALYZE Procedure

                   Multiple Imputation Parameter Estimates

Parameter        Estimate      Std Error    95% Confidence Limits        DF

Oxygen          47.180993       0.990266      45.1466      49.2154   26.298
RunTime         10.511906       0.276910       9.9432      11.0806   26.503
RunPulse       171.517500       1.828591     167.7549     175.2801   25.463

                   Multiple Imputation Parameter Estimates

                   Parameter         Minimum        Maximum

                   Oxygen          47.004201      47.499541
                   RunTime         10.444149      10.592244
                   RunPulse       171.146171     172.071730

                   Multiple Imputation Parameter Estimates

                                            t for H0:
          Parameter          Theta0   Parameter=Theta0   Pr > |t|

          Oxygen                  0              47.64     <.0001
          RunTime                 0              37.96     <.0001
          RunPulse                0              93.80     <.0001
```

The “Multiple Imputation Parameter Estimates” table shown in Output 45.1.3 displays the estimated mean and corresponding standard error for each variable. The table also displays a 95% confidence interval for the mean and a *t* statistic with the associated *p*-value for testing the hypothesis that the mean is equal to the value specified. You can use the THETA0= option to specify the value for the null hypothesis, which is zero by default. The table also displays the minimum and maximum parameter estimates from the imputed data sets.

Note that the results in this example could also have been obtained with the MI procedure.

Example 45.2. Reading Means and Covariance Matrices from a DATA= COV Data Set

This example creates a COV type data set that contains sample means and covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the population means.

The following statements use the CORR procedure to generate sample means and a covariance matrix for the variables in each imputed data set.

```
proc corr data=outmi cov nocorr noprint out=outcov(type=cov);
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display sample means and covariance matrices from the first two imputed data sets in Output 45.2.1.

```
proc print data=outcov(obs=12);
   title 'CORR Means and Covariance Matrices'
         ' (First Two Imputations)';
run;
```

Output 45.2.1. COV Data Set

```
          CORR Means and Covariance Matrices (First Two Imputations)

Obs    _Imputation_    _TYPE_     _NAME_       Oxygen      RunTime     RunPulse

  1          1          COV      Oxygen       28.5603      -7.2652      -11.812
  2          1          COV      RunTime      -7.2652       2.5214        2.536
  3          1          COV      RunPulse    -11.8121       2.5357       79.271
  4          1          MEAN                  47.0120      10.4441      171.216
  5          1          STD                    5.3442       1.5879        8.903
  6          1          N                     31.0000      31.0000       31.000
  7          2          COV      Oxygen       27.1240      -6.6761      -10.217
  8          2          COV      RunTime      -6.6761       2.2035        2.611
  9          2          COV      RunPulse    -10.2170       2.6114       95.631
 10          2          MEAN                  47.2407      10.5040      171.244
 11          2          STD                    5.2081       1.4844        9.779
 12          2          N                     31.0000      31.0000       31.000
```

Note that the covariance matrices in the data set outcov are estimated covariance matrices of variables, $V(\mathbf{y})$. The estimated covariance matrix of the sample means is $V(\bar{\mathbf{y}}) = V(\mathbf{y})/n$, where n is the sample size, and is not the same as an estimated covariance matrix for variables.

The following statements combine the results for the imputed data sets, and derive both univariate and multivariate inferences about the means. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outcov edf=30 wcov bcov tcov mult;
   modeleffects Oxygen RunTime RunPulse;
run;
```

The “Multiple Imputation Variance Information” and “Multiple Imputation Parameter Estimates” tables display the same results as in Output 45.1.2 and Output 45.1.3 in Example 45.1.

With the WCOV, BCOV, and TCOV options, the procedure displays the between-imputation covariance matrix, within-imputation covariance matrix, and total covariance matrix assuming that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix in Output 45.2.2.

Output 45.2.2. Covariance Matrices

```
                        The MIANALYZE Procedure

                  Within-Imputation Covariance Matrix

                         Oxygen            RunTime           RunPulse

          Oxygen        0.930852655      -0.226506411      -0.461022083
          RunTime      -0.226506411       0.073141598       0.080316017
          RunPulse     -0.461022083       0.080316017       3.114441784


                  Between-Imputation Covariance Matrix

                         Oxygen            RunTime           RunPulse

          Oxygen       0.0414778123      0.0099248946      0.0183701754
          RunTime      0.0099248946      0.0029478891      0.0091684769
          RunPulse     0.0183701754      0.0091684769      0.1910855259


                        Total Covariance Matrix

                         Oxygen            RunTime           RunPulse

          Oxygen        1.202882661      -0.292700068      -0.595750001
          RunTime      -0.292700068       0.094516313       0.103787365
          RunPulse     -0.595750001       0.103787365       4.024598310
```

With the MULT option, the procedure assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the parameters taken jointly.

Output 45.2.3. Multiple Imputation Multivariate Inference

```
                        The MIANALYZE Procedure

               Multiple Imputation Multivariate Inference
          Assuming Proportionality of Between/Within Covariance Matrices

           Avg Relative
               Increase                              F for H0:
            in Variance   Num DF   Den DF   Parameter=Theta0     Pr > F

               0.292237        3   122.68            12519.7     <.0001
```

The “Multiple Imputation Multivariate Inference” table displayed in Output 45.2.3 shows a significant p-value for the null hypothesis that the population means are all equal to zero.

Example 45.3. Reading Regression Results from a DATA= EST Data Set

This example creates an EST type data set that contains regression coefficients and their corresponding covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model.

The following statements use the REG procedure to generate regression coefficients:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display regression coefficients and their covariance matrices from the first two imputed data sets in Output 45.3.1.

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'REG Model Coefficients and Covariance matrices'
         ' (First Two Imputations)';
run;
```

Output 45.3.1. EST Type Data Set

```
    REG Model Coefficients and Covariance matrices (First Two Imputations)

Obs    _Imputation_    _TYPE_     _NAME_      Intercept     RunTime    RunPulse

  1          1         PARMS                     86.544    -2.82231    -0.05873
  2          1         COV       Intercept      100.145    -0.53519    -0.55077
  3          1         COV       RunTime         -0.535     0.10774    -0.00345
  4          1         COV       RunPulse        -0.551    -0.00345     0.00343
  5          2         PARMS                     83.021    -3.00023    -0.02491
  6          2         COV       Intercept       79.032    -0.66765    -0.41918
  7          2         COV       RunTime         -0.668     0.11456    -0.00313
  8          2         COV       RunPulse        -0.419    -0.00313     0.00264
```

The following statements combine the results for the imputed data sets. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For a regression model with three independent variables (including the Intercept) and 31 observations, the complete-data error degrees of freedom is 28.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Output 45.3.2. Multiple Imputation Variance Information

```
                         The MIANALYZE Procedure

                  Multiple Imputation Variance Information

                   -----------------Variance----------------
      Parameter          Between          Within           Total       DF

      Intercept        45.529229       76.543614      131.178689   9.1917
      RunTime           0.019390        0.106220        0.129487   18.311
      RunPulse          0.001007        0.002537        0.003746   12.137

                 Multiple Imputation Variance Information

                           Relative       Fraction
                           Increase        Missing        Relative
          Parameter     in Variance    Information      Efficiency

          Intercept        0.713777       0.461277        0.915537
          RunTime          0.219051       0.192620        0.962905
          RunPulse         0.476384       0.355376        0.933641
```

The “Multiple Imputation Variance Information” table shown in Output 45.3.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 45.3.3. Multiple Imputation Parameter Estimates

```
                         The MIANALYZE Procedure

                  Multiple Imputation Parameter Estimates

Parameter         Estimate      Std Error    95% Confidence Limits        DF

Intercept        90.837440      11.453327     65.01034     116.6645   9.1917
RunTime          -3.032870       0.359844     -3.78795      -2.2778   18.311
RunPulse         -0.068578       0.061204     -0.20176       0.0646   12.137

                  Multiple Imputation Parameter Estimates

                  Parameter          Minimum         Maximum

                  Intercept        83.020730      100.839807
                  RunTime          -3.204426       -2.822311
                  RunPulse         -0.112840       -0.024910

                  Multiple Imputation Parameter Estimates

                                          t for H0:
          Parameter          Theta0   Parameter=Theta0   Pr > |t|

          Intercept               0               7.93     <.0001
          RunTime                 0              -8.43     <.0001
          RunPulse                0              -1.12     0.2842
```

The “Multiple Imputation Parameter Estimates” table shown in Output 45.3.3 displays the estimated mean and standard error of the regression coefficients. The inferences are based on the *t* distribution. The table also displays a 95% mean confidence interval and a *t* test with the associated *p*-value for the hypothesis that the regression coefficient is equal to zero. Since the *p*-value for RunPulse is 0.1597, this variable can be removed from the regression model.

Example 45.4. Reading Mixed Model Results from PARMS= and COVB= Data Sets

This example creates data sets containing parameter estimates and covariance matrices computed by a mixed model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the parameters.

The following PROC MIXED statements generate the fixed-effect parameter estimates and covariance matrix for each imputed data set:

```
proc mixed data=outmi;
   model Oxygen= RunTime RunPulse RunTime*RunPulse/solution covb;
   by _Imputation_;
   ods output SolutionF=mixparms CovB=mixcovb;
run;
```

The following statements display parameter estimates from the first two imputed data sets in Output 45.4.1.

```
proc print data=mixparms (obs=8);
   var _Imputation_ Effect Estimate StdErr;
   title ’MIXED Model Coefficients (First Two Imputations)’;
run;
```

Output 45.4.1. PROC MIXED Model Coefficients

```
                MIXED Model Coefficients (First Two Imputations)

        Obs    _Imputation_    Effect                Estimate      StdErr

          1          1         Intercept               148.09     81.5231
          2          1         RunTime                -8.8115      7.8794
          3          1         RunPulse               -0.4123      0.4684
          4          1         RunTime*RunPulse       0.03437     0.04517
          5          2         Intercept              64.3607     64.6034
          6          2         RunTime                -1.1270      6.4307
          7          2         RunPulse               0.08160      0.3688
          8          2         RunTime*RunPulse      -0.01069     0.03664
```

The following statements display the covariance matrices associated with the parameter estimates from the first two imputed data sets in Output 45.4.2. Note that the variables Col1, Col2, Col3, and Col4 are used to identify the effects Intercept, RunTime, RunPulse, and RunTime*RunPulse through the variable Row.

```
proc print data=mixcovb (obs=8);
  var _Imputation_ Row Effect Col1 Col2 Col3 Col4;
  title 'Covariance Matrices (First Two Imputations)';
run;
```

Output 45.4.2. PROC MIXED Covariance Matrices

```
                 Covariance Matrices (First Two Imputations)

 Obs _Imputation_  Row Effect                Col1     Col2     Col3     Col4

   1       1         1 Intercept         6646.01  -637.40 -38.1515   3.6542
   2       1         2 RunTime           -637.40  62.0842   3.6548  -0.3556
   3       1         3 RunPulse         -38.1515   3.6548   0.2194 -0.02099
   4       1         4 RunTime*RunPulse   3.6542  -0.3556 -0.02099 0.002040
   5       2         1 Intercept         4173.59  -411.46 -23.7889   2.3441
   6       2         2 RunTime           -411.46  41.3545   2.3414  -0.2353
   7       2         3 RunPulse         -23.7889   2.3414   0.1360 -0.01338
   8       2         4 RunTime*RunPulse   2.3441  -0.2353 -0.01338 0.001343
```

For univariate inference, only parameter estimates and their associated standard errors are needed. The following statements use the MIANALYZE procedure with the input PARMS= data set to produce univariate results.

```
proc mianalyze parms=mixparms edf=28;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

Output 45.4.3. Multiple Imputation Variance Information

```
                           The MIANALYZE Procedure

                  Multiple Imputation Variance Information

                     -----------------Variance-----------------
  Parameter                 Between          Within           Total       DF

  Intercept             1972.654530     4771.948777     7139.134213    11.82
  RunTime                 14.712602       45.549686       63.204808   13.797
  RunPulse                 0.062941        0.156717        0.232247   12.046
  RunTime*RunPulse         0.000470        0.001490        0.002055   13.983

                   Multiple Imputation Variance Information

                              Relative        Fraction
                              Increase         Missing        Relative
       Parameter           in Variance     Information      Efficiency

       Intercept              0.496063        0.365524        0.931875
       RunTime                0.387601        0.305893        0.942348
       RunPulse               0.481948        0.358274        0.933136
       RunTime*RunPulse       0.378863        0.300674        0.943276
```

The “Multiple Imputation Variance Information” table shown in Output 45.4.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 45.4.4. Multiple Imputation Parameter Estimates

```
                             The MIANALYZE Procedure

                     Multiple Imputation Parameter Estimates

Parameter              Estimate      Std Error    95% Confidence Limits        DF

Intercept            136.071356      84.493397     -48.3352     320.4779    11.82
RunTime               -7.457186       7.950145     -24.5322       9.6178   13.797
RunPulse              -0.328104       0.481920      -1.3777       0.7215   12.046
RunTime*RunPulse       0.025364       0.045328      -0.0719       0.1226   13.983

                     Multiple Imputation Parameter Estimates

                 Parameter               Minimum        Maximum

                 Intercept             64.360719     186.549814
                 RunTime              -11.514341      -1.127010
                 RunPulse              -0.602162       0.081597
                 RunTime*RunPulse      -0.010690       0.047429

                     Multiple Imputation Parameter Estimates

                                                     t for H0:
          Parameter                Theta0   Parameter=Theta0   Pr > |t|

          Intercept                     0               1.61     0.1337
          RunTime                       0              -0.94     0.3644
          RunPulse                      0              -0.68     0.5089
          RunTime*RunPulse              0               0.56     0.5846
```

The “Multiple Imputation Parameter Estimates” table shown in Output 45.4.4 displays the estimated mean and standard error of the regression coefficients.

Since each covariance matrix contains variables Row, Col1, Col2, Col3, and Col4 for parameters, the EFFECTVAR=ROWCOL option is needed when specifying the COVB= option. The following statements illustrate the use of the MIANALYZE procedure with input PARMS= and COVB(EFFECTVAR=ROWCOL)= data sets:

```
proc mianalyze parms=mixparms edf=28
               covb(effectvar=rowcol)=mixcovb;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

Example 45.5. Reading Generalized Linear Model Results from PARMS=, PARMINFO=, and COVB= Data Sets

This example creates data sets containing parameter estimates and corresponding covariance matrices computed by a generalized linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GENMOD to generate the parameter estimates and covariance matrix for each imputed data set:

```
proc genmod data=outmi;
   model Oxygen= RunTime RunPulse/covb;
   by _Imputation_;
   ods output ParameterEstimates=gmparms
              ParmInfo=gmpinfo
              CovB=gmcovb;
run;
```

The following statements print parameter estimates and covariance matrix from the first two imputed data sets in Output 45.5.1.

```
proc print data=gmparms (obs=8);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GENMOD Model Coefficients (First Two Imputations)';
run;
```

Output 45.5.1. PROC GENMOD Model Coefficients

```
                 GENMOD Model Coefficients (First Two Imputations)

            Obs    _Imputation_    Parameter     Estimate      StdErr

              1          1         Intercept      86.5440      9.5107
              2          1         RunTime        -2.8223      0.3120
              3          1         RunPulse       -0.0587      0.0556
              4          1         Scale           2.6692      0.3390
              5          2         Intercept      83.0207      8.4489
              6          2         RunTime        -3.0002      0.3217
              7          2         RunPulse       -0.0249      0.0488
              8          2         Scale           2.5727      0.3267
```

The following statements display the parameter information table in Output 45.5.2. The table identifies parameter names used in the covariance matrices. The parameters Prm1, Prm2, and Prm3 are used for effects Intercept, RunTime, and RunPulse in each covariance matrix.

```
proc print data=gmpinfo (obs=6);
   title 'GENMOD Parameter Information (First Two Imputations)';
run;
```

Output 45.5.2. PROC GENMOD Model Information

```
              GENMOD Parameter Information (First Two Imputations)

                 Obs    _Imputation_    Parameter     Effect

                   1          1            Prm1       Intercept
                   2          1            Prm2       RunTime
                   3          1            Prm3       RunPulse
                   4          2            Prm1       Intercept
                   5          2            Prm2       RunTime
                   6          2            Prm3       RunPulse
```

The following statements display the covariance matrices from the first two imputed data sets in Output 45.5.3. Note that the GENMOD procedure computes maximum likelihood estimates for each covariance matrix.

```
proc print data=gmcovb (obs=8);
   var _Imputation_ RowName Prm1 Prm2 Prm3;
   title 'GENMOD Covariance Matrices (First Two Imputations)';
run;
```

Output 45.5.3. PROC GENMOD Covariance Matrices

```
               GENMOD Covariance Matrices (First Two Imputations)

                                Row
      Obs    _Imputation_    Name          Prm1          Prm2          Prm3

        1          1         Prm1     90.453923     -0.483394     -0.497473
        2          1         Prm2     -0.483394     0.0973159     -0.003113
        3          1         Prm3     -0.497473     -0.003113     0.0030954
        4          1         Scale    2.765E-17     -3.05E-17     2.759E-18
        5          2         Prm1     71.383332     -0.603037     -0.378616
        6          2         Prm2     -0.603037     0.1034766     -0.002826
        7          2         Prm3     -0.378616     -0.002826     0.0023843
        8          2         Scale    1.132E-14     2.181E-16     -7.62E-17
```

The following statements use the MIANALYZE procedure with input PARMS=, PARMINFO=, and COVB= data sets:

```
proc mianalyze parms=gmparms covb=gmcovb parminfo=gmpinfo;
   modeleffects Intercept RunTime RunPulse;
run;
```

Since the GENMOD procedure computes maximum likelihood estimates for the covariance matrix, the EDF= option is not used. The resulting model coefficients are identical to the estimates in Example 45.3 in Output 45.3.3 but the standard errors are slightly different because in this example, maximum likelihood estimates for the standard errors are combined without the EDF= option, whereas in Example 45.3, unbiased estimates for the standard errors are combined with the EDF= option.

Example 45.6. Reading GLM Results from PARMS= and XPXI= Data Sets

This example creates data sets containing parameter estimates and corresponding $(X'X)^{-1}$ matrices computed by a general linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GLM to generate the parameter estimates and $(X'X)^{-1}$ matrix for each imputed data set:

```
proc glm data=outmi;
   model Oxygen= RunTime RunPulse/inverse;
   by _Imputation_;
   ods output ParameterEstimates=glmparms
              InvXPX=glmxpxi;
quit;
```

The following statements display parameter estimates and standard errors from imputed data sets in Output 45.6.1.

```
proc print data=glmparms (obs=6);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GLM Model Coefficients (First Two Imputations)';
run;
```

Output 45.6.1. PROC GLM Model Coefficients

```
                 GLM Model Coefficients (First Two Imputations)

        Obs    _Imputation_    Parameter       Estimate         StdErr

          1          1         Intercept      86.5440339     10.00726811
          2          1         RunTime        -2.8223108      0.32824165
          3          1         RunPulse       -0.0587292      0.05854109
          4          2         Intercept      83.0207303      8.88996885
          5          2         RunTime        -3.0002288      0.33847204
          6          2         RunPulse       -0.0249103      0.05137859
```

The following statements display $(X'X)^{-1}$ matrices from imputed data sets in Output 45.6.2.

```
proc print data=glmxpxi (obs=8);
   var _Imputation_ Parameter Intercept RunTime RunPulse;
   title ’GLM X’’X Inverse Matrices (First Two Imputations)’;
run;
```

Output 45.6.2. PROC GLM $(X'X)^{-1}$ Matrices

```
                 GLM X’X Inverse Matrices (First Two Imputations)

Obs    _Imputation_    Parameter       Intercept         RunTime        RunPulse

  1          1         Intercept    12.696250656    -0.067849956    -0.069826009
  2          1         RunTime      -0.067849956    0.0136594055    -0.000436938
  3          1         RunPulse     -0.069826009    -0.000436938    0.0004344762
  4          1         Oxygen       86.544033929    -2.822310769    -0.058729234
  5          2         Intercept    10.784620785    -0.091107072    -0.057201387
  6          2         RunTime      -0.091107072    0.0156332765    -0.000426902
  7          2         RunPulse     -0.057201387    -0.000426902    0.0003602208
  8          2         Oxygen       83.020730343    -3.000228818    -0.024910305
```

The standard errors for the estimates in the output glmparms data set are needed to create the covariance matrix from the $(X'X)^{-1}$ matrix. The following statements use the MIANALYZE procedure with input PARMS= and XPXI= data sets to produce the same results as displayed in Example 45.3 in Output 45.3.2 and Output 45.3.3:

```
proc mianalyze parms=glmparms xpxi=glmxpxi edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Example 45.7. Reading Logistic Model Results from PARMS= and COVB= Data Sets

This example creates data sets containing parameter estimates and corresponding covariance matrices computed by a logistic regression analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC LOGISTIC to generate the parameter estimates and covariance matrix for each imputed data set.

```
proc logistic data=outfish;
   class Species;
   model Species= Height Width Height*Width/ covb;
   by _Imputation_;
   ods output ParameterEstimates=lgsparms
              CovB=lgscovb;
run;
```

The following statements displays the logistic regression coefficients from the first two imputations in Output 45.7.1.

```
proc print data=lgsparms (obs=8);
   title 'LOGISTIC Model Coefficients (First Two Imputations)';
run;
```

Output 45.7.1. PROC LOGISTIC Model Coefficients

```
              LOGISTIC Model Coefficients (First Two Imputations)

                                                                   Prob
Obs   _Imputation_   Variable      DF   Estimate    StdErr   WaldChiSq   ChiSq

  1         1        Intercept      1    -4.2188    7.8679      0.2875  0.5918
  2         1        Height         1     2.4568    1.0579      5.3929  0.0202
  3         1        Width          1    -3.3480    2.8541      1.3761  0.2408
  4         1        Height*Width   1    -0.1331    0.1441      0.8527  0.3558
  5         2        Intercept      1   -10.9235    9.1880      1.4135  0.2345
  6         2        Height         1     3.1578    1.5208      4.3116  0.0379
  7         2        Width          1    -1.7683    2.9749      0.3533  0.5522
  8         2        Height*Width   1    -0.2714    0.1892      2.0575  0.1515
```

The following statements displays the covariance matrices associated with parameter estimates from the first two imputations in Output 45.7.2.

```
proc print data=lgscovb (obs=8);
   title 'LOGISTIC Model Covariance Matrices (First Two Imputations)';
run;
```

Output 45.7.2. PROC LOGISTIC Covariance Matrices

```
           LOGISTIC Model Covariance Matrices (First Two Imputations)

                                                                    Height
Obs   _Imputation_   Parameter     Intercept     Height      Width     Width

  1         1        Intercept     61.90439    -2.39611   -18.8182   0.923732
  2         1        Height        -2.39611    1.119218   -0.76837   -0.11322
  3         1        Width         -18.8182    -0.76837   8.145619   -0.18386
  4         1        HeightWidth   0.923732    -0.11322   -0.18386   0.020762
  5         2        Intercept     84.41847    -5.94636   -20.9352   1.389396
  6         2        Height        -5.94636    2.312748   -1.08263   -0.24839
  7         2        Width         -20.9352    -1.08263   8.849757    -0.1547
  8         2        HeightWidth   1.389396    -0.24839    -0.1547   0.035796
```

The following statements use the MIANALYZE procedure with input PARMS= and COVB= data sets.

```
proc mianalyze parms=lgsparms
               covb(effectvar=stacking)=lgscovb;
   modeleffects Intercept Height Width Height*Width;
run;
```

Output 45.7.3. Multiple Imputation Variance Information

```
                          The MIANALYZE Procedure

                 Multiple Imputation Variance Information

                    -----------------Variance-----------------
    Parameter            Between          Within           Total       DF

    Intercept          15.218807       70.592292       88.854861   94.689
    Height              0.181361        1.626663        1.844296   287.26
    Width               0.804258        8.428402        9.393511   378.93
    Height*Width        0.006765        0.026888        0.035006    74.37

                  Multiple Imputation Variance Information

                             Relative       Fraction
                             Increase        Missing       Relative
         Parameter        in Variance    Information     Efficiency

         Intercept           0.258705       0.221798       0.957525
         Height              0.133791       0.124081       0.975785
         Width               0.114507       0.107441       0.978964
         Height*Width        0.301942       0.251772       0.952060
```

The “Multiple Imputation Variance Information” table shown in Output 45.7.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 45.7.4. Multiple Imputation Parameter Estimates

```
                         The MIANALYZE Procedure

                 Multiple Imputation Parameter Estimates

Parameter           Estimate      Std Error    95% Confidence Limits        DF

Intercept          -7.085702       9.426286     -25.8000      11.62863   94.689
Height              2.757779       1.358049       0.0848       5.43077   287.26
Width              -2.678006       3.064884      -8.7043       3.34830   378.93
Height*Width       -0.191947       0.187099      -0.5647       0.18083    74.37

                 Multiple Imputation Parameter Estimates

                Parameter             Minimum         Maximum

                Intercept          -11.769173       -4.203658
                Height               2.439954        3.285454
                Width               -3.349258       -1.626538
                Height*Width        -0.291998       -0.131535

                 Multiple Imputation Parameter Estimates

                                                t for H0:
          Parameter              Theta0   Parameter=Theta0   Pr > |t|

          Intercept                   0              -0.75     0.4541
          Height                      0               2.03     0.0432
          Width                       0              -0.87     0.3828
          Height*Width                0              -1.03     0.3083
```

The “Multiple Imputation Parameter Estimates” table shown in Output 45.7.4 displays the combined parameter estimates with associated standard errors.

Example 45.8. Reading Mixed Model Results with CLASS Variables

This example creates data sets containing parameter estimates and corresponding covariance matrices with CLASS variables computed by a mixed regression model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC MIXED to generate the parameter estimates and covariance matrix for each imputed data set:

```
proc mixed data=outfish;
   class Species;
   model Length3= Species Height Width/ solution covb;
   by _Imputation_;
   ods output SolutionF=mxparms CovB=mxcovb;
run;
```

The following statements displays the mixed model coefficients from the first two imputations in Output 45.8.1.

```
proc print data=mxparms (obs=10);
   var _Imputation_ Effect Species Estimate StdErr;
   title 'MIXED Model Coefficients (First Two Imputations)';
run;
```

Output 45.8.1. PROC MIXED Model Coefficients

```
                 MIXED Model Coefficients (First Two Imputations)

      Obs    _Imputation_    Effect       Species    Estimate      StdErr

        1          1         Intercept                 6.8381      1.0290
        2          1         Species        Gp1      -0.05924      0.7253
        3          1         Species        Gp2             0           .
        4          1         Height                    0.9185      0.1732
        5          1         Width                     3.2526      0.5321
        6          2         Intercept                 6.9417      0.9868
        7          2         Species        Gp1       -0.3178      0.7290
        8          2         Species        Gp2             0           .
        9          2         Height                    0.9544      0.1683
       10          2         Width                     3.1697      0.5079
```

The following statements use the MIANALYZE procedure with input PARMS= data set.

```
proc mianalyze parms(classvar=full)=mxparms;
   class Species;
   modeleffects Intercept Species Height Width;
run;
```

Output 45.8.2. Multiple Imputation Variance Information

```
                            The MIANALYZE Procedure

                   Multiple Imputation Variance Information

                          -----------------Variance-----------------
Parameter    Species          Between          Within           Total        DF

Intercept                    0.013257        1.017462        1.033370     16879
Species      Gp1             0.068045        0.519627        0.601281     216.9
Species      Gp2                    0               .               .         .
Height                       0.002691        0.028993        0.032222    398.26
Width                        0.014947        0.270396        0.288332    1033.6

                    Multiple Imputation Variance Information

                                 Relative        Fraction
                                 Increase         Missing        Relative
     Parameter    Species     in Variance     Information      Efficiency

     Intercept                   0.015635        0.015511        0.996907
     Species      Gp1            0.157139        0.143659        0.972071
     Species      Gp2                   .               .               .
     Height                      0.111380        0.104703        0.979489
     Width                       0.066334        0.064017        0.987358
```

The “Multiple Imputation Variance Information” table shown in Output 45.8.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 45.8.3. Multiple Imputation Parameter Estimates

```
                            The MIANALYZE Procedure

                    Multiple Imputation Parameter Estimates

Parameter  Species      Estimate      Std Error   95% Confidence Limits       DF

Intercept               6.844098       1.016548     4.85156    8.836638    16879
Species    Gp1         -0.184298       0.775423    -1.71263    1.344030    216.9
Species    Gp2                 0              .           .           .        .
Height                  0.928624       0.179506     0.57573    1.281522   398.26
Width                   3.237105       0.536966     2.18344    4.290772   1033.6

                    Multiple Imputation Parameter Estimates

                Parameter  Species       Minimum        Maximum

                Intercept               6.713049       6.976758
                Species    Gp1         -0.580012       0.033160
                Species    Gp2                 0              0
                Height                  0.879314       1.004623
                Width                   3.064954       3.360809

                    Multiple Imputation Parameter Estimates

                                                 t for H0:
         Parameter  Species        Theta0   Parameter=Theta0   Pr > |t|

         Intercept                      0               6.73     <.0001
         Species    Gp1                 0              -0.24     0.8124
         Species    Gp2                 0                .         .
         Height                         0               5.17     <.0001
         Width                          0               6.03     <.0001
```

The “Multiple Imputation Parameter Estimates” table shown in Output 45.8.3 displays the combined parameter estimates with associated standard errors.

Example 45.9. Using a TEST statement

This example creates an EST type data set that contains regression coefficients and their corresponding covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model. A TEST statement is used to test linear hypotheses about the parameters.

The following statements use the REG procedure to generate regression coefficients:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements combine the results for the imputed data sets. A TEST statement is used to test linear hypotheses of Intercept=0 and RunTime=RunPulse.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
   test Intercept, RunTime=RunPulse / mult;
run;
```

Output 45.9.1. Test Specification

```
                              The MIANALYZE Procedure
                                  Test: Test 1

                                Test Specification

             ------------------L Matrix-----------------
Parameter       Intercept         RunTime        RunPulse               C

TestPrm1         1.000000               0               0               0
TestPrm2                0        1.000000       -1.000000               0
```

The “Test Specification” table shown in Output 45.9.1 displays the **L** matrix and the **c** vector in a TEST statement. Since there is no label specified for the TEST statement, “Test 1” is used as the label.

Output 45.9.2. Multiple Imputation Variance Information

```
                              The MIANALYZE Procedure
                                  Test: Test 1

                       Multiple Imputation Variance Information

                      -----------------Variance----------------
     Parameter          Between          Within           Total       DF

     TestPrm1         45.529229       76.543614      131.178689   9.1917
     TestPrm2          0.014715        0.114324        0.131983   20.598

                      Multiple Imputation Variance Information

                         Relative        Fraction
                         Increase         Missing        Relative
         Parameter    in Variance     Information      Efficiency

         TestPrm1        0.713777        0.461277        0.915537
         TestPrm2        0.154459        0.141444        0.972490
```

The “Multiple Imputation Variance Information” table shown in Output 45.9.2 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. A detailed description of these statistics is provided in the “Combining Inferences from Imputed Data Sets” section on page 2624 and the “Multiple Imputation Efficiency” section on page 2626.

Output 45.9.3. Multiple Imputation Parameter Estimates

```
                              The MIANALYZE Procedure
                                    Test: Test 1

                       Multiple Imputation Parameter Estimates

  Parameter        Estimate      Std Error    95% Confidence Limits        DF

  TestPrm1        90.837440      11.453327     65.01034     116.6645   9.1917
  TestPrm2        -2.964292       0.363294     -3.72070      -2.2079   20.598

                       Multiple Imputation Parameter Estimates

                                                               t for H0:
Parameter          Minimum        Maximum             C   Parameter=C   Pr > |t|

TestPrm1         83.020730     100.839807             0          7.93     <.0001
TestPrm2         -3.091586      -2.763582             0         -8.16     <.0001
```

The "Multiple Imputation Parameter Estimates" table shown in Output 45.9.3 displays the estimated mean and standard error of the linear components. The inferences are based on the *t* distribution. The table also displays a 95% mean confidence interval and a *t* test with the associated *p*-value for the hypothesis that each linear component of $\mathbf{L}\boldsymbol{\beta}$ is equal to zero.

With the MULT option, the procedure assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the linear components taken jointly in Output 45.9.4.

Output 45.9.4. Multiple Imputation Multivariate Inference

```
                              The MIANALYZE Procedure
                                    Test: Test 1

                      Multiple Imputation Multivariate Inference
             Assuming Proportionality of Between/Within Covariance Matrices

           Avg Relative
               Increase                                F for H0:
            in Variance   Num DF   Den DF   Parameter=Theta0     Pr > F

               0.419868        2   35.053              60.34     <.0001
```

Example 45.10. Combining Correlation Coefficients

This example combines sample correlation coefficients computed from a set of imputed data sets using Fisher's z transformation.

Fisher's z transformation of the sample correlation r is

$$z = \frac{1}{2} \log \left(\frac{1+r}{1-r} \right)$$

The statistic z is approximately normally distributed with mean

$$\log\left(\frac{1+\rho}{1-\rho}\right)$$

and variance $1/(n-3)$, where ρ is the population correlation coefficient and n is the number of observations.

The following statements use the CORR procedure to compute the correlation r and its associated Fisher's z statistic between variables Oxygen and RunTime for each imputed data set. The ODS statement is used to save Fisher's z statistic in an output data set.

```
proc corr data=outmi fisher(biasadj=no);
   var Oxygen RunTime;
   by _Imputation_;
   ods output FisherPearsonCorr= outz;
run;
```

The following statements display the number of observations and Fisher's z statistic for each imputed data set in Output 45.10.1.

```
proc print data=outz;
   title 'Fisher''s Correlation Statistics';
   var _Imputation_ NObs ZVal;
run;
```

Output 45.10.1. Output z Statistics

```
                     Fisher's Correlation Statistics

               Obs    _Imputation_        NObs          ZVal

                 1          1               31      -1.27869
                 2          2               31      -1.30715
                 3          3               31      -1.27922
                 4          4               31      -1.39243
                 5          5               31      -1.40146
```

The following statements generate the standard error associated with the z statistic, $1/\sqrt{n-3}$:

```
data outz;
   set outz;
   StdZ= 1. / sqrt(NObs-3);
run;
```

The following statements use the MIANALYZE procedure to generate a combined parameter estimate $\hat{z}$ and its variance, as shown in Output 45.10.2. The ODS statement is used to save the parameter estimates in an output data set.

```
proc mianalyze data=outz;
   ods output ParameterEstimates=parms;
   modeleffects ZVal;
   stderr StdZ;
run;
```

Output 45.10.2. Combining Fisher's z statistics

```
                              The MIANALYZE Procedure

                        Multiple Imputation Parameter Estimates

Parameter        Estimate      Std Error    95% Confidence Limits        DF

ZVal            -1.331787       0.200327     -1.72587     -0.93771   330.23

                        Multiple Imputation Parameter Estimates

                        Parameter         Minimum        Maximum

                        ZVal            -1.401459      -1.278686

                        Multiple Imputation Parameter Estimates

                                                    t for H0:
          Parameter          Theta0   Parameter=Theta0   Pr > |t|

          ZVal                    0              -6.65     <.0001
```

In addition to the estimate for z, PROC MIANALYZE also generates 95% confidence limits for z, $\hat{z}_{.025}$ and $\hat{z}_{.975}$. The following statements print the estimate and 95% confidence limits for z in Output 45.10.3.

```
proc print data=parms;
   title 'Parameter Estimates with 95% Confidence Limits';
   var Estimate LCLMean UCLMean;
run;
```

Output 45.10.3. Parameter Estimates with 95% Confidence Limits

```
                 Parameter Estimates with 95% Confidence Limits

                  Obs        Estimate      LCLMean      UCLMean

                   1        -1.331787     -1.72587     -0.93771
```

An estimate of the correlation coefficient and 95% confidence limits are then generated from the following inverse transformation as described in the "Correlation Coefficients" section on page 2630

$$r = tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

for $z = \hat{z}$, $\hat{z}_{.025}$, and $\hat{z}_{.975}$.

The following statements generate and display an estimate of the correlation coefficient and its 95% confidence limits.

```
data corr_ci;
   set parms;
   r=       tanh( Estimate);
   r_lower= tanh( LCLMean);
   r_upper= tanh( UCLMean);
run;

proc print data=corr_ci;
   title 'Estimated Correlation Coefficient'
         ' with 95% Confidence Limits';
   var r r_lower r_upper;
run;
```

Output 45.10.4. Estimated Correlation Coefficient

```
         Estimated Correlation Coefficient with 95% Confidence Limits

                       Obs        r         r_lower      r_upper

                        1     -0.86969     -0.93857     -0.73417
```

References

In addition to the research literature listed below, the development of the MI and MIANALYZE procedures has benefited significantly from the advice and expertise of

- Paul Allison, University of Pennsylvania
- John Barnard Jr., Cleveland Clinic Foundation
- Nick Horton, Boston University
- Donald B. Rubin, Harvard University
- Joseph L. Schafer, Penn State University

We gratefully acknowledge the time and effort that these researchers have contributed.

Allison, P.D. (2000), “Multiple Imputation for Missing Data: A Cautionary Tale,” *Sociological Methods and Research*, 28, 301–309.

Allison, P.D. (2001), “Missing Data,” Thousand Oaks, CA: Sage Publications.

Barnard, J. and Rubin, D.B. (1999), “Small-Sample Degrees of Freedom with Multiple Imputation,” *Biometrika*, 86, 948–955.

Cochran, W.J. (1977), *Sampling Techniques*, Second Edition, New York: John Wiley & Sons, Inc.

Gadbury, G.L., Coffey, C.S., and Allison, D.B. (2003), “Modern Statistical Methods for Handling Missing Repeated Measurements in Obesity Trial Data: Beyond LOCF,” *Obesity Reviews*, 4, 175–184.

Horton, N.J. and Lipsitz, S.R. (2001), “Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables,” *Journal of the American Statistical Association*, 55, 244–254.

Li, K.H., Raghunathan, T.E., and Rubin, D.B. (1991), “Large-Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics and an F Reference Distribution,” *Journal of the American Statistical Association*, 86, 1065–1073.

Little, R.J.A. and Rubin, D.B. (2002), *Statistical Analysis with Missing Data*, Second Edition, New York: John Wiley & Sons, Inc.

Rubin, D.B. (1976), “Inference and Missing Data,” *Biometrika*, 63, 581–592.

Rubin, D.B. (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: John Wiley & Sons, Inc.

Rubin, D.B. (1996), “Multiple Imputation After 18+ Years,” *Journal of the American Statistical Association*, 91, 473–489.

Schafer, J.L. (1997), *Analysis of Incomplete Multivariate Data*, New York: Chapman and Hall.

Chapter 46
The MIXED Procedure

Chapter Contents

Chapter 46
The MIXED Procedure

Overview

The MIXED procedure fits a variety of mixed linear models to data and enables you to use these fitted models to make statistical inferences about the data. A *mixed linear model* is a generalization of the standard linear model used in the GLM procedure, the generalization being that the data are permitted to exhibit correlation and nonconstant variability. The mixed linear model, therefore, provides you with the flexibility of modeling not only the means of your data (as in the standard linear model) but their variances and covariances as well.

The primary assumptions underlying the analyses performed by PROC MIXED are as follows:

- The data are normally distributed (Gaussian).
- The means (expected values) of the data are linear in terms of a certain set of parameters.
- The variances and covariances of the data are in terms of a different set of parameters, and they exhibit a structure matching one of those available in PROC MIXED.

Since Gaussian data can be modeled entirely in terms of their means and variances/covariances, the two sets of parameters in a mixed linear model actually specify the complete probability distribution of the data. The parameters of the mean model are referred to as *fixed-effects parameters*, and the parameters of the variance-covariance model are referred to as *covariance parameters*.

The fixed-effects parameters are associated with known explanatory variables, as in the standard linear model. These variables can be either qualitative (as in the traditional analysis of variance) or quantitative (as in standard linear regression). However, the covariance parameters are what distinguishes the mixed linear model from the standard linear model.

The need for covariance parameters arises quite frequently in applications, the following being the two most typical scenarios:

- The experimental units on which the data are measured can be grouped into clusters, and the data from a common cluster are correlated.
- Repeated measurements are taken on the same experimental unit, and these repeated measurements are correlated or exhibit variability that changes.

The first scenario can be generalized to include one set of clusters nested within another. For example, if students are the experimental unit, they can be clustered into classes, which in turn can be clustered into schools. Each level of this hierarchy can introduce an additional source of variability and correlation. The second scenario occurs in longitudinal studies, where repeated measurements are taken over time. Alternatively, the repeated measures could be spatial or multivariate in nature.

PROC MIXED provides a variety of covariance structures to handle the previous two scenarios. The most common of these structures arises from the use of *random-effects parameters*, which are additional unknown random variables assumed to impact the variability of the data. The variances of the random-effects parameters, commonly known as *variance components*, become the covariance parameters for this particular structure. Traditional mixed linear models contain both fixed- and random-effects parameters, and, in fact, it is the combination of these two types of effects that led to the name *mixed model*. PROC MIXED fits not only these traditional variance component models but numerous other covariance structures as well.

PROC MIXED fits the structure you select to the data using the method of *restricted maximum likelihood (REML)*, also known as *residual maximum likelihood*. It is here that the Gaussian assumption for the data is exploited. Other estimation methods are also available, including *maximum likelihood* and *MIVQUE0*. The details behind these estimation methods are discussed in subsequent sections.

Once a model has been fit to your data, you can use it to draw statistical inferences via both the fixed-effects and covariance parameters. PROC MIXED computes several different statistics suitable for generating hypothesis tests and confidence intervals. The validity of these statistics depends upon the mean and variance-covariance model you select, so it is important to choose the model carefully. Some of the output from PROC MIXED helps you assess your model and compare it with others.

Basic Features

PROC MIXED provides easy accessibility to numerous mixed linear models that are useful in many common statistical analyses. In the style of the GLM procedure, PROC MIXED fits the specified mixed linear model and produces appropriate statistics.

Some basic features of PROC MIXED are

- covariance structures, including variance components, compound symmetry, unstructured, AR(1), Toeplitz, spatial, general linear, and factor analytic
- GLM-type grammar, using MODEL, RANDOM, and REPEATED statements for model specification and CONTRAST, ESTIMATE, and LSMEANS statements for inferences
- appropriate standard errors for all specified estimable linear combinations of fixed and random effects, and corresponding *t*- and *F*-tests
- subject and group effects that enable blocking and heterogeneity, respectively
- REML and ML estimation methods implemented with a Newton-Raphson algorithm

- capacity to handle unbalanced data
- ability to create a SAS data set corresponding to any table

PROC MIXED uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC MIXED into a SAS data set. See the "ODS Table Names" section on page 2752.

Experimental graphics are now available with the MIXED procedure. For more information, see the "ODS Graphics" section on page 2757.

Notation for the Mixed Model

This section introduces the mathematical notation used throughout this chapter to describe the mixed linear model. You should be familiar with basic matrix algebra (refer to Searle 1982). A more detailed description of the mixed model is contained in the "Mixed Models Theory" section on page 2731.

A statistical model is a mathematical description of how data are generated. The standard linear model, as used by the GLM procedure, is one of the most common statistical models:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

In this expression, $\mathbf{y}$ represents a vector of observed data, $\boldsymbol{\beta}$ is an unknown vector of fixed-effects parameters with known design matrix $\mathbf{X}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector modeling the statistical noise around $\mathbf{X}\boldsymbol{\beta}$. The focus of the standard linear model is to model the mean of $\mathbf{y}$ by using the fixed-effects parameters $\boldsymbol{\beta}$. The residual errors $\boldsymbol{\epsilon}$ are assumed to be independent and identically distributed Gaussian random variables with mean 0 and variance σ^2.

The mixed model generalizes the standard linear

model as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

Here, $\boldsymbol{\gamma}$ is an unknown vector of random-effects parameters with known design matrix $\mathbf{Z}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector whose elements are no longer required to be independent and homogeneous.

To further develop this notion of variance modeling, assume that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are Gaussian random variables that are uncorrelated and have expectations $\mathbf{0}$ and variances $\mathbf{G}$ and $\mathbf{R}$, respectively. The variance of $\mathbf{y}$ is thus

$$\mathbf{V} = \mathbf{ZGZ}' + \mathbf{R}$$

Note that, when $\mathbf{R} = \sigma^2\mathbf{I}$ and $\mathbf{Z} = \mathbf{0}$, the mixed model reduces to the standard linear model.

You can model the variance of the data, $\mathbf{y}$, by specifying the structure (or form) of $\mathbf{Z}$, $\mathbf{G}$, and $\mathbf{R}$. The model matrix $\mathbf{Z}$ is set up in the same fashion as $\mathbf{X}$, the model matrix for the fixed-effects parameters. For $\mathbf{G}$ and $\mathbf{R}$, you must select some *covariance structure.* Possible covariance structures include

- variance components
- compound symmetry (common covariance plus diagonal)
- unstructured (general covariance)
- autoregressive
- spatial
- general linear
- factor analytic

By appropriately defining the model matrices $\mathbf{X}$ and $\mathbf{Z}$, as well as the covariance structure matrices $\mathbf{G}$ and $\mathbf{R}$, you can perform numerous mixed model analyses.

PROC MIXED Contrasted with Other SAS Procedures

PROC MIXED is a generalization of the GLM procedure in the sense that PROC GLM fits standard linear models, and PROC MIXED fits the wider class of mixed linear models. Both procedures have similar CLASS, MODEL, CONTRAST, ESTIMATE, and LSMEANS statements, but their RANDOM and REPEATED statements differ (see the following paragraphs). Both procedures use the non-full-rank model parameterization, although the sorting of classification levels can differ between the two. PROC MIXED computes only Type I–Type III tests of fixed effects, while PROC GLM offers Types I–IV.

The RANDOM statement in PROC MIXED incorporates random effects constituting the γ vector in the mixed model. However, in PROC GLM, effects specified in the RANDOM statement are still treated as fixed as far as the model fit is concerned, and they serve only to produce corresponding expected mean squares. These expected mean squares lead to the traditional ANOVA estimates of variance components. PROC MIXED computes REML and ML estimates of variance parameters, which are generally preferred to the ANOVA estimates (Searle 1988; Harville 1988; Searle, Casella, and McCulloch 1992). Optionally, PROC MIXED also computes MIVQUE0 estimates, which are similar to ANOVA estimates.

The REPEATED statement in PROC MIXED is used to specify covariance structures for repeated measurements on subjects, while the REPEATED statement in PROC GLM is used to specify various transformations with which to conduct the traditional univariate or multivariate tests. In repeated measures situations, the mixed model approach used in PROC MIXED is more flexible and more widely applicable than either the univariate or multivariate approaches. In particular, the mixed model approach provides a larger class of covariance structures and a better mechanism for handling missing values (Wolfinger and Chang 1995).

PROC MIXED subsumes the VARCOMP procedure. PROC MIXED provides a wide variety of covariance structures, while PROC VARCOMP estimates only simple random effects. PROC MIXED carries out several analyses that are absent in PROC VARCOMP, including the estimation and testing of linear combinations of fixed and random effects.

The ARIMA and AUTOREG procedures provide more time series structures than PROC MIXED, although they do not fit variance component models. The CALIS procedure fits general covariance matrices, but it does not allow fixed effects as does PROC MIXED. The LATTICE and NESTED procedures fit special types of mixed linear models that can also be handled in PROC MIXED, although PROC MIXED may run slower because of its more general algorithm. The TSCSREG procedure analyzes time-series cross-sectional data, and it fits some structures not available in PROC MIXED.

Getting Started

Clustered Data Example

Consider the following SAS data set as an introductory example:

```
data heights;
   input Family Gender$ Height @@;
   datalines;
1 F 67   1 F 66   1 F 64   1 M 71   1 M 72   2 F 63
2 F 63   2 F 67   2 M 69   2 M 68   2 M 70   3 F 63
3 M 64   4 F 67   4 F 66   4 M 67   4 M 67   4 M 69
run;
```

The response variable Height measures the heights (in inches) of 18 individuals. The individuals are classified according to Family and Gender. You can perform a traditional two-way analysis of variance of these data with the following PROC MIXED code:

```
proc mixed;
   class Family Gender;
   model Height = Gender Family Family*Gender;
run;
```

The PROC MIXED statement invokes the procedure. The CLASS statement instructs PROC MIXED to consider both Family and Gender as classification variables. Dummy (indicator) variables are, as a result, created corresponding to all of the distinct levels of Family and Gender. For these data, Family has four levels and Gender has two levels.

The MODEL statement first specifies the response (dependent) variable Height. The explanatory (independent) variables are then listed after the equal (=) sign. Here, the two explanatory variables are Gender and Family, and they comprise the main effects of the design. The third explanatory term, Family*Gender, models an interaction between the two main effects.

PROC MIXED uses the dummy variables associated with Gender, Family, and Family*Gender to construct the $\mathbf{X}$ matrix for the linear model. A column of 1s is also included as the first column of $\mathbf{X}$ to model a global intercept. There are no $\mathbf{Z}$ or $\mathbf{G}$ matrices for this model, and $\mathbf{R}$ is assumed to equal $\sigma^2\mathbf{I}$, where $\mathbf{I}$ is an 18×18 identity matrix.

The RUN statement completes the specification. The coding is precisely the same as with the GLM procedure. However, much of the output from PROC MIXED is different from that produced by PROC GLM.

The following is the output from PROC MIXED.

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.HEIGHTS
              Dependent Variable           Height
              Covariance Structure         Diagonal
              Estimation Method            REML
              Residual Variance Method     Profile
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Residual
```

Figure 46.1. Model Information

The "Model Information" table describes the model, some of the variables that it involves, and the method used in fitting it. This table also lists the method (profile, factor, or fit) for handling the residual variance.

```
                              The Mixed Procedure

                            Class Level Information

                Class      Levels    Values

                Family          4    1 2 3 4
                Gender          2    F M
```

Figure 46.2. Class Level Information

The "Class Level Information" table lists the levels of all variables specified in the CLASS statement. You can check this table to make sure that the data are correct.

```
                      The Mixed Procedure

                          Dimensions

             Covariance Parameters             1
             Columns in X                     15
             Columns in Z                      0
             Subjects                          1
             Max Obs Per Subject              18
```

Figure 46.3. Dimensions

The “Dimensions” table lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements.

```
                      The Mixed Procedure

                    Number of Observations

         Number of Observations Read             18
         Number of Observations Used             18
         Number of Observations Not Used          0
```

Figure 46.4. Number of Observations

The “Number of Observations” table displays information about the sample size being processed.

```
                      The Mixed Procedure

                     Covariance Parameter
                           Estimates

                     Cov Parm     Estimate

                     Residual       2.1000
```

Figure 46.5. Covariance Parameter Estimates

The “Covariance Parameter Estimates” table displays the estimate of σ^2 for the model.

```
                      The Mixed Procedure

                        Fit Statistics

             -2 Res Log Likelihood           41.6
             AIC (smaller is better)         43.6
             AICC (smaller is better)        44.1
             BIC (smaller is better)         43.9
```

Figure 46.6. Fit Statistics

The “Fit Statistics” table lists several pieces of information about the fitted mixed model, including values derived from the computed value of the restricted/residual likelihood.

```
                          The Mixed Procedure

                     Type 3 Tests of Fixed Effects

                            Num      Den
        Effect               DF       DF    F Value    Pr > F

        Gender                1       10      17.63    0.0018
        Family                3       10       5.90    0.0139
        Family*Gender         3       10       2.89    0.0889
```

Figure 46.7. Tests of Fixed Effects

The “Type 3 Tests of Fixed Effects” table displays significance tests for the three effects listed in the MODEL statement. The Type III F-statistics and p-values are the same as those produced by the GLM procedure. However, because PROC MIXED uses a likelihood-based estimation scheme, it does not directly compute or display sums of squares for this analysis.

The Type 3 test for Family*Gender effect is not significant at the 5% level, but the tests for both main effects are significant.

The important assumptions behind this analysis are that the data are normally distributed and that they are independent with constant variance. For these data, the normality assumption is probably realistic since the data are observed heights. However, since the data occur in clusters (families), it is very likely that observations from the same family are statistically correlated, that is, not independent.

The methods implemented in PROC MIXED are still based on the assumption of normally distributed data, but you can drop the assumption of independence by modeling statistical correlation in a variety of ways. You can also model variances that are heterogeneous, that is, nonconstant.

For the height data, one of the simplest ways of modeling correlation is through the use of *random effects*. Here the family effect is assumed to be normally distributed with zero mean and some unknown variance. This is in contrast to the previous model in which the family effects are just constants, or *fixed effects*. Declaring Family as a random effect sets up a common correlation among all observations having the same level of Family.

Declaring Family*Gender as a random effect models an additional correlation between all observations that have the same level of both Family and Gender. One interpretation of this effect is that a female in a certain family exhibits more correlation with the other females in that family than with the other males, and likewise for a male. With the height data, this model seems reasonable.

The code to fit this correlation model in PROC MIXED is as follows:

```
proc mixed;
   class Family Gender;
   model Height = Gender;
   random Family Family*Gender;
run;
```

Note that Family and Family*Gender are now listed in the RANDOM statement. The dummy variables associated with them are used to construct the **Z** matrix in the mixed model. The **X** matrix now consists of a column of 1s and the dummy variables for Gender.

The **G** matrix for this model is diagonal, and it contains the variance components for both Family and Family*Gender. The **R** matrix is still assumed to equal $\sigma^2 I$, where I is an identity matrix.

The output from this analysis is as follows.

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.HEIGHTS
              Dependent Variable           Height
              Covariance Structure         Variance Components
              Estimation Method            REML
              Residual Variance Method     Profile
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Containment
```

Figure 46.8. Model Information

The “Model Information” table shows that the containment method is used to compute the degrees of freedom for this analysis. This is the default method when a RANDOM statement is used; see the description of the DDFM= option on page 2693 for more information.

```
                              The Mixed Procedure

                            Class Level Information

                Class     Levels    Values

                Family         4    1 2 3 4
                Gender         2    F M
```

Figure 46.9. Class Levels Information

The “Class Levels Information” table is the same as before.

```
                        The Mixed Procedure

                            Dimensions

              Covariance Parameters             3
              Columns in X                      3
              Columns in Z                     12
              Subjects                          1
              Max Obs Per Subject              18


                      Number of Observations

          Number of Observations Read              18
          Number of Observations Used              18
          Number of Observations Not Used           0
```

Figure 46.10. Dimensions and Number of Observations

The “Dimensions” table displays the new sizes of the **X** and **Z** matrices.

```
                        The Mixed Procedure

                         Iteration History

      Iteration    Evaluations    -2 Res Log Like       Criterion

              0              1        74.11074833
              1              2        71.51614003      0.01441208
              2              1        71.13845990      0.00412226
              3              1        71.03613556      0.00058188
              4              1        71.02281757      0.00001689
              5              1        71.02245904      0.00000002
              6              1        71.02245869      0.00000000


                      Convergence criteria met.
```

Figure 46.11. REML Estimation Iteration History

The “Iteration History” table displays the results of the numerical optimization of the restricted/residual likelihood. Six iterations are required to achieve the default convergence criterion of 1E−8.

```
                        The Mixed Procedure

                        Covariance Parameter
                             Estimates

                     Cov Parm          Estimate

                     Family              2.4010
                     Family*Gender       1.7657
                     Residual            2.1668
```

Figure 46.12. Covariance Parameter Estimates (REML)

The “Covariance Parameter Estimates” table displays the results of the REML fit. The Estimate column contains the estimates of the variance components for Family and Family*Gender, as well as the estimate of σ^2.

```
                        The Mixed Procedure

                          Fit Statistics

              -2 Res Log Likelihood            71.0
              AIC (smaller is better)          77.0
              AICC (smaller is better)         79.0
              BIC (smaller is better)          75.2
```

Figure 46.13. Fit Statistics

The “Fit Statistics” table contains basic information about the REML fit.

```
                        The Mixed Procedure

                    Type 3 Tests of Fixed Effects

                          Num     Den
          Effect           DF      DF    F Value    Pr > F

          Gender            1       3       7.95    0.0667
```

Figure 46.14. Type 3 Tests of Fixed Effects

The “Type 3 Tests of Fixed Effects” table contains a significance test for the lone fixed effect, Gender. Note that the associated p-value is not nearly as significant as in the previous analysis. This illustrates the importance of correctly modeling correlation in your data.

An additional benefit of the random effects analysis is that it enables you to make inferences about gender that apply to an entire population of families, whereas the inferences about gender from the analysis where Family and Family*Gender are fixed effects apply only to the particular families in the data set.

PROC MIXED thus offers you the ability to model correlation directly and to make inferences about fixed effects that apply to entire populations of random effects.

Syntax

The following statements are available in PROC MIXED.

PROC MIXED < *options* > **;**

BY *variables* **;**

CLASS *variables* **;**

ID *variables* **;**

MODEL *dependent* = < *fixed-effects* > < / *options* > **;**

RANDOM *random-effects* < / *options* > **;**

REPEATED < *repeated-effect* >< / *options* > **;**

PARMS *(value-list)* . . . < / *options* > **;**

PRIOR < *distribution* >< / *options* > **;**

CONTRAST *'label'* < *fixed-effect values* . . . >
< | *random-effect values* . . . > , . . . < / *options* > **;**

ESTIMATE *'label'* < *fixed-effect values* . . . >
< | *random-effect values* . . . >< / *options* > **;**

LSMEANS *fixed-effects* < / *options* > **;**

WEIGHT *variable* **;**

Items within angle brackets (< >) are optional. The CONTRAST, ESTIMATE, LSMEANS, and RANDOM statements can appear multiple times; all other statements can appear only once.

The PROC MIXED and MODEL statements are required, and the MODEL statement must appear after the CLASS statement if a CLASS statement is included. The CONTRAST, ESTIMATE, LSMEANS, RANDOM, and REPEATED statements must follow the MODEL statement. The CONTRAST and ESTIMATE statements must also follow any RANDOM statements.

Table 46.1 summarizes the basic functions and important options of each PROC MIXED statement. The syntax of each statement in Table 46.1 is described in the following sections in alphabetical order after the description of the PROC MIXED statement.

Table 46.1. Summary of PROC MIXED Statements

Statement	Description	Important Options
PROC MIXED	invokes the procedure	DATA= specifies input data set, METHOD= specifies estimation method
BY	performs multiple PROC MIXED analyses in one invocation	none
CLASS	declares qualitative variables that create indicator variables in design matrices	none
ID	lists additional variables to be included in predicted values tables	none
MODEL	specifies dependent variable and fixed effects, setting up $\mathbf{X}$	S requests solution for fixed-effects parameters, DDFM= specifies denominator degrees of freedom method, OUTP= outputs predicted values to a data set, INFLUENCE computes influence diagnostics
RANDOM	specifies random effects, setting up $\mathbf{Z}$ and $\mathbf{G}$	SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, S requests solution for random-effects parameters, G displays estimated $\mathbf{G}$
REPEATED	sets up $\mathbf{R}$	SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, R displays estimated blocks of $\mathbf{R}$, GROUP= enables between-subject heterogeneity, LOCAL adds a diagonal matrix to $\mathbf{R}$
PARMS	specifies a grid of initial values for the covariance parameters	HOLD= and NOITER hold the covariance parameters or their ratios constant, PDATA= reads the initial values from a SAS data set
PRIOR	performs a sampling-based Bayesian analysis for variance component models	NSAMPLE= specifies the sample size, SEED= specifies the starting seed
CONTRAST	constructs custom hypothesis tests	E displays the $\mathbf{L}$ matrix coefficients
ESTIMATE	constructs custom scalar estimates	CL produces confidence limits
LSMEANS	computes least squares means for classification fixed effects	DIFF computes differences of the least squares means, ADJUST= performs multiple comparisons adjustments, AT changes covariates, OM changes weighting, CL produces confidence limits, SLICE= tests simple effects
WEIGHT	specifies a variable by which to weight $\mathbf{R}$	none

PROC MIXED Statement

PROC MIXED < *options* >**;**

The PROC MIXED statement invokes the procedure. You can specify the following options.

ABSOLUTE

makes the convergence criterion absolute. By default, it is relative (divided by the current objective function value). See the CONVF, CONVG, and CONVH options in this section for a description of various convergence criteria.

ALPHA=*number*

requests that confidence limits be constructed for the covariance parameter estimates with confidence level 1 − *number*. The value of *number* must be between 0 and 1; the default is 0.05.

ASYCORR

produces the asymptotic correlation matrix of the covariance parameter estimates. It is computed from the corresponding asymptotic covariance matrix (see the description of the ASYCOV option, which follows). For ODS purposes, the label of the "Asymptotic Correlation" table is "AsyCorr."

ASYCOV

requests that the asymptotic covariance matrix of the covariance parameters be displayed. By default, this matrix is the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian (second derivative) matrix of the objective function. See the "Covariance Parameter Estimates" section on page 2750 for more information about this matrix. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. For ODS purposes, the label of the "Asymptotic Covariance" table is "AsyCov."

CL<**=WALD**>

requests confidence limits for the covariance parameter estimates. A Satterthwaite approximation is used to construct limits for all parameters that have a lower boundary constraint of zero. These limits take the form

$$\frac{\nu\widehat{\sigma}^2}{\chi^2_{\nu,1-\alpha/2}} \leq \sigma^2 \leq \frac{\nu\widehat{\sigma}^2}{\chi^2_{\nu,\alpha/2}}$$

where $\nu = 2Z^2$, Z is the Wald statistic $\widehat{\sigma}^2/\mathrm{se}(\widehat{\sigma}^2)$, and the denominators are quantiles of the χ^2-distribution with ν degrees of freedom. Refer to Milliken and Johnson (1992) and Burdick and Graybill (1992) for similar techniques.

For all other parameters, Wald Z-scores and normal quantiles are used to construct the limits. Wald limits are also provided for variance components if you specify the NOBOUND option. The optional =WALD specification requests Wald limits for all parameters.

The confidence limits are displayed as extra columns in the “Covariance Parameter Estimates” table. The confidence level is $1-\alpha = 0.95$ by default; this can be changed with the ALPHA= option.

CONVF<*=number*>

requests the relative function convergence criterion with tolerance *number*. The relative function convergence criterion is

$$\frac{|f_k - f_{k-1}|}{|f_k|} \le number$$

where f_k is the value of the objective function at iteration *k*. To prevent the division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E$-$8.

CONVG <*=number*>

requests the relative gradient convergence criterion with tolerance *number*. The relative gradient convergence criterion is

$$\frac{\max_j |g_{jk}|}{|f_k|} \le number$$

where f_k is the value of the objective function, and g_{jk} is the *j*th element of the gradient (first derivative) of the objective function, both at iteration *k*. To prevent division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E$-$8.

CONVH<*=number*>

requests the relative Hessian convergence criterion with tolerance *number*. The relative Hessian convergence criterion is

$$\frac{\mathbf{g}_k{}' \mathbf{H}_k^{-1} \mathbf{g}_k}{|f_k|} \le number$$

where f_k is the value of the objective function, $\mathbf{g}_k$ is the gradient (first derivative) of the objective function, and $\mathbf{H}_k$ is the Hessian (second derivative) of the objective function, all at iteration *k*.

If $\mathbf{H}_k$ is singular, then PROC MIXED uses the following relative criterion:

$$\frac{\mathbf{g}_k' \mathbf{g}_k}{|f_k|} \le number$$

To prevent the division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E$-$8.

COVTEST

produces asymptotic standard errors and Wald Z-tests for the covariance parameter estimates.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC MIXED. The default is the most recently created data set.

DFBW

has the same effect as the DDFM=BW option in the MODEL statement.

EMPIRICAL

computes the estimated variance-covariance matrix of the fixed-effects parameters by using the asymptotically consistent estimator described in Huber (1967), White (1980), Liang and Zeger (1986), and Diggle, Liang, and Zeger (1994). This estimator is commonly referred to as the “sandwich” estimator, and it is computed as follows:

$$(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-} \left(\sum_{i=1}^{S} \mathbf{X}_i' \widehat{\mathbf{V}_i}^{-1} \widehat{\boldsymbol{\epsilon}}_i \widehat{\boldsymbol{\epsilon}}_i{}' \widehat{\mathbf{V}_i}^{-1} \mathbf{X}_i \right) (\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$$

Here, $\widehat{\boldsymbol{\epsilon}}_i = y_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}$, S is the number of subjects, and matrices with an i subscript are those for the ith subject. You must include the SUBJECT= option in either a RANDOM or REPEATED statement for this option to take effect.

When you specify the EMPIRICAL option, PROC MIXED adjusts all standard errors and test statistics involving the fixed-effects parameters. This changes output in the following tables (listed in Table 46.8 on page 2752): Contrast, CorrB, CovB, Diffs, Estimates, InvCovB, LSMeans, MMEq, MMEqSol, Slices, SolutionF, Tests1–Tests3. The OUTP= and OUTPM= data sets are also affected. Finally, the Satterthwaite and Kenward-Roger degrees of freedom methods are not available if you specify EMPIRICAL.

IC

displays a table of various information criteria. The criteria are all in smaller-is-better form, and are described in Table 46.2.

Table 46.2. Information Criteria

Criteria	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$	Hurvich and Tsai (1989)
		Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log\log n$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log n$	Schwarz (1978)
CAIC	$-2\ell + d(\log n + 1)$	Bozdogan (1987)

Here ℓ denotes the maximum value of the (possibly restricted) log likelihood, d the dimension of the model, and n the number of observations. In Version 6 of SAS/STAT software, n equals the number of valid observations for maximum likelihood estimation and $n - p$ for restricted maximum likelihood estimation, where p equals the rank of $\mathbf{X}$. In later versions, n equals the number of effective subjects as displayed in the

"Dimensions" table, unless this value equals 1, in which case n equals the number of levels of the first RANDOM effect you specify. If the number of effective subjects equals 1 and you have no RANDOM statements, then n reverts to the Version 6 values. For AICC (a finite-sample corrected version of AIC), n^* equals the Version 6 values of n, unless this number is less than $d + 2$, in which case it equals $d + 2$.

For restricted likelihood estimation, d equals q the effective number of estimated covariance parameters. In Version 6, when a parameter estimate lies on a boundary constraint, then it is still included in the calculation of d, but in later versions it is not. The most common example of this behavior is when a variance component is estimated to equal zero. For maximum likelihood estimation, d equals $q + p$.

For ODS purposes, the name of the "Information Criteria" table is "InfoCrit."

INFO

is a default option. The creation of the "Model Information", "Dimensions", and "Number of Observations" tables can be suppressed using the NOINFO option.

Note that, in Version 6, this option displays the "Model Information" and "Dimensions" tables.

ITDETAILS

displays the parameter values at each iteration and enables the writing of notes to the SAS log pertaining to "infinite likelihood" and "singularities" during Newton-Raphson iterations.

LOGNOTE

writes periodic notes to the log describing the current status of computations. It is designed for use with analyses requiring extensive CPU resources.

MAXFUNC=*number*

specifies the maximum number of likelihood evaluations in the optimization process. The default is 150.

MAXITER=*number*

specifies the maximum number of iterations. The default is 50.

METHOD=REML
METHOD=ML
METHOD=MIVQUE0
METHOD=TYPE1
METHOD=TYPE2
METHOD=TYPE3

specifies the estimation method for the covariance parameters. The REML specification performs residual (restricted) maximum likelihood, and it is the default method. The ML specification performs maximum likelihood, and the MIVQUE0 specification performs minimum variance quadratic unbiased estimation of the covariance parameters.

The METHOD=TYPEn specifications apply only to variance component models with no SUBJECT= effects and no REPEATED statement. An analysis of variance table is included in the output, and the expected mean squares are used to estimate the variance components (see Chapter 32, “The GLM Procedure,” for further explanation). The resulting method-of-moment variance component estimates are used in subsequent calculations, including standard errors computed from ESTIMATE and LSMEANS statements. For ODS purposes, the new table names are “Type1,” “Type2,” and “Type3,” respectively.

MMEQ

requests that coefficients of the mixed model equations be displayed. These are

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}+\widehat{\mathbf{G}}^{-1} \end{bmatrix}, \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

assuming that $\widehat{\mathbf{G}}$ is nonsingular. If $\widehat{\mathbf{G}}$ is singular, PROC MIXED produces the following coefficients

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}}+\widehat{\mathbf{G}} \end{bmatrix}, \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

See the “Estimating β and γ in the Mixed Model” section on page 2739 for further information on these equations.

MMEQSOL

requests that a solution to the mixed model equations be produced, as well as the inverted coefficients matrix. Formulas for these equations are provided in the preceding description of the MMEQ option.

When $\widehat{\mathbf{G}}$ is singular, $\widehat{\boldsymbol{\tau}}$ and a generalized inverse of the left-hand-side coefficient matrix are transformed using $\widehat{\mathbf{G}}$ to produce $\widehat{\gamma}$ and $\widehat{\mathbf{C}}$, respectively, where $\widehat{\mathbf{C}}$ is a generalized inverse of the left-hand-side coefficient matrix of the original equations.

NAMELEN<=*number*>

specifies the length to which long effect names are shortened. The default and minimum value is 20.

NOBOUND

has the same effect as the NOBOUND option in the PARMS statement (see page 2707).

NOCLPRINT<=*number*>

suppresses the display of the “Class Level Information” table if you do not specify *number*. If you do specify *number*, only levels with totals that are less than *number* are listed in the table.

NOINFO

suppresses the display of the “Model Information”, “Dimensions”, and “Number of Observations” tables.

NOITPRINT

suppresses the display of the “Iteration History” table.

NOPROFILE

includes the residual variance as part of the Newton-Raphson iterations. This option applies only to models that have a residual variance parameter. By default, this parameter is profiled out of the likelihood calculations, except when you have specified the HOLD= or NOITER option in the PARMS statement.

ORD

displays ordinates of the relevant distribution in addition to *p*-values. The ordinate can be viewed as an approximate odds ratio of hypothesis probabilities.

ORDER=DATA
ORDER=FORMATTED
ORDER=FREQ
ORDER=INTERNAL

specifies the sorting order for the levels of all CLASS variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use CONTRAST or ESTIMATE statements.

The default is ORDER=FORMATTED, and its behavior has been modified for Version 8. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values. In order to revert to the previous method you can specify this format explicitly for the CLASS variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and required you to use an explicit format or ORDER=INTERNAL to get the more natural ordering.

The following table shows how PROC MIXED interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter on the SORT procedure in the *SAS*

Procedures Guide and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

RATIO

produces the ratio of the covariance parameter estimates to the estimate of the residual variance when the latter exists in the model.

RIDGE=*number*

specifies the starting value for the minimum ridge value used in the Newton-Raphson algorithm. The default is 0.3125.

SCORING<=*number*>

requests that Fisher scoring be used in association with the estimation method up to iteration *number*, which is 0 by default. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute approximate standard errors for the covariance parameters instead of the observed Hessian. The output from the ASYCOV and ASYCORR options is similarly adjusted.

SIGITER

is an alias for the NOPROFILE option.

UPDATE

is an alias for the LOGNOTE option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MIXED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the MIXED procedure. The NOTSORTED option does not mean that the data are unsorted but rather means that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

Since sorting the data changes the order in which PROC MIXED reads observations, the sorting order for the levels of the CLASS variable may be affected if you have specified ORDER=DATA in the PROC MIXED statement. This, in turn, affects specifications in the CONTRAST statements.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC MIXED statement.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE MIXED specifies that class levels should be determined using only no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

CONTRAST Statement

CONTRAST *'label'* < *fixed-effect values . . .* >
< | *random-effect values . . .* > , . . . < */ options* > **;**

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC GLM, although it has been extended to include random effects. This enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991).

You can test the hypothesis $\mathbf{L}'\phi = 0$, where $\mathbf{L}' = (\mathbf{K}' \ \mathbf{M}')$ and $\phi' = (\beta' \ \gamma')$, in several inference spaces. The inference space corresponds to the choice of $\mathbf{M}$. When $\mathbf{M} = \mathbf{0}$, your inferences apply to the entire population from which the random effects are sampled; this is known as the *broad* inference space. When all elements of $\mathbf{M}$ are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the *narrow* inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by zeroing portions of $\mathbf{M}$ corresponding to selected main effects and interactions, you can choose *intermediate* inference spaces. The broad inference space is usually the most appropriate, and it is used when you do not specify any random effects in the CONTRAST statement.

In the CONTRAST statement,

label identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.

fixed-effect identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

random-effect identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (|); however, random effects do not have to be specified.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the fixed and random effects.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas. The rows of the $\mathbf{K}'$ component of $\mathbf{L}'$ are specified on the left side of the vertical bars (|). These rows test the fixed effects and are, therefore, checked for estimability. The rows of the $\mathbf{M}'$ component of $\mathbf{L}'$ are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC MIXED finds the fixed-effects portion of the specified contrast to be nonestimable (see the SINGULAR= option on page 2684), then it displays “Non-est” for the contrast entries.

The following CONTRAST statement reproduces the *F*-test for the effect A in the split-plot example (see Example 46.1 on page 2777):

```
contrast 'A broad'
         A  1 -1 0    A*B  .5 .5 -.5 -.5 0 0 ,
         A  1 0 -1    A*B  .5 .5 0 0 -.5 -.5 / df=6;
```

Note that no random effects are specified in the preceding contrast; thus, the inference space is broad. The resulting *F*-test has two numerator degrees of freedom because $\mathbf{L}'$ has two rows. The denominator degrees of freedom is, by default, the residual degrees of freedom (9), but the DF= option changes the denominator degrees of freedom to 6.

The following CONTRAST statement reproduces the *F*-test for A when Block and A*Block are considered fixed effects (the narrow inference space):

```
contrast 'A narrow'
          A        1 -1 0
          A*B     .5 .5 -.5 -.5 0 0 |
          A*Block .25  .25  .25  .25
                 -.25 -.25 -.25 -.25
                   0    0    0    0 ,
          A        1 0 -1
          A*B     .5 .5 0 0 -.5 -.5 |
          A*Block .25  .25  .25  .25
                   0    0    0    0
                 -.25 -.25 -.25 -.25 ;
```

The preceding contrast does not contain coefficients for B and Block because they cancel out in estimated differences between levels of A. Coefficients for B and Block are necessary when estimating the mean of one of the levels of A in the narrow inference space (see Example 46.1 on page 2777).

If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are automatically "filled in" over the levels of the higher-order effect. This feature is designed to preserve estimability for cases when there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect as in the construction of least squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in **L** are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the RANDOM statement, then the coefficients specified for the effects in the RANDOM statement are equitably distributed across the levels of the SUBJECT effect. You can use the E option to see exactly what **L** matrix is used.

The SUBJECT and GROUP options in the CONTRAST statement are useful for the case when a SUBJECT= or GROUP= variable appears in the RANDOM statement, and you want to contrast different subjects or groups. By default, CONTRAST statement coefficients on random effects are distributed equally across subjects and groups.

PROC MIXED handles missing level combinations of classification variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{\mathrm{rank}(\mathbf{L})}$$

and approximates its distribution with an *F*-distribution. In this expression, $\widehat{\mathbf{C}}$ is an estimate of the generalized inverse of the coefficient matrix in the mixed model equations. See the "Inference and Test Statistics" section on page 2741 for more information on this *F*-statistic.

The numerator degrees of freedom in the *F*-approximation is *rank*(**L**), and the denominator degrees of freedom is taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the DF= option.

You can specify the following options in the CONTRAST statement after a slash (/).

CHISQ

requests that χ^2-tests be performed in addition to any *F*-tests. A χ^2-statistic equals its corresponding F-statistic times the associate numerator degrees of freedom, and this same degrees of freedom is used to compute the p-value for the χ^2-test. This p-value will always be less than that for the F-test, as it effectively corresponds to an F-test with infinite denominator degrees of freedom.

DF=*number*

specifies the denominator degrees of freedom for the *F*-test. The default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement.

E

requests that the **L** matrix coefficients for the contrast be displayed. For ODS purposes, the label of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*
GRP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across groups.

SINGULAR=*number*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the absolute value of the element of **v** with the largest absolute value. If ABS($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than C**number* for any row of $\mathbf{K}'$ in the contrast, then $\mathbf{K}$ is declared nonestimable. Here $\mathbf{T}$ is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, and C is ABS($\mathbf{K}'$) except when it equals 0, and then C is 1. The value for *number* must be between 0 and 1; the default is 1E−4.

SUBJECT *coeffs*
SUB *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears on the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across subjects.

ESTIMATE Statement

ESTIMATE *'label'* < *fixed-effect values . . .* >
< | *random-effect values . . .* > , *. . .* < */ options* > **;**

The ESTIMATE statement is exactly like a CONTRAST statement, except only one-row $\mathbf{L}$ matrices are permitted. The actual estimate, $\mathbf{L}'\widehat{\mathbf{p}}$, is displayed along with its approximate standard error. An approximate t-test that $\mathbf{L}'\widehat{\mathbf{p}} = 0$ is also produced.

PROC MIXED selects the degrees of freedom to match those displayed in the "Tests of Fixed Effects" table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom using the DF= option.

If PROC MIXED finds the fixed-effects portion of the specified estimate to be non-estimable, then it displays "Non-est" for the estimate entries.

The following examples of ESTIMATE statements compute the mean of the first level of A in the split-plot example (see Example 46.1 on page 2777) for various inference spaces.

```
estimate 'A1 mean narrow'   intercept 1
                            A 1 B .5 .5 A*B .5 .5 |
                            block .25 .25 .25 .25
                            A*Block .25 .25 .25 .25
                                    0 0 0 0
                                    0 0 0 0;
estimate 'A1 mean intermed' intercept 1
                            A 1 B .5 .5 A*B .5 .5 |
                            Block .25 .25 .25 .25;
estimate 'A1 mean broad'    intercept 1
                            A 1 B .5 .5 A*B .5 .5;
```

The construction of the $\mathbf{L}$ vector for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.

You can specify the following options in the ESTIMATE statement after a slash (/).

ALPHA=*number*

requests that a t-type confidence interval be constructed with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

CL

requests that t-type confidence limits be constructed. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

DF=*number*

specifies the degrees of freedom for the *t*-test and confidence limits. The default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the ESTIMATE statement.

DIVISOR=*number*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integer numerators.

E

requests that the **L** matrix coefficients be displayed. For ODS purposes, the label of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*
GRP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across groups.

LOWER
LOWERTAILED

requests that the *p*-value for the *t*-test be based only on values less than the *t*-statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL option.

SINGULAR=*number*

tunes the estimability checking as documented for the CONTRAST statement.

SUBJECT *coeffs*
SUB *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across subjects.

For example, the ESTIMATE statement in the following code from Example 46.5 constructs the difference between the random slopes of the first two batches.

```
proc mixed data=rc;
   class batch;
   model y = month / s;
   random int month / type=un sub=batch s;
   estimate 'slope b1 - slope b2' | month 1 / subject 1 -1;
run;
```

UPPER
UPPERTAILED

requests that the *p*-value for the *t*-test be based only on values greater than the *t*-statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the CL option.

ID Statement

ID *variables* **;**

The ID statement specifies which variables from the input data set are to be included in the OUTP= and OUTPM= data sets from the MODEL statement. If you do not specify an ID statement, then all variables are included in these data sets. Otherwise, only the variables you list in the ID statement are included. Specifying an ID statement with no variables prevents any variables from being included in these data sets.

LSMEANS Statement

LSMEANS *fixed-effects* < */ options* > **;**

The LSMEANS statement computes least-squares means (LS-means) of fixed effects. As in the GLM procedure, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs. The $\mathbf{L}$ matrix constructed to compute them is the same as the $\mathbf{L}$ matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model.

Each LS-mean is computed as $\mathbf{L}\widehat{\boldsymbol{\beta}}$ where $\mathbf{L}$ is the coefficient matrix associated with the least-squares mean and $\widehat{\boldsymbol{\beta}}$ is the estimate of the fixed-effects parameter vector (see the "Estimating β and γ in the Mixed Model" section on page 2739). The approximate standard errors for the LS-mean is computed as the square root of $\mathbf{L}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}'$.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. As in the ESTIMATE statement, the $\mathbf{L}$ matrix is tested for estimability, and if this test fails, PROC MIXED displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC MIXED constructs an approximate *t*-test to test the null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the "Tests of Fixed Effects" table (see the "Default Output" section on page 2748).

You can specify the following options in the LSMEANS statement after a slash (/).

ADJUST=BON
ADJUST=DUNNETT
ADJUST=SCHEFFE
ADJUST=SIDAK
ADJUST=SIMULATE<*(simoptions)*>
ADJUST=SMM | GT2
ADJUST=TUKEY

requests a multiple comparison adjustment for the *p*-values and confidence limits for the differences of LS-means. By default, PROC MIXED adjusts all pairwise differences unless you specify ADJUST=DUNNETT, in which case PROC MIXED analyzes all differences with a control level. The ADJUST= option implies the DIFF option (see page 2690).

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in Chapter 32, "The GLM Procedure," and Chapter 48, "The MULTTEST Procedure," ; also refer to Westfall and Young (1993) and Westfall *et al.* (1999). When you specify ADJUST=TUKEY and your data are unbalanced, PROC MIXED uses the approximation described in Kramer (1956). Similarly, when you specify ADJUST=DUNNETT and the LS-means are correlated, PROC MIXED uses the factor-analytic covariance approximation described in Hsu (1992). The preceding references also describe the SCHEFFE and SMM adjustments.

The SIMULATE adjustment computes adjusted *p*-values and confidence limits from the simulated distribution of the maximum or maximum absolute value of a multivariate *t* random vector. All covariance parameters except the residual variance are fixed at their estimated values throughout the simulation, potentially resulting in some underdispersion. The simulation estimates q, the true $(1-\alpha)$th quantile, where $1-\alpha$ is the confidence coefficient. The default α is 0.05, and you can change this value with the ALPHA= option in the LSMEANS statement.

The number of samples is set so that the tail area for the simulated q is within γ of $1-\alpha$ with $100(1-\epsilon)$% confidence. In equation form,

$$P(|F(\widehat{q})-(1-\alpha)|\leq\gamma) \quad = \quad 1-\epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; refer to Edwards and Berry (1987) for details. By default, γ = 0.005 and ϵ = 0.01, placing the tail area of $\hat{q}$ within 0.005 of 0.95 with 99% confidence. The ACC= and EPS= *simoptions* reset γ and ϵ, respectively; the NSAMP= *simoption* sets the sample size directly; and the SEED= *simoption* specifies an integer used to start the pseudorandom number generator for the simulation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. For additional descriptions of these and other simulation options, see the "LSMEANS Statement" section on page 1753 in Chapter 32, "The GLM Procedure."

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the LS-means with confidence level $1-$ *number*. The value of *number* must be between 0 and 1; the default is 0.05.

AT ***variable*** **=** ***value***
AT ***(variable-list)*** **=** ***(value-list)***
AT MEANS

enables you to modify the values of the covariates used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to assign arbitrary values to the covariates. Additional columns in the output table indicate the values of the covariates.

If there is an effect containing two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option sets covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to cross products of covariates.

As an example, consider the following invocation of PROC MIXED:

```
proc mixed;
   class A;
   model Y = A X1 X2 X1*X2;
   lsmeans A;
   lsmeans A / at means;
   lsmeans A / at X1=1.2;
   lsmeans A / at (X1 X2)=(1.2 0.3);
run;
```

For the first two LSMEANS statements, the LS-means coefficient for X1 is $\overline{x_1}$ (the mean of X1) and for X2 is $\overline{x_2}$ (the mean of X2). However, for the first LSMEANS statement, the coefficient for X1*X2 is $\overline{x_1 x_2}$, but for the second LSMEANS statement, the coefficient is $\overline{x_1} \cdot \overline{x_2}$. The third LSMEANS statement sets the coefficient for X1 equal to 1.2 and leaves it at $\overline{x_2}$ for X2, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

If a WEIGHT variable is present, it is used in processing AT variables. Also, observations with missing dependent variables are included in computing the covariate means, unless these observations form a missing cell and the FULLX option in the MODEL statement is not in effect. You can use the E option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you desire.

The AT option is disabled if you specify the BYLEVEL option.

BYLEVEL

requests PROC MIXED to process the OM data set by each level of the LS-mean effect (LSMEANS effect) in question. For more details, see the OM option later in this section.

CL

requests that *t*-type confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR

displays the estimated correlation matrix of the least-squares means as part of the “Least Squares Means” table.

COV

displays the estimated covariance matrix of the least-squares means as part of the “Least Squares Means” table.

DF=*number*

specifies the degrees of freedom for the *t*-test and confidence limits. The default is the denominator degrees of freedom taken from the “Tests of Fixed Effects” table corresponding to the LS-means effect unless the DDFM=SATTERTH or DDFM=KENWARDROGER option is in effect on the MODEL statement. For these DDFM= methods degrees of freedom are determined separately for each test; see the DDFM= option on page 2693 for more information.

DIFF<=*difftype*>
PDIFF<=*difftype*>

requests that differences of the LS-means be displayed. The optional *difftype* specifies which differences to produce, with possible values being ALL, CONTROL, CONTROLL, and CONTROLU. The *difftype* ALL requests all pairwise differences, and it is the default. The *difftype* CONTROL requests the differences with a control, which, by default, is the first level of each of the specified LSMEANS effects.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are class variables, each having two levels, 1 and 2, the following LSMEANS statement specifies the (1,2) level of A*B and the (2,1) level of B*C as controls:

```
lsmeans A*B B*C / diff=control('1' '2' '2' '1');
```

For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL *difftype*. For one-tailed results, use either the CONTROLL or CONTROLU *difftype*. The CONTROLL *difftype* tests whether the noncontrol levels are significantly smaller than the control; the upper confidence limits for the control minus the noncontrol levels are considered to be infinity and are displayed as missing. Conversely, the CONTROLU *difftype* tests whether the noncontrol levels are significantly larger than the control; the upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.

If you want to perform multiple comparison adjustments on the differences of LS-Means, you must specify the ADJUST= option.

The differences of the LS-means are displayed in a table titled “Differences of Least Squares Means.” For ODS purposes, the table name is “Diffs.”

E

requests that the **L** matrix coefficients for all LSMEANS effects be displayed. For ODS purposes, the label of this “**L** Matrix Coefficients” table is “Coef.”

OM<=*OM-data-set*>
OBSMARGINS<=*OM-data-set*>

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in *OM-data-set*. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in *OM-data-set*.

By default, *OM-data-set* is the same as the analysis data set. You can optionally specify another data set that describes the population for which you want to make inferences. This data set must contain all model variables except for the dependent variable (which is ignored if it is present). In addition, the levels of all CLASS variables must be the same as those occurring in the analysis data set. Specifying an *OM-data-set* enables you to construct arbitrarily weighted LS-means.

In computing the observed margins, PROC MIXED uses all observations for which there are no missing or invalid independent variables, including those for which there are missing dependent variables. Also, if *OM-data-set* has a WEIGHT variable, PROC MIXED uses weighted margins to construct the LS-means coefficients. If *OM-data-set* is balanced, the LS-means are unchanged by the OM option.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across all of the *OM-data-set*, PROC MIXED computes separate margins for each level of the LSMEANS effect in question. In this case the resulting LS-means are actually equal to raw means for fixed effects models and certain balanced random effects models, but their estimated standard errors account for the covariance structure that you have specified. If the AT option is specified, the BYLEVEL option disables it.

You can use the E option in conjunction with either the OM or BYLEVEL option to check that the modified LS-means coefficients are the ones you desire. It is possible that the modified LS-means are not estimable when the standard ones are, or vice versa. Nonestimable LS-means are noted as “Non-est” in the output.

PDIFF

is the same as the DIFF option. See the description of the DIFF option on page 2690.

SINGULAR=*number*

tunes the estimability checking as documented on the “CONTRAST Statement” section on page 2681.

SLICE= *fixed-effect*
SLICE= *(fixed-effects)*

specifies effects by which to partition interaction LSMEANS effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant, and you want to test the effect of A for each level of B. The appropriate LSMEANS statement is

```
lsmeans A*B / slice=B;
```

This code tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and using them to form an *F*-test. See the “Inference and Test Statistics” section on page 2741 for more information on this *F*-test.

The SLICE option produces a table titled “Tests of Effect Slices.” For ODS purposes, the table name is “Slices.”

MODEL Statement

MODEL *dependent* = < *fixed-effects* >< / *options* >**;**

The MODEL statement names a single dependent variable and the fixed effects, which determine the $\mathbf{X}$ matrix of the mixed model (see the "Parameterization of Mixed Models" section on page 2743 for details). The specification of effects is the same as in the GLM procedure; however, unlike PROC GLM, you do not specify random effects in the MODEL statement. The MODEL statement is required.

An intercept is included in the fixed-effects model by default. If no fixed effects are specified, only this intercept term is fit. The intercept can be removed by using the NOINT option.

You can specify the following options in the MODEL statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the fixed-effects parameters with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

ALPHAP=*number*

requests that a *t*-type confidence interval be constructed for the predicted values with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

CHISQ

requests that χ^2-tests be performed for all specified effects in addition to the *F*-tests. Type III tests are the default; you can produce the Type I and Type II tests using the HTYPE= option.

CL

requests that *t*-type confidence limits be constructed for each of the fixed-effects parameter estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CONTAIN

has the same effect as the DDFM=CONTAIN option.

CORRB

produces the approximate correlation matrix of the fixed-effects parameter estimates. For ODS purposes, the label for this table is "CorrB."

COVB

produces the approximate variance-covariance matrix of the fixed-effects parameter estimates $\widehat{\boldsymbol{\beta}}$. By default, this matrix equals $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$ and results from sweeping $(\mathbf{X}\ \mathbf{y})'\widehat{\mathbf{V}}^{-1}(\mathbf{X}\ \mathbf{y})$ on all but its last pivot and removing the y border. The EMPIRICAL option in the PROC MIXED statement changes this matrix into "empirical sandwich" form, as described on page 2676. For ODS purposes, the label for this table is "CovB."

COVBI

produces the inverse of the approximate variance-covariance matrix of the fixed-effects parameter estimates. For ODS purposes, the label for this table is “InvCovB.”

DDF=*value-list*

enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the “Tests of Fixed Effects” table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. For example,

```
model Y = A B A*B / ddf=3,.,4.7;
```

assigns 3 denominator degrees of freedom to A and 4.7 to A*B, while those for B remain the same. If you specify DDFM=SATTERTH or DDFM=KENWARDROGER the DDF= option has no effect.

DDFM=CONTAIN
DDFM=BETWITHIN
DDFM=RESIDUAL
DDFM=SATTERTH
DDFM=KENWARDROGER

specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, and LSMEANS statements.

The DDFM=CONTAIN option invokes the *containment method* to compute denominator degrees of freedom, and it is the default when you specify a RANDOM statement. The containment method is carried out as follows: Denote the fixed effect in question A, and search the RANDOM effect list for the effects that *syntactically* contain A. For example, the RANDOM effect B(A) contains A, but the RANDOM effect C does not, even if it has the same levels as B(A).

Among the RANDOM effects that contain A, compute their rank contribution to the ($\mathbf{X}\ \mathbf{Z}$) matrix. The DDF assigned to A is the smallest of these rank contributions. If no effects are found, the DDF for A is set equal to the residual degrees of freedom, $N - \text{rank}(\mathbf{X}\ \mathbf{Z})$. This choice of DDF matches the tests performed for balanced split-plot designs and should be adequate for moderately unbalanced designs.

Caution: If you have a $\mathbf{Z}$ matrix with a large number of columns, the overall memory requirements and the computing time after convergence can be substantial for the containment method. If it is too large, you may want to use the DDFM=BETWITHIN option.

The DDFM=BETWITHIN option is the default for REPEATED statement specifications (with no RANDOM statements). It is computed by dividing the residual degrees of freedom into between-subject and within-subject portions. PROC MIXED then checks whether a fixed effect changes within any subject. If so, it assigns within-subject degrees of freedom to the effect; otherwise, it assigns the between-subject degrees of freedom to the effect (refer to Schluchter and Elashoff 1990). If there are multiple within-subject effects containing classification variables, the within-subject degrees of freedom is partitioned into components corresponding to the subject-by-effect interactions.

One exception to the preceding method is the case when you have specified no RANDOM statements and a REPEATED statement with the TYPE=UN option. In this case, all effects are assigned the between-subject degrees of freedom to provide for better small-sample approximations to the relevant sampling distributions. DDFM=KENWARDROGER may be a better option to try for this case.

The DDFM=RESIDUAL option performs all tests using the residual degrees of freedom, $n - \mathrm{rank}(\mathbf{X})$, where n is the number of observations.

The DDFM=SATTERTH option performs a general Satterthwaite approximation for the denominator degrees of freedom, computed as follows. Suppose θ is the vector of unknown parameters in V and suppose $C = (X'V^{-1}X)^{-}$, where $^{-}$ denotes a generalized inverse. Let $\hat{C}$ and $\hat{\theta}$ be the corresponding estimates.

Consider the one-dimensional case, and consider ℓ to be a vector defining an estimable linear combination of β. The Satterthwaite degrees of freedom for the t-statistic

$$t = \frac{\ell\hat{\beta}}{\sqrt{\ell\hat{C}\ell'}}$$

is computed as

$$\nu = \frac{2(\ell\hat{C}\ell')^2}{g'Ag}$$

where g is the gradient of $\ell C\ell'$ with respect to θ, evaluated at $\hat{\theta}$, and A is the asymptotic variance-covariance matrix of $\hat{\theta}$ obtained from the second derivative matrix of the likelihood equations.

For the multi-dimensional case, let L be an estimable contrast matrix of rank $q > 1$. The Satterthwaite denominator degrees of freedom for the F-statistic

$$F = \frac{\hat{\beta}'L'(L\hat{C}L')^{-1}L\hat{\beta}}{q}$$

is computed by first performing the spectral decomposition $L\hat{C}L' = P'DP$ where P is an orthogonal matrix of eigenvectors and D is a diagonal matrix of eigenvalues, both of dimension $q \times q$. Define ℓ_m to be the mth row of PL, and let

$$\nu_m = \frac{2(D_m)^2}{g'_m A g_m}$$

where D_m is the mth diagonal element of D and g_m is the gradient of $\ell_m C\ell'_m$ with respect to θ, evaluated at $\hat{\theta}$. Then let

$$E = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2} I(\nu_m > 2)$$

where the indicator function eliminates terms for which $\nu_m \leq 2$. The degrees of freedom for F are then computed as

$$\nu = \frac{2E}{E - q}$$

provided $E > q$; otherwise ν is set to zero.

This method is a generalization of the techniques described in Giesbrecht and Burns (1985), McLean and Sanders (1988), and Fai and Cornelius (1996). The method can also include estimated random effects. In this case, append $\hat{\gamma}$ to $\hat{\beta}$ and change $\hat{C}$ to be the inverse of the coefficient matrix in the mixed model equations. The calculations require extra memory to hold c matrices that are the size of the mixed model equations, where c is the number of covariance parameters. In the notation of Table 46.12 on page 2773, this is approximately $8q(p+g)(p+g)/2$ bytes. Extra computing time is also required to process these matrices. The Satterthwaite method implemented here is intended to produce an accurate *F*-approximation; however, the results may differ from those produced by PROC GLM. Also, the small sample properties of this approximation have not been extensively investigated for the various models available with PROC MIXED.

The DDFM=KENWARDROGER option performs the degrees-of-freedom calculations detailed by Kenward and Roger (1997). This approximation involves inflating the estimated variance-covariance matrix of the fixed and random effects by the method proposed by Prasad and Rao (1990) and Harville and Jeske (1992); refer also to Kackar and Harville (1984). Satterthwaite-type degrees of freedom are then computed based on this adjustment. By default, the observed information matrix of the covariance parameter estimates is used in the calculations.

When the asymptotic variance matrix of the covariance parameters is found to be singular, a generalized inverse is used. Covariance parameters with zero variance then do not contribute to the degrees-of-freedom adjustment for DDFM=SATTERTH and DDFM=KENWARDROGER, and a message is written to the LOG.

This method changes output in the following tables (listed in Table 46.8 on page 2752): Contrast, CorrB, CovB, Diffs, Estimates, InvCovB, LSMeans, MMEq, MMEqSol, Slices, SolutionF, SolutionR, Tests1–Tests3. The OUTP= and OUTPM= data sets are also affected.

E

requests that Type I, Type II, and Type III **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the labels of the tables are “Coef.”

E1

requests that Type I **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the label of this table is “Coef.”

E2

requests that Type II **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the label of this table is “Coef.”

E3

requests that Type III **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the label of this table is “Coef.”

FULLX

requests that columns of the **X** matrix that consist entirely of zeros not be eliminated from **X**; otherwise, they are eliminated by default. For a column corresponding to a missing cell to be added to **X**, its particular levels must be present in at least one observation in the analysis data set along with a missing dependent variable. The use of the FULLX option can impact coefficient specifications in the CONTRAST and ESTIMATE statements, as well as covariate coefficients from LSMEANS statements specified with the AT MEANS option.

HTYPE=*value-list*

indicates the type of hypothesis test to perform on the fixed effects. Valid entries for *value* are 1, 2, and 3; the default value is 3. You can specify several types by separating the values with a comma or a space. The ODS table names are “Tests1” for the Type 1 tests, “Tests2” for the Type 2 tests, and “Tests3” for Type 3 tests.

Experimental

INFLUENCE<(<EFFECT=*effect***>**
<ESTIMATES|EST>
<ITER=*number***>**
<KEEP=*number***>**
<SELECT=*value-list***>**
<SIZE=*number***>)>**

specifies that influence and case deletion diagnostics are to be computed.

The INFLUENCE option of the MODEL statement in the MIXED procedure computes influence diagnostics by noniterative or iterative methods. The noniterative diagnostics rely on recomputation formulas under the assumption that covariance parameters or their ratios remain fixed. With the possible exception of a profiled residual variance, no covariance parameters are updated. This is the default behavior because of its computational efficiency. However, the impact of an observation on the overall analysis can be underestimated if its effect on covariance parameters is not assessed. Toward this end, iterative methods can be applied to gauge the overall impact of observations and to obtain influence diagnostics for the covariance parameter estimates.

If you specify the INFLUENCE option without further suboptions, PROC MIXED computes single-case deletion diagnostics and influence statistics for each observation in the data set by updating estimates for the fixed effects parameter estimates, and also the residual variance, if it is profiled. The EFFECT=, SELECT=, ITER=, SIZE=, and KEEP= suboptions provide additional flexibility in the computation and reporting of influence statistics.

Description	**Suboption**
Compute influence diagnostics for individual observations	default
Measure influence of sets of observations chosen according to a classification variable or effect	EFFECT=
Remove pairs of observations and report the results sorted by degree of influence	SIZE=2
Remove triples, quadruples of observations,...	SIZE=
Allow selection of individual observations, observations sharing specific levels of effects, and construction of tuples from specified subsets of observations	SELECT=
Update fixed effects and covariance parameters by refitting the mixed model, adding up to n iterations	ITER=$n > 0$
Compute influence diagnostics for the covariance parameters	ITER=$n > 0$
Update only fixed effects and the residual variance, if it is profiled	ITER=0
Add the reduced-data estimates to the data set created with ODS OUTPUT	ESTIMATES

The modifiers and their default values are discussed in the following paragraphs. The set of computed influence diagnostics varies with the suboptions. The most extensive set of influence diagnostics is obtained when ITER=n with $n > 0$.

You can produce statistical graphics of influence diagnostics when the experimental ODS GRAPHICS statement is specified. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the "ODS Graphics" section on page 2757.

EFFECT=*effect* specifies an effect according to which observations are grouped. Observations sharing the same level of the *effect* are removed from the analysis as a group. The *effect* must contain only class variables, but need not be contained in the model.

Removing observations can change the rank of the $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^-$ matrix. This is particularly likely to happen when multiple observations are eliminated from the analysis. If the rank of the estimated variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ changes or its singularity pattern is altered, no influence diagnostics are computed.

ESTIMATES|EST specifies that the updated parameter estimates should be written to the ODS output data set. The values are not displayed in the "Influence" table, but if you use ODS OUTPUT to create a data set from the listing, the estimates are added to the data set. If ITER=0, only the fixed effects estimates are saved. In iterative influence analyses, fixed effects and covariance parameters are stored. The p fixed effects parameter estimates are named Parm1–Parm$p$, and the q covariance parameter estimates are named CovP1–CovPq. The order corresponds to that in the “Solution for Fixed Effects” and “Covariance Parameter Estimates” tables. If parameter updates fail, for example, because of a loss of rank or a nonpositive definite Hessian, missing values are reported.

ITER=*n* controls the maximum number of additional iterations PROC MIXED performs to update the fixed effects and covariance parameter estimates following data point removal. If you specify $n > 0$, then statistics such as DFFITS, MDFFITS, and the likelihood distances measure the impact of observation(s) on all aspects of the analysis. Typically, the influence will grow compared to values at ITER=0. In models without RANDOM or REPEATED effects, the ITER= option has no effect.

This documentation refers to analyses when $n > 0$ simply as iterative influence analysis, even if final covariance parameter estimates can be updated in a single step (for example, when METHOD=MIVQUE0 or METHOD=TYPE3). This nomenclature reflects the fact that only if $n > 0$ will all model parameters be updated, which may require additional iterations. If $n > 0$ and METHOD=REML (default) or METHOD=ML, the procedure updates fixed effects *and* variance-covariance parameters after removing the selected observations with additional Newton-Raphson iterations, starting from the converged estimates for the entire data. The process stops for each observation or set of observations if the convergence criterion is satisfied or the number of further iterations exceeds *n*. If *n* > 0 and METHOD=TYPE1, TYPE2, or TYPE3, ANOVA estimates of the covariance parameters are recomputed in a single step.

Compared to noniterative updates the computations are more involved. In particular for large data sets and/or a large number of random effects, iterative updates require considerably more resources. A one-step (ITER=1) or two-step update may be a good compromise. The output includes the number of iterations performed, which is less than n if the iteration converges. If the process does not converge in n iterations, you should be careful in interpreting the results, especially if n is fairly large.

Bounds and other restrictions on the covariance parameters carry over from the full-data model. Covariance parameters that are not iterated in the model fit to the full data (the NOITER or HOLD option of the PARMS statement) are likewise not updated in the refit. In certain models, for example, random effects models, the ratios between the covariance parameters and the residual variance are maintained rather than the actual value of the covariance parameter estimate (see the section “Influence Diagnostics” on page 2765 in the "Details" section).

KEEP=*n* determines how many observations are retained for display and in the output data set or how many tuples if you specify SIZE=. The output is sorted by an influence statistic as discussed for the SIZE= suboption.

SELECT = *value-list* specifies which observations or effect levels are chosen for influence calculations. If SELECT= is not specified, diagnostics are computed for all possible subsets, that is

- all observations, if EFFECT= or SIZE= are not given
- all levels of the specified effect, if EFFECT= is specified
- all tuples of size k formed from the observations in *value-list*, if SIZE=k is specified

When you specify an effect with the EFFECT= option, the values in *value-list* represent indices of the levels in the order in which PROC MIXED builds classification effects. Which observations in the data set correspond to this index depends on the order of the variables in the CLASS statement, not the order in which the variables appear in the interaction effect. See the section "Parameterization of Mixed Models" on page 2743 on precisely how the procedure indexes nested and crossed effects and how levels of classification variables are ordered. The actual values of the classification variables involved in the effect are shown on the output so you can determine which observations were removed.

If the EFFECT= suboption is not specified, the SELECT= value list refers to the sequence in which observations are read from the input data set or from the current BY group if there is a BY statement. This indexing is not necessarily the same as the observation numbers in the input data set, for example, if a WHERE clause is specified or during BY processing.

SIZE= *n* instructs PROC MIXED to remove groups of observations formed as tuples of size *n*. For example, SIZE=2 specifies all $n \times (n-1)/2$ unique pairs of observations. The number of tuples for SIZE=k is $n!/(k!(n-k)!)$ and grows quickly with n and k. Using the SIZE= option can result in considerable computing time. The MIXED procedure displays by default only the 50 tuples with the greatest influence. Use the KEEP= option to override this default and to retain a different number of tuples in the listing or ODS output data set. Regardless of the KEEP= specification, all tuples are evaluated and the results are ordered according to an influence statistic. This statistic is the (restricted) likelihood distance as a measure of overall influence if ITER = $n > 0$ or when a residual variance is profiled. When likelihood distances are unavailable, the results are ordered by the PRESS statistic.

To reduce computational burden, the SIZE= option can be combined with the SELECT=*value-list* modifier. For example,

```
proc mixed data=aerosol;
   class aerosol manufacturer filter;
   model penetration = aerosol manufacturer /
      influence(size=2 keep=5
                select=13,14,18,30,31,33);
   random filter(manuf);
run;
```

evaluates all $15 = 6 \times 5/2$ pairs formed from observations 13, 14, 18, 30, 31, and 33 and displays the five pairs with the greatest influence. If any observation in a tuple contains missing values or has otherwise not contributed to the analysis, the tuple is not evaluated. This guarantees that the displayed results refer to the same number of observations, so that meaningful statistics are available by which to order the results. If computations fail for a particular tuple, for example, because the $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$ matrix changes rank or the $\mathbf{G}$ matrix is not positive definite, no results are produced. Results are retained when the maximum number of iterative updates is exceeded in iterative influence analyses.

The SIZE= suboption cannot be combined with the EFFECT= suboption.

As in the case of the EFFECT= suboption, the statistics being computed are those appropriate for removal of multiple data points, even if SIZE=1.

For ODS purposes the label of the “Influence Diagnostics” table is “Influence.” The variables in this table depend on whether you specify the EFFECT=, SIZE=, or KEEP= suboption and whether covariance parameters are iteratively updated. When ITER=0 (the default) certain influence diagnostics are only meaningful if the residual variance is profiled. Table 46.3 and Table 46.4 summarize the statistics obtained depending on the model and modifiers. The last column in these tables gives the variable name in the ODS OUTPUT INFLUENCE= data set. Restricted likelihood distances are reported instead of the likelihood distance unless METHOD=ML. See the “Influence Diagnostics” section beginning on page 2765 for details on the individual statistics.

Table 46.3. Statistics Computed with INFLUENCE Option, Noniterative Analysis (ITER=0)

Suboption	σ^2 **profiled**	**Statistic**	**Variable Name**
Default	Yes	Observed value	Observed
		Predicted value	Predicted
		Residual	Residual
		Leverage	Leverage
		PRESS residual	PRESSRes
		Internally studentized residual	Student
		Externally studentized residual	RStudent
		RMSE without deleted obs	RMSE

Table 46.3. (continued)

Suboption	σ^2 **profiled**	**Statistic**	**Variable Name**
		Cook's D	CookD
		DFFITS	DFFITS
		COVRATIO	COVRATIO
		(Restricted) likelihood distance	RLD, LD
Default	No	Observed value	Observed
		Predicted value	Predicted
		Residual	Residual
		Leverage	Leverage
		PRESS residual	PRESSRes
		Internally studentized residual	Student
		Cook's D	CookD
EFFECT=, SIZE=, or KEEP=	Yes	Observations in level (tuple)	Nobs
		PRESS statistic	PRESS
		Cook's D	CookD
		MDFFITS	MDFFITS
		COVRATIO	COVRATIO
		COVTRACE	COVTRACE
		RMSE without deleted level (tuple)	RMSE
		(Restricted) likelihood distance	RLD, LD
EFFECT=, SIZE=, or KEEP=	No	Observations in level (tuple)	Nobs
		PRESS statistic	PRESS
		Cook's D	CookD

Table 46.4. Statistics Computed with INFLUENCE Option, Iterative Analysis (ITER=$n > 0$)

Suboption	**Statistic**	**Variable Name**
Default	Number of iterations	Iter
	Observed value	Observed
	Predicted value	Predicted
	Residual	Residual
	Leverage	Leverage
	PRESS residual	PRESSres
	Internally studentized residual	Student
	Externally studentized residual	RStudent
	RMSE without deleted obs (if possible)	RMSE
	Cook's D	CookD
	DFFITS	DFFITS
	COVRATIO	COVRATIO
	Cook's D CovParms	CookDCP
	COVRATIO CovParms	COVRATIOCP
	(Restricted) likelihood distance	RLD, LD

Table 46.4. (continued)

Suboption	**Statistic**	**Variable Name**
EFFECT=,	Number of iterations	Iter
SIZE=,	PRESS statistic	PRESS
or KEEP=	RMSE without deleted level (tuple)	RMSE
	Cook's D	CookD
	MDFFITS	MDFFITS
	COVRATIO	COVRATIO
	COVTRACE	COVTRACE
	Cook's D CovParms	CookDCP
	COVRATIO CovParms	COVRATIOCP
	(Restricted) likelihood distance	RLD, LD

INTERCEPT

adds a row to the tables for Type 1, 2, and 3 tests corresponding to the overall intercept.

LCOMPONENTS

requests an estimate for each row of the **L** matrix used to form tests of fixed effects. Components corresponding to Type 3 tests are the default; you can produce the Type 1 and Type 2 component estimates with the HTYPE= option.

Tests of fixed effects involve testing of linear hypotheses of the form $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$. The matrix **L** is constructed from Type 1, 2, or 3 estimable functions. By default the MIXED procedure constructs Type 3 tests. In many situations, the individual rows of the matrix **L** represent contrasts of interest. For example, in a one-way classification model, the Type 3 estimable functions define differences of factor level means. In a balanced two-way layout, the rows of **L** correspond to differences of cell means.

For example, if factors A and B have a and b levels, respectively, the statements

```
class A B;
model y = A B x / htype=1,3 lcomponents;
```

produce $(a-1)$ one degree of freedom tests for the rows of **L** associated with the Type 1 and Type 3 estimable functions for factor A, $(b-1)$ tests for the rows of **L** associated with factor B, and a single test for the Type 1 and Type 3 coefficients associated with regressor X.

The denominator degrees of freedom associated with a row of **L** are the same as those in the corresponding "Tests of Fixed Effects" table, except for DDFM=KENWARDROGER and DDFM=SATTERTH. For these degree of freedom methods, the denominator degrees of freedom are computed separately for each row of **L**.

For ODS purposes, the name of the table containing all requested component tests is "LComponents." See Example 46.9 on page 2839 for examples incorporating the LCOMPONENTS option.

NOCONTAIN

has the same effect as the DDFM=RESIDUAL option.

NOINT

requests that no intercept be included in the model. An intercept is included by default.

NOTEST

specifies that no hypothesis tests be performed for the fixed effects.

OUTP=*SAS-data-set*

OUTPRED=*SAS-data-set*

specifies an output data set containing predicted values and related quantities. This option replaces the P option from Version 6.

Predicted values are formed by using the rows from $(\mathbf{X}\ \mathbf{Z})$ as $\mathbf{L}$ matrices. Thus, predicted values from the original data are $\mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{Z}\widehat{\boldsymbol{\gamma}}$. Their approximate standard errors of prediction are formed from the quadratic form of $\mathbf{L}$ with $\widehat{\mathbf{C}}$ defined in the "Statistical Properties" section on page 2740. The L95 and U95 variables provide a *t*-type confidence interval for the predicted values, and they correspond to the L95M and U95M variables from the GLM and REG procedures for fixed-effect models. The residuals are the observed minus the predicted values. Predicted values for data points other than those observed can be obtained by using missing dependent variables in your input data set.

Specifications that have a REPEATED statement with the SUBJECT= option and missing dependent variables compute predicted values using empirical best linear unbiased prediction (EBLUP). Using hats ($\hat{}$) to denote estimates, the EBLUP formula is

$$\hat{\mathbf{m}} = \mathbf{X}_{\mathbf{m}}\hat{\boldsymbol{\beta}} + \hat{\mathbf{C}}_{\mathbf{m}}\hat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

where $\mathbf{m}$ represents a hypothetical realization of a missing data vector with associated design matrix $\mathbf{X}_{\mathbf{m}}$. The matrix $\mathbf{C}_{\mathbf{m}}$ is the model-based covariance matrix between $\mathbf{m}$ and the observed data $\mathbf{y}$, and other notation is as presented in the "Mixed Models Theory" section beginning on page 2731.

The estimated prediction variance is as follows:

$$\begin{aligned}\hat{\text{Var}}(\hat{\mathbf{m}} - \mathbf{m}) &= \hat{\mathbf{V}}_{\mathbf{m}} - \hat{\mathbf{C}}_{\mathbf{m}}\hat{\mathbf{V}}^{-1}\hat{\mathbf{C}}_{\mathbf{m}}^{T} + \\ &\quad [\mathbf{X}_{\mathbf{m}} - \hat{\mathbf{C}}_{\mathbf{m}}\hat{\mathbf{V}}^{-1}\mathbf{X}](\mathbf{X}^{T}\hat{\mathbf{V}}^{-1}\mathbf{X})^{-}[\mathbf{X}_{\mathbf{m}} - \hat{\mathbf{C}}_{\mathbf{m}}\hat{\mathbf{V}}^{-1}\mathbf{X}]^{T}\end{aligned}$$

where $\mathbf{V}_{\mathbf{m}}$ is the model-based variance matrix of $\mathbf{m}$. For further details, refer to Henderson (1984) and Harville (1990). This feature can be useful for forecasting time series or for computing spatial predictions.

By default, all variables from the input data set are included in the OUTP= data set. You can select a subset of these variables using the ID statement.

OUTPM=*SAS-data-set*
OUTPREDM=*SAS-data-set*

specifies an output data set containing predicted means and related quantities. This option replaces the PM option from Version 6.

The output data set is of the same form as that resulting from the OUTP= option, except that the predicted values do not incorporate the EBLUP values $\mathbf{Z}\widehat{\gamma}$. They also do not use the EBLUPs for specifications that have a REPEATED statement with the SUBJECT= option and missing dependent variables. The predicted values are formed as $\mathbf{X}\widehat{\boldsymbol{\beta}}$ in the OUTPM= data set, and standard errors are quadratic forms in the approximate variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ as displayed by the COVB option.

By default, all variables from the input data set are included in the OUTPM= data set. You can select a subset of these variables using the ID statement.

Experimental

RESIDUAL

requests that Pearson-type and (internally) studentized residuals be added to the OUTP= and OUTPM= data sets. Studentized residuals are raw residuals standardized by their estimated standard error. When residuals are internally studentized, the data point in question has contributed to the estimation of the covariance parameter estimates on which the standard error of the residual is based. Externally studentized residuals can be computed with the INFLUENCE option. Pearson-type residuals scale the residual by the standard deviation of the response.

The RESIDUAL option adds the variables PearsonResid and StudentResid to the OUTP= and OUTPM= data sets. In the former case, the standardization draws on the variance of the conditional distribution (given the random effects). The standardization of residuals in the OUTPM= data set draws on the variance of the marginal distribution of the response.

The option has no effect unless the OUTP= or OUTPM= option are specified or you request statistical graphics with the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the MIXED procedure, see the “ODS Graphics” section on page 2757. For computational details about studentized and Pearson residuals in MIXED, see the “Residual Diagnostics” section beginning on page 2763.

SINGULAR=*number*

tunes the sensitivity in sweeping. If a diagonal pivot element is less than D**number* as PROC MIXED sweeps a matrix, the associated column is declared to be linearly dependent upon previous columns, and the associated parameter is set to 0. The value D is the original diagonal element of the matrix. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGCHOL=*number*

tunes the sensitivity in computing Cholesky roots. If a diagonal pivot element is less than D**number* as PROC MIXED performs the Cholesky decomposition on a matrix, the associated column is declared to be linearly dependent upon previous columns and is set to **0**. The value D is the original diagonal element of the matrix. The default for *number* is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGRES=*number*

sets the tolerance for which the residual variance is considered to be zero. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SOLUTION
S

requests that a solution for the fixed-effects parameters be produced. Using notation from the "Mixed Models Theory" section beginning on page 2731, the fixed-effects parameter estimates are $\widehat{\mathbf{b}}$ and their approximate standard errors are the square roots of the diagonal elements of $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$. You can output this approximate variance matrix with the COVB option or modify it with the EMPIRICAL option in the PROC MIXED statement.

Along with the estimates and their approximate standard errors, a *t*-statistic is computed as the estimate divided by its standard error. The degrees of freedom for this *t*-statistic matches the one appearing in the "Tests of Fixed Effects" table under the effect containing the parameter. The "Pr > |t|" column contains the two-tailed p-value corresponding to the *t*-statistic and associated degrees of freedom. You can use the CL option to request confidence intervals for all of the parameters; they are constructed around the estimate by using a radius of the standard error times a percentage point from the *t*-distribution.

VCIRY *Experimental*

requests that responses and marginal residuals be scaled by the inverse Cholesky root of the marginal variance-covariance matrix. The variables ScaledDep and ScaledResid are added to the OUTPM= data set. These quantities can be important in bootstrapping of data or residuals. Examination of the scaled residuals is also helpful in diagnosing departures from normality. Notice that the results of this scaling operation can depend on the order in which the MIXED procedure processes the data.

The VCIRY option has no effect unless you also use the OUTPM= option or you request statistical graphics with the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the "ODS Graphics" section on page 2757.

XPVIX

is an alias for the COVBI option.

XPVIXI

is an alias for the COVB option.

ZETA=*number*

tunes the sensitivity in forming Type III functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E−8.

PARMS Statement

PARMS *(value-list) . . . < / options >* **;**

The PARMS statement specifies initial values for the covariance parameters, or it requests a grid search over several values of these parameters. You must specify the values in the order in which they appear in the "Covariance Parameter Estimates" table.

The *value-list* specification can take any of several forms:

m a single value

$m_1, m_2, \ldots, m_n$ several values

m to *n* a sequence where *m* equals the starting value, *n* equals the ending value, and the increment equals 1

m to *n* by *i* a sequence where *m* equals the starting value, *n* equals the ending value, and the increment equals *i*

m_1, m_2 to m_3 mixed values and sequences

You can use the PARMS statement to input known parameters. Referring to the split-plot example (Example 46.1 on page 2777), suppose the three variance components are known to be 60, 20, and 6. The SAS statements to fix the variance components at these values is as follows:

```
proc mixed data=sp noprofile;
   class Block A B;
   model Y = A B A*B;
   random Block A*Block;
   parms (60) (20) (6) / noiter;
run;
```

The NOPROFILE option requests PROC MIXED to refrain from profiling the residual variance parameter during its calculations, thereby enabling its value to be held at 6 as specified in the PARMS statement. The NOITER option prevents any Newton-Raphson iterations so that the subsequent results are based on the given variance components. You can also specify known parameters of $\mathbf{G}$ using the GDATA= option in the RANDOM statement.

If you specify more than one set of initial values, PROC MIXED performs a grid search of the likelihood surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times. The grid search feature is also useful for exploring the likelihood surface. See Example 46.3 on page 2795.

The results from the PARMS statement are the values of the parameters on the specified grid (denoted by CovP1–CovP*n*), the residual variance (possibly estimated) for models with a residual variance parameter, and various functions of the likelihood.

For ODS purposes, the label of the "Parameter Search" table is "ParmSearch."

You can specify the following options in the PARMS statement after a slash (/).

HOLD=*value-list*
EQCONS=*value-list*

specifies which parameter values PROC MIXED should hold to equal the specified values. For example, the statement

```
parms (5) (3) (2) (3) / hold=1,3;
```

constrains the first and third covariance parameters to equal 5 and 2, respectively.

LOGDETH

evaluates the log determinant of the Hessian matrix for each point specified in the PARMS statement. A Log Det H column is added to the “Parameter Search” table.

LOWERB=*value-list*

enables you to specify lower boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC MIXED uses for the covariance parameters, and each number corresponds to the lower boundary constraint. A missing value instructs PROC MIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes the remaining ones are missing.

An example for which this option is useful is when you want to constrain the $\mathbf{G}$ matrix to be positive definite in order to avoid the more computationally intensive algorithms required when $\mathbf{G}$ becomes singular. The corresponding code for a random coefficients model is as follows:

```
proc mixed;
   class person;
   model y = time;
   random int time / type=fa0(2) sub=person;
   parms / lowerb=1e-4,.,1e-4;
run;
```

Here the FA0(2) structure is used in order to specify a Cholesky root parameterization for the 2×2 unstructured blocks in $\mathbf{G}$. This parameterization ensures that the $\mathbf{G}$ matrix is nonnegative definite, and the PARMS statement then ensures that it is positive definite by constraining the two diagonal terms to be greater than or equal to $1\mathrm{E}{-4}$.

NOBOUND

requests the removal of boundary constraints on covariance parameters. For example, variance components have a default lower boundary constraint of 0, and the NOBOUND option allows their estimates to be negative.

NOITER

requests that no Newton-Raphson iterations be performed and that PROC MIXED use the best value from the grid search to perform inferences. By default, iterations begin at the best value from the PARMS grid search.

NOPROFILE

specifies a different computational method for the residual variance during the grid search. By default, PROC MIXED estimates this parameter using the profile likelihood when appropriate. This estimate is displayed in the Variance column of the "Parameter Search" table. The NOPROFILE option suppresses the profiling and uses the actual value of the specified variance in the likelihood calculations.

OLS

requests starting values corresponding to the usual general linear model. Specifically, all variances and covariances are set to zero except for the residual variance, which is set equal to its ordinary least-squares (OLS) estimate. This option is useful when the default MIVQUE0 procedure produces poor starting values for the optimization process.

PARMSDATA=*SAS-data-set*
PDATA=*SAS-data-set*

reads in covariance parameter values from a SAS data set. The data set should contain the EST or COVP1–COVP*n* variables.

RATIOS

indicates that ratios with the residual variance are specified instead of the covariance parameters themselves. The default is to use the individual covariance parameters.

UPPERB=*value-list*

enables you to specify upper boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC MIXED uses for the covariance parameters, and each number corresponds to the upper boundary constraint. A missing value instructs PROC MIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes that the remaining ones are missing.

PRIOR Statement

PRIOR $<$ *distribution* $><$ */ options* $>$ **;**

The PRIOR statement enables you to carry out a sampling-based Bayesian analysis in PROC MIXED. It currently operates only with variance component models. The analysis produces a SAS data set containing a pseudo-random sample from the joint posterior density of the variance components and other parameters in the mixed model.

The posterior analysis is performed after all other PROC MIXED computations. It begins with the "Posterior Sampling Information" table, which provides basic information about the posterior sampling analysis, including the prior densities, sampling algorithm, sample size, and random number seed. For ODS purposes, the name of this table is “Posterior.”

By default, PROC MIXED uses an independence chain algorithm in order to generate the posterior sample (Tierney 1994). This algorithm works by generating a pseudo-random proposal from a convenient base distribution, chosen to be as close as possible to the posterior. The proposal is then retained in the sample with probability proportional to the ratio of weights constructed by taking the ratio of the true posterior to the base density. If a proposal is not accepted, then a duplicate of the previous observation is added to the chain.

In selecting the base distribution, PROC MIXED makes use of the fact that the fixed-effects parameters can be analytically integrated out of the joint posterior, leaving the marginal posterior density of the variance components. In order to better approximate the marginal posterior density of the variance components, PROC MIXED transforms them using the MIVQUE(0) equations. You can display the selected transformation with the PTRANS option or specify your own with the TDATA= option. The density of the transformed parameters is then approximated by a product of inverted gamma densities (refer to Gelfand et al. 1990).

To determine the parameters for the inverted gamma densities, PROC MIXED evaluates the logarithm of the posterior density over a grid of points in each of the transformed parameters, and you can display the results of this search with the PSEARCH option. PROC MIXED then performs a linear regression of these values on the logarithm of the inverted gamma density. The resulting base densities are displayed in the “Base Densities” table; for ODS purposes, the name of this table is “BaseDen.” You can input different base densities with the BDATA= option.

At the end of the sampling, the “Acceptance Rates” table displays the acceptance rate computed as the number of accepted samples divided by the total number of samples generated. For ODS purposes, the label of the “Acceptance Rates” table is “AcceptanceRates.”

The OUT= option specifies the output data set containing the posterior sample. PROC MIXED automatically includes all variance component parameters in this data set (labeled COVP1–COVPn), the Type III *F*-statistics constructed as in Ghosh (1992) discussing Schervish (1992) (labeled T3Fn), the log values of the posterior (labeled LOGF), the log of the base sampling density (labeled LOGG), and the log of their ratio (labeled LOGRATIO). If you specify the SOLUTION option in the MODEL statement, the data set also contains a random sample from the posterior density of the fixed-effects parameters (labeled BETAn), and if you specify the SOLUTION option in the RANDOM statement, the table contains a random sample from the posterior density of the random-effects parameters (labeled GAMn). PROC MIXED also generates additional variables corresponding to any CONTRAST, ESTIMATE, or LSMEANS statement that you specify.

Subsequently, you can use SAS/INSIGHT, or the UNIVARIATE, CAPABILITY, or KDE procedures to analyze the posterior sample.

The prior density of the variance components is, by default, a noninformative version of Jeffreys’ prior (Box and Tiao 1973). You can also specify informative priors with the DATA= option or a flat (equal to 1) prior for the variance components. The prior density of the fixed-effects parameters is assumed to be flat (equal to 1), and the resulting posterior is conditionally multivariate normal (conditioning on the variance component parameters) with mean $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$ and variance $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$.

The *distribution* argument in the PRIOR statement determines the prior density for the variance component parameters of your mixed model. Valid values are as follows.

DATA=

enables you to input the prior densities of the variance components used by the sampling algorithm. This data set must contain the TYPE and PARM1–PARMn variables, where n is the largest number of parameters among each of the base densities. The format of the DATA= data set matches that created by PROC MIXED in the “Base Densities” table, so you can output the densities from one run and use them as input for a subsequent run.

JEFFREYS

specifies a noninformative reference version of Jeffreys’ prior constructed using the square root of the determinant of the expected information matrix as in (1.3.92) of Box and Tiao (1973). This is the default prior.

FLAT

specifies a prior density equal to 1 everywhere, making the likelihood function the posterior.

You can specify the following options in the PRIOR statement after a slash (/).

ALG=IC | INDCHAIN
ALG=IS | IMPSAMP
ALG=RS | REJSAMP
ALG=RWC | RWCHAIN

specifies the algorithm used for generating the posterior sample. The ALG=IC option requests an independence chain algorithm, and it is the default. The option ALG=IS requests importance sampling, ALG=RS requests rejection sampling, and ALG=RWC requests a random walk chain. For more information on these techniques, refer to Ripley (1987), Smith and Gelfand (1992), and Tierney (1994).

BDATA=

enables you to input the base densities used by the sampling algorithm. This data set must contain the TYPE and PARM1–PARMn variables, where n is the largest number of parameters among each of the base densities. The format of the BDATA= data set matches that created by PROC MIXED in the “Base Densities” table, so you can output the densities from one run and use them as input for a subsequent run.

GRID=*(value-list)*

specifies a grid of values over which to evaluate the posterior density. The *value-list* syntax is the same as in the PARMS statement (see page 2706), and you must specify an output data set name with the OUTG= option.

GRIDT=*(value-list)*

specifies a transformed grid of values over which to evaluate the posterior density. The *value-list* syntax is the same as in the PARMS statement (see page 2706), and you must specify an output data set name with the OUTGT= option.

IFACTOR=*number*

is an alias for the SFACTOR= option.

LOGNOTE=*number*

instructs PROC MIXED to write a note to the SAS log after it generates the sample corresponding to each multiple of *number*. This is useful for monitoring the progress of CPU-intensive runs.

LOGRBOUND=*number*

specifies the bounding constant for rejection sampling. The value of *number* equals the maximum of $\log(f/g)$ over the variance component parameter space, where f is the posterior density and g is the product inverted gamma densities used to perform rejection sampling.

When performing the rejection sampling, you may encounter the message

```
WARNING: The log ratio bound of LL was violated at sample XX.
```

When this occurs, PROC MIXED reruns an optimization algorithm to determine a new log upper bound and then restarts the rejection sampling. The resulting OUT= data set contains all observations that have been generated; therefore, assuming that you have requested N samples, you should retain only the final N observations in this data set for analysis purposes.

NSAMPLE=*number*

specifies the number of posterior samples to generate. The default is 1000, but more accurate results are obtained with larger samples such as 10000.

NSEARCH=*number*

specifies the number of posterior evaluations PROC MIXED makes for each transformed parameter in determining the parameters for the inverted gamma densities. The default is 20.

OUT=*SAS-data-set*

creates an output data set containing the sample from the posterior density.

OUTG=*SAS-data-set*

creates an output data set from the grid evaluations specified in the GRID= option.

OUTGT=*SAS-data-set*

creates an output data set from the transformed grid evaluations specified in the GRIDT= option.

PSEARCH

displays the search used to determine the parameters for the inverted gamma densities. For ODS purposes, the name of the table is “Search.”

PTRANS

displays the transformation of the variance components. For ODS purposes, the name of the table is "Trans."

SEED=*number*

specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer clock. You should use a positive seed (less than $2^{31} - 1$) whenever you want to duplicate the sample in another run of PROC MIXED.

SFACTOR=*number*

enables you to adjust the range over which PROC MIXED searches the transformed parameters in order to determine the parameters for the inverted gamma densities. PROC MIXED determines the range by first transforming the estimates from the standard PROC MIXED analysis (REML, ML, or MIVQUE0, depending upon which estimation method you select). It then multiplies and divides the transformed estimates by $2*$*number* to obtain upper and lower bounds, respectively. Transformed values that produce negative variance components in the original scale are not included in the search. The default value is 1; *number* must be greater than 0.5.

TDATA=

enables you to input the transformation of the covariance parameters used by the sampling algorithm. This data set should contain the CovP1–CovPn variables. The format of the TDATA= data set matches that created by PROC MIXED in the "Trans" table, so you can output the transformation from one run and use is as input for a subsequent run.

TRANS=EXPECTED
TRANS=MIVQUE0
TRANS=OBSERVED

specifies the particular algorithm used to determine the transformation of the covariance parameters. The default is MIVQUE0, indicating a transformation based on the MIVQUE(0) equations. The other two options indicate the type of Hessian matrix used in constructing the transformation via a Cholesky root.

UPDATE=*number*

is an alias for the LOGNOTE= option.

RANDOM Statement

RANDOM *random-effects* < */ options* > **;**

The RANDOM statement defines the random effects constituting the γ vector in the mixed model. It can be used to specify traditional variance component models (as in the VARCOMP procedure) and to specify random coefficients. The random effects can be classification or continuous, and multiple RANDOM statements are possible.

Using notation from the "Mixed Models Theory" section beginning on page 2731, the purpose of the RANDOM statement is to define the $\mathbf{Z}$ matrix of the mixed model, the random effects in the γ vector, and the structure of $\mathbf{G}$. The $\mathbf{Z}$ matrix is constructed exactly as the $\mathbf{X}$ matrix for the fixed effects, and the $\mathbf{G}$ matrix is constructed to correspond with the effects constituting $\mathbf{Z}$. The structure of $\mathbf{G}$ is defined by using the TYPE= option described on page 2715.

You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC MIXED does not include the intercept in the RANDOM statement by default as it does in the MODEL statement.

You can specify the following options in the RANDOM statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the random effect estimates with confidence level 1 – *number*. The value of *number* must be between 0 and 1; the default is 0.05.

CL

requests that *t*-type confidence limits be constructed for each of the random effect estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

G

requests that the estimated $\mathbf{G}$ matrix be displayed. PROC MIXED displays blanks for values that are 0. If you specify the SUBJECT= option, then the block of the $\mathbf{G}$ matrix corresponding to the first subject is displayed. For ODS purposes, the name of the table is “G.”

GC

displays the lower-triangular Cholesky root of the estimated $\mathbf{G}$ matrix according to the rules listed under the G option. For ODS purposes, the name of the table is “CholG.”

GCI

displays the inverse Cholesky root of the estimated $\mathbf{G}$ matrix according to the rules listed under the G option. For ODS purposes, the name of the table is “InvCholG.”

GCORR

displays the correlation matrix corresponding to the estimated $\mathbf{G}$ matrix according to the rules listed under the G option. For ODS purposes, the name of the table is “GCorr.”

GDATA=*SAS-data-set*

requests that the $\mathbf{G}$ matrix be read in from a SAS data set. This $\mathbf{G}$ matrix is assumed to be known; therefore, only $\mathbf{R}$-side parameters from effects in the REPEATED statement are included in the Newton-Raphson iterations. If no REPEATED statement is specified, then only a residual variance is estimated.

The information in the GDATA= data set can appear in one of two ways. The first is a sparse representation for which you include ROW, COL, and VALUE variables to indicate the row, column, and value of $\mathbf{G}$. All unspecified locations are assumed to be 0. The second representation is for dense matrices. In it you include ROW and COL1–COL*n* variables to indicate the row and columns of $\mathbf{G}$, which is a symmetric matrix of order *n*. For both representations, you must specify effects in the RANDOM statement that generate a $\mathbf{Z}$ matrix that contains *n* columns. See Example 46.4 on page 2802.

If you have more than one RANDOM statement, only one GDATA= option is required on any one of them, and the data set you specify must contain the entire $\mathbf{G}$ matrix defined by all of the RANDOM statements.

If the GDATA= data set contains variance ratios instead of the variances themselves, then use the RATIOS option.

Known parameters of $\mathbf{G}$ can also be input using the PARMS statement with the HOLD= option.

GI

displays the inverse of the estimated $\mathbf{G}$ matrix according to the rules listed under the G option. For ODS purposes, the name of the table is “InvG.”

GROUP=*effect*

GRP=*effect*

defines an effect specifying heterogeneity in the covariance structure of $\mathbf{G}$. All observations having the same level of the group effect have the same covariance parameters. Each new level of the group effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in defining the group effect, as strange covariance patterns can result with its misuse. Also, the group effect can greatly increase the number of estimated covariance parameters, which may adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large “Class Levels Information” table.

LDATA=*SAS-data-set*

reads the coefficient matrices associated with the TYPE=LIN(*number*) option. The data set must contain the variables PARM, ROW, COL1–COLn, or PARM, ROW, COL, VALUE. The PARM variable denotes which of the *number* coefficient matrices is currently being constructed, and the ROW, COL1–COLn, or ROW, COL, VALUE variables specify the matrix values, as they do with the GDATA= option. Unspecified values of these matrices are set equal to 0.

NOFULLZ

eliminates the columns in $\mathbf{Z}$ corresponding to missing levels of random effects involving CLASS variables. By default, these columns are included in $\mathbf{Z}$.

RATIOS

indicates that ratios with the residual variance are specified in the GDATA= data set instead of the covariance parameters themselves. The default GDATA= data set contains the individual covariance parameters.

SOLUTION
S

requests that the solution for the random-effects parameters be produced. Using notation from the "Mixed Models Theory" section beginning on page 2731, these estimates are the empirical best linear unbiased predictors (EBLUPs) $\widehat{\gamma} = \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. They can be useful for comparing the random effects from different experimental units and can also be treated as residuals in performing diagnostics for your mixed model.

The numbers displayed in the SE Pred column of the "Solution for Random Effects" table are not the standard errors of the $\widehat{\gamma}$ displayed in the Estimate column; rather, they are the standard errors of predictions $\widehat{\gamma}_i - \gamma_i$, where $\widehat{\gamma}_i$ is the *i*th EBLUP and γ_i is the *i*th random-effect parameter.

SUBJECT=*effect*
SUB=*effect*

identifies the subjects in your mixed model. Complete independence is assumed across subjects; thus, for the RANDOM statement, the SUBJECT= option produces a block-diagonal structure in $\mathbf{G}$ with identical blocks. The $\mathbf{Z}$ matrix is modified to accommodate this block-diagonality. In fact, specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Levels Information" table.

When you specify the SUBJECT= option and a classification random effect, computations are usually much quicker if the levels of the random effect are duplicated within each level of the SUBJECT= effect.

TYPE=*covariance-structure*

specifies the covariance structure of $\mathbf{G}$. Valid values for *covariance-structure* and their descriptions are listed in Table 46.5 on page 2721 and Table 46.6 on page 2722. Although a variety of structures are available, most applications call for either TYPE=VC or TYPE=UN. The TYPE=VC (variance components) option is the default structure, and it models a different variance component for each random effect.

The TYPE=UN (unstructured) option is useful for correlated random coefficient models. For example,

```
random intercept age / type=un subject=person;
```

specifies a random intercept-slope model that has different variances for the intercept and slope and a covariance between them. You can also use TYPE=FA0(2) here to request a $\mathbf{G}$ estimate that is constrained to be nonnegative definite.

If you are constructing your own columns of $\mathbf{Z}$ with continuous variables, you can use the TYPE=TOEP(1) structure to group them together to have a common variance component. If you desire to have different covariance structures in different parts of $\mathbf{G}$, you must use multiple RANDOM statements with different TYPE= options.

V<=*value-list*>

requests that blocks of the estimated $\mathbf{V}$ matrix be displayed. The first block determined by the SUBJECT= effect is the default displayed block. PROC MIXED displays entries that are 0 as blanks in the table.

You can optionally use the *value-list* specification, which indicates the subjects for which blocks of $\mathbf{V}$ are to be displayed. For example, the statement

```
random int time / type=un subject=person v=1,3,7;
```

displays block matrices for the first, third, and seventh persons. The table name for ODS purposes is “V.”

VC<=*value-list*>

displays the Cholesky root of the blocks of the estimated $\mathbf{V}$ matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is “CholV.”

VCI<=*value-list*>

displays the inverse of the Cholesky root of the blocks of the estimated $\mathbf{V}$ matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is “InvCholV.”

VCORR<=*value-list*>

displays the correlation matrix corresponding to the blocks of the estimated $\mathbf{V}$ matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is “VCorr.”

VI<=*value-list*>

displays the inverse of the blocks of the estimated $\mathbf{V}$ matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is “InvV.”

REPEATED Statement

REPEATED < *repeated-effect* >< / *options* > **;**

The REPEATED statement is used to specify the $\mathbf{R}$ matrix in the mixed model. Its syntax is different from that of the REPEATED statement in PROC GLM. If no REPEATED statement is specified, $\mathbf{R}$ is assumed to be equal to $\sigma^2\mathbf{I}$.

For many repeated measures models, no repeated effect is required in the REPEATED statement. Simply use the SUBJECT= option to define the blocks of $\mathbf{R}$ and the TYPE= option to define their covariance structure. In this case, the repeated measures data must be similarly ordered for each subject, and you must indicate all missing response variables with periods in the input data set unless they all fall at the end of a subject's repeated response profile. These requirements are necessary in order to inform PROC MIXED of the proper location of the observed repeated responses.

Specifying a repeated effect is useful when you do not want to indicate missing values with periods in the input data set. The repeated effect must contain only classification variables. Make sure that the levels of the repeated effect are different for each observation within a subject; otherwise, PROC MIXED constructs identical rows in $\mathbf{R}$ corresponding to the observations with the same level. This results in a singular $\mathbf{R}$ and an infinite likelihood.

Whether you specify a REPEATED effect or not, the rows of $\mathbf{R}$ for each subject are constructed in the order that they appear in the input data set.

You can specify the following options in the REPEATED statement after a slash (/).

GROUP=*effect*
GRP=*effect*

defines an effect specifying heterogeneity in the covariance structure of $\mathbf{R}$. All observations having the same level of the GROUP effect have the same covariance parameters. Each new level of the GROUP effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in properly defining the GROUP effect, as strange covariance patterns can result with its misuse. Also, the GROUP effect can greatly increase the number of estimated covariance parameters, which may adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Levels Information" table.

HLM

produces a table of Hotelling-Lawley-McKeon statistics (McKeon 1974) for all fixed effects whose levels change across data having the same level of the SUBJECT= effect (the *within-subject* fixed effects). This option applies only when you specify a REPEATED statement with the TYPE=UN option and no RANDOM statements. For balanced data, this model is equivalent to the multivariate model for repeated measures in PROC GLM.

The Hotelling-Lawley-McKeon statistic has a slightly better F approximation than the Hotelling-Lawley-Pillai-Samson statistic (see the description of the HLPS option, which follows). Both of the Hotelling-Lawley statistics can perform much better in small samples than the default F statistic (Wright 1994).

Separate tables are produced for Type I, II, and III tests, according to the ones you select. For ODS purposes, the labels for these tables are "HLM1," "HLM2," and "HLM3," respectively.

HLPS

produces a table of Hotelling-Lawley-Pillai-Samson statistics (Pillai and Samson 1959) for all fixed effects whose levels change across data having the same level of the SUBJECT= effect (the *within-subject* fixed effects). This option applies only when you specify a REPEATED statement with the TYPE=UN option and no RANDOM statements. For balanced data, this model is equivalent to the multivariate model for repeated measures in PROC GLM, and this statistic is the same as the Hotelling-Lawley Trace statistic produced by PROC GLM.

Separate tables are produced for Type I, II, and III tests, according to the ones you select. For ODS purposes, the labels for these tables are "HLPS1," "HLPS2," and "HLPS3," respectively.

LDATA=*SAS-data-set*

reads the coefficient matrices associated with the TYPE=LIN(*number*) option. The data set must contain the variables PARM, ROW, COL1–COLn, or PARM, ROW, COL, VALUE. The PARM variable denotes which of the *number* coefficient matrices is currently being constructed, and the ROW, COL1–COLn, or ROW, COL, VALUE variables specify the matrix values, as they do with the RANDOM statement option GDATA=. Unspecified values of these matrices are set equal to 0.

LOCAL
LOCAL=EXP(<*effects*>**)**
LOCAL=POM(*POM-data-set***)**

requests that a diagonal matrix be added to $\mathbf{R}$. With just the LOCAL option, this diagonal matrix equals $\sigma^2\mathbf{I}$, and σ^2 becomes an additional variance parameter that PROC MIXED profiles out of the likelihood provided that you do not specify the NOPROFILE option in the PROC MIXED statement. The LOCAL option is useful if you want to add an observational error to a time series structure (Jones and Boadi-Boateng 1991) or a nugget effect to a spatial structure (Cressie 1991).

The LOCAL=EXP(<*effects*>) option produces exponential local effects, also known as dispersion effects, in a log-linear variance model. These local effects have the form

$$\sigma^2 \mathrm{diag}[\exp(\mathbf{U}\delta)]$$

where $\mathbf{U}$ is the full-rank design matrix corresponding to the effects that you specify and δ are the parameters that PROC MIXED estimates. An intercept is not included in $\mathbf{U}$ because it is accounted for by σ^2. PROC MIXED constructs the full-rank $\mathbf{U}$ in terms of 1s and -1s for classification effects. Be sure to scale continuous effects in $\mathbf{U}$ sensibly.

The LOCAL=POM(*POM-data-set*) option specifies the power-of-the-mean structure. This structure possesses a variance of the form $\sigma^2|\mathbf{x}_i'\boldsymbol{\beta}^*|^\theta$ for the ith observation, where $\mathbf{x}_i$ is the ith row of $\mathbf{X}$ (the design matrix of the fixed effects), and $\boldsymbol{\beta}^*$ is an estimate of the fixed-effects parameters that you specify in *POM-data-set*.

The SAS data set specified by *POM-data-set* contains the numeric variable Estimate (in previous releases, the variable name was required to be EST), and it has at least as many observations as there are fixed-effects parameters. The first p observations of the Estimate variable in *POM-data-set* are taken to be the elements of $\boldsymbol{\beta}^*$, where p is the number of columns of $\mathbf{X}$. You must order these observations according to the non-full-rank parameterization of the MIXED procedure. One easy way to set up *POM-data-set* for a $\boldsymbol{\beta}^*$ corresponding to ordinary least squares is illustrated by the following code:

```
ods output SolutionF=sf;
proc mixed;
   class a;
   model y = a x / s;
run;

proc mixed;
   class a;
   model y = a x;
   repeated / local=pom(sf);
run;
```

Note that the generalized least-squares estimate of the fixed-effects parameters from the second PROC MIXED step usually is not the same as your specified $\boldsymbol{\beta}^*$. However, you can iterate the POM fitting until the two estimates agree. Continuing from the previous example, the code for performing one step of this iteration is as follows.

```
ods output SolutionF=sf1;
proc mixed;
   class a;
   model y = a x / s;
   repeated / local=pom(sf);
run;

proc compare brief data=sf compare=sf1;
   var estimate;
run;

data sf;
   set sf1;
run;
```

Unfortunately, this iterative process does not always converge. For further details, refer to the description of pseudo-likelihood in Chapter 3 of Carroll and Ruppert (1988).

LOCALW

specifies that only the local effects and no others be weighted. By default, all effects are weighted. The LOCALW option is used in connection with the WEIGHT statement and the LOCAL option in the REPEATED statement

NONLOCALW

specifies that only the nonlocal effects and no others be weighted. By default, all effects are weighted. The NONLOCALW option is used in connection with the WEIGHT statement and the LOCAL option in the REPEATED statement

R<=*value-list*>

requests that blocks of the estimated $\mathbf{R}$ matrix be displayed. The first block determined by the SUBJECT= effect is the default displayed block. PROC MIXED displays blanks for value-lists that are 0.

The *value-list* indicates the subjects for which blocks of $\mathbf{R}$ are to be displayed. For example,

```
repeated / type=cs subject=person r=1,3,5;
```

displays block matrices for the first, third, and fifth persons. See the "PARMS Statement" section on page 2706 for the possible forms of *value-list*. The table name for ODS purposes is "R."

RC<=*value-list*>

produces the Cholesky root of blocks of the estimated $\mathbf{R}$ matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "CholR."

RCI<=*value-list*>

produces the inverse Cholesky root of blocks of the estimated $\mathbf{R}$ matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "InvCholR."

RCORR<=*value-list*>

produces the correlation matrix corresponding to blocks of the estimated $\mathbf{R}$ matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "RCorr."

RI<=*value-list*>

produces the inverse of blocks of the estimated $\mathbf{R}$ matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "InvR."

SSCP

requests that an unstructured $\mathbf{R}$ matrix be estimated from the sum-of-squares-and-crossproducts matrix of the residuals. It applies only when you specify TYPE=UN and have no RANDOM statements. Also, you must have a sufficient number of subjects for the estimate to be positive definite.

This option is useful when the size of the blocks of $\mathbf{R}$ are large (for example, greater than 10) and you want to use or inspect an unstructured estimate that is much quicker to compute than the default REML estimate. The two estimates will agree for certain balanced data sets when you have a classification fixed effect defined across all time points within a subject.

SUBJECT=*effect*
SUB=*effect*

identifies the subjects in your mixed model. Complete independence is assumed across subjects; therefore, the SUBJECT= option produces a block-diagonal structure in $\mathbf{R}$ with identical blocks. When the SUBJECT= effect consists entirely of classification variables, the blocks of $\mathbf{R}$ correspond to observations sharing the same level of that effect. These blocks are sorted according to this effect as well.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Levels Information" table.

If you want to model nonzero covariance among all of the observations in your SAS data set, specify SUBJECT=INTERCEPT to treat the data as if they are all from one subject. Be aware though that, in this case, PROC MIXED manipulates an $\mathbf{R}$ matrix with dimensions equal to the number of observations. If no SUBJECT= effect is specified, then every observation is assumed to be from a different subject and $\mathbf{R}$ is assumed to be diagonal. For this reason, you usually want to use the SUBJECT= option in the REPEATED statement.

TYPE=*covariance-structure*

specifies the covariance structure of the $\mathbf{R}$ matrix. The SUBJECT= option defines the blocks of $\mathbf{R}$, and the TYPE= option specifies the structure of these blocks. Valid values for *covariance-structure* and their descriptions are provided in Table 46.5 and Table 46.6. The default structure is VC.

Table 46.5. Covariance Structures

Structure	**Description**	**Parms**	(i,j)**th element**
ANTE(1)	Ante-Dependence	$2t-1$	$\sigma_i\sigma_j \prod_{k=i}^{j-1} \rho_k$
AR(1)	Autoregressive(1)	2	$\sigma^2\rho^{\lvert i-j\rvert}$
ARH(1)	Heterogeneous AR(1)	$t+1$	$\sigma_i\sigma_j\rho^{\lvert i-j\rvert}$
ARMA(1,1)	ARMA(1,1)	3	$\sigma^2[\gamma\rho^{\lvert i-j\rvert-1}1(\mathrm{i}\neq \mathrm{j}) + 1(\mathrm{i{=}j})]$
CS	Compound Symmetry	2	$\sigma_1 + \sigma^2 1(i=j)$
CSH	Heterogeneous CS	$t+1$	$\sigma_i\sigma_j[\rho 1(\mathrm{i}\neq \mathrm{j}) + 1(\mathrm{i{=}j})]$
FA(q)	Factor Analytic	$\frac{q}{2}(2t-q+1)+t$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk} + \sigma_i^2 1(i=j)$
FA0(q)	No Diagonal FA	$\frac{q}{2}(2t-q+1)$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk}$
FA1(q)	Equal Diagonal FA	$\frac{q}{2}(2t-q+1)+1$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk} + \sigma^2 1(i=j)$
HF	Huynh-Feldt	$t+1$	$(\sigma_i^2+\sigma_j^2)/2 + \lambda 1(\mathrm{i}\neq \mathrm{j})$
LIN(q)	General Linear	q	$\Sigma_{k=1}^{q}\theta_k\mathbf{A}_{ij}$
TOEP	Toeplitz	t	$\sigma_{\lvert i-j\rvert+1}$
TOEP(q)	Banded Toeplitz	q	$\sigma_{\lvert i-j\rvert+1}1(\lvert i-j\rvert < q)$
TOEPH	Heterogeneous TOEP	$2t-1$	$\sigma_i\sigma_j\rho_{\lvert i-j\rvert}$
TOEPH(q)	Banded Hetero TOEP	$t+q-1$	$\sigma_i\sigma_j\rho_{\lvert i-j\rvert}1(\lvert i-j\rvert < q)$
UN	Unstructured	$t(t+1)/2$	σ_{ij}
UN(q)	Banded	$\frac{q}{2}(2t-q+1)$	$\sigma_{ij}1(\lvert i-j\rvert < q)$
UNR	Unstructured Corrs	$t(t+1)/2$	$\sigma_i\sigma_j\rho_{\max(i,j)\,\min(i,j)}$

Table 46.5. (continued)

Structure	**Description**	**Parms**	**(i,j)th element**
UNR(q)	Banded Correlations	$\frac{q}{2}(2t-q+1)$	$\sigma_i \sigma_j \rho_{\max(i,j)\,\min(i,j)}$
UN@AR(1)	Direct Product AR(1)	$t_1(t_1+1)/2+1$	$\sigma_{i_1 j_1} \rho^{\lvert i_2-j_2 \rvert}$
UN@CS	Direct Product CS	$t_1(t_1+1)/2+1$	$\sigma_{i_1 j_1}(1-\sigma^2 1(i_2 \neq j_2)), 0 \leq \sigma^2 \leq 1$
UN@UN	Direct Product UN	$t_1(t_1+1)/2 +$ $t_2(t_2+1)/2-1$	$\sigma_{1,i_1 j_1}\sigma_{2,i_2 j_2}$
VC	Variance Components	q	$\sigma_k^2 1(i=j)$ and i corresponds to kth effect

In Table 46.5, "Parms" is the number of covariance parameters in the structure, t is the overall dimension of the covariance matrix, and $1(A)$ equals 1 when A is true and 0 otherwise. For example, $1(i=j)$ equals 1 when $i=j$ and 0 otherwise, and $1(\lvert i-j \rvert < q)$ equals 1 when $\lvert i-j \rvert < q$ and 0 otherwise. For the TOEPH structures, $\rho_0 = 1$, and for the UNR structures, $\rho_{ii} = 1$ for all i. For the direct product structures, the subscripts "1" and "2" refer to the first and second structure in the direct product, respectively, and $i_1 = \text{int}((i+t_2-1)/t_2)$, $j_1 = \text{int}((j+t_2-1)/t_2)$, $i_2 = \text{mod}(i-1,t_2)+1$, and $j_2 = \text{mod}(j-1,t_2)+1$.

Table 46.6. Spatial Covariance Structures

Structure	**Description**	**Parms**	**(i,j)th element**
SP(EXP)(*c-list*)	Exponential	2	$\sigma^2[\exp(-d_{ij}/\theta)]$
SP(EXPA)(*c-list*)	Anisotropic Exponential	$2c+1$	$\sigma^2 \prod_{k=1}^{c} \exp[-\theta_k d(i,j,k)^{p_k}]$
SP(EXPGA)(c_1 c_2)	2D Exponential, Geometrically Anisotropic	4	$\sigma^2[\exp(-d_{ij}(\theta,\lambda)/\rho)]$
SP(GAU)(*c-list*)	Gaussian	2	$\sigma^2[\exp(-d_{ij}^2/\rho^2)]$
SP(GAUGA)(c_1 c_2)	2D Gaussian, Geometrically Anisotropic	4	$\sigma^2[\exp(-d_{ij}(\theta,\lambda)^2/\rho^2)]$
SP(LIN)(*c-list*)	Linear	2	$\sigma^2(1-\rho d_{ij})\, 1(\rho d_{ij} \leq 1)$
SP(LINL)(*c-list*)	Linear log	2	$\sigma^2(1-\rho\log(d_{ij}))$ $\times 1(\rho\log(d_{ij}) \leq 1)$
SP(MATERN)(*c-list*)	Matérn	3	$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}}{2\rho}\right)^{\nu} 2K_\nu(d_{ij}/\rho)$
SP(MATHSW)(*c-list*)	Matérn (Handcock-Stein-Wallis)	3	$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}\sqrt{\nu}}{\rho}\right)^{\nu} 2K_\nu \left(\frac{2d_{ij}\sqrt{\nu}}{\rho}\right)$
SP(POW)(*c-list*)	Power	2	$\sigma^2 \rho^{d_{ij}}$
SP(POWA)(*c-list*)	Anisotropic Power	$c+1$	$\sigma^2 \rho_1^{d(i,j,1)} \rho_2^{d(i,j,2)} \ldots \rho_c^{d(i,j,c)}$
SP(SPH)(*c-list*)	Spherical	2	$\sigma^2[1-(\frac{3d_{ij}}{2\rho})+(\frac{d_{ij}^3}{2\rho^3})]\, 1(d_{ij} \leq \rho)$
SP(SPHGA)(c_1 c_2)	2D Spherical, Geometrically Anisotropic	4	$\sigma^2[1-(\frac{3d_{ij}(\theta,\lambda)}{2\rho})+(\frac{d_{ij}(\theta,\lambda)^3}{2\rho^3})]$ $\times 1(d_{ij}(\theta,\lambda) \leq \rho)$

In Table 46.6, *c-list* contains the names of the numeric variables used as coordinates of the location of the observation in space, and d_{ij} is the Euclidean distance between the *i*th and *j*th vectors of these coordinates, which correspond to the *i*th and *j*th observations in the input data set. For SP(POWA) and SP(EXPA), *c* is the number of coordinates, and $d(i,j,k)$ is the absolute distance between the *k*th coordinate,

$k = 1, \ldots, c$, of the *i*th and *j*th observations in the input data set. For the geometrically anisotropic structures SP(EXPGA), SP(GAUGA), and SP(SPHGA), exactly two spatial coordinate variables must be specified as c_1 and c_2. Geometric anisotropy is corrected by applying a rotation θ and scaling λ to the coordinate system, and $d_{ij}(\theta, \lambda)$ represents the Euclidean distance between two points in the transformed space. SP(MATERN) and SP(MATHSW) represent covariance structures in a class defined by Matérn (refer to Matérn 1986, Handcock and Stein 1993, Handcock and Wallis 1994). The function K_ν is the modified Bessel function of the second kind of (real) order $\nu > 0$; the parameter ν governs the smoothness of the process (see below for more details).

Table 46.7 lists some examples of the structures in Table 46.5 and Table 46.6.

Table 46.7. Covariance Structure Examples

Description	**Structure**	**Example**
Variance Components	VC (default)	$\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$
Compound Symmetry	CS	$\begin{bmatrix} \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 \end{bmatrix}$
Unstructured	UN	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{43} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$
Banded Main Diagonal	UN(1)	$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive	AR(1)	$\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$
Toeplitz	TOEP	$\begin{bmatrix} \sigma^2 & \sigma_1 & \sigma_2 & \sigma_3 \\ \sigma_1 & \sigma^2 & \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 & \sigma^2 & \sigma_1 \\ \sigma_3 & \sigma_2 & \sigma_1 & \sigma^2 \end{bmatrix}$
Toeplitz with Two Bands	TOEP(2)	$\begin{bmatrix} \sigma^2 & \sigma_1 & 0 & 0 \\ \sigma_1 & \sigma^2 & \sigma_1 & 0 \\ 0 & \sigma_1 & \sigma^2 & \sigma_1 \\ 0 & 0 & \sigma_1 & \sigma^2 \end{bmatrix}$

Table 46.7. (continued)

Description	Structure	Example
Spatial Power	SP(POW)(c)	$\sigma^2 \begin{bmatrix} 1 & \rho^{d_{12}} & \rho^{d_{13}} & \rho^{d_{14}} \\ \rho^{d_{21}} & 1 & \rho^{d_{23}} & \rho^{d_{24}} \\ \rho^{d_{31}} & \rho^{d_{32}} & 1 & \rho^{d_{34}} \\ \rho^{d_{41}} & \rho^{d_{42}} & \rho^{d_{43}} & 1 \end{bmatrix}$
Heterogeneous AR(1)	ARH(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho^2 & \sigma_1\sigma_4\rho^3 \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho^2 \\ \sigma_3\sigma_1\rho^2 & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho^3 & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive Moving-Average	ARMA(1,1)	$\sigma^2 \begin{bmatrix} 1 & \gamma & \gamma\rho & \gamma\rho^2 \\ \gamma & 1 & \gamma & \gamma\rho \\ \gamma\rho & \gamma & 1 & \gamma \\ \gamma\rho^2 & \gamma\rho & \gamma & 1 \end{bmatrix}$
Heterogeneous CS	CSH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho & \sigma_1\sigma_4\rho \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho \\ \sigma_3\sigma_1\rho & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
First-Order Factor Analytic	FA(1)	$\begin{bmatrix} \lambda_1^2 + d_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\ \lambda_2\lambda_1 & \lambda_2^2 + d_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\ \lambda_3\lambda_1 & \lambda_3\lambda_2 & \lambda_3^2 + d_3 & \lambda_3\lambda_4 \\ \lambda_4\lambda_1 & \lambda_4\lambda_2 & \lambda_4\lambda_3 & \lambda_4^2 + d_4 \end{bmatrix}$
Huynh-Feldt	HF	$\begin{bmatrix} \sigma_1^2 & \frac{\sigma_1^2+\sigma_2^2}{2} - \lambda & \frac{\sigma_1^2+\sigma_3^2}{2} - \lambda \\ \frac{\sigma_2^2+\sigma_1^2}{2} - \lambda & \sigma_2^2 & \frac{\sigma_2^2+\sigma_3^2}{2} - \lambda \\ \frac{\sigma_3^2+\sigma_1^2}{2} - \lambda & \frac{\sigma_3^2+\sigma_2^2}{2} - \lambda & \sigma_3^2 \end{bmatrix}$
First-Order Ante-dependence	ANTE(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_1\rho_2 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_2 \\ \sigma_3\sigma_1\rho_2\rho_1 & \sigma_3\sigma_2\rho_2 & \sigma_3^2 \end{bmatrix}$
Heterogeneous Toeplitz	TOEPH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_2 & \sigma_1\sigma_4\rho_3 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_1 & \sigma_2\sigma_4\rho_2 \\ \sigma_3\sigma_1\rho_2 & \sigma_3\sigma_2\rho_1 & \sigma_3^2 & \sigma_3\sigma_4\rho_1 \\ \sigma_4\sigma_1\rho_3 & \sigma_4\sigma_2\rho_2 & \sigma_4\sigma_3\rho_1 & \sigma_4^2 \end{bmatrix}$
Unstructured Correlations	UNR	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{21} & \sigma_1\sigma_3\rho_{31} & \sigma_1\sigma_4\rho_{41} \\ \sigma_2\sigma_1\rho_{21} & \sigma_2^2 & \sigma_2\sigma_3\rho_{32} & \sigma_2\sigma_4\rho_{42} \\ \sigma_3\sigma_1\rho_{31} & \sigma_3\sigma_2\rho_{32} & \sigma_3^2 & \sigma_3\sigma_4\rho_{43} \\ \sigma_4\sigma_1\rho_{41} & \sigma_4\sigma_2\rho_{42} & \sigma_4\sigma_3\rho_{43} & \sigma_4^2 \end{bmatrix}$
Direct Product AR(1)	UN@AR(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} \otimes \begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix} =$

Table 46.7. (continued)

Description	**Structure**	**Example**
		$\begin{bmatrix} \sigma_1^2 & \sigma_1^2\rho & \sigma_1^2\rho^2 & \sigma_{21} & \sigma_{21}\rho & \sigma_{21}\rho^2 \\ \sigma_1^2\rho & \sigma_1^2 & \sigma_1^2\rho & \sigma_{21}\rho & \sigma_{21} & \sigma_{21}\rho \\ \sigma_1^2\rho^2 & \sigma_1^2\rho & \sigma_1^2 & \sigma_{21}\rho^2 & \sigma_{21}\rho & \sigma_{21} \\ \sigma_{21} & \sigma_{21}\rho & \sigma_{21}\rho^2 & \sigma_2^2 & \sigma_2^2\rho & \sigma_2^2\rho^2 \\ \sigma_{21}\rho & \sigma_{21} & \sigma_{21}\rho & \sigma_2^2\rho & \sigma_2^2 & \sigma_2^2\rho \\ \sigma_{21}\rho^2 & \sigma_{21}\rho & \sigma_{21} & \sigma_2^2\rho^2 & \sigma_2^2\rho & \sigma_2^2 \end{bmatrix}$

The following provides some further information about these covariance structures:

TYPE=ANTE(1) specifies the first-order antedependence structure (refer to Kenward 1987, Patel 1991, and Macchiavelli and Arnold 1994). In Table 46.5, σ_i^2 is the ith variance parameter, and ρ_k is the kth autocorrelation parameter satisfying $|\rho_k| < 1$.

TYPE=AR(1) specifies a first-order autoregressive structure. PROC MIXED imposes the constraint $|\rho| < 1$ for stationarity.

TYPE=ARH(1) specifies a heterogeneous first-order autoregressive structure. As with TYPE=AR(1), PROC MIXED imposes the constraint $|\rho| < 1$ for stationarity.

TYPE=ARMA(1,1) specifies the first-order autoregressive moving average structure. In Table 46.5, ρ is the autoregressive parameter, γ models a moving average component, and σ^2 is the residual variance. In the notation of Fuller (1976, p. 68), $\rho = \theta_1$ and

$$\gamma = \frac{(1 + b_1\theta_1)(\theta_1 + b_1)}{1 + b_1^2 + 2b_1\theta_1}$$

The example in Table 46.7 and $|b_1| < 1$ imply that

$$b_1 = \frac{\beta - \sqrt{\beta^2 - 4\alpha^2}}{2\alpha}$$

where $\alpha = \gamma - \rho$ and $\beta = 1 + \rho^2 - 2\gamma\rho$. PROC MIXED imposes the constraints $|\rho| < 1$ and $|\gamma| < 1$ for stationarity, although for some values of ρ and γ in this region the resulting covariance matrix is not positive definite. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

TYPE=CS specifies the compound-symmetry structure, which has constant variance and constant covariance.

TYPE=CSH specifies the heterogeneous compound-symmetry structure. This structure has a different variance parameter for each diagonal element, and it uses the square roots of these parameters in the off-diagonal entries. In Table 46.5, σ_i^2 is the ith variance parameter, and ρ is the correlation parameter satisfying $|\rho| < 1$.

TYPE=FA(q) specifies the factor-analytic structure with q factors (Jennrich and Schluchter 1986). This structure is of the form $\Lambda\Lambda' + \mathbf{D}$, where Λ is a $t \times q$ rectangular matrix and $\mathbf{D}$ is a $t \times t$ diagonal matrix with t different parameters. When $q > 1$, the elements of Λ in its upper right-hand corner (that is, the elements in the ith row and jth column for $j > i$) are set to zero to fix the rotation of the structure.

TYPE=FA0(q) is similar to the FA(q) structure except that no diagonal matrix $\mathbf{D}$ is included. When $q < t$, that is, when the number of factors is less than the dimension of the matrix, this structure is nonnegative definite but not of full rank. In this situation, you can use it for approximating an unstructured $\mathbf{G}$ matrix in the RANDOM statement or for combining with the LOCAL option in the REPEATED statement. When $q = t$, you can use this structure to constrain $\mathbf{G}$ to be nonnegative definite in the RANDOM statement.

TYPE=FA1(q) is similar to the FA(q) structure except that all of the elements in $\mathbf{D}$ are constrained to be equal. This offers a useful and more parsimonious alternative to the full factor-analytic structure.

TYPE=HF specifies the Huynh-Feldt covariance structure (Huynh and Feldt 1970). This structure is similar to the CSH structure in that it has the same number of parameters and heterogeneity along the main diagonal. However, it constructs the off-diagonal elements by taking arithmetic rather than geometric means.

You can perform a likelihood ratio test of the Huynh-Feldt conditions by running PROC MIXED twice, once with TYPE=HF and once with TYPE=UN, and then subtracting their respective values of -2 times the maximized likelihood.

If PROC MIXED does not converge under your Huynh-Feldt model, you can specify your own starting values with the PARMS statement. The default MIVQUE(0) starting values can sometimes be poor for this structure. A good choice for starting values is often the parameter estimates corresponding to an initial fit using TYPE=CS.

TYPE=LIN(q) specifies the general linear covariance structure with q parameters (Helms and Edwards 1991). This structure consists of a linear combination of known matrices that are input with the LDATA= option. This structure is very general, and you need to make sure that the variance matrix is positive definite. By default, PROC MIXED sets the initial values of the parameters to 1. You can use the PARMS statement to specify other initial values.

TYPE=SIMPLE is an alias for TYPE=VC.

TYPE=SP(EXPA)(*c-list*) specifies the spatial anisotropic exponential structure, where *c-list* is a list of variables indicating the coordinates. This structure has (i, j)th element equal to

$$\sigma^2 \prod_{k=1}^{c} \exp[-\theta_k d(i,j,k)^{p_k}]$$

where c is the number of coordinates and $d(i,j,k)$ is the absolute distance between the kth coordinate ($k = 1, \ldots, c$) of the ith and jth observations in the input data set. There are $2c+1$ parameters to be estimated: θ_k, p_k ($k = 1, \ldots, c$), and σ^2.

You may want to constrain some of the EXPA parameters to known values. For example, suppose you have three coordinate variables C1, C2, and C3 and you want to constrain the powers p_k to equal 2, as in Sacks et al. (1989). Suppose further that you want to model covariance across the entire input data set and you suspect the θ_k and σ^2 estimates are close to 3, 4, 5, and 1, respectively. Then specify

```
repeated / type=sp(expa)(c1 c2 c3)
   subject=intercept;
parms (3) (4) (5) (2) (2) (2) (1) /
   hold=4,5,6;
```

TYPE=SP(EXPGA)(c_1 c_2)

TYPE=SP(GAUGA)(c_1 c_2)

TYPE=SP(SPHGA)(c_1 c_2) specify modifications of the isotropic SP(EXP), SP(SPH), and SP(GAU) covariance structures that allow for geometric anisotropy in two dimensions. The coordinates are specified by the variables c1 and c2.

If the spatial process is geometrically anisotropic in $\mathbf{c} = [c_{i1}, c_{i2}]$, then it is isotropic in the coordinate system

$$\mathbf{Ac} = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{c} = \mathbf{c}^*$$

for a properly chosen angle θ and scaling factor λ. Elliptical isocorrelation contours are thereby transformed to spherical contours, adding two parameters to the respective isotropic covariance structures. Euclidean distances (see Table 46.6 on page 2722) are expressed in terms of $\mathbf{c}^*$.

The angle θ of the clockwise rotation is reported in radians, $0 \leq \theta \leq 2\pi$. The scaling parameter λ represents the ratio of the range parameters in the direction of the major and minor axis of the correlation contours. In other words, following a rotation of the coordinate system by angle θ, isotropy is achieved by compressing or magnifying distances in one coordinate by the factor λ.

Fixing $\lambda = 1.0$ reduces the models to isotropic ones for any angle of rotation. If the scaling parameter is held constant at 1.0, you should also hold constant the angle of rotation, e.g.,

```
repeated / type=sp(expga)(gxc gyc)
           subject=intercept;
parms (6) (1.0) (0.0) (1) / hold=2,3;
```

If λ is fixed at any other value than 1.0, the angle of rotation can be estimated. Specifying a starting grid of angles and scaling factors can considerably improve the convergence properties of the optimization algorithm for these models. Only a single random effect with geometrically anisotropic structure is permitted.

TYPE=SP(MATERN)(*c-list*)

TYPE=SP(MATHSW)(*c-list*) specifies covariance structures in the Matérn class of covariance functions (Matérn 1986). Two observations for the same subject (block of $\mathbf{R}$) that are Euclidean distance d_{ij} apart have covariance

$$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}}{2\rho}\right)^{\nu} 2K_{\nu}(d_{ij}/\rho) \qquad \nu > 0,\ \rho > 0,$$

where K_ν is the modified Bessel function of the second kind of (real) order $\nu > 0$. The smoothness (continuity) of a stochastic process with covariance function in this class increases with ν. The Matérn class thus enables data-driven estimation of the smoothness properties. The covariance is identical to the exponential model for $\nu = 0.5$ (TYPE=SP(EXP)(*c-list*)), while for $\nu = 1$ the model advocated by Whittle (1954) results. As $\nu \to \infty$ the model approaches the gaussian covariance structure (TYPE=SP(GAU)(*c-list*)).

The MATHSW structure represents the Matérn class in the parameterization of Handcock and Stein (1993) and Handcock and Wallis (1994),

$$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}\sqrt{\nu}}{\rho}\right)^{\nu} 2K_{\nu}\left(\frac{2d_{ij}\sqrt{\nu}}{\rho}\right)$$

Since computation of the function K_ν and its derivatives is numerically very intensive, fitting models with Matérn covariance structures can be more time consuming than for other spatial covariance structures. Good starting values are essential.

TYPE=SP(POW)(*c-list*)

TYPE=SP(POWA)(*c-list*) specifies the spatial power structures. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

TYPE=TOEP<(q)> specifies a banded Toeplitz structure. This can be viewed as a moving-average structure with order equal to $q - 1$. The TYPE=TOEP option is a full Toeplitz matrix, which can be viewed as an autoregressive structure with order equal to the dimension of the matrix. The specification TYPE=TOEP(1) is the same as $\sigma^2 I$, where I is an identity matrix, and it can be useful for specifying the same variance component for several effects.

TYPE=TOEPH<(q)> specifies a heterogeneous banded Toeplitz structure. In Table 46.5, σ_i^2 is the ith variance parameter and ρ_j is the jth correlation parameter satisfying $|\rho_j| < 1$. If you specify the order parameter q, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to 0. The option TOEPH(1) is equivalent to both the UN(1) and UNR(1) options.

TYPE=UN<(q)> specifies a completely general (unstructured) covariance matrix parameterized directly in terms of variances and covariances. The variances are constrained to be nonnegative, and the covariances are unconstrained. This structure is not constrained to be nonnegative definite in order to avoid nonlinear constraints; however, you can use the FA0 structure if you want this constraint to be imposed by a Cholesky factorization. If you specify the order parameter q, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to 0.

TYPE=UNR<(q)> specifies a completely general (unstructured) covariance matrix parameterized in terms of variances and correlations. This structure fits the same model as the TYPE=UN(q) option but with a different parameterization. The ith variance parameter is σ_i^2. The parameter ρ_{jk} is the correlation between the jth and kth measurements; it satisfies $|\rho_{jk}| < 1$. If you specify the order parameter r, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to zero.

TYPE=UN@AR(1)

TYPE=UN@CS

TYPE=UN@UN specify direct (Kronecker) product structures designed for multivariate repeated measures (refer to Galecki 1994). These structures are constructed by taking the Kronecker product of an unstructured matrix (modeling covariance across the multivariate observations) with an additional covariance matrix (modeling covariance across time or another factor). The upper left value in the second matrix is constrained to equal 1 to identify the model. Refer to *SAS/IML User's Guide, First Edition,* for more details on direct products.

To use these structures in the REPEATED statement, you must specify two distinct REPEATED effects, both of which must be included in the CLASS statement. The first effect indicates the multivariate observations, and the second identifies the levels of time or some additional factor. Note that the input data set must still be constructed in "univariate" format; that is, all dependent observations are still listed observation-wise in one single variable. Although this construction provides for general modeling possibilities, it forces you to construct variables indicating both dimensions of the Kronecker product.

For example, suppose your observed data consist of heights and weights of several children measured over several successive years.

Your input data set should then contain variables similar to the following:

- Y, all of the heights and weights, with a separate observation for each
- Var, indicating whether the measurement is a height or a weight
- Year, indicating the year of measurement
- Child, indicating the child on which the measurement was taken

Your PROC MIXED code for a Kronecker AR(1) structure across years would then be

```
proc mixed;
   class Var Year Child;
   model Y = Var Year Var*Year;
   repeated Var Year / type=un@ar(1)
                       subject=Child;
run;
```

You should nearly always want to model different means for the multivariate observations, hence the inclusion of Var in the MODEL statement. The preceding mean model consists of cell means for all combinations of VAR and YEAR.

TYPE=VC specifies standard variance components and is the default structure for both the RANDOM and REPEATED statements. In the RANDOM statement, a distinct variance component is assigned to each effect. In the REPEATED statement, this structure is usually used only with the GROUP= option to specify a heterogeneous variance model.

Jennrich and Schluchter (1986) provide general information about the use of covariance structures, and Wolfinger (1996) presents details about many of the heterogeneous structures. Marx and Thompson (1987), Cressie (1991), and Zimmerman and Harville (1991) discuss spatial structures.

WEIGHT Statement

WEIGHT *variable* **;**

If you do not specify a REPEATED statement, the WEIGHT statement operates exactly like the one in PROC GLM. In this case PROC MIXED replaces $\mathbf{X}'\mathbf{X}$ and $\mathbf{Z}'\mathbf{Z}$ with $\mathbf{X}'\mathbf{W}\mathbf{X}$ and $\mathbf{Z}'\mathbf{W}\mathbf{Z}$, where $\mathbf{W}$ is the diagonal weight matrix. If you specify a REPEATED statement, then the WEIGHT statement replaces $\mathbf{R}$ with $\mathbf{LRL}$, where $\mathbf{L}$ is a diagonal matrix with elements $\mathbf{W}^{-1/2}$. Observations with nonpositive or missing weights are not included in the PROC MIXED analysis.

Details

Mixed Models Theory

This section provides an overview of a likelihood-based approach to general linear mixed models. This approach simplifies and unifies many common statistical analyses, including those involving repeated measures, random effects, and random coefficients. The basic assumption is that the data are linearly related to unobserved multivariate normal random variables. Extensions to nonlinear and nonnormal situations are possible but are not discussed here. Additional theory and examples are provided in Littell et al. (1996), Verbeke and Molenberghs (1997 2000), and Brown and Prescott (1999).

Matrix Notation

Suppose that you observe n data points $y_1, \ldots, y_n$ and that you want to explain them using n values for each of p explanatory variables $x_{11}, \ldots, x_{1p}$, $x_{21}, \ldots, x_{2p}$, $\ldots, x_{n1}, \ldots, x_{np}$. The x_{ij} values may be either regression-type continuous variables or dummy variables indicating class membership. The standard linear model for this setup is

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i \qquad\qquad i = 1, \ldots, n$$

where $\beta_1, \ldots, \beta_p$ are unknown *fixed-effects* parameters to be estimated and $\epsilon_1, \ldots, \epsilon_n$ are unknown independent and identically distributed normal (Gaussian) random variables with mean 0 and variance σ^2.

The preceding equations can be written simultaneously using vectors and a matrix, as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

For convenience, simplicity, and extendibility, this entire system is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ denotes the vector of observed y_i's, $\mathbf{X}$ is the known matrix of x_{ij}'s, $\boldsymbol{\beta}$ is the unknown fixed-effects parameter vector, and $\boldsymbol{\epsilon}$ is the unobserved vector of independent and identically distributed Gaussian random errors.

In addition to denoting data, random variables, and explanatory variables in the preceding fashion, the subsequent development makes use of basic matrix operators such as transpose ($'$), inverse ($^{-1}$), generalized inverse ($^{-}$), determinant ($|\cdot|$), and matrix multiplication. Refer to Searle (1982) for details on these and other matrix techniques.

Formulation of the Mixed Model

The previous general linear model is certainly a useful one (Searle 1971), and it is the one fitted by the GLM procedure. However, many times the distributional assumption about ϵ is too restrictive. The mixed model extends the general linear model by allowing a more flexible specification of the covariance matrix of ϵ. In other words, it allows for both correlation and heterogeneous variances, although you still assume normality.

The mixed model is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where everything is the same as in the general linear model except for the addition of the known design matrix, $\mathbf{Z}$, and the vector of unknown *random-effects parameters*, $\boldsymbol{\gamma}$. The matrix $\mathbf{Z}$ can contain either continuous or dummy variables, just like $\mathbf{X}$. The name *mixed model* comes from the fact that the model contains both fixed-effects parameters, $\boldsymbol{\beta}$, and random-effects parameters, $\boldsymbol{\gamma}$. Refer to Henderson (1990) and Searle, Casella, and McCulloch (1992) for historical developments of the mixed model.

A key assumption in the foregoing analysis is that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed with

$$\begin{aligned} \mathrm{E}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \\ \mathrm{Var}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} &= \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix} \end{aligned}$$

The variance of $\mathbf{y}$ is, therefore, $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$. You can model $\mathbf{V}$ by setting up the random-effects design matrix $\mathbf{Z}$ and by specifying covariance structures for $\mathbf{G}$ and $\mathbf{R}$.

Note that this is a general specification of the mixed model, in contrast to many texts and articles that discuss only simple random effects. Simple random effects are a special case of the general specification with $\mathbf{Z}$ containing dummy variables, $\mathbf{G}$ containing variance components in a diagonal structure, and $\mathbf{R} = \sigma^2\mathbf{I}_n$, where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. The general linear model is a further special case with $\mathbf{Z} = \mathbf{0}$ and $\mathbf{R} = \sigma^2\mathbf{I}_n$.

The following two examples illustrate the most common formulations of the general linear mixed model.

Example: Growth Curve with Compound Symmetry

Suppose that you have three growth curve measurements for s individuals and that you want to fit an overall linear trend in time. Your $\mathbf{X}$ matrix is as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

The first column (coded entirely with 1s) fits an intercept, and the second column (coded with times of $1, 2, 3$) fits a slope. Here, $n = 3s$ and $p = 2$.

Suppose further that you want to introduce a common correlation among the observations from a single individual, with correlation being the same for all individuals. One way of setting this up in the general mixed model is to eliminate the $\mathbf{Z}$ and $\mathbf{G}$ matrices and let the $\mathbf{R}$ matrix be block diagonal with blocks corresponding to the individuals and with each block having the *compound-symmetry* structure. This structure has two unknown parameters, one modeling a common covariance and the other a residual variance. The form for $\mathbf{R}$ would then be as follows:

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 + \sigma^2 & \sigma_1^2 & \sigma_1^2 & & & & \\ \sigma_1^2 & \sigma_1^2 + \sigma^2 & \sigma_1^2 & & & & \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma^2 & & & & \\ & & & \ddots & & & \\ & & & & \sigma_1^2 + \sigma^2 & \sigma_1^2 & \sigma_1^2 \\ & & & & \sigma_1^2 & \sigma_1^2 + \sigma^2 & \sigma_1^2 \\ & & & & \sigma_1^2 & \sigma_1^2 & \sigma_1^2 + \sigma^2 \end{bmatrix}$$

where blanks denote zeroes. There are $3s$ rows and columns altogether, and the common correlation is $\sigma_1^2/(\sigma_1^2 + \sigma^2)$.

The PROC MIXED code to fit this model is as follows:

```
proc mixed;
   class indiv;
   model y = time;
   repeated / type=cs subject=indiv;
run;
```

Here, indiv is a classification variable indexing individuals. The MODEL statement fits a straight line for time; the intercept is fit by default just as in PROC GLM. The REPEATED statement models the $\mathbf{R}$ matrix: TYPE=CS specifies the compound symmetry structure, and SUBJECT=INDIV specifies the blocks of $\mathbf{R}$.

An alternative way of specifying the common intra-individual correlation is to let

$$
\begin{aligned}
\mathbf{Z} &= \begin{bmatrix}
1 & & & \\
1 & & & \\
1 & & & \\
 & 1 & & \\
 & 1 & & \\
 & 1 & & \\
 & & \ddots & \\
 & & & 1 \\
 & & & 1 \\
 & & & 1
\end{bmatrix} \\
\mathbf{G} &= \begin{bmatrix}
\sigma_1^2 & & & \\
 & \sigma_1^2 & & \\
 & & \ddots & \\
 & & & \sigma_1^2
\end{bmatrix}
\end{aligned}
$$

and $\mathbf{R} = \sigma^2 \mathbf{I}_n$. The $\mathbf{Z}$ matrix has $3s$ rows and s columns, and $\mathbf{G}$ is $s \times s$.

You can set up this model in PROC MIXED in two different but equivalent ways:

```
proc mixed;
   class indiv;
   model y = time;
   random indiv;
run;

proc mixed;
   class indiv;
   model y = time;
   random intercept / subject=indiv;
run;
```

Both of these specifications fit the same model as the previous one that used the REPEATED statement; however, the RANDOM specifications constrain the correlation to be positive whereas the REPEATED specification leaves the correlation unconstrained.

Example: Split-Plot Design

The split-plot design involves two experimental treatment factors, A and B, and two different sizes of experimental units to which they are applied (refer to Winer 1971, Snedecor and Cochran 1980, Milliken and Johnson 1992, and Steel, Torrie, and Dickey 1997). The levels of A are randomly assigned to the larger sized experimental unit, called *whole plots*, whereas the levels of B are assigned to the smaller sized experimental unit, the *subplots*. The subplots are assumed to be nested within the whole plots, so that a whole plot consists of a cluster of subplots and a level of A is applied to the entire cluster.

Such an arrangement is often necessary by nature of the experiment, the classical example being the application of fertilizer to large plots of land and different crop varieties planted in subdivisions of the large plots. For this example, fertilizer is the whole plot factor A and variety is the subplot factor B.

The first example is a split-plot design for which the whole plots are arranged in a randomized block design. The appropriate PROC MIXED code is as follows:

```
proc mixed;
   class a b block;
   model y = a|b;
   random block a*block;
run;
```

Here

$$\mathbf{R} = \sigma^2 \mathbf{I}_{24}$$

and $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{G}$ have the following form:

$$\mathbf{X} = \begin{bmatrix}
1 & 1 & & & 1 & & 1 & & & & & \\
1 & 1 & & & & 1 & & 1 & & & & \\
1 & & 1 & & 1 & & & & 1 & & & \\
1 & & 1 & & & 1 & & & & 1 & & \\
1 & & & 1 & 1 & & & & & & 1 & \\
1 & & & 1 & & 1 & & & & & & 1 \\
\vdots & & \vdots & & \vdots & & & & & \vdots & & \\
1 & 1 & & & 1 & & 1 & & & & & \\
1 & 1 & & & & 1 & & 1 & & & & \\
1 & & 1 & & 1 & & & & 1 & & & \\
1 & & 1 & & & 1 & & & & 1 & & \\
1 & & & 1 & 1 & & & & & & 1 & \\
1 & & & 1 & & 1 & & & & & & 1
\end{bmatrix}$$

$$
\mathbf{Z} = \begin{bmatrix}
1 & & & & 1 \\
1 & & & & 1 \\
1 & & & & & 1 \\
1 & & & & & 1 \\
1 & & & & & & 1 \\
1 & & & & & & 1 \\
& 1 & & & & & & 1 \\
& 1 & & & & & & 1 \\
& 1 & & & & & & & 1 \\
& 1 & & & & & & & 1 \\
& 1 & & & & & & & & 1 \\
& 1 & & & & & & & & 1 \\
& & 1 & & & & & & & & 1 \\
& & 1 & & & & & & & & 1 \\
& & 1 & & & & & & & & & 1 \\
& & 1 & & & & & & & & & 1 \\
& & 1 & & & & & & & & & & 1 \\
& & 1 & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & & & 1 \\
& & & 1 & & & & & & & & & & & & 1
\end{bmatrix}
$$

$$
\mathbf{G} = \begin{bmatrix}
\sigma_B^2 \\
& \sigma_B^2 \\
& & \sigma_B^2 \\
& & & \sigma_B^2 \\
& & & & \sigma_{AB}^2 \\
& & & & & \sigma_{AB}^2 \\
& & & & & & \ddots \\
& & & & & & & \sigma_{AB}^2
\end{bmatrix}
$$

where σ_B^2 is the variance component for Block and σ_{AB}^2 is the variance component for A*Block. Changing the RANDOM statement to

```
random int a / subject=block;
```

fits the same model, but with $\mathbf{Z}$ and $\mathbf{G}$ sorted differently.

$$\mathbf{Z} = \begin{bmatrix}
1 & 1 & & & & & & & & & & & & & & \\
1 & 1 & & & & & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & & & & \\
1 & & & 1 & & & & & & & & & & & & \\
1 & & & 1 & & & & & & & & & & & & \\
& & & & 1 & 1 & & & & & & & & & & \\
& & & & 1 & 1 & & & & & & & & & & \\
& & & & 1 & & 1 & & & & & & & & & \\
& & & & 1 & & 1 & & & & & & & & & \\
& & & & 1 & & & 1 & & & & & & & & \\
& & & & 1 & & & 1 & & & & & & & & \\
& & & & & & & & 1 & 1 & & & & & & \\
& & & & & & & & 1 & 1 & & & & & & \\
& & & & & & & & 1 & & 1 & & & & & \\
& & & & & & & & 1 & & 1 & & & & & \\
& & & & & & & & 1 & & & 1 & & & & \\
& & & & & & & & 1 & & & 1 & & & & \\
& & & & & & & & & & & & 1 & 1 & & \\
& & & & & & & & & & & & 1 & 1 & & \\
& & & & & & & & & & & & 1 & & 1 & \\
& & & & & & & & & & & & 1 & & 1 & \\
& & & & & & & & & & & & 1 & & & 1 \\
& & & & & & & & & & & & 1 & & & 1
\end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix}
\sigma_B^2 & & & & & & & & \\
& \sigma_{AB}^2 & & & & & & & \\
& & \sigma_{AB}^2 & & & & & & \\
& & & \sigma_{AB}^2 & & & & & \\
& & & & \ddots & & & & \\
& & & & & \sigma_B^2 & & & \\
& & & & & & \sigma_{AB}^2 & & \\
& & & & & & & \sigma_{AB}^2 & \\
& & & & & & & & \sigma_{AB}^2
\end{bmatrix}$$

Estimating G and R in the Mixed Model

Estimation is more difficult in the mixed model than in the general linear model. Not only do you have $\boldsymbol{\beta}$ as in the general linear model, but you have unknown parameters in $\boldsymbol{\gamma}$, $\mathbf{G}$, and $\mathbf{R}$ as well. Least squares is no longer the best method. *Generalized least squares* (GLS) is more appropriate, minimizing

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

However, it requires knowledge of $\mathbf{V}$ and, therefore, knowledge of $\mathbf{G}$ and $\mathbf{R}$. Lacking such information, one approach is to use *estimated* GLS, in which you insert some reasonable estimate for $\mathbf{V}$ into the minimization problem. The goal thus becomes finding a reasonable estimate of $\mathbf{G}$ and $\mathbf{R}$.

In many situations, the best approach is to use *likelihood-based* methods, exploiting the assumption that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed (Hartley and Rao 1967; Patterson and Thompson 1971; Harville 1977; Laird and Ware 1982; Jennrich and Schluchter 1986). PROC MIXED implements two likelihood-based methods: *maximum likelihood* (ML) and *restricted/residual maximum likelihood* (REML). A favorable theoretical property of ML and REML is that they accommodate data that are missing at random (Rubin 1976; Little 1995).

PROC MIXED constructs an objective function associated with ML or REML and maximizes it over all unknown parameters. Using calculus, it is possible to reduce this maximization problem to one over only the parameters in $\mathbf{G}$ and $\mathbf{R}$. The corresponding log-likelihood functions are as follows:

$$\begin{aligned}
\text{ML:} \quad l(\mathbf{G},\mathbf{R}) &= -\frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\mathbf{r}'\mathbf{V}^{-1}\mathbf{r} - \frac{n}{2}\log(2\pi) \\
\text{REML:} \quad l_R(\mathbf{G},\mathbf{R}) &= -\frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\log|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}| \\
&\quad -\frac{1}{2}\mathbf{r}'\mathbf{V}^{-1}\mathbf{r} - \frac{n-p}{2}\log(2\pi)\}
\end{aligned}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$ and p is the rank of $\mathbf{X}$. PROC MIXED actually minimizes -2 times these functions using a ridge-stabilized Newton-Raphson algorithm. Lindstrom and Bates (1988) provide reasons for preferring Newton-Raphson to the Expectation-Maximum (EM) algorithm described in Dempster, Laird, and Rubin (1977) and Laird, Lange, and Stram (1987), as well as analytical details for implementing a QR-decomposition approach to the problem. Wolfinger, Tobias, and Sall (1994) present the sweep-based algorithms that are implemented in PROC MIXED.

One advantage of using the Newton-Raphson algorithm is that the second derivative matrix of the objective function evaluated at the optima is available upon completion. Denoting this matrix $\mathbf{H}$, the asymptotic theory of maximum likelihood (refer to Serfling 1980) shows that $2\mathbf{H}^{-1}$ is an asymptotic variance-covariance matrix of the estimated parameters of $\mathbf{G}$ and $\mathbf{R}$. Thus, tests and confidence intervals based on asymptotic normality can be obtained. However, these can be unreliable in small samples, especially for parameters such as variance components which have sampling distributions that tend to be skewed to the right.

If a residual variance σ^2 is a part of your mixed model, it can usually be *profiled* out of the likelihood. This means solving analytically for the optimal σ^2 and plugging this expression back into the likelihood formula (refer to Wolfinger, Tobias, and Sall 1994). This reduces the number of optimization parameters by one and can improve convergence properties. PROC MIXED profiles the residual variance out of the log likelihood whenever it appears reasonable to do so. This includes the case when $\mathbf{R}$ equals $\sigma^2\mathbf{I}$ and when it has blocks with a compound symmetry, time series, or spatial structure. PROC MIXED does not profile the log likelihood when $\mathbf{R}$ has unstructured blocks, when you use the HOLD= or NOITER option in the PARMS statement, or when you use the NOPROFILE option in the PROC MIXED statement.

Instead of ML or REML, you can use the noniterative MIVQUE0 method to estimate $\mathbf{G}$ and $\mathbf{R}$ (Rao 1972; LaMotte 1973; Wolfinger, Tobias, and Sall 1994). In fact, by default PROC MIXED uses MIVQUE0 estimates as starting values for the ML and REML procedures. For variance component models, another estimation method involves equating Type I, II, or III expected mean squares to their observed values and solving the resulting system. However, Swallow and Monahan (1984) present simulation evidence favoring REML and ML over MIVQUE0 and other method-of-moment estimators.

Estimating β and γ in the Mixed Model

ML, REML, MIVQUE0, or Type1–Type3 provide estimates of $\mathbf{G}$ and $\mathbf{R}$, which are denoted $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$, respectively. To obtain estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the standard method is to solve the *mixed model equations* (Henderson 1984):

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}+\widehat{\mathbf{G}}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

The solutions can also be written as

$$\begin{aligned} \widehat{\boldsymbol{\beta}} &= (\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{y} \\ \widehat{\boldsymbol{\gamma}} &= \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}) \end{aligned}$$

and have connections with empirical Bayes estimators (Laird and Ware 1982, Carlin and Louis 1996).

Note that the mixed model equations are extended normal equations and that the preceding expression assumes that $\widehat{\mathbf{G}}$ is nonsingular. For the extreme case when the eigenvalues of $\widehat{\mathbf{G}}$ are very large, $\widehat{\mathbf{G}}^{-1}$ contributes very little to the equations and $\widehat{\boldsymbol{\gamma}}$ is close to what it would be if $\boldsymbol{\gamma}$ actually contained fixed-effects parameters. On the other hand, when the eigenvalues of $\widehat{\mathbf{G}}$ are very small, $\widehat{\mathbf{G}}^{-1}$ dominates the equations and $\widehat{\boldsymbol{\gamma}}$ is close to 0. For intermediate cases, $\widehat{\mathbf{G}}^{-1}$ can be viewed as shrinking the fixed-effects estimates of $\boldsymbol{\gamma}$ towards 0 (Robinson 1991).

If $\widehat{\mathbf{G}}$ is singular, then the mixed model equations are modified (Henderson 1984) as follows:

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{L}} \\ \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{L}}+\mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

where $\widehat{\mathbf{L}}$ is the lower-triangular Cholesky root of $\widehat{\mathbf{G}}$, satisfying $\widehat{\mathbf{G}} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}'$. Both $\widehat{\boldsymbol{\tau}}$ and a generalized inverse of the left-hand-side coefficient matrix are then transformed using $\widehat{\mathbf{L}}$ to determine $\widehat{\boldsymbol{\gamma}}$.

An example of when the singular form of the equations is necessary is when a variance component estimate falls on the boundary constraint of 0.

Model Selection

The previous section on estimation assumes the specification of a mixed model in terms of $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{G}$, and $\mathbf{R}$. Even though $\mathbf{X}$ and $\mathbf{Z}$ have known elements, their specific form and construction is flexible, and several possibilities may present themselves for a particular data set. Likewise, several different covariance structures for $\mathbf{G}$ and $\mathbf{R}$ might be reasonable.

Space does not permit a thorough discussion of model selection, but a few brief comments and references are in order. First, subject matter considerations and objectives are of great importance when selecting a model; refer to Diggle (1988) and Lindsey (1993).

Second, when the data themselves are looked to for guidance, many of the graphical methods and diagnostics appropriate for the general linear model extend to the mixed model setting as well (Christensen, Pearson, and Johnson 1992).

Finally, a likelihood-based approach to the mixed model provides several statistical measures for model adequacy as well. The most common of these are the likelihood ratio test and Akaike's and Schwarz's criteria (Bozdogan 1987; Wolfinger 1993, Keselman et al. 1998, 1999).

Statistical Properties

If $\mathbf{G}$ and $\mathbf{R}$ are known, $\widehat{\boldsymbol{\beta}}$ is the *best linear unbiased estimator* (BLUE) of $\boldsymbol{\beta}$, and $\widehat{\gamma}$ is the *best linear unbiased predictor* (BLUP) of γ (Searle 1971; Harville 1988, 1990; Robinson 1991; McLean, Sanders, and Stroup 1991). Here, "best" means minimum mean squared error. The covariance matrix of $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}, \widehat{\gamma} - \gamma)$ is

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix}^{-}$$

where $^{-}$ denotes a generalized inverse (refer to Searle 1971).

However, $\mathbf{G}$ and $\mathbf{R}$ are usually unknown and are estimated using one of the aforementioned methods. These estimates, $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$, are therefore simply substituted into the preceding expression to obtain

$$\widehat{\mathbf{C}} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} + \widehat{\mathbf{G}}^{-1} \end{bmatrix}^{-}$$

as the approximate variance-covariance matrix of $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}, \widehat{\gamma} - \gamma)$. In this case, the BLUE and BLUP acronyms no longer apply, but the word *empirical* is often added to indicate such an approximation. The appropriate acronyms thus become EBLUE and EBLUP.

McLean and Sanders (1988) show that $\widehat{\mathbf{C}}$ can also be written as

$$\widehat{\mathbf{C}} = \begin{bmatrix} \widehat{\mathbf{C}}_{11} & \widehat{\mathbf{C}}'_{21} \\ \widehat{\mathbf{C}}_{21} & \widehat{\mathbf{C}}_{22} \end{bmatrix}$$

where

$$\begin{aligned}
\widehat{\mathbf{C}}_{11} &= (\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-} \\
\widehat{\mathbf{C}}_{21} &= -\widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}\mathbf{X}\widehat{\mathbf{C}}_{11} \\
\widehat{\mathbf{C}}_{22} &= (\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} + \widehat{\mathbf{G}}^{-1})^{-1} - \widehat{\mathbf{C}}_{21}\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{Z}\widehat{\mathbf{G}}
\end{aligned}$$

Note that $\widehat{\mathbf{C}}_{11}$ is the familiar estimated generalized least-squares formula for the variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$.

As a cautionary note, $\widehat{\mathbf{C}}$ tends to underestimate the true sampling variability of $(\widehat{\boldsymbol{\beta}}\ \widehat{\gamma})$ because no account is made for the uncertainty in estimating $\mathbf{G}$ and $\mathbf{R}$. Although inflation factors have been proposed (Kackar and Harville 1984; Kass and Steffey 1989; Prasad and Rao 1990), they tend to be small for data sets that are fairly well balanced. PROC MIXED does not compute any inflation factors by default, but rather accounts for the downward bias by using the approximate t and F statistics described subsequently. The DDFM=KENWARDROGER option in the MODEL statement prompts PROC MIXED to compute a specific inflation factor along with Satterthwaite-based degrees of freedom.

Inference and Test Statistics

For inferences concerning the covariance parameters in your model, you can use likelihood-based statistics. One common likelihood-based statistic is the *Wald* Z, which is computed as the parameter estimate divided by its asymptotic standard error. The asymptotic standard errors are computed from the inverse of the second derivative matrix of the likelihood with respect to each of the covariance parameters. The Wald Z is valid for large samples, but it can be unreliable for small data sets and for parameters such as variance components, which are known to have a skewed or bounded sampling distribution.

A better alternative is the likelihood ratio χ^2. This statistic compares two covariance models, one a special case of the other. To compute it, you must run PROC MIXED twice, once for each of the two models, and then subtract the corresponding values of -2 times the log likelihoods. You can use either ML or REML to construct this statistic, which tests whether the full model is necessary beyond the reduced model.

As long as the reduced model does not occur on the boundary of the covariance parameter space, the χ^2 statistic computed in this fashion has a large-sample sampling distribution that is χ^2 with degrees of freedom equal to the difference in the number of covariance parameters between the two models. If the reduced model does occur on the boundary of the covariance parameter space, the asymptotic distribution becomes a mixture of χ^2 distributions (Self and Liang 1987). A common example of this is when you are testing that a variance component equals its lower boundary constraint of 0.

A final possibility for obtaining inferences concerning the covariance parameters is to simulate or resample data from your model and construct empirical sampling distributions of the parameters. The SAS macro language and the ODS system are useful tools in this regard.

For inferences concerning the fixed- and random-effects parameters in the mixed model, consider estimable linear combinations of the following form:

$$\mathbf{L}\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{bmatrix}$$

The estimability requirement (Searle 1971) applies only to the $\boldsymbol{\beta}$-portion of $\mathbf{L}$, as any linear combination of $\boldsymbol{\gamma}$ is estimable. Such a formulation in terms of a general $\mathbf{L}$ matrix encompasses a wide variety of common inferential procedures such as those employed with Type I–Type III tests and LS-means. The CONTRAST and ESTIMATE statements in PROC MIXED enable you to specify your own $\mathbf{L}$ matrices. Typically, inference on fixed-effects is the focus, and, in this case, the $\boldsymbol{\gamma}$-portion of $\mathbf{L}$ is assumed to contain all 0s.

Statistical inferences are obtained by testing the hypothesis

$$H : \mathbf{L}\begin{bmatrix}\boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{bmatrix} = 0$$

or by constructing point and interval estimates.

When $\mathbf{L}$ consists of a single row, a general t-statistic can be constructed as follows (refer to McLean and Sanders 1988, Stroup 1989a):

$$t = \frac{\mathbf{L}\begin{bmatrix}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\gamma}}\end{bmatrix}}{\sqrt{\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}'}}$$

Under the assumed normality of $\boldsymbol{\gamma}$ and ϵ, t has an exact t-distribution only for data exhibiting certain types of balance and for some special unbalanced cases. In general, t is only approximately t-distributed, and its degrees of freedom must be estimated. See the DDFM= option on page 2693 for a description of the various degrees-of-freedom methods available in PROC MIXED.

With $\widehat{\nu}$ being the approximate degrees of freedom, the associated confidence interval is

$$\mathbf{L}\begin{bmatrix}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\gamma}}\end{bmatrix} \pm t_{\widehat{\nu},\alpha/2}\sqrt{\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}'}$$

where $t_{\widehat{\nu},\alpha/2}$ is the $(1-\alpha/2)100$th percentile of the $t_{\widehat{\nu}}$-distribution.

When the rank of $\mathbf{L}$ is greater than 1, PROC MIXED constructs the following general F-statistic:

$$F = \frac{\begin{bmatrix}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\gamma}}\end{bmatrix}'\mathbf{L}'(\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}')^{-1}\mathbf{L}\begin{bmatrix}\widehat{\boldsymbol{\beta}}\\ \widehat{\boldsymbol{\gamma}}\end{bmatrix}}{\mathrm{rank}(\mathbf{L})}$$

Analogous to t, F in general has an approximate F-distribution with $\mathrm{rank}(\mathbf{L})$ numerator degrees of freedom and $\widehat{\nu}$ denominator degrees of freedom.

The t- and F-statistics enable you to make inferences about your fixed effects, which account for the variance-covariance model you select. An alternative is the χ^2 statistic associated with the likelihood ratio test. This statistic compares two fixed-effects models, one a special case of the other. It is computed just as when comparing different covariance models, although you should use ML and not REML here because the penalty term associated with restricted likelihoods depends upon the fixed-effects specification.

Parameterization of Mixed Models

Recall that a mixed model is of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ represents univariate data, $\boldsymbol{\beta}$ is an unknown vector of fixed effects with known model matrix $\mathbf{X}$, $\boldsymbol{\gamma}$ is an unknown vector of random effects with known model matrix $\mathbf{Z}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector.

PROC MIXED constructs a mixed model according to the specifications in the MODEL, RANDOM, and REPEATED statements. Each effect in the MODEL statement generates one or more columns in the model matrix $\mathbf{X}$, and each effect in the RANDOM statement generates one or more columns in the model matrix $\mathbf{Z}$. Effects in the REPEATED statement do not generate model matrices; they serve only to index observations within subjects. This section shows precisely how PROC MIXED builds $\mathbf{X}$ and $\mathbf{Z}$.

Intercept

By default, all models automatically include a column of 1s in $\mathbf{X}$ to estimate a fixed-effect intercept parameter μ. You can use the NOINT option in the MODEL statement to suppress this intercept. The NOINT option is useful when you are specifying a classification effect in the MODEL statement and you want the parameter estimate to be in terms of the mean response for each level of that effect, rather than in terms of a deviation from an overall mean.

By contrast, the intercept is not included by default in $\mathbf{Z}$. To obtain a column of 1s in $\mathbf{Z}$, you must specify in the RANDOM statement either the INTERCEPT effect or some effect that has only one level.

Regression Effects

Numeric variables, or polynomial terms involving them, may be included in the model as regression effects (covariates). The actual values of such terms are included as columns of the model matrices $\mathbf{X}$ and $\mathbf{Z}$. You can use the bar operator with a regression effect to generate polynomial effects. For instance, X|X|X expands to X X*X X*X*X, a cubic model.

Main Effects

If a class variable has *m* levels, PROC MIXED generates *m* columns in the model matrix for its main effect. Each column is an indicator variable for a given level. The order of the columns is the sort order of the values of their levels and can be controlled with the ORDER= option in the PROC MIXED statement. The following table is an example.

Data		**I**	A		B		
A	B	μ	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

Typically, there are more columns for these effects than there are degrees of freedom for them. In other words, PROC MIXED uses an over-parameterized model.

Interaction Effects

Often a model includes interaction (crossed) effects. With an interaction, PROC MIXED first reorders the terms to correspond to the order of the variables in the CLASS statement. Thus, B*A becomes A*B if A precedes B in the CLASS statement. Then, PROC MIXED generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the cross index faster than the leftmost variables. Empty columns (that would contain all 0s) are not generated for **X**, but they are for **Z**.

Data		**I**	A		B			A*B					
A	B	μ	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In the preceding matrix, main-effects columns are not linearly independent of crossed-effect columns; in fact, the column space for the crossed effects contains the space of the main effect.

When your model contains many interaction effects, you may be able to code them more parsimoniously using the bar operator (|). The bar operator generates all possible interaction effects. For example, A|B|C expands to A B A*B C A*C B*C A*B*C. To eliminate higher-order interaction effects, use the at sign (@) in conjunction with the bar operator. For instance, A|B|C|D@2 expands to A B A*B C A*C B*C D A*D B*D C*D.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following two statements are the same (but the ordering of the columns is different):

```
model Y=A B(A);

model Y=A A*B;
```

The nesting operator in PROC MIXED is more a notational convenience than an operation distinct from crossing. Nested effects are typically characterized by the property that the nested variables never appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables.

Data		**I**	A		B(A)					
A	B	μ	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

Note that nested effects are often distinguished from interaction effects by the implied randomization structure of the design. That is, they usually indicate random effects within a fixed-effects framework. The fact that random effects can be modeled directly in the RANDOM statement may make the specification of nested effects in the MODEL statement unnecessary.

Continuous-Nesting-Class Effects

When a continuous variable nests with a class variable, the design columns are constructed by multiplying the continuous values into the design columns for the class effect.

Data		**I**	A		X(A)	
X	A	μ	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate slope for X within each level of A.

Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. The two models are made different by the presence of the continuous variable as a regressor by itself, as well as a contributor to a compound effect.

Data		**I**	**X**	A		X*A	
X	A	μ	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

You can use continuous-by-class effects to test for homogeneity of slopes.

General Effects

An example that combines all the effects is X1*X2*A*B*C(D E). The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses. You should be aware of the sequencing of parameters when you use the CONTRAST or ESTIMATE statements to compute some function of the parameter estimates.

Effects may be renamed by PROC MIXED to correspond to ordering rules. For example, B*A(E D) may be renamed A*B(D E) to satisfy the following:

- Class variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in the CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed list index faster than variables in the nested list.

- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. If the CLASS statement is

```
class A B C D;
```

then the order of the parameters for the effect B*A(C D), which is renamed A*B(C D), is

$$\begin{array}{llll}
A_1B_1C_1D_1 \rightarrow & A_1B_2C_1D_1 \rightarrow & A_2B_1C_1D_1 \rightarrow & A_2B_2C_1D_1 \rightarrow \\
A_1B_1C_1D_2 \rightarrow & A_1B_2C_1D_2 \rightarrow & A_2B_1C_1D_2 \rightarrow & A_2B_2C_1D_2 \rightarrow \\
A_1B_1C_2D_1 \rightarrow & A_1B_2C_2D_1 \rightarrow & A_2B_1C_2D_1 \rightarrow & A_2B_2C_2D_1 \rightarrow \\
A_1B_1C_2D_2 \rightarrow & A_1B_2C_2D_2 \rightarrow & A_2B_1C_2D_2 \rightarrow & A_2B_2C_2D_2
\end{array}$$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect moves fastest because it is rightmost in the cross list. Then A moves next fastest, and D moves next fastest. The C effect is the slowest since it is leftmost in the nested list.

When numeric levels are used, levels are sorted by their character format, which may not correspond to their numeric sort sequence (for example, noninteger levels). Therefore, it is advisable to include a desired format for numeric levels or to use the ORDER=INTERNAL option in the PROC MIXED statement to ensure that levels are sorted by their internal values.

Implications of the Non-Full-Rank Parameterization

For models with fixed-effects involving class variables, there are more design columns in $\mathbf{X}$ constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns of $\mathbf{X}$. In this event, all of the parameters are not estimable; there is an infinite number of solutions to the mixed model equations. PROC MIXED uses a generalized (g2) inverse to obtain values for the estimates (Searle 1971). The solution values are not displayed unless you specify the SOLUTION option in the MODEL statement. The solution has the characteristic that estimates are 0 whenever the design column for that parameter is a linear combination of previous columns. With this parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Some procedures (such as the CATMOD procedure) reparameterize models to full rank using restrictions on the parameters. PROC GLM and PROC MIXED do not reparameterize, making the hypotheses that are commonly tested more understandable. Refer to Goodnight (1978) for additional reasons for not reparameterizing.

Missing Level Combinations

PROC MIXED handles missing level combinations of classification variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST and ESTIMATE coefficients.

Default Output

The following sections describe the output PROC MIXED produces by default. This output is organized into various tables, and they are discussed in order of appearance.

Model Information

The “Model Information” table describes the model, some of the variables it involves, and the method used in fitting it. It also lists the method (profile, fit, factor, or none) for handling the residual variance in the model. The *profile* method concentrates the residual variance out of the optimization problem, whereas the *fit* method retains it as a parameter in the optimization. The *factor* method keeps the residual fixed, and *none* is displayed when a residual variance is not a part of the model.

The “Model Information” table also has a row labeled Fixed Effects SE Method. This row describes the method used to compute the approximate standard errors for the fixed-effects parameter estimates and related functions of them. The two possibilities for this row are Model-Based, which is the default method, and Empirical, which results from using the EMPIRICAL option in the PROC MIXED statement.

For ODS purposes, the label of the “Model Information” table is “ModelInfo.”

Class Level Information

The “Class Level Information” table lists the levels of every variable specified in the CLASS statement. You should check this information to make sure the data are correct. You can adjust the order of the CLASS variable levels with the ORDER= option in the PROC MIXED statement. For ODS purposes, the label of the “Class Level Information” table is “ClassLevels.”

Dimensions

The “Dimensions” table lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements. For ODS purposes, the label of the “Dimensions” table is “Dimensions.”

Number of Observations

The “Number of Observations” table shows the number of observations read from the data set and the number of observations used in fitting the model.

Iteration History

The “Iteration History” table describes the optimization of the residual log likelihood or log likelihood described on page 2738. The function to be minimized (the *objective function*) is $-2l$ for ML and $-2l_R$ for REML; the column name of the objective function in the “Iteration History” table is “-2 Log Like” for ML and “-2 Res Log Like” for REML. The minimization is performed using a ridge-stabilized Newton-Raphson algorithm, and the rows of this table describe the iterations that this algorithm takes in order to minimize the objective function.

The Evaluations column of the “Iteration History” table tells how many times the objective function is evaluated during each iteration.

The Criterion column of the “Iteration History” table is, by default, a relative Hessian convergence quantity given by

$$\frac{\mathbf{g}_k'\mathbf{H}_k^{-1}\mathbf{g}_k}{|f_k|}$$

where f_k is the value of the objective function at iteration *k*, $\mathbf{g}_k$ is the gradient (first derivative) of f_k, and $\mathbf{H}_k$ is the Hessian (second derivative) of f_k. If $\mathbf{H}_k$ is singular, then PROC MIXED uses the following relative quantity:

$$\frac{\mathbf{g}_k'\mathbf{g}_k}{|f_k|}$$

To prevent the division by $|f_k|$, use the ABSOLUTE option in the PROC MIXED statement. To use a relative function or gradient criterion, use the CONVF or CONVG options, respectively.

The Hessian criterion is considered superior to function and gradient criteria because it measures orthogonality rather than lack of progress (Bates and Watts 1988). Provided the initial estimate is feasible and the maximum number of iterations is not exceeded, the Newton-Raphson algorithm is considered to have converged when the criterion is less than the tolerance specified with the CONVF, CONVG, or CONVH option in the PROC MIXED statement. The default tolerance is 1E$-$8. If convergence is not achieved, PROC MIXED displays the estimates of the parameters at the last iteration.

A convergence criterion that is missing indicates that a boundary constraint has been dropped; it is usually not a cause for concern.

If you specify the ITDETAILS option in the PROC MIXED statement, then the covariance parameter estimates at each iteration are included as additional columns in the “Iteration History” table.

For ODS purposes, the label of the “Iteration History” table is “IterHistory.”

Covariance Parameter Estimates

The “Covariance Parameter Estimates” table contains the estimates of the parameters in $\mathbf{G}$ and $\mathbf{R}$ (see the “Estimating G and R in the Mixed Model” section on page 2737). Their values are labeled in the “Cov Parm” table along with Subject and Group information if applicable. The estimates are displayed in the Estimate column and are the results of one of the following estimation methods: REML, ML, MIVQUE0, SSCP, Type1, Type2, or Type3.

If you specify the RATIO option in the PROC MIXED statement, the Ratio column is added to the table listing the ratios of each parameter estimate to that of the residual variance.

Requesting the COVTEST option in the PROC MIXED statement produces the Std Error, Z Value, and Pr Z columns. The Std Error column contains the approximate standard errors of the covariance parameter estimates. These are the square roots of the diagonal elements of the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian matrix. The $\mathbf{H}$ matrix consists of the second derivatives of the objective function with respect to the covariance parameters; refer to Wolfinger, Tobias, and Sall (1994) for formulas. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. The observed or expected inverse Fisher information matrix can be viewed as an asymptotic covariance matrix of the estimates.

The Z Value column is the estimate divided by its approximate standard error, and the Pr Z column is the one- or two-tailed area of the standard Gaussian density outside of the Z-value. The MIXED procedure computes one-sided p-values for the residual variance and for covariance parameters with a lower bound of 0. The procedure computes two-sided p-values otherwise. These statistics constitute Wald tests of the covariance parameters, and they are valid only asymptotically.

Caution: Wald tests can be unreliable in small samples.

For ODS purposes, the label of the “Covariance Parameter Estimates” table is “CovParms.”

Fit Statistics

The “Fit Statistics” table provides some statistics about the estimated mixed model. Expressions for the -2 times the log likelihood are provided in the “Estimating G and R in the Mixed Model” section on page 2737. If the log likelihood is an extremely large number, then PROC MIXED has deemed the estimated $\mathbf{V}$ matrix to be singular. In this case, all subsequent results should be viewed with caution.

In addition, the “Fit Statistics” table lists three information criteria: AIC, AICC, and BIC, all in smaller-is-better form. Expressions for these criteria are described under the IC option on page 2676.

For ODS purposes, the label of the “Model Fitting Information” table is “FitStatistics.”

Null Model Likelihood Ratio Test

If one covariance model is a submodel of another, you can carry out a likelihood ratio test for the significance of the more general model by computing -2 times the difference between their log likelihoods. Then compare this statistic to the χ^2 distribution with degrees of freedom equal to the difference in the number of parameters for the two models.

This test is reported in the “Null Model Likelihood Ratio Test” table to determine whether it is necessary to model the covariance structure of the data at all. The “Chi-Square” value is -2 times the log likelihood from the null model minus -2 times the log likelihood from the fitted model, where the null model is the one with only the fixed effects listed in the MODEL statement and $\mathbf{R} = \sigma^2\mathbf{I}$. This statistic has an asymptotic χ^2-distribution with $q - 1$ degrees of freedom, where *q* is the effective number of covariance parameters (those not estimated to be on a boundary constraint). The Pr > ChiSq column contains the upper-tail area from this distribution. This *p*-value can be used to assess the significance of the model fit.

This test is not produced for cases where the null hypothesis lies on the boundary of the parameter space, which is typically for variance component models. This is because the standard asymptotic theory does not apply in this case (Self and Liang 1987, Case 5).

If you specify a PARMS statement, PROC MIXED constructs a likelihood ratio test between the best model from the grid search and the final fitted model and reports the results in the “Parameter Search” table.

For ODS purposes, the label of the “Null Model Likelihood Ratio Test” table is “LRT.”

Type 3 Tests of Fixed Effects

The “Type 3 Tests of Fixed Effects” table contains hypothesis tests for the significance of each of the fixed effects, that is, those effects you specify in the MODEL statement. By default, PROC MIXED computes these tests by first constructing a Type III **L** matrix (see Chapter 11, “The Four Types of Estimable Functions,”) for each effect. This **L** matrix is then used to compute the following *F*-statistic:

$$F = \frac{\widehat{\boldsymbol{\beta}}'\mathbf{L}'[\mathbf{L}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}']^{-}\mathbf{L}\widehat{\boldsymbol{\beta}}}{\mathrm{rank}(\mathbf{L})}$$

A p-value for the test is computed as the tail area beyond this statistic from an *F*-distribution with NDF and DDF degrees of freedom. The numerator degrees of freedom (NDF) is the row rank of **L**, and the denominator degrees of freedom is computed using one of the methods described under the DDFM= option on page 2693. Small values of the p-value (typically less than 0.05 or 0.01) indicate a significant effect.

You can use the HTYPE= option in the MODEL statement to obtain tables of Type I (sequential) tests and Type II (adjusted) tests in addition to or instead of the table of Type III (partial) tests.

You can use the CHISQ option in the MODEL statement to obtain Wald χ^2 tests of the fixed effects. These are carried out by using the numerator of the F-statistic and comparing it with the χ^2 distribution with NDF degrees of freedom. It is more liberal than the F-test because it effectively assumes an infinite denominator degrees of freedom.

For ODS purposes, the label of the “Type 1 Tests of Fixed Effects” through the “Type 3 Tests of Fixed Effects” tables are “Tests1” through “Tests3,” respectively.

ODS Table Names

Each table created by PROC MIXED has a name associated with it, and you must use this name to reference the table when using ODS statements. These names are listed in Table 46.8.

Table 46.8. ODS Tables Produced in PROC MIXED

Table Name	Description	Required Statement / Option
AccRates	acceptance rates for posterior sampling	PRIOR
AsyCorr	asymptotic correlation matrix of covariance parameters	PROC MIXED ASYCORR
AsyCov	asymptotic covariance matrix of covariance parameters	PROC MIXED ASYCOV
Base	base densities used for posterior sampling	PRIOR
Bound	computed bound for posterior rejection sampling	PRIOR
CholG	Cholesky root of the estimated **G** matrix	RANDOM / GC
CholR	Cholesky root of blocks of the estimated **R** matrix	REPEATED / RC
CholV	Cholesky root of blocks of the estimated **V** matrix	RANDOM / VC
ClassLevels	level information from the CLASS statement	default output
Coef	**L** matrix coefficients	E option on MODEL, CONTRAST, ESTIMATE, or LSMEANS
Contrasts	results from the CONTRAST statements	CONTRAST
ConvergenceStatus	convergence status	default
CorrB	approximate correlation matrix of fixed-effects parameter estimates	MODEL / CORRB
CovB	approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVB
CovParms	estimated covariance parameters	default output
Diffs	differences of LS-means	LSMEANS / DIFF (or PDIFF)
Dimensions	dimensions of the model	default output
Estimates	results from ESTIMATE statements	ESTIMATE

Table 46.8. (continued)

Table Name	**Description**	**Required Statement / Option**
FitStatistics	fit statistics	default
G	estimated **G** matrix	RANDOM / G
GCorr	correlation matrix from the estimated **G** matrix	RANDOM / GCORR
HLM1	Type 1 Hotelling-Lawley-McKeon tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLM TYPE=UN
HLM2	Type 2 Hotelling-Lawley-McKeon tests of fixed effects	MODEL / HTYPE=2 and REPEATED / HLM TYPE=UN
HLM3	Type 3 Hotelling-Lawley-McKeon tests of fixed effects	REPEATED / HLM TYPE=UN
HLPS1	Type 1 Hotelling-Lawley-Pillai-Samson tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLPS TYPE=UN
HLPS2	Type 2 Hotelling-Lawley-Pillai-Samson tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLPS TYPE=UN
HLPS3	Type 3 Hotelling-Lawley-Pillai-Samson tests of fixed effects	REPEATED / HLPS TYPE=UN
Influence	influence diagnostics	MODEL / INFLUENCE
InfoCrit	information criteria	PROC MIXED IC
InvCholG	inverse Cholesky root of the estimated **G** matrix	RANDOM / GCI
InvCholR	inverse Cholesky root of blocks of the estimated **R** matrix	REPEATED / RCI
InvCholV	inverse Cholesky root of blocks of the estimated **V** matrix	RANDOM / VCI
InvCovB	inverse of approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVBI
InvG	inverse of the estimated **G** matrix	RANDOM / GI
InvR	inverse of blocks of the estimated **R** matrix	REPEATED / RI
InvV	inverse of blocks of the estimated **V** matrix	RANDOM / VI
IterHistory	iteration history	default output
LComponents	single degree of freedom estimates corresponding to rows of the **L** matrix for fixed effects	MODEL / LCOMPONENTS
LRT	likelihood ratio test	default output
LSMeans	LS-means	LSMEANS
MMEq	mixed model equations	PROC MIXED MMEQ
MMEqSol	mixed model equations solution	PROC MIXED MMEQSOL
ModelInfo	model information	default output
NObs	number of observations read and used	default output
ParmSearch	parameter search values	PARMS

Table 46.8. (continued)

Table Name	Description	Required Statement / Option
Posterior	posterior sampling information	PRIOR
R	blocks of the estimated $\mathbf{R}$ matrix	REPEATED / R
RCorr	correlation matrix from blocks of the estimated $\mathbf{R}$ matrix	REPEATED / RCORR
Search	posterior density search table	PRIOR / PSEARCH
Slices	tests of LS-means slices	LSMEANS / SLICE=
SolutionF	fixed effects solution vector	MODEL / S
SolutionR	random effects solution vector	RANDOM / S
Tests1	Type 1 tests of fixed effects	MODEL / HTYPE=1
Tests2	Type 2 tests of fixed effects	MODEL / HTYPE=2
Tests3	Type 3 tests of fixed effects	default output
Type1	Type 1 analysis of variance	PROC MIXED METHOD=TYPE1
Type2	Type 2 analysis of variance	PROC MIXED METHOD=TYPE2
Type3	Type 3 analysis of variance	PROC MIXED METHOD=TYPE3
Trans	transformation of covariance parameters	PRIOR / PTRANS
V	blocks of the estimated $\mathbf{V}$ matrix	RANDOM / V
VCorr	correlation matrix from blocks of the estimated $\mathbf{V}$ matrix	RANDOM / VCORR

In Table 46.8 , “Coef” refers to multiple tables produced by the E, E1, E2, or E3 options in the MODEL statement and the E option in the CONTRAST, ESTIMATE, and LSMEANS statements. You can create one large data set of these tables with a statement similar to

```
ods output Coef=c;
```

To create separate data sets, use

```
ods output Coef(match_all)=c;
```

Here the resulting data sets are named C, C1, C2, etc. The same principles apply to data sets created from the “R,” “CholR,” “InvCholR,” “RCorr,” “InvR,” “V,” “CholV,” “InvCholV,” “VCorr,” and “InvV” tables.

In Table 46.8 , the following changes have occurred from Version 6. The “Predicted,” “PredMeans,” and “Sample” tables from Version 6 no longer exist and have been replaced by output data sets; see descriptions of the MODEL statement options OUTPRED= on page 2703 and OUTPREDM= on page 2704 and the PRIOR statement option OUT= on page 2711 for more details. The “ML” and “REML” tables from Version 6 have been replaced by the “IterHistory” table. The “Tests,” “HLM,” and “HLPS” tables from Version 6 have been renamed “Tests3,” “HLM3,” and “HLPS3.”

Table 46.9 lists the variable names associated with the data sets created when you use the ODS OUTPUT option in conjunction with the preceding tables. In Table 46.9 , n is used to denote a generic number that is dependent upon the particular data set and model you select, and it can assume a different value each time it is used (even within the same table). The phrase *model specific* appears in rows of the affected tables to indicate that columns in these tables depend upon the variables you specify in the model.

Caution: There exists a danger of name collisions with the variables in the *model specific* tables in Table 46.9 and variables in your input data set. You should avoid using input variables with the same names as the variables in these tables.

Table 46.9. Variable Names for the ODS Tables Produced in PROC MIXED

Table Name	Variables
AsyCorr	Row, CovParm, CovP1–CovPn
AsyCov	Row, CovParm, CovP1–CovPn
BaseDen	Type, Parm1–Parm$n$
Bound	Technique, Converge, Iterations, Evaluations, LogBound, CovP1–CovPn, TCovP1–TCovPn
CholG	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
CholR	Index, Row, Col1–Col$n$
CholV	Index, Row, Col1–Col$n$
ClassLevels	Class, Levels, Values
Coef	*model specific*, LMatrix, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row1–Row$n$
Contrasts	Label, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
CorrB	*model specific*, Effect, Row, Col1–Col$n$
CovB	*model specific*, Effect, Row, Col1–Col$n$
CovParms	CovParm, Subject, Group, Estimate, StandardError, ZValue, ProbZ, Alpha, Lower, Upper
Diffs	*model specific*, Effect, Margins, ByLevel, AT variables, Diff, StandardError, DF, tValue, Tails, Probt, Adjustment, Adjp, Alpha, Lower, Upper, AdjLow, AdjUpp
Dimensions	Descr, Value
Estimates	Label, Estimate, StandardError, DF, tValue, Tails, Probt, Alpha, Lower, Upper
FitStatistics	Descr, Value
G	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
GCorr	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
HLM1	Effect, NumDF, DenDF, FValue, ProbF
HLM2	Effect, NumDF, DenDF, FValue, ProbF
HLM3	Effect, NumDF, DenDF, FValue, ProbF
HLPS1	Effect, NumDF, DenDF, FValue, ProbF
HLPS2	Effect, NumDF, DenDF, FValue, ProbF
HLPS3	Effect, NumDF, DenDF, FValue, ProbF

Table 46.9. (continued)

Table Name	**Variables**
Influence	*dependent on option modifiers*, Effect, Tuple, Obs1–Obs$k$, Level, Iter, Index, Predicted, Residual, Leverage, PressRes, PRESS, Student, RMSE, RStudent, CookD, DFFITS, MDFFITS, CovRatio, CovTrace, CookDCP, MDFFITSCP, CovRatioCP, CovTraceCP, LD, RLD, Parm1–Parm$p$, CovP1–CovPq, Notes
InfoCrit	Neg2LogLike, Parms, AIC, AICC, HQIC, BIC, CAIC
InvCholG	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
InvCholR	Index, Row, Col1–Col$n$
InvCholV	Index, Row, Col1–Col$n$
InvCovB	*model specific*, Effect, Row, Col1–Col$n$
InvG	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
InvR	Index, Row, Col1–Col$n$
InvV	Index, Row, Col1–Col$n$
IterHistory	CovP1–CovPn, Iteration, Evaluations, M2ResLogLike, M2LogLike, Criterion
LComponents	Effect, TestType, LIndex, Estimate, StdErr, DF, tValue, Probt
LRT	DF, ChiSquare, ProbChiSq
LSMeans	*model specific*, Effect, Margins, ByLevel, AT variables, Estimate, StandardError, DF, tValue, Probt, Alpha, Lower, Upper, Cov1–Cov$n$, Corr1–Corr$n$
MMEq	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
MMEqSol	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Row, Col1–Col$n$
ModelInfo	Descr, Value
Nobs	Label, N, NObsRead, NObsUsed, SumFreqsRead, SumFreqsUsed
ParmSearch	CovP1–CovPn, Var, ResLogLike, M2ResLogLike2, LogLike, M2LogLike, LogDetH
Posterior	Descr, Value
R	Index, Row, Col1–Col$n$
RCorr	Index, Row, Col1–Col$n$
Search	Parm, TCovP1–TCovPn, Posterior
Slices	*model specific*, Effect, Margins, ByLevel, AT variables, NumDF, DenDF, FValue, ProbF
SolutionF	*model specific*, Effect, Estimate, StandardError, DF, tValue, Probt, Alpha, Lower, Upper
SolutionR	*model specific*, Effect, Subject, Sub1–Sub$n$, Group, Group1–Group$n$, Estimate, StdErrPred, DF, tValue, Probt, Alpha, Lower, Upper
Tests1	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Tests2	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Tests3	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Type1	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF

Table 46.9. (continued)

Table Name	Variables
Type2	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF
Type3	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF
Trans	Prior, TCovP, CovP1–CovPn
V	Index, Row, Col1–Col$n$
VCorr	Index, Row, Col1–Col$n$

Some of the variables listed in Table 46.9 are created only when you have specified certain options in the relevant PROC MIXED statements.

Converting from Previous Releases

The following changes have occurred in variables listed in Table 46.9 from Version 6. Nearly all underscores have been removed from variable names in order to be compatible and consistent with other procedures. Some of the variable names have been changed (for example, T has been changed to tValue and PT to Probt) for the same reason. You may have to modify some of your Version 6 code to accommodate these changes.

In Version 6, PROC MIXED used a MAKE statement to save displayed output as data sets. The MAKE statement is now obsolete and may not be supported in future releases. Use the ODS OUTPUT statement instead. The following table shows typical conversions in order to replace the MAKE statement in Version 6 code with ODS statements.

Table 46.10. ODS Conversions for PROC MIXED

Version 6 Syntax	Versions 7 and 8 Syntax
`make ’covparms’ out=cp;`	`ods output covparms=cp;`

`make ’covparms’ out=cp noprint;`	`ods listing exclude covparms;`
	`ods output covparms=cp;`

`%global _print_; %let _print_=off;`	`ods listing close;`
`%global _print_; %let _print_=on;`	`ods listing;`

ODS Graphics (Experimental)

This section describes the use of ODS for creating diagnostic plots with the MIXED procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement. In addition you must specify the relevant options of the PROC MIXED or MODEL statement (Table 46.11). To request plots of studentized residuals, for example, specify the experimental RESIDUAL option of the MODEL statement. To obtain graphical displays of leave-one-out parameter estimates, specify the experimental INFLUENCE option with the ESTIMATES suboption.

For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.” ODS names of the various graphics are given in the “ODS Graph Names” section (page 2762).

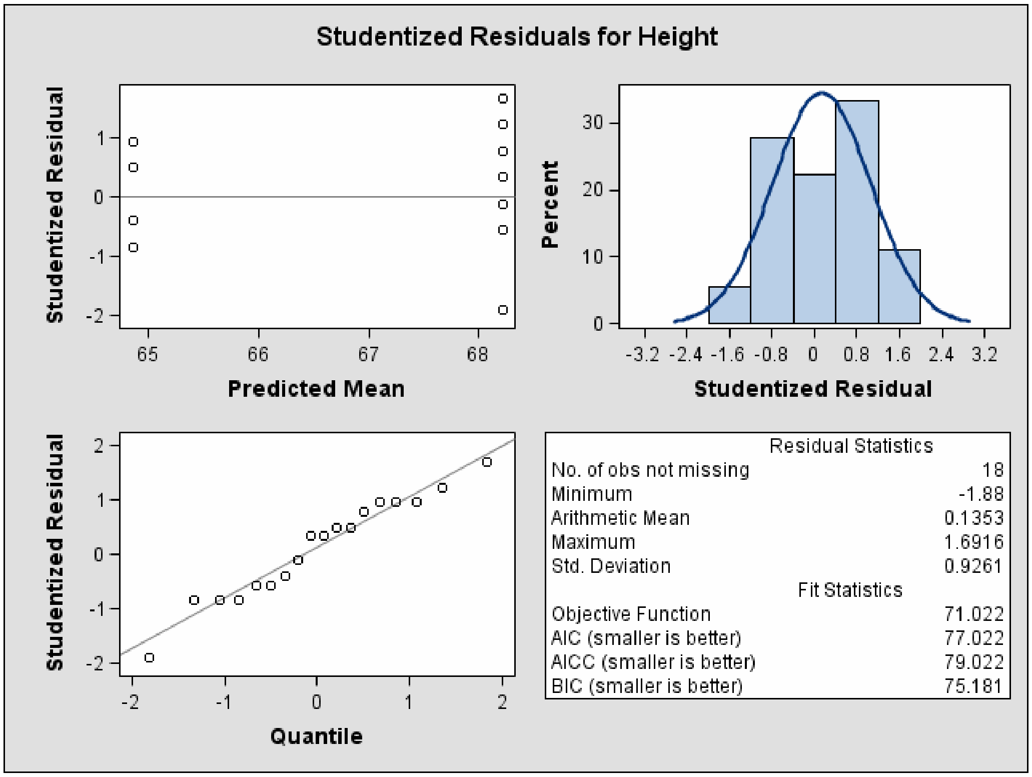
Residual Plots

With the experimental graphics features the MIXED procedure can generate panels of residual diagnostics. Each panel consists of a plot of residuals versus predicted values, a histogram with Normal density overlaid, a Q-Q plot, and summary residual and fit statistics (Figure 46.15). The plots are produced even if the OUTP= and OUTPM= options of the MODEL statement are not specified. Three such panels are produced for the marginal residuals which would be added to the OUTPM= data set. The panels display the raw, studentized, and Pearson residuals (see “Residual Diagnostics”). In models with RANDOM effects where EBLUPs can be used for prediction, the raw, studentized and Pearson conditional residuals are also plotted.

Recall the example in the “Getting Started” section on page 2665. The following statements generate six 2×2 panels of residual graphs.

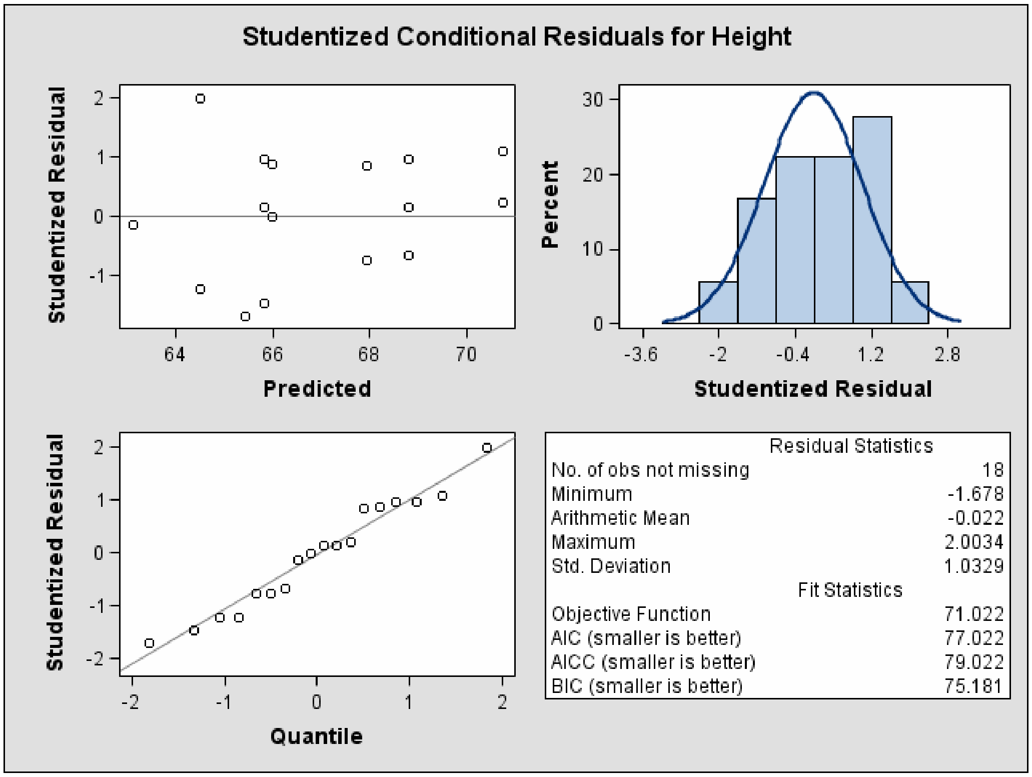
```
ods html;
ods graphics on;

proc mixed;
   class Family Gender;
   model Height = Gender / residual;
   random Family Family*Gender;
run;

ods graphics off;
ods html close;
```

The graphical displays are requested by specifying the experimental ODS GRAPHICS statement. The panel for the marginal studentized residuals is shown in Figure 46.15 and the panel for the conditional studentized residuals in Figure 46.16.



Residual Statistics	
No. of obs not missing	18
Minimum	-1.88
Arithmetic Mean	0.1353
Maximum	1.6916
Std. Deviation	0.9261
Fit Statistics	
Objective Function	71.022
AIC (smaller is better)	77.022
AICC (smaller is better)	79.022
BIC (smaller is better)	75.181

Figure 46.15. Marginal Studentized Residual Panel (Experimental)



Residual Statistics	
No. of obs not missing	18
Minimum	-1.678
Arithmetic Mean	-0.022
Maximum	2.0034
Std. Deviation	1.0329
Fit Statistics	
Objective Function	71.022
AIC (smaller is better)	77.022
AICC (smaller is better)	79.022
BIC (smaller is better)	75.181

Figure 46.16. Conditional Studentized Residual Panel (Experimental)

A similar panel display for the scaled residuals is produced when you specify the experimental VCIRY option of the MODEL statement; see option VCIRY on page 2705 for more details.

The “Residual Statistics” in the lower right-hand corner inset provides descriptive statistics for the set of residuals that is displayed. Note that residuals in a mixed model do not necessarily sum to zero, even if the model contains an intercept.

Influence Plots

With the experimental graphics features the MIXED procedure can generate one or more panels of influence graphics. The type and number of panels produced depends on the modifiers of the INFLUENCE option. Plots related to covariance parameters are produced when diagnostics are computed by iterative methods (ITER=). The estimates of the fixed effects—and covariance parameters when updates are iterative—are plotted when the ESTIMATES modifier is specified.

The two types of panel graphs produced by the INFLUENCE option are shown in Figure 46.17 and Figure 46.18. The diagnostics panel shows an overall influence statistic (likelihood distance) and diagnostics for the fixed effects (CookD and COVRATIO). The statistics produced depend on suboptions of the INFLUENCE option (see Example 46.8 for the statements and options that produced Figure 46.17). Reference lines are drawn at zero for PRESS residuals and COVTRACE, and at one for COVRATIO. A reference line for likelihood distances is drawn at the 75th percentile of a chi-square distribution with m degrees of freedom if the largest displacement value in the “Influence” table is close to or larger than that percentile. The number m equals the number of parameters being updated.

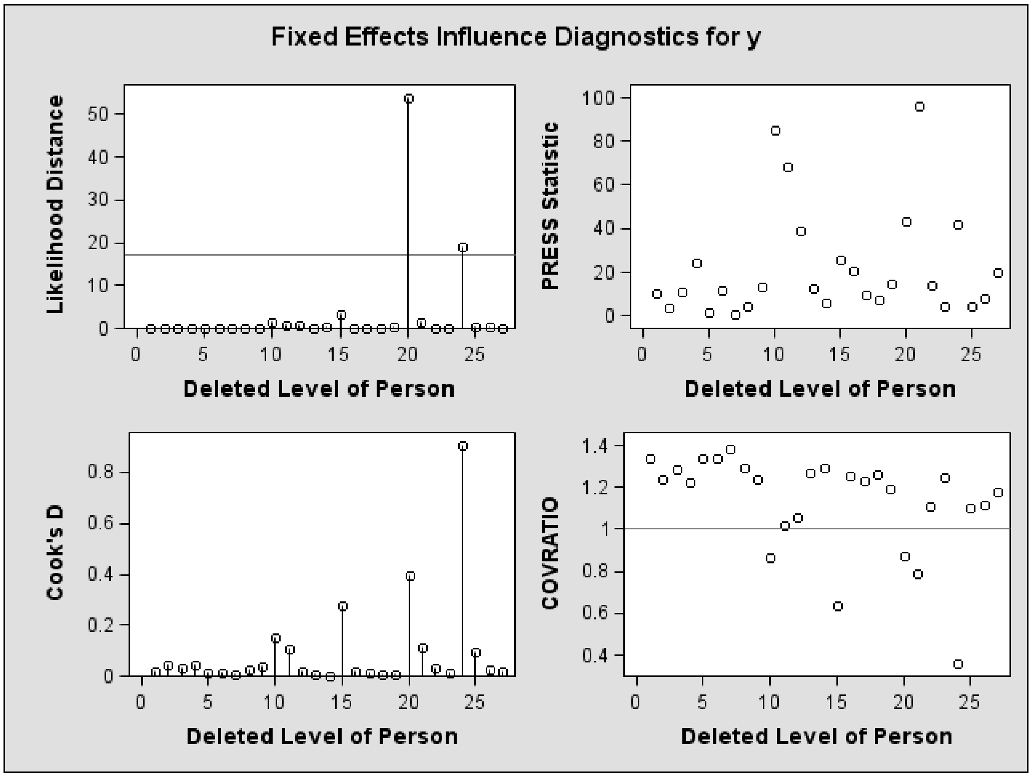


Figure 46.17. Influence Diagnostics (Experimental)

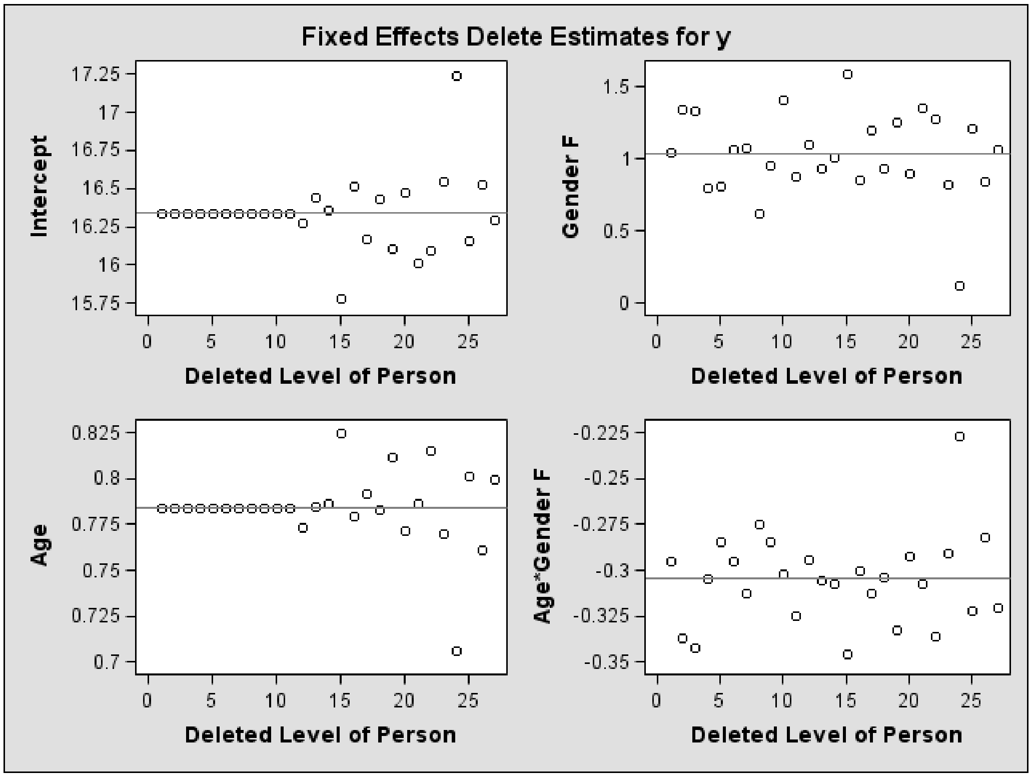


Figure 46.18. Delete Estimates (Experimental)

The second type of influence panel plot is obtained when you specify the ESTIMATES suboption (Figure 46.18). It shows the delete-estimates for each updated model parameter. Reference lines are drawn at the full data estimates. For noniterative influence analyses with profiled residual variance, the delete-case root mean square error is also plotted.

For the SAS statements that produce influence plots and for variations of these graphs see Example 46.7 and Example 46.8.

Box Plots

You can specify the BOXPLOT option in the PROC MIXED statement.

BOXPLOT <*(suboptions)*>

requests box plots of observed and residual values $\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$ for effects that consist of single CLASS variables. This includes SUBJECT= and GROUP= effects.

For models with a RANDOM statement you also obtain box plots of the conditional residuals $\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{Z}\widehat{\boldsymbol{\gamma}}$. The box plots are constructed from studentized residuals when the RESIDUAL option of the MODEL statement is specified.

The following suboptions modify the appearance of the plots:

DATALABEL | NODATALABEL

determines whether to place observation labels next to far outliers. Far outliers are labeled by default.

FILL | NOFILL

determines whether the boxes are filled. The default is FILL.

NPANEL=*n*

limits the number of boxes per plot. The default is to place box plots for all levels of a factor in a common panel.

See Example 46.8 for an application.

ODS Graph Names

The MIXED procedure assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 46.11.

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 46.11. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Table 46.11. ODS Graphics Produced by PROC MIXED

ODS Graph Name	Plot Description	Option
BoxPlot1,2,...	Box plots	BOXPLOT
InfluenceEstCovPPanel1,2,...	Covariance parameter delete estimates	MODEL / INFLUENCE(EST ITER=*n*)
InfluenceEstParmPanel1,2,...	Fixed effects delete estimates	MODEL / INFLUENCE(EST)
InfluenceStatCovPPanel	Diagnostics for covariance parameters	MODEL / INFLUENCE(ITER=*n*)
InfluenceStatParmPanel	Diagnostics for overall influence and fixed effects	MODEL / INFLUENCE
PearsonCondResidualPanel	Pearson conditional residuals	MODEL / RESIDUAL
PearsonResidualPanel	Pearson marginal residuals	MODEL / RESIDUAL
RawCondResidualPanel	Conditional residuals	MODEL / RESIDUAL
RawResidualPanel	Marginal residuals	MODEL / RESIDUAL
ScaledResidualPanel	Scaled residuals	MODEL / VCIRY
StudentizedCondResidualPanel	Studentized conditional residuals	MODEL / RESIDUAL
StudentizedResidualPanel	Studentized marginal residuals	MODEL / RESIDUAL

Residuals and Influence Diagnostics (Experimental)

Residual Diagnostics

Consider a residual vector of the form $\widetilde{\mathbf{e}} = \mathbf{PY}$, where $\mathbf{P}$ is a projection matrix, possibly an oblique projector. A typical element $\widetilde{e}_i$ with variance v_i and estimated variance $\widehat{v}_i$ is said to be *standardized* as

$$\frac{\widetilde{e}_i}{\sqrt{\mathrm{var}[\widetilde{e}_i]}} = \frac{\widetilde{e}_i}{\sqrt{v_i}}$$

and *studentized* as

$$\frac{\widetilde{e}_i}{\sqrt{\widehat{v}_i}}$$

External studentization uses an estimate of $\mathrm{var}[\widetilde{e}_i]$ which does not involve the ith observation. Externally studentized residuals are often preferred over studentized residuals because they have well-known distributional properties in standard linear models for independent data.

Residuals that are scaled by the estimated variance of the response, i.e., $\widetilde{e}_i/\sqrt{\widehat{\mathrm{var}}[Y_i]}$, are referred to as Pearson-type residuals.

Marginal and Conditional Residuals

The marginal and conditional means in the linear mixed model are $\mathrm{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$ and $\mathrm{E}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$, respectively. Accordingly, the vector $\mathbf{r}_m$ of marginal residuals is defined as

$$\mathbf{r}_m = \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$$

and the vector $\mathbf{r}_c$ of conditional residuals is

$$\mathbf{r}_c = \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{Z}\widehat{\boldsymbol{\gamma}} = \mathbf{r}_m - \mathbf{Z}\widehat{\boldsymbol{\gamma}}$$

Following Gregoire, Schabenberger, and Barrett (1995), let $\mathbf{Q} = \mathbf{X}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{X}'$ and $\mathbf{K} = \mathbf{I} - \mathbf{Z}\widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}$. Then

$$\begin{aligned} \widehat{\mathrm{var}}[\mathbf{r}_m] &= \widehat{\mathbf{V}} - \mathbf{Q} \\ \widehat{\mathrm{var}}[\mathbf{r}_c] &= \mathbf{K}(\widehat{\mathbf{V}} - \mathbf{Q})\mathbf{K}' \end{aligned}$$

For an individual observation the raw, studentized, and Pearson residuals computed by the RESIDUAL option of the MODEL statement are given in the following table.

Type of Residual	Marginal	Conditional
Raw	$r_{mi} = Y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$	$r_{ci} = r_{mi} - \mathbf{z}_i'\widehat{\boldsymbol{\gamma}}$
Studentized	$r_{mi}^{student} = \frac{r_{mi}}{\sqrt{\widehat{\mathrm{var}}[r_{mi}]}}$	$r_{ci}^{student} = \frac{r_{ci}}{\sqrt{\widehat{\mathrm{var}}[r_{ci}]}}$
Pearson	$r_{mi}^{pearson} = \frac{r_{mi}}{\sqrt{\widehat{\mathrm{var}}[Y_i]}}$	$r_{ci}^{pearson} = \frac{r_{ci}}{\sqrt{\widehat{\mathrm{var}}[Y_i\|\gamma]}}$

When the OUTPM= option of the MODEL statement is specified in addition to the RESIDUAL option, r_{mi}, $r_{mi}^{student}$, and $r_{mi}^{pearson}$ are added to the data set as variables Resid, StudentResid, and PearsonResid, respectively. When the OUTP= option is specified, r_{ci}, $r_{ci}^{student}$, and $r_{ci}^{pearson}$ are added to the data set.

Scaled Residuals

For correlated data, a set of scaled quantities can be defined through the Cholesky decomposition of the variance-covariance matrix. Since fitted residuals in linear models are rank-deficient, it is customary to draw on the variance-covariance matrix of the data. If $\mathrm{var}[\mathbf{Y}] = \mathbf{V}$ and $\mathbf{C}'\mathbf{C} = \mathbf{V}$, then $\mathbf{C}'^{-1}\mathbf{Y}$ has uniform dispersion and its elements are uncorrelated.

Scaled residuals in a mixed model are meaningful for quantities based on the marginal distribution of the data. Let $\widehat{\mathbf{C}}$ denote the Cholesky root of $\widehat{\mathbf{V}}$, so that $\widehat{\mathbf{C}}'\widehat{\mathbf{C}} = \widehat{\mathbf{V}}$, and define

$$\begin{aligned} \mathbf{Y}_c &= \widehat{\mathbf{C}}'^{-1}\mathbf{Y} \\ \mathbf{r}_{m(c)} &= \widehat{\mathbf{C}}'^{-1}\mathbf{r}_m \end{aligned}$$

By analogy with other scalings, the inverse Cholesky decomposition can also be applied to the residual vector, $\widehat{\mathbf{C}}'^{-1}\mathbf{r}_m$, although $\mathbf{V}$ is not the variance-covariance matrix of $\mathbf{r}_m$.

To diagnose whether the covariance structure of the model has been specified correctly can be difficult based on $\mathbf{Y}_c$, since the inverse Cholesky transformation affects the expected value of $\mathbf{Y}_c$. You can draw on $\mathbf{r}_{m(c)}$ as a vector of (approximately) uncorrelated data with constant mean.

When the OUTPM= option of the MODEL statement is specified in addition to the VCIRY option, $\mathbf{Y}_c$ is added as variable ScaledDep and $\mathbf{r}_{m(c)}$ is added as ScaledResid to the data set.

Influence Diagnostics

The Basic Idea and Statistics

The general idea of quantifying the influence of one or more observations relies on computing parameter estimates based on all data points, removing the cases in question from the data, refitting the model, and computing statistics based on the change between full-data and reduced-data estimation. Influence statistics can be coarsely grouped by the aspect of estimation which is their primary target:

- overall measures compare changes in objective functions: (restricted) likelihood distance (Cook and Weisberg 1982, Ch. 5.2)
- influence on parameter estimates: Cook's D (Cook 1977, 1979), MDFFITS (Belsley, Kuh, and Welsch 1980, p. 32)
- influence on precision of estimates: CovRatio and CovTrace
- influence on fitted and predicted values: PRESS residual, PRESS statistic (Allen 1974), DFFITS (Belsley, Kuh, and Welsch 1980, p. 15)
- outlier properties: Internally and externally studentized residuals, leverage

For linear models for uncorrelated data, it is not necessary to refit the model after removing a data point in order to measure the impact of an observation on the model. The change in fixed effect estimates, residuals, residual sums of squares, and the variance-covariance matrix of the fixed effects can be computed based on the fit to the full data alone. By contrast, in mixed models several important complications arise. Data points can impact not only the fixed effects but also the covariance parameter estimates on which the fixed effects estimates depend. Furthermore, closed-form expressions for computing the change in important model quantities may not be available.

This section provides background material for the various influence diagnostics available with the MIXED procedure. See the section "Mixed Models Theory" beginning on page 2731 for relevant expressions and definitions. The parameter vector $\boldsymbol{\theta}$ denotes all unknown parameters in the **R** and **G** matrix.

The observations whose influence is being ascertained are represented by the set U and referred to simply as "the observations in U." The estimate of a parameter vector, for example, $\boldsymbol{\beta}$, obtained from all observations except those in the set U is denoted $\widehat{\boldsymbol{\beta}}_{(U)}$. In case of a matrix $\mathbf{A}$, the notation $\mathbf{A}_{(U)}$ represents the matrix with the rows in U removed; these rows are collected in $\mathbf{A}_U$. If $\mathbf{A}$ is symmetric, then notation $\mathbf{A}_{(U)}$ implies removal of rows and columns. The vector Y_U comprises the responses of the data points being removed, and $\mathbf{V}_{(U)}$ is the variance-covariance matrix of the remaining observations. When $k = 1$, lowercase notation emphasizes that single points are removed, e.g., $\mathbf{A}_{(u)}$.

Managing the Covariance Parameters

An important component of influence diagnostics in the mixed model is the estimated variance-covariance matrix $\mathbf{V} = \mathbf{ZGZ}' + \mathbf{R}$. To make the dependence on the vector of covariance parameters explicit, write it as $\mathbf{V}(\boldsymbol{\theta})$. If one parameter, σ^2, is profiled or factored out of $\mathbf{V}$, the remaining parameters are denoted as $\boldsymbol{\theta}^*$. Notice that in a model where $\mathbf{G}$ is diagonal and $\mathbf{R} = \sigma^2\mathbf{I}$, the parameter vector $\boldsymbol{\theta}^*$ contains the ratios of each variance component and σ^2 (see Wolfinger, Tobias, and Sall 1994). When ITER=0, two scenarios are distinguished:

1. If the residual variance is not profiled, either because the model does not contain a residual variance or because it is part of the Newton-Raphson iterations, then $\widehat{\boldsymbol{\theta}}_{(U)} \equiv \widehat{\boldsymbol{\theta}}$.
2. If the residual variance is profiled then $\widehat{\boldsymbol{\theta}}^*_{(U)} \equiv \widehat{\boldsymbol{\theta}}^*$ and $\widehat{\sigma}^2_{(U)} \neq \widehat{\sigma}^2$. Influence statistics such as Cook's D and internally studentized residuals are based on $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ whereas externally studentized residuals and the DFFITS statistic are based on $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U) = \sigma^2_{(U)}\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$. In a random components model with uncorrelated errors, for example, the computation of $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$ involves scaling of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$ by the full data estimate $\widehat{\sigma}^2$ and multiplying the result with the reduced-data estimate $\widehat{\sigma}^2_{(U)}$.

Certain statistics, such as MDFFITS, COVRATIO, and COVTRACE, require an estimate of the variance of the fixed effects that is based on the reduced number of observations. For example, $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$ is evaluated at the reduced-data parameter estimates but computed for the entire data set. The matrix $\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}_{(U)})$, on the other hand, has rows and columns corresponding to the points in U removed. The resulting matrix is evaluated at the delete-case estimates.

When influence analysis is iterative, the entire vector $\boldsymbol{\theta}$ is updated, whether the residual variance is profiled or not. The matrices to be distinguished here are $\mathbf{V}(\widehat{\boldsymbol{\theta}})$, $\mathbf{V}(\widehat{\boldsymbol{\theta}}_{(U)})$, and $\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}_{(U)})$, with unambiguous notation.

Predicted Values, PRESS Residual, and PRESS Statistic

An unconditional predicted value is $\widehat{y}_i = \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$, where the vector $\mathbf{x}_i$ is the ith row of $\mathbf{X}$. The (raw) residual is given as $\widehat{\epsilon}_i = y_i - \widehat{y}_i$ and the PRESS *residual* is

$$\widehat{\epsilon}_{i(U)} = y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}_{(U)}$$

The PRESS *statistic* is the sum of the squared PRESS residuals,

$$PRESS = \sum_{i \in U} \widehat{\epsilon}^2_{i(U)}$$

where the sum is over the observations in U.

If EFFECT=, SIZE=, or KEEP= are not specified, PROC MIXED computes the PRESS residual for each observation selected through SELECT= (or all observations if SELECT= is not given). If EFFECT=, SIZE=, or KEEP= are specified, the procedure computes $PRESS$.

Leverage

For the general mixed model, leverage can be defined through the projection matrix that results from a transformation of the model with the inverse of the Cholesky decomposition of $\mathbf{V}$, or through an oblique projector. The MIXED procedure follows the latter path in the computation of influence diagnostics. The leverage value reported for the ith observation is the ith diagonal entry of the matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}$$

which is the weight of the observation in contributing to its own predicted value, $\mathbf{H} = d\widehat{\mathbf{Y}}/d\mathbf{Y}$.

While $\mathbf{H}$ is idempotent, it is generally not symmetric and thus not a projection matrix in the narrow sense.

The properties of these leverages are generalizations of the properties in models with diagonal variance-covariance matrices. For example, $\widehat{\mathbf{Y}} = \mathbf{HY}$, and in a model with intercept and $\mathbf{V} = \sigma^2\mathbf{I}$, the leverage values

$$h_{ii} = \mathbf{x}_i'(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i$$

are $h^l_{ii} = 1/n \le h_{ii} \le 1 = h^u_{ii}$ and $\sum_{i=1}^n h_{ii} = \text{rank}(\mathbf{X})$. The lower bound for h_{ii} is achieved in an intercept-only model, the upper bound in a saturated model. The trace of $\mathbf{H}$ equals the rank of $\mathbf{X}$.

If ν_{ij} denotes the element in row i, column j of $\mathbf{V}^{-1}$, then for a model containing only an intercept the diagonal elements of $\mathbf{H}$ are

$$h_{ii} = \frac{\sum_{j=1}^n \nu_{ij}}{\sum_{i=1}^n \sum_{j=1}^n \nu_{ij}}$$

Because $\sum_{j=1}^n \nu_{ij}$ is a sum of elements in the ith row of the *inverse* variance-covariance matrix h_{ii} can be negative, even if the correlations among data points are nonnegative. In case of a saturated model with $\mathbf{X} = \mathbf{I}$, $h_{ii} = 1.0$.

Internally and Externally Studentized Residuals

See the section "Residual Diagnostics" on page 2763 for the distinction between standardization, studentization, and scaling of residuals. Internally studentized marginal and conditional residuals are computed with the RESIDUAL option of the MIXED procedure (p. 2763). The INFLUENCE option computes internally and externally studentized marginal residuals.

The computation of internally studentized residuals relies on the diagonal entries of $\mathbf{V}(\widehat{\boldsymbol{\theta}}) - \mathbf{Q}(\widehat{\boldsymbol{\theta}})$ where $\mathbf{Q}(\widehat{\boldsymbol{\theta}}) = \mathbf{X}(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}\mathbf{X}'$. Externally studentized residuals require iterative influence analysis or a profiled residual variance. In the former case the studentization is based on $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$; in the latter case it is based on $\sigma^2_{(U)}\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$.

Cook's D

Cook's D statistic is an invariant norm that measures the influence of observations in U on a vector of parameter estimates (Cook 1977). In case of the fixed effects coefficients, let

$$\boldsymbol{\delta}_{(U)} = \widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_{(U)}$$

Then the MIXED procedure computes

$$D(\boldsymbol{\beta}) = \boldsymbol{\delta}'_{(U)}\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}]^{-}\boldsymbol{\delta}_{(U)}/\text{rank}(\mathbf{X})$$

where $\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}]^{-}$ is the matrix that results from sweeping $(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}$.

If $\mathbf{V}$ is known, Cook's D can be calibrated according to a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{X}$ (Christensen, Pearson, and Johnson 1992). For estimated $\mathbf{V}$ the calibration can be carried out according to an $F(\text{rank}(\mathbf{X}), n - \text{rank}(\mathbf{X}))$ distribution. To interpret D on a familiar scale, Cook (1979) and Cook and Weisberg (1982, p. 116) refer to the 50th percentile of the reference distribution. If D is equal to that percentile, then removing the points in U moves the fixed effects coefficient vector from the center of the confidence region to the 50% confidence ellipsoid (Myers 1990, p. 262).

In the case of iterative influence analysis, the MIXED procedure also computes a D-type statistic for the covariance parameters. If $\boldsymbol{\Gamma}$ is the asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\theta}}$, then MIXED computes

$$D_{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U))})'\widehat{\boldsymbol{\Gamma}}^{-1}(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})$$

DFFITS and MDFFITS

A DFFIT measures the change in predicted values due to removal of data points. If this change is standardized by the externally estimated standard error of the predicted value in the full data, the DFFITS statistic of Belsley, Kuh, and Welsch (1980, p. 15) results:

$$\text{DFFITS}_i = (\widehat{y}_i - \widehat{y}_{i(u)})/\text{ese}(\widehat{y}_i)$$

The MIXED procedure computes DFFITS when the EFFECT= or SIZE= modifiers of the INFLUENCE option are not in effect. In general, an external estimate of the estimated standard error is used. When ITER > 0, the estimate is

$$\text{ese}(\widehat{y}_i) = \sqrt{\mathbf{x}_i'(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}}_{(u)})^{-}\mathbf{X})^{-1}\mathbf{x}_i}$$

When ITER=0 and σ^2 is profiled, then

$$\text{ese}(\widehat{y}_i) = \widehat{\sigma}_{(u)}\sqrt{\mathbf{x}_i'(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)^{-1}\mathbf{X})^{-}\mathbf{x}_i}$$

When the EFFECT=, SIZE=, or KEEP= modifiers are specified, the MIXED procedure computes a multivariate version suitable for the deletion of multiple data points. The statistic, termed MDFFITS after the MDFFIT statistic of Belsley, Kuh, and Welsch (1980, p. 32), is closely related to Cook's D. Consider the case $\mathbf{V} = \sigma^2\mathbf{V}(\boldsymbol{\theta}^*)$ so that

$$\text{var}[\widehat{\boldsymbol{\beta}}] = \sigma^2(\mathbf{X}'\mathbf{V}(\boldsymbol{\theta}^*)^{-1}\mathbf{X})^{-}$$

and let $\widetilde{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}]$ be an estimate of $\text{var}[\widehat{\boldsymbol{\beta}}_{(U)}]$ that does not use the observations in U. The MDFFITS statistic is then computed as

$$\text{MDFFITS}(\boldsymbol{\beta}) = \boldsymbol{\delta}_{(U)}'\widetilde{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}]^{-}\boldsymbol{\delta}_{(U)}/\text{rank}(\mathbf{X})$$

If ITER=0 and σ^2 is profiled, then $\widetilde{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}]^{-}$ is obtained by sweeping

$$\widehat{\sigma}^2_{(U)}(\mathbf{X}_{(U)}'\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}^*)^{-}\mathbf{X}_{(U)})^{-}$$

The underlying idea is that if $\boldsymbol{\theta}^*$ were known, then

$$(\mathbf{X}_{(U)}'\mathbf{V}_{(U)}(\boldsymbol{\theta}^*)^{-1}\mathbf{X}_{(U)})^{-}$$

would be $\text{var}[\widehat{\boldsymbol{\beta}}]/\sigma^2$ in a generalized least squares regression with all but the data in U.

In the case of iterative influence analysis, $\widetilde{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}]$ is evaluated at $\widehat{\boldsymbol{\theta}}_{(U)}$. Furthermore, a MDFFITS-type statistic is then computed for the covariance parameters:

$$\text{MDFFITS}(\boldsymbol{\theta}) = (\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})'\widehat{\text{var}}[\widehat{\boldsymbol{\theta}}_{(U)}]^{-1}(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})$$

Covariance Ratio and Trace

These statistics depend on the availability of an external estimate of $\mathbf{V}$, or at least of σ^2. Whereas Cook's D and MDFFITS measure the impact of data points on a vector of parameter estimates, the covariance-based statistics measure impact on their precision. Following Christensen, Pearson, and Johnson (1992), the MIXED procedure computes

$$\begin{aligned}\text{COVTRACE}(\boldsymbol{\beta}) &= |\text{trace}(\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}]^{-}\ \widehat{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}]) - \text{rank}(\mathbf{X})| \\ \text{COVRATIO}(\boldsymbol{\beta}) &= \frac{\det_{ns}(\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}_{(U)}])}{\det_{ns}(\widehat{\text{var}}[\widehat{\boldsymbol{\beta}}])}\end{aligned}$$

where $\det_{ns}(\mathbf{M})$ denotes the determinant of the nonsingular part of matrix $\mathbf{M}$.

In the case of iterative influence analysis these statistics are also computed for the covariance parameter estimates. If q denotes the rank of $\text{var}[\widehat{\boldsymbol{\theta}}]$, then

$$\begin{aligned}\text{COVTRACE}(\boldsymbol{\theta}) &= |\text{trace}(\widehat{\text{var}}[\widehat{\boldsymbol{\theta}}]^{-}\ \widehat{\text{var}}[\widehat{\boldsymbol{\theta}}_{(U)}]) - q| \\ \text{COVRATIO}(\boldsymbol{\theta}) &= \frac{\det_{ns}(\widehat{\text{var}}[\widehat{\boldsymbol{\theta}}_{(U)}])}{\det_{ns}(\widehat{\text{var}}[\widehat{\boldsymbol{\theta}}])}\end{aligned}$$

Likelihood Distances

The log-likelihood function l and restricted log-likelihood function l_R of the linear mixed model are given in the section "Estimating G and R in the Mixed Model" beginning on page 2737. Denote as $\boldsymbol{\psi}$ the collection of all parameters, i.e., the fixed effects $\boldsymbol{\beta}$ and the covariance parameters $\boldsymbol{\theta}$. Twice the difference between the (restricted) log-likelihood evaluated at the full-data estimates $\widehat{\boldsymbol{\psi}}$ and at the reduced-data estimates $\widehat{\boldsymbol{\psi}}_{(U)}$ is known as the (restricted) likelihood distance:

$$\begin{aligned}RLD_{(U)} &= 2\{l_R(\widehat{\boldsymbol{\psi}}) - l_R(\widehat{\boldsymbol{\psi}}_{(U)})\} \\ LD_{(U)} &= 2\{l(\widehat{\boldsymbol{\psi}}) - l(\widehat{\boldsymbol{\psi}}_{(U)})\}\end{aligned}$$

Cook and Weisberg (1982, Ch. 5.2) refer to these differences as *likelihood distances*, Beckman, Nachtsheim, and Cook (1987) term the measures *likelihood displacements*. If the number of elements in $\boldsymbol{\psi}$ that are subject to updating following point removal is q, then likelihood displacements can be compared against cutoffs from a chi-square distribution with q degrees of freedom. Notice that this reference distribution does not depend on the number of observations removed from the analysis, but rather on the number of model parameters that are updated. The likelihood displacement gives twice the amount by which the log-likelihood of the full data changes if one were to utilize an estimate based on fewer data points. It is thus a global, summary measure of the influence of the observations in U jointly on all parameters.

Unless METHOD=ML, the MIXED procedure computes the likelihood displacement based on the residual (=restricted) log likelihood, even if METHOD=MIVQUE0 or METHOD=TYPE1, TYPE2, or TYPE3.

Noniterative Update Formulas

Update formulas that do not require refitting of the model are available for the cases where $\mathbf{V} = \sigma^2\mathbf{I}$, $\mathbf{V}$ is known, or $\mathbf{V}^*$ is known. When ITER=0 and these update formulas can be invoked, the MIXED procedure uses the computational devices that are outlined in the following paragraphs. It is then assumed that the variance-covariance matrix of the fixed effects has the form $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^-$. When DDFM=KENWARDROGER, this is not the case; the estimated variance-covariance matrix is then inflated to better represent the uncertainty in the estimated covariance parameters. Influence statistics when DDFM=KENWARDROGER should iteratively update the covariance parameters (ITER > 0). The dependence of $\mathbf{V}$ on $\boldsymbol{\theta}$ is suppressed in the sequel for brevity.

Updating the Fixed Effects

Denote by $\mathbf{U}$ the $(n \times k)$ matrix that is assembled from k columns of the identity matrix. Each column of $\mathbf{U}$ corresponds to the removal of one data point. The point being targeted by the ith column of $\mathbf{U}$ corresponds to the row in which a 1 appears. Furthermore, define

$$\begin{aligned}
\boldsymbol{\Omega} &= (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^- \\
\mathbf{Q} &= \mathbf{X}\boldsymbol{\Omega}\mathbf{X}' \\
\mathbf{P} &= \mathbf{V}^{-1}(\mathbf{V} - \mathbf{Q})\mathbf{V}^{-1}
\end{aligned}$$

The change in the fixed effects estimates following removal of the observations in U is

$$\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_{(U)} = \boldsymbol{\Omega}\mathbf{X}'\mathbf{V}^{-1}\mathbf{U}(\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\mathbf{U}'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$$

Using results in Cook and Weisberg (1982, A2) you can further compute

$$\widetilde{\boldsymbol{\Omega}} = (\mathbf{X}'_{(U)}\mathbf{V}^{-1}_{(U)}\mathbf{X}_{(U)})^- = \boldsymbol{\Omega} + \boldsymbol{\Omega}\mathbf{X}'\mathbf{V}^{-1}\mathbf{U}(\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\mathbf{U}'\mathbf{V}^{-1}\mathbf{X}\boldsymbol{\Omega}$$

If $\mathbf{X}$ is $(n \times p)$ of rank $m < p$, then $\boldsymbol{\Omega}$ is deficient in rank and the MIXED procedure computes needed quantities in $\widetilde{\boldsymbol{\Omega}}$ by sweeping (Goodnight 1979). If the rank of the $(k \times k)$ matrix $\mathbf{U}'\mathbf{P}\mathbf{U}$ is less than k, the removal of the observations introduces a new singularity, whether $\mathbf{X}$ is of full rank or not. The solution vectors $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}_{(U)}$ then do not have the same expected values and should not be compared. When the MIXED procedure encounters this situation, influence diagnostics that depend on the choice of generalized inverse are not computed. The procedure also monitors the singularity criteria when sweeping the rows of $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^-$ and of $(\mathbf{X}'_{(U)}\mathbf{V}^{-1}_{(U)}\mathbf{X}_{(U)})^-$. If a new singularity is encountered or a former singularity disappears, no influence statistics are computed.

Residual Variance

When σ^2 is profiled out of the marginal variance-covariance matrix, a closed-form estimate of σ^2 that is based on only the remaining observations can be computed provided $\mathbf{V}^* = \mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$ is known. Hurtado (1993, Thm. 5.2) shows that

$$(n-q-r)\widehat{\sigma}^2_{(U)} = (n-q)\widehat{\sigma}^2 - \widehat{\boldsymbol{\epsilon}}_U'(\widehat{\sigma}^2\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\widehat{\boldsymbol{\epsilon}}_U$$

and $\widehat{\boldsymbol{\epsilon}}_U = \mathbf{U}'\mathbf{V}^{*-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}})$. In the case of maximum likelihood estimation $q=0$ and for REML estimation $q = \text{rank}(\mathbf{X})$. The constant r equals the rank of $(\mathbf{U}'\mathbf{P}\mathbf{U})$ for REML estimation and the number of effective observations that are removed if METHOD=ML.

Likelihood Distances

For noniterative methods the following computational devices are used to compute (restricted) likelihood distances provided that the residual variance σ^2 is profiled.

The log-likelihood function $l(\widehat{\boldsymbol{\theta}})$ evaluated at the full-data and reduced-data estimates can be written as

$$\begin{aligned} l(\widehat{\boldsymbol{\psi}}) &= -\frac{n}{2}\log(\widehat{\sigma}^2) - \frac{1}{2}\log|\mathbf{V}^*| - \frac{1}{2}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}})'\mathbf{V}^{*-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}})/\widehat{\sigma}^2 \\ &\quad -\frac{n}{2}\log(2\pi) \\ l(\widehat{\boldsymbol{\psi}}_{(U)}) &= -\frac{n}{2}\log(\widehat{\sigma}^2_{(U)}) - \frac{1}{2}\log|\mathbf{V}^*| \\ &\quad -\frac{1}{2}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})'\mathbf{V}^{*-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})/\widehat{\sigma}^2_{(U)} - \frac{n}{2}\log(2\pi) \end{aligned}$$

Notice that $l(\widehat{\boldsymbol{\theta}}_{(U)})$ evaluates the log-likelihood for n data points at the reduced-data estimates. It is not the log-likelihood obtained by fitting the model to the reduced data. The likelihood distance is then

$$LD_{(U)} = n\log\{\frac{\widehat{\sigma}^2_{(U)}}{\widehat{\sigma}^2}\} - n + (\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})'\mathbf{V}^{*-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})/\widehat{\sigma}^2_{(U)}$$

Expressions for $RLD_{(U)}$ in noniterative influence analysis are derived along the same lines.

Computational Issues

Computational Method

In addition to numerous matrix-multiplication routines, PROC MIXED frequently uses the sweep operator (Goodnight 1979) and the Cholesky root (Golub and Van Loan 1989). The routines perform a modified W transformation (Goodnight and Hemmerle 1979) for $\mathbf{G}$-side likelihood calculations and a direct method for $\mathbf{R}$-side likelihood calculations. For the Type III F-tests, PROC MIXED uses the algorithm described in Chapter 32, “The GLM Procedure.”

PROC MIXED uses a ridge-stabilized Newton-Raphson algorithm to optimize either a full (ML) or residual (REML) likelihood function. The Newton-Raphson algorithm is preferred to the EM algorithm (Lindstrom and Bates 1988). PROC MIXED profiles the likelihood with respect to the fixed effects and also with respect to the residual variance whenever it appears reasonable to do so. The residual profiling can be avoided by using the NOPROFILE option of the PROC MIXED statement. PROC MIXED uses the MIVQUE0 method (Rao 1972; Giesbrecht 1989) to compute initial values.

The likelihoods that PROC MIXED optimizes are usually well-defined continuous functions with a single optimum. The Newton-Raphson algorithm typically performs well and finds the optimum in a few iterations. It is a quadratically converging algorithm, meaning that the error of the approximation near the optimum is squared at each iteration. The quadratic convergence property is evident when the convergence criterion drops to zero by factors of 10 or more.

Table 46.12. Notation for Order Calculations

Symbol	Number
p	columns of **X**
g	columns of **Z**
N	observations
q	covariance parameters
t	maximum observations per subject
S	subjects

Using the notation from Table 46.12, the following are estimates of the computational speed of the algorithms used in PROC MIXED. For likelihood calculations, the crossproducts matrix construction is of order $N(p+g)^2$ and the sweep operations are of order $(p+g)^3$. The first derivative calculations for parameters in $\mathbf{G}$ are of order qg^3 for ML and $q(g^3+pg^2+p^2g)$ for REML. If you specify a subject effect in the RANDOM statement and if you are not using the REPEATED statement, then replace g by g/S and q by qS in these calculations. The first derivative calculations for parameters in $\mathbf{R}$ are of order $qS(t^3+gt^2+g^2t)$ for ML and $qS(t^3+(p+g)t^2+(p^2+g^2)t)$ for REML. For the second derivatives, replace q by $q(q+1)/2$ in the first derivative expressions. When you specify both $\mathbf{G}$- and $\mathbf{R}$-side parameters (that is, when you use both the RANDOM and REPEATED statements), then additional calculations are required of an order equal to the sum of the orders for $\mathbf{G}$ and $\mathbf{R}$. Considerable execution times may result in this case.

For further details about the computational techniques used in PROC MIXED, refer to Wolfinger, Tobias, and Sall (1994).

Parameter Constraints

By default, some covariance parameters are assumed to satisfy certain boundary constraints during the Newton-Raphson algorithm. For example, variance components are constrained to be nonnegative and autoregressive parameters are constrained to be between -1 and 1. You can remove these constraints with the NOBOUND option in the PARMS statement, but this may lead to estimates that produce an infinite likelihood. You can also introduce or change boundary constraints with the LOWERB= and UPPERB= options in the PARMS statement.

During the Newton-Raphson algorithm, a parameter may be set equal to one of its boundary constraints for a few iterations and then it may move away from the boundary. You see a missing value in the Criterion column of the “Iteration History” table whenever a boundary constraint is dropped.

For some data sets the final estimate of a parameter may equal one of its boundary constraints. This is usually not a cause for concern, but it may lead you to consider a different model. For instance, a variance component estimate can equal zero; in this case, you may want to drop the corresponding random effect from the model. However, be aware that changing the model in this fashion can impact degrees of freedom calculations.

Convergence Problems

For some data sets, the Newton-Raphson algorithm can fail to converge. Nonconvergence can result from a number of causes, including flat or ridged likelihood surfaces and ill-conditioned data.

It is also possible for PROC MIXED to converge to a point that is not the global optimum of the likelihood, although this usually occurs only with the spatial covariance structures.

If you experience convergence problems, the following points may be helpful:

- One useful tool is the PARMS statement, which lets you input initial values for the covariance parameters and performs a grid search over the likelihood surface.
- Sometimes the Newton-Raphson algorithm does not perform well when two of the covariance parameters are on a different scale; that is, they are several orders of magnitude apart. This is because the Hessian matrix is processed jointly for the two parameters, and elements of it corresponding to one of the parameters can become close to internal tolerances in PROC MIXED. In this case, you can improve stability by rescaling the effects in the model so that the covariance parameters are on the same scale.
- Data that is extremely large or extremely small can adversely affect results because of the internal tolerances in PROC MIXED. Rescaling it can improve stability.
- For stubborn problems, you may want to specify ODS OUTPUT COVPARMS= data-set-name to output the “CovParms” table as a precautionary measure. That way, if the problem does not converge, you can read the final parameter values back into a new run with the PARMSDATA= option in the PARMS statement.
- Fisher scoring can be more robust than Newton-Raphson to poor MIVQUE(0) starting values. Specifying a SCORING= value of 5 or so may help to recover from poor starting values.
- Tuning the singularity options SINGULAR=, SINGCHOL=, and SINGRES= in the MODEL statement may improve the stability of the optimization process.

- Tuning the MAXITER= and MAXFUNC= options in the PROC MIXED statement can save resources. Also, the ITDETAILS option displays the values of all of the parameters at each iteration.
- Using the NOPROFILE and NOBOUND options in the PROC MIXED statement may help convergence, although they can produce unusual results.
- Although the CONVH convergence criterion usually gives the best results, you may want to try CONVF or CONVG, possibly along with the ABSOLUTE option.
- If the convergence criterion bottoms out at a relatively small value such as 1E$-$7 but never gets less than 1E$-$8, you may want to specify CONVH=1E$-$6 in the PROC MIXED statement to get results; however, interpret the results with caution.
- An infinite likelihood during the iteration process means that the Newton-Raphson algorithm has stepped into a region where either the $\mathbf{R}$ or $\mathbf{V}$ matrix is nonpositive definite. This is usually no cause for concern as long as iterations continue. If PROC MIXED stops because of an infinite likelihood, recheck your model to make sure that no observations from the same subject are producing identical rows in $\mathbf{R}$ or $\mathbf{V}$ and that you have enough data to estimate the particular covariance structure you have selected. Any time that the final estimated likelihood is infinite, subsequent results should be interpreted with caution.
- A nonpositive definite Hessian matrix can indicate a surface saddlepoint or linear dependencies among the parameters.
- A warning message about the singularities of $\mathbf{X}$ changing indicates that there is some linear dependency in the estimate of $\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X}$ that is not found in $\mathbf{X}'\mathbf{X}$. This can adversely affect the likelihood calculations and optimization process. If you encounter this problem, make sure that your model specification is reasonable and that you have enough data to estimate the particular covariance structure you have selected. Rearranging effects in the MODEL statement so that the most significant ones are first can help because PROC MIXED sweeps the estimate of $\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}$ in the order of the MODEL effects and the sweep is more stable if larger pivots are dealt with first. If this does not help, specifying starting values with the PARMS statement can place the optimization on a different and possibly more stable path.
- Lack of convergence may indicate model misspecification or a violation of the normality assumption.

Memory

Let p be the number of columns in $\mathbf{X}$, and let g be the number of columns in $\mathbf{Z}$. For large models, most of the memory resources are required for holding symmetric matrices of order p, g, and $p + g$. The approximate memory requirement in bytes is

$$40(p^2 + g^2) + 32(p + g)^2$$

If you have a large model that exceeds the memory capacity of your computer, see the suggestions listed under “Computing Time.”

Computing Time

PROC MIXED is computationally intensive, and execution times can be long. In addition to the CPU time used in collecting sums and cross products and in solving the mixed model equations (as in PROC GLM), considerable CPU time is often required to compute the likelihood function and its derivatives. These latter computations are performed for every Newton-Raphson iteration.

If you have a model that takes too long to run, the following suggestions may be helpful:

- Examine the "Model Information" table to find out the number of columns in the $\mathbf{X}$ and $\mathbf{Z}$ matrices. A large number of columns in either matrix can greatly increase computing time. You may want to eliminate some higher order effects if they are too large.
- If you have a $\mathbf{Z}$ matrix with a lot of columns, use the DDFM=BW option in the MODEL statement to eliminate the time required for the containment method.
- If possible, "factor out" a common effect from the effects in the RANDOM statement and make it the SUBJECT= effect. This creates a block-diagonal $\mathbf{G}$ matrix and can often speed calculations.
- If possible, use the same or nested SUBJECT= effects in all RANDOM and REPEATED statements.
- If your data set is very large, you may want to analyze it in pieces. The BY statement can help implement this strategy.
- In general, specify random effects with a lot of levels in the REPEATED statement and those with a few levels in the RANDOM statement.
- The METHOD=MIVQUE0 option runs faster than either the METHOD=REML or METHOD=ML option because it is noniterative.
- You can specify known values for the covariance parameters using the HOLD= or NOITER option in the PARMS statement or the GDATA= option in the RANDOM statement. This eliminates the need for iteration.
- The LOGNOTE option in the PROC MIXED statement writes periodic messages to the SAS log concerning the status of the calculations. It can help you diagnose where the slow down is occurring.

Examples

The following are basic examples of the use of PROC MIXED. More examples and details can be found in Littell et al. (1996), Wolfinger (1997), Verbeke and Molenberghs (1997, 2000), Murray (1998), Singer (1998), Sullivan, Dukes, and Losina (1999), and Brown and Prescott (1999).

Example 46.1. Split-Plot Design

PROC MIXED can fit a variety of mixed models. One of the most common mixed models is the split-plot design. The split-plot design involves two experimental factors, A and B. Levels of A are randomly assigned to whole plots (main plots), and levels of B are randomly assigned to split plots (subplots) within each whole plot. The design provides more precise information about B than about A, and it often arises when A can be applied only to large experimental units. An example is where A represents irrigation levels for large plots of land and B represents different crop varieties planted in each large plot.

Consider the following data from Stroup (1989a), which arise from a balanced split-plot design with the whole plots arranged in a randomized complete-block design. The variable A is the whole-plot factor, and the variable B is the subplot factor. A traditional analysis of these data involves the construction of the whole-plot error (A*Block) to test A and the pooled residual error (B*Block and A*B*Block) to test B and A*B. To carry out this analysis with PROC GLM, you must use a TEST statement to obtain the correct *F*-test for A.

Performing a mixed model analysis with PROC MIXED eliminates the need for the error term construction. PROC MIXED estimates variance components for Block, A*Block, and the residual, and it automatically incorporates the correct error terms into test statistics.

```
data sp;
   input Block A B Y @@;
   datalines;
1 1 1  56  1 1 2  41
1 2 1  50  1 2 2  36
1 3 1  39  1 3 2  35
2 1 1  30  2 1 2  25
2 2 1  36  2 2 2  28
2 3 1  33  2 3 2  30
3 1 1  32  3 1 2  24
3 2 1  31  3 2 2  27
3 3 1  15  3 3 2  19
4 1 1  30  4 1 2  25
4 2 1  35  4 2 2  30
4 3 1  17  4 3 2  18
;
```

```
proc mixed;
   class A B Block;
   model Y = A B A*B;
   random Block A*Block;
run;
```

The variables A, B, and Block are listed as classification variables in the CLASS statement. The columns of model matrix **X** consist of indicator variables corresponding to the levels of the fixed effects A, B, and A*B listed on the right-hand side in the MODEL statement. The dependent variable Y is listed on the left-hand side in the MODEL statement.

The columns of the model matrix **Z** consist of indicator variables corresponding to the levels of the random effects Block and A*Block. The **G** matrix is diagonal and contains the variance components of Block and A*Block. The **R** matrix is also diagonal and contains the residual variance.

The SAS code produces Output 46.1.1.

Output 46.1.1. Split-Plot Example

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.SP
              Dependent Variable           Y
              Covariance Structure         Variance Components
              Estimation Method            REML
              Residual Variance Method     Profile
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Containment
```

The “Model Information” table lists basic information about the split-plot model. REML is used to estimate the variance components, and the residual variances are profiled out of the optimization.

```
                              The Mixed Procedure

                            Class Level Information

                Class    Levels    Values

                A             3    1 2 3
                B             2    1 2
                Block         4    1 2 3 4
```

The “Class Level Information” table lists the levels of all variables specified in the CLASS statement. You can check this table to make sure that the data are correct.

```
                              The Mixed Procedure

                                  Dimensions

                    Covariance Parameters             3
                    Columns in X                     12
                    Columns in Z                     16
                    Subjects                          1
                    Max Obs Per Subject              24
```

The “Dimensions” table lists the magnitudes of various vectors and matrices. The $\mathbf{X}$ matrix is seen to be 24×12, and the $\mathbf{Z}$ matrix is 24×16.

```
                              The Mixed Procedure

                            Number of Observations

                Number of Observations Read              24
                Number of Observations Used              24
                Number of Observations Not Used           0
```

The “Number of Observations” table shows that all observations read from the data set are used in the analysis.

```
                              The Mixed Procedure

                               Iteration History

           Iteration    Evaluations    -2 Res Log Like       Criterion

                   0              1       139.81461222
                   1              1       119.76184570      0.00000000


                           Convergence criteria met.
```

PROC MIXED estimates the variance components for Block, A*Block, and the residual by REML. The REML estimates are the values that maximize the likelihood of a set of linearly independent error contrasts, and they provide a correction for the downward bias found in the usual maximum likelihood estimates. The objective function is -2 times the logarithm of the restricted likelihood, and PROC MIXED minimizes this objective function to obtain the estimates.

The minimization method is the Newton-Raphson algorithm, which uses the first and second derivatives of the objective function to iteratively find its minimum. The “Iteration History” table records the steps of that optimization process. For this example, only one iteration is required to obtain the estimates. The Evaluations column reveals that the restricted likelihood is evaluated once for each of the iterations. A criterion of 0 indicates that the Newton-Raphson algorithm has converged.

```
                         The Mixed Procedure

                        Covariance Parameter
                              Estimates

                        Cov Parm     Estimate

                        Block         62.3958
                        A*Block       15.3819
                        Residual       9.3611
```

The REML estimates for the variance components of Block, A*Block, and the residual are 62.40, 15.38, and 9.36, respectively, as listed in the Estimate column of the "Covariance Parameter Estimates" table.

```
                         The Mixed Procedure

                           Fit Statistics

                -2 Res Log Likelihood           119.8
                AIC (smaller is better)         125.8
                AICC (smaller is better)        127.5
                BIC (smaller is better)         123.9
```

The "Fitting Information" table lists several pieces of information about the fitted mixed model, including the residual log likelihood. Akaike's and Schwarz's criteria can be used to compare different models; the ones with smaller values are preferred.

```
                         The Mixed Procedure

                    Type 3 Tests of Fixed Effects

                       Num     Den
            Effect      DF      DF    F Value    Pr > F

            A            2       6       4.07    0.0764
            B            1       9      19.39    0.0017
            A*B          2       9       4.02    0.0566
```

Finally, the fixed effects are tested using Type III estimable functions. The tests match the one obtained from the following PROC GLM code:

```
proc glm data=sp;
   class A B Block;
   model Y = A B A*B Block A*Block;
   test h=A e=A*Block;
run;
```

You can continue this analysis by producing solutions for the fixed and random effects and then testing various linear combinations of them by using the CONTRAST and ESTIMATE statements. If you use the same CONTRAST and ESTIMATE statements with PROC GLM, the test statistics correspond to the fixed-effects-only model. The test statistics from PROC MIXED incorporate the random effects.

The various "inference space" contrasts given by Stroup (1989a) can be implemented via the ESTIMATE statement. Consider the following examples:

```
estimate 'a1 mean narrow'
          intercept 1 A 1 B .5 .5 A*B .5 .5 |
          Block     .25 .25 .25 .25
          A*Block   .25 .25 .25 .25 0 0 0 0 0 0 0 0;

estimate 'a1 mean intermed'
          intercept 1 A 1 B .5 .5 A*B .5 .5 |
          Block     .25 .25 .25 .25;
estimate 'a1 mean broad'
          intercept 1 a 1 b .5 .5 A*B .5 .5;
```

These statements result in Output 46.1.2.

Output 46.1.2. Inference Space Results

```
                              The Mixed Procedure

                                   Estimates

                                   Standard
Label                 Estimate        Error       DF    t Value     Pr > |t|

a1 mean narrow         32.8750       1.0817        9      30.39       <.0001
a1 mean intermed       32.8750       2.2396        9      14.68       <.0001
a1 mean broad          32.8750       4.5403        9       7.24       <.0001
```

Note that all the estimates are equal, but their standard errors increase with the size of the inference space. The narrow inference space consists of the observed levels of Block and A*Block, and the *t*-statistic value of 30.39 applies only to these levels. This is the same *t*-statistic computed by PROC GLM, because it computes standard errors from the narrow inference space. The intermediate inference space consists of the observed levels of Block and the entire population of levels from which A*Block are sampled. The *t*-statistic value of 14.68 applies to this intermediate space. The broad inference space consists of arbitrary random levels of both Block and A*Block, and the *t*-statistic value of 7.24 is appropriate. Note that the larger the inference space, the weaker the conclusion. However, the broad inference space is usually the one of interest, and even in this space conclusive results are common. The highly significant *p*-value for 'a1 mean broad' is an example. You can also obtain the 'a1 mean broad' result by specifying A in an LSMEANS statement. For more discussion of the inference space concept, refer to McLean, Sanders, and Stroup (1991).

The following statements illustrate another feature of the RANDOM statement. Recall that the basic code for a split-plot design with whole plots arranged in randomized blocks is as follows.

```
proc mixed;
   class A B Block;
   model Y = A B A*B;
   random Block A*Block;
run;
```

An equivalent way of specifying this model is

```
proc mixed data=sp;
   class A B Block;
   model Y = A B A*B;
   random intercept A / subject=Block;
run;
```

In general, if all of the effects in the RANDOM statement can be nested within one effect, you can specify that one effect using the SUBJECT= option. The subject effect is, in a sense, “factored out” of the random effects. The specification using the SUBJECT= effect can result in quicker execution times for large problems because PROC MIXED is able to perform the likelihood calculations separately for each subject.

Example 46.2. Repeated Measures

The following data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14. Some of the observations are suspect (for example, the third observation for person 20); however, all of the data are used here for comparison purposes.

The analysis strategy employs a linear growth curve model for the boys and girls as well as a variance-covariance model that incorporates correlations for all of the observations arising from the same person. The data are assumed to be Gaussian, and their likelihood is maximized to estimate the model parameters. Refer to Jennrich and Schluchter (1986), Louis (1988), Crowder and Hand (1990), Diggle, Liang, and Zeger (1994), and Everitt (1995) for overviews of this approach to repeated measures. Jennrich and Schluchter present results for the Pothoff and Roy data from various covariance structures. The PROC MIXED code to fit an unstructured variance matrix (their Model 2) is as follows:

```
data pr;
   input Person Gender $ y1 y2 y3 y4;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
   datalines;
 1   F   21.0    20.0    21.5    23.0
 2   F   21.0    21.5    24.0    25.5
 3   F   20.5    24.0    24.5    26.0
 4   F   23.5    24.5    25.0    26.5
 5   F   21.5    23.0    22.5    23.5
 6   F   20.0    21.0    21.0    22.5
 7   F   21.5    22.5    23.0    25.0
 8   F   23.0    23.0    23.5    24.0
 9   F   20.0    21.0    22.0    21.5
10   F   16.5    19.0    19.0    19.5
11   F   24.5    25.0    28.0    28.0
12   M   26.0    25.0    29.0    31.0
13   M   21.5    22.5    23.0    26.5
14   M   23.0    22.5    24.0    27.5
15   M   25.5    27.5    26.5    27.0
16   M   20.0    23.5    22.5    26.0
17   M   24.5    25.5    27.0    28.5
18   M   22.0    22.0    24.5    26.5
19   M   24.0    21.5    24.5    25.5
20   M   23.0    20.5    31.0    26.0
21   M   27.5    28.0    31.0    31.5
22   M   23.0    23.0    23.5    25.0
23   M   21.5    23.5    24.0    28.0
24   M   17.0    24.5    26.0    29.5
25   M   22.5    25.5    25.5    26.0
26   M   23.0    24.5    26.0    30.0
27   M   22.0    21.5    23.5    25.0
;

proc mixed data=pr method=ml covtest;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=un subject=Person r;
run;
```

To follow Jennrich and Schluchter, this example uses maximum likelihood (METHOD=ML) instead of the default REML to estimate the unknown covariance parameters. The COVTEST option requests asymptotic tests of all of the covariance parameters.

The MODEL statement first lists the dependent variable Y. The fixed effects are then listed after the equals sign. The variable Gender requests a different intercept for the girls and boys, Age models an overall linear growth trend, and Gender*Age makes the slopes different over time. It is actually not necessary to specify Age separately, but doing so enables PROC MIXED to carry out a test for heterogeneous slopes. The S option requests the display of the fixed-effects solution vector.

The REPEATED statement contains no effects, taking advantage of the default assumption that the observations are ordered similarly for each subject. The TYPE=UN option requests an unstructured block for each SUBJECT=Person. The **R** matrix is, therefore, block diagonal with 27 blocks, each block consisting of identical 4×4 unstructured matrices. The 10 parameters of these unstructured blocks make up the covariance parameters estimated by maximum likelihood. The R option requests that the first block of **R** be displayed.

The results from this analysis are shown in Output 46.2.1.

Output 46.2.1. Repeated Measures with Unstructured Covariance Matrix

```
                               The Mixed Procedure

                                Model Information

              Data Set                     WORK.PR
              Dependent Variable           y
              Covariance Structure         Unstructured
              Subject Effect               Person
              Estimation Method            ML
              Residual Variance Method     None
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Between-Within
```

The covariance structure is listed as “Unstructured” here, and no residual variance is used with this structure. The default degrees-of-freedom method here is “Between-Within.”

```
                               The Mixed Procedure

                             Class Level Information

                Class     Levels    Values

                Person        27    1 2 3 4 5 6 7 8 9 10 11 12 13
                                    14 15 16 17 18 19 20 21 22 23
                                    24 25 26 27
                Gender         2    F M
```

Note that Person has 27 levels and Gender has 2.

```
                               The Mixed Procedure

                                   Dimensions

                       Covariance Parameters           10
                       Columns in X                     6
                       Columns in Z                     0
                       Subjects                        27
                       Max Obs Per Subject              4
```

The 10 covariance parameters result from the 4×4 unstructured blocks of $\mathbf{R}$. There is no $\mathbf{Z}$ matrix for this model, and each of the 27 subjects has a maximum of 4 observations.

```
                           The Mixed Procedure

                         Number of Observations

               Number of Observations Read             108
               Number of Observations Used             108
               Number of Observations Not Used           0
```

```
                           The Mixed Procedure

                            Iteration History

        Iteration    Evaluations        -2 Log Like       Criterion

                0              1       478.24175986
                1              2       419.47721707      0.00000152
                2              1       419.47704812      0.00000000


                        Convergence criteria met.
```

Three Newton-Raphson iterations are required to find the maximum likelihood estimates. The default relative Hessian criterion has a final value less than 1E$-$8, indicating the convergence of the Newton-Raphson algorithm and the attainment of an optimum.

```
                           The Mixed Procedure

                    Estimated R Matrix for Person 1

             Row         Col1         Col2         Col3         Col4

               1       5.1192       2.4409       3.6105       2.5222
               2       2.4409       3.9279       2.7175       3.0624
               3       3.6105       2.7175       5.9798       3.8235
               4       2.5222       3.0624       3.8235       4.6180
```

The preceding 4×4 matrix is the estimated unstructured covariance matrix. It is the estimate of the first block of $\mathbf{R}$, and the other 26 blocks all have the same estimate.

```
                              The Mixed Procedure

                        Covariance Parameter Estimates

                                              Standard         Z
      Cov Parm     Subject     Estimate        Error       Value        Pr Z

      UN(1,1)      Person        5.1192       1.4169        3.61      0.0002
      UN(2,1)      Person        2.4409       0.9835        2.48      0.0131
      UN(2,2)      Person        3.9279       1.0824        3.63      0.0001
      UN(3,1)      Person        3.6105       1.2767        2.83      0.0047
      UN(3,2)      Person        2.7175       1.0740        2.53      0.0114
      UN(3,3)      Person        5.9798       1.6279        3.67      0.0001
      UN(4,1)      Person        2.5222       1.0649        2.37      0.0179
      UN(4,2)      Person        3.0624       1.0135        3.02      0.0025
      UN(4,3)      Person        3.8235       1.2508        3.06      0.0022
      UN(4,4)      Person        4.6180       1.2573        3.67      0.0001
```

The preceding table lists the 10 estimated covariance parameters in order; note their correspondence to the first block of $\mathbf{R}$ displayed previously. The parameter estimates are labeled according to their location in the block in the Cov Parm column, and all of these estimates are associated with Person as the subject effect. The Std Error column lists approximate standard errors of the covariance parameters obtained from the inverse Hessian matrix. These standard errors lead to approximate Wald Z-statistics, which are compared with the standard normal distribution. The results of these tests indicate that all the parameters are significantly different from 0; however, the Wald test can be unreliable in small samples.

To carry out Wald tests of various linear combinations of these parameters, use the following procedure. First, run the code again, adding the ASYCOV option and an ODS statement:

```
ods output CovParms=cp AsyCov=asy;
proc mixed data=pr method=ml covtest asycov;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=un subject=Person r;
run;
```

This creates two data sets, cp and asy, which contain the covariance parameter estimates and their asymptotic variance covariance matrix, respectively. Then read these data sets into the SAS/IML matrix programming language as follows:

```
proc iml;
   use cp;
   read all var {Estimate} into est;
   use asy;
   read all var ('CovP1':'CovP10') into asy;
```

You can then construct your desired linear combinations and corresponding quadratic forms with the asy matrix.

```
                    The Mixed Procedure

                      Fit Statistics

          -2 Log Likelihood               419.5
          AIC (smaller is better)         447.5
          AICC (smaller is better)        452.0
          BIC (smaller is better)         465.6


             Null Model Likelihood Ratio Test

               DF    Chi-Square      Pr > ChiSq

                9         58.76          <.0001
```

The null model likelihood ratio test (LRT) is highly significant for this model, indicating that the unstructured covariance matrix is preferred to the diagonal one of the ordinary least-squares null model. The degrees of freedom for this test is 9, which is the difference between 10 and the 1 parameter for the null model's diagonal matrix.

```
                              The Mixed Procedure

                          Solution for Fixed Effects

                                       Standard
Effect        Gender    Estimate        Error      DF    t Value    Pr > |t|

Intercept                15.8423       0.9356      25      16.93      <.0001
Gender        F           1.5831       1.4658      25       1.08      0.2904
Gender        M                0            .       .        .          .
Age                       0.8268      0.07911      25      10.45      <.0001
Age*Gender    F          -0.3504       0.1239      25      -2.83      0.0091
Age*Gender    M                0            .       .        .          .
```

The preceding table lists the solution vector for the fixed effects. The estimate of the boys' intercept is 15.84, while that for the girls is $15.84+1.58 = 17.42$. Similarly, the estimate for the boys' slope is 0.827, while that for the girls is $0.827-0.350 = 0.477$. Thus the girls' starting point is larger than that for the boys, but their growth rate is about half that of the boys.

Note that two of the estimates equal 0; this is a result of the overparameterized model used by PROC MIXED. You can obtain a full-rank parameterization by using the following MODEL statement:

```
   model y = Gender Gender*Age / noint s;
```

Here, the NOINT option causes the different intercepts to be fit directly as the two levels of Gender. However, this alternative specification results in different tests for these effects.

```
                        The Mixed Procedure

                   Type 3 Tests of Fixed Effects

                         Num     Den
          Effect          DF      DF    F Value    Pr > F

          Gender           1      25       1.17    0.2904
          Age              1      25     110.54    <.0001
          Age*Gender       1      25       7.99    0.0091
```

The “Type 3 Tests of Fixed Effects” table displays Type III tests for all of the fixed effects. These tests are partial in the sense that they account for all of the other fixed effects in the model. In addition, you can use the HTYPE= option in the MODEL statement to obtain Type I (sequential) or Type II (also partial) tests of effects.

It is usually best to consider higher-order terms first, and in this case the Age*Gender test reveals a difference between the slopes that is statistically significant at the 1% level. Note that the p-value for this test (0.0091) is the same as the p-value in the “Age*Gender F” row in the “Solution for Fixed Effects” table and that the F-statistic (7.99) is the square of the t-statistic (-2.83), ignoring rounding error. Similar connections are evident among the other rows in these two tables.

The Age test is one for an overall growth curve accounting for possible heterogeneous slopes, and it is highly significant. Finally, the Gender row tests the null hypothesis of a common intercept, and this hypothesis cannot be rejected from these data.

As an alternative to the F-tests shown here, you can carry out likelihood ratio tests of various hypotheses by fitting the reduced models, subtracting -2 log likelihoods, and comparing the resulting statistics with χ^2 distributions.

Since the different levels of the repeated effect represent different years, it is natural to try fitting a time series model to the data within each subject. To obtain time series structures in $\mathbf{R}$, you can replace TYPE=UN with TYPE=AR(1) or TYPE=TOEP to obtain the first- or *n*th-order autoregressive covariance matrices, respectively. For example, the code to fit an AR(1) structure is

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=ar(1) sub=Person r;
run;
```

To fit a random coefficients model, use the following code:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   random intercept Age / type=un sub=Person g;
run;
```

This specifies an unstructured covariance matrix for the random intercept and slope. In mixed model notation, $\mathbf{G}$ is block diagonal with identical 2×2 unstructured blocks for each person. By default, $\mathbf{R}$ becomes $\sigma^2\mathbf{I}$. See Example 46.5 for further information on this model.

Finally, you can fit a compound symmetry structure by using TYPE=CS.

```
proc mixed data=pr method=ml covtest;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=cs subject=Person r;
run;
```

The results from this analysis are shown in Output 46.2.2.

Output 46.2.2. Repeated Measures with Compound Symmetry Structure

```
                              The Mixed Procedure

                               Model Information

           Data Set                     WORK.PR
           Dependent Variable           y
           Covariance Structure         Compound Symmetry
           Subject Effect               Person
           Estimation Method            ML
           Residual Variance Method     Profile
           Fixed Effects SE Method      Model-Based
           Degrees of Freedom Method    Between-Within
```

The “Model Information” table is the same as before except for the change in “Covariance Structure.”

```
                          The Mixed Procedure

                        Class Level Information

               Class     Levels    Values

               Person        27    1 2 3 4 5 6 7 8 9 10 11 12 13
                                   14 15 16 17 18 19 20 21 22 23
                                   24 25 26 27
               Gender         2    F M


                              Dimensions

                      Covariance Parameters             2
                      Columns in X                      6
                      Columns in Z                      0
                      Subjects                         27
                      Max Obs Per Subject               4


                        Number of Observations

                  Number of Observations Read             108
                  Number of Observations Used             108
                  Number of Observations Not Used           0
```

The compound symmetry structure has two parameters.

```
                          The Mixed Procedure

                           Iteration History

          Iteration    Evaluations        -2 Log Like       Criterion

                  0              1       478.24175986
                  1              1       428.63905802      0.00000000


                       Convergence criteria met.
```

Since the data are balanced, only one step is required to find the estimates.

```
                          The Mixed Procedure

                     Estimated R Matrix for Person 1

               Row        Col1        Col2        Col3        Col4

                 1      4.9052      3.0306      3.0306      3.0306
                 2      3.0306      4.9052      3.0306      3.0306
                 3      3.0306      3.0306      4.9052      3.0306
                 4      3.0306      3.0306      3.0306      4.9052
```

Note the compound symmetry structure here, which consists of a common covariance with a diagonal enhancement.

```
                            The Mixed Procedure

                       Covariance Parameter Estimates

                                         Standard         Z
     Cov Parm     Subject    Estimate       Error     Value         Pr Z

     CS           Person       3.0306      0.9552      3.17       0.0015
     Residual                  1.8746      0.2946      6.36       <.0001
```

The common covariance is estimated to be 3.0306, as listed in the CS row of the preceding table, and the residual variance is estimated to be 1.8746, as listed in the Residual row. You can use these two numbers to estimate the intraclass correlation coefficient (ICC) for this model. Here, the ICC estimate equals 3.0306/(3.0306 + 1.8746) = 0.6178. You can also obtain this number by adding the RCORR option to the REPEATED statement.

```
                            The Mixed Procedure

                              Fit Statistics

                   -2 Log Likelihood               428.6
                   AIC (smaller is better)         440.6
                   AICC (smaller is better)        441.5
                   BIC (smaller is better)         448.4


                      Null Model Likelihood Ratio Test

                        DF    Chi-Square      Pr > ChiSq

                         1         49.60          <.0001
```

In this case, the null model LRT has only one degree of freedom, corresponding to the common covariance parameter. The test indicates that modeling this extra covariance is superior to fitting the simple null model.

```
                            The Mixed Procedure

                        Solution for Fixed Effects

                                    Standard
Effect        Gender    Estimate       Error      DF    t Value    Pr > |t|

Intercept                16.3406      0.9631      25      16.97      <.0001
Gender        F           1.0321      1.5089      25       0.68      0.5003
Gender        M                0           .       .        .         .
Age                       0.7844     0.07654      79      10.25      <.0001
Age*Gender    F          -0.3048      0.1199      79      -2.54      0.0130
Age*Gender    M                0           .       .        .         .
```

Note that the fixed effects estimates and their standard errors are not very different from those in the preceding unstructured example.

```
                              The Mixed Procedure

                         Type 3 Tests of Fixed Effects

                               Num     Den
             Effect             DF      DF    F Value    Pr > F

             Gender              1      25       0.47    0.5003
             Age                 1      79     111.10    <.0001
             Age*Gender          1      79       6.46    0.0130
```

The *F*-tests are also similar to those from the preceding unstructured example. Again, the slopes are significantly different but the intercepts are not.

You can fit the same compound symmetry model with the following specification using the RANDOM statement:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   random Person;
run;
```

Compound symmetry is the structure that Jennrich and Schluchter deemed best among the ones they fit. To carry the analysis one step further, you can use the GROUP= option to specify heterogeneity of this structure across girls and boys.

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=cs subject=Person group=Gender;
run;
```

The results from this analysis are shown in Output 46.2.3.

Output 46.2.3. Repeated Measures with Heterogeneous Structures

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.PR
              Dependent Variable           y
              Covariance Structure         Compound Symmetry
              Subject Effect               Person
              Group Effect                 Gender
              Estimation Method            ML
              Residual Variance Method     None
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Between-Within
```

Note that Gender is listed as a “Group Effect.”

```
                              The Mixed Procedure

                            Class Level Information

                Class     Levels    Values

                Person        27    1 2 3 4 5 6 7 8 9 10 11 12 13
                                    14 15 16 17 18 19 20 21 22 23
                                    24 25 26 27
                Gender         2    F M


                                  Dimensions

                       Covariance Parameters             4
                       Columns in X                      6
                       Columns in Z                      0
                       Subjects                         27
                       Max Obs Per Subject               4


                            Number of Observations

                   Number of Observations Read             108
                   Number of Observations Used             108
                   Number of Observations Not Used           0
```

The four covariance parameters result from the two compound symmetry structures corresponding to the two levels of Gender.

```
                              The Mixed Procedure

                               Iteration History

          Iteration    Evaluations        -2 Log Like       Criterion

                  0              1       478.24175986
                  1              1       408.81297228      0.00000000


                           Convergence criteria met.
```

Even with the heterogeneity, only one iteration is required for convergence.

```
                              The Mixed Procedure

                         Covariance Parameter Estimates

                 Cov Parm     Subject    Group       Estimate

                 Variance     Person     Gender F      0.5900
                 CS           Person     Gender F      3.8804
                 Variance     Person     Gender M      2.7577
                 CS           Person     Gender M      2.4463
```

The preceding table lists the heterogeneous estimates. Note that both the common covariance and the diagonal enhancement differ between girls and boys.

```
                              The Mixed Procedure

                                Fit Statistics

                    -2 Log Likelihood              408.8
                    AIC (smaller is better)        424.8
                    AICC (smaller is better)       426.3
                    BIC (smaller is better)        435.2


                       Null Model Likelihood Ratio Test

                         DF    Chi-Square      Pr > ChiSq

                          3         69.43          <.0001
```

Both Akaike's Information Criterion (424.8) and Schwarz's Bayesian Criterion (435.2) are smaller for this model than for the homogeneous compound symmetry model (440.6 and 448.4, respectively). This indicates that the heterogeneous model is more appropriate. To construct the likelihood ratio test between the two models, subtract the -2 log likelihood values: $428.6 - 408.8 = 19.8$. Comparing this value with the χ^2 distribution with two degrees of freedom yields a *p*-value less than 0.0001, again favoring the heterogeneous model.

```
                              The Mixed Procedure

                          Solution for Fixed Effects

                                            Standard
Effect          Gender     Estimate           Error      DF    t Value     Pr > |t|

Intercept                   16.3406          1.1130      25      14.68       <.0001
Gender          F            1.0321          1.3890      25       0.74       0.4644
Gender          M                 0               .       .          .            .
Age                          0.7844         0.09283      79       8.45       <.0001
Age*Gender      F           -0.3048          0.1063      79      -2.87       0.0053
Age*Gender      M                 0               .       .          .            .
```

Note that the fixed effects estimates are the same as in the homogeneous case, but the standard errors are different.

```
                              The Mixed Procedure

                         Type 3 Tests of Fixed Effects

                              Num      Den
               Effect          DF       DF    F Value    Pr > F

               Gender           1       25       0.55    0.4644
               Age              1       79     141.37    <.0001
               Age*Gender       1       79       8.22    0.0053
```

The fixed effects tests are similar to those from previous models, although the p-values do change as a result of specifying a different covariance structure. It is important for you to select a reasonable covariance structure in order to obtain valid inferences for your fixed effects.

Example 46.3. Plotting the Likelihood

The data for this example are from Hemmerle and Hartley (1973) and are also used as an example for the VARCOMP procedure. The response variable consists of measurements from an oven experiment, and the model contains a fixed effect A and random effects B and A*B.

The SAS code is as follows:

```
data hh;
   input a b y @@;
   datalines;
1 1 237   1 1 254    1 1 246
1 2 178   1 2 179
2 1 208   2 1 178    2 1 187
2 2 146   2 2 145    2 2 141
3 1 186   3 1 183
3 2 142   3 2 125    3 2 136
;
```

```
ods output ParmSearch=parms;
proc mixed data=hh asycov mmeq mmeqsol covtest;
   class a b;
   model y = a / outp=predicted;
   random b a*b;
   lsmeans a;
   parms (17 to 20 by .1) (.3 to .4 by .005) (1.0);
run;

proc print data=predicted;
run;
```

The ASYCOV option in the PROC statement requests the asymptotic variance matrix of the covariance parameter estimates. This matrix is the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian matrix of the objective function evaluated at the final covariance parameter estimates. The MMEQ and MMEQSOL options in the PROC statement request that the mixed model equations and their solution be displayed.

The OUTP= option in the MODEL statement produces the data set predicted, containing the predicted values. Least-squares means (LSMEANS) are requested for A. The PARMS and ODS statements are used to construct a data set containing the likelihood surface.

The results from this analysis are shown in Output 46.3.1.

Output 46.3.1. Plotting the Likelihood

```
                            The Mixed Procedure

                             Model Information

          Data Set                        WORK.HH
          Dependent Variable              y
          Covariance Structure            Variance Components
          Estimation Method               REML
          Residual Variance Method        Profile
          Fixed Effects SE Method         Model-Based
          Degrees of Freedom Method       Containment
```

The “Model Information” table lists details about this variance components model.

```
                            The Mixed Procedure

                          Class Level Information

            Class     Levels    Values

            a              3    1 2 3
            b              2    1 2
```

The “Class Level Information” table lists the levels for A and B.

```
                              The Mixed Procedure

                                  Dimensions

                      Covariance Parameters            3
                      Columns in X                     4
                      Columns in Z                     8
                      Subjects                         1
                      Max Obs Per Subject             16


                            Number of Observations

                  Number of Observations Read              16
                  Number of Observations Used              16
                  Number of Observations Not Used           0
```

The “Dimensions” table reveals that **X** is 16×4 and **Z** is 16×8. Since there are no SUBJECT= effects, PROC MIXED considers the data effectively to be from one subject with 16 observations.

```
                              The Mixed Procedure

                                Parameter Search

    CovP1     CovP2     CovP3  Variance         Res Log Like  -2 Res Log Like

  17.0000    0.3000    1.0000   80.1400             -52.4699         104.9399
  17.0000    0.3050    1.0000   80.0466             -52.4697         104.9393
  17.0000    0.3100    1.0000   79.9545             -52.4694         104.9388
  17.0000    0.3150    1.0000   79.8637             -52.4692         104.9384
  17.0000    0.3200    1.0000   79.7742             -52.4691         104.9381
  17.0000    0.3250    1.0000   79.6859             -52.4690         104.9379
  17.0000    0.3300    1.0000   79.5988             -52.4689         104.9378
  17.0000    0.3350    1.0000   79.5129             -52.4689         104.9377
  17.0000    0.3400    1.0000   79.4282             -52.4689         104.9377
  17.0000    0.3450    1.0000   79.3447             -52.4689         104.9378
"     .         .         .         .                    .                . "
"     .         .         .         .                    .                . "
"     .         .         .         .                    .                . "
  20.0000    0.3550    1.0000   78.2003             -52.4683         104.9366
  20.0000    0.3600    1.0000   78.1201             -52.4684         104.9368
  20.0000    0.3650    1.0000   78.0409             -52.4685         104.9370
  20.0000    0.3700    1.0000   77.9628             -52.4687         104.9373
  20.0000    0.3750    1.0000   77.8857             -52.4689         104.9377
  20.0000    0.3800    1.0000   77.8096             -52.4691         104.9382
  20.0000    0.3850    1.0000   77.7345             -52.4693         104.9387
  20.0000    0.3900    1.0000   77.6603             -52.4696         104.9392
  20.0000    0.3950    1.0000   77.5871             -52.4699         104.9399
  20.0000    0.4000    1.0000   77.5148             -52.4703         104.9406
```

Only a portion of the “Parameter Search” table is shown because the full listing has 651 rows.

```
                               The Mixed Procedure

                                Iteration History

           Iteration    Evaluations    -2 Res Log Like       Criterion

                   1              2       104.93416367      0.00000000


                            Convergence criteria met.
```

Convergence is quick because PROC MIXED starts from the best value from the grid search.

```
                               The Mixed Procedure

                         Covariance Parameter Estimates

                                         Standard         Z
             Cov Parm     Estimate        Error      Value         Pr Z

             b             1464.36      2098.01       0.70       0.2426
             a*b           26.9581      59.6570       0.45       0.3257
             Residual      78.8426      35.3512       2.23       0.0129
```

The preceding table lists the variance components estimates. Note that B is much more variable than A*B.

```
                               The Mixed Procedure

                      Asymptotic Covariance Matrix of Estimates

               Row    Cov Parm         CovP1        CovP2        CovP3

                 1    b              4401640       1.2831      -273.32
                 2    a*b             1.2831      3558.96      -502.84
                 3    Residual       -273.32      -502.84      1249.71
```

The asymptotic covariance matrix also reflects the large variability of B relative to A*B.

```
         The Mixed Procedure

           Fit Statistics

-2 Res Log Likelihood           104.9
AIC (smaller is better)         110.9
AICC (smaller is better)        113.6
BIC (smaller is better)         107.0


   PARMS Model Likelihood Ratio Test

     DF    Chi-Square      Pr > ChiSq

      2          0.00          1.0000
```

The PARMS likelihood ratio test (LRT) compares the best model from the grid search with the final fitted model. Since these models are nearly the same, the LRT is not significant.

```
                                   The Mixed Procedure

                                  Mixed Model Equations

Row  Effect     a  b      Col1      Col2      Col3      Col4      Col5      Col6      Col7

  1  Intercept          0.2029   0.06342   0.07610   0.06342    0.1015    0.1015   0.03805
  2  a          1      0.06342   0.06342                       0.03805   0.02537   0.03805
  3  a          2      0.07610             0.07610             0.03805   0.03805
  4  a          3      0.06342                       0.06342   0.02537   0.03805
  5  b             1    0.1015   0.03805   0.03805   0.02537    0.1022             0.03805
  6  b             2    0.1015   0.02537   0.03805   0.03805              0.1022
  7  a*b        1  1   0.03805   0.03805                       0.03805             0.07515
  8  a*b        1  2   0.02537   0.02537                                 0.02537
  9  a*b        2  1   0.03805             0.03805             0.03805
 10  a*b        2  2   0.03805             0.03805                       0.03805
 11  a*b        3  1   0.02537                       0.02537   0.02537
 12  a*b        3  2   0.03805                       0.03805             0.03805

                                  Mixed Model Equations

        Row      Col8        Col9       Col10       Col11       Col12       Col13

          1   0.02537     0.03805     0.03805     0.02537     0.03805     36.4143
          2   0.02537                                                     13.8757
          3               0.03805     0.03805                             12.7469
          4                                       0.02537     0.03805      9.7917
          5               0.03805                 0.02537                 21.2956
          6   0.02537                 0.03805                 0.03805     15.1187
          7                                                                9.3477
          8   0.06246                                                      4.5280
          9               0.07515                                          7.2676
         10                           0.07515                              5.4793
         11                                       0.06246                  4.6802
         12                                                   0.07515      5.1115
```

The mixed model equations are analogous to the normal equations in the standard linear model. For this example, rows 1–4 correspond to the fixed effects, rows 5–12 correspond to the random effects, and Col13 corresponds to the dependent variable.

```
                                The Mixed Procedure

                           Mixed Model Equations Solution

 Row  Effect     a  b      Col1      Col2      Col3      Col4      Col5      Col6      Col7

   1  Intercept          761.84  -29.7718  -29.6578             -731.14   -733.22   -0.4680
   2  a          1     -29.7718   59.5436   29.7718             -2.0764    2.0764  -14.0239
   3  a          2     -29.6578   29.7718   56.2773             -1.0382    1.0382    0.4680
   4  a          3
   5  b             1   -731.14   -2.0764   -1.0382              741.63    722.73   -4.2598
   6  b             2   -733.22    2.0764    1.0382              722.73    741.63    4.2598
   7  a*b        1  1   -0.4680  -14.0239    0.4680             -4.2598    4.2598   22.8027
   8  a*b        1  2    0.4680  -12.9342   -0.4680              4.2598   -4.2598    4.1555
   9  a*b        2  1   -0.5257    1.0514  -12.9534             -4.7855    4.7855    2.1570
  10  a*b        2  2    0.5257   -1.0514  -14.0048              4.7855   -4.7855   -2.1570
  11  a*b        3  1  -12.4663   12.9342   12.4663             -4.2598    4.2598    1.9200
  12  a*b        3  2  -14.4918   14.0239   14.4918              4.2598   -4.2598   -1.9200

                           Mixed Model Equations Solution

         Row      Col8        Col9       Col10       Col11       Col12       Col13

           1    0.4680     -0.5257      0.5257    -12.4663    -14.4918      159.61
           2  -12.9342      1.0514     -1.0514     12.9342     14.0239     53.2049
           3   -0.4680    -12.9534    -14.0048     12.4663     14.4918      7.8856
           4
           5    4.2598     -4.7855      4.7855     -4.2598      4.2598     26.8837
           6   -4.2598      4.7855     -4.7855      4.2598     -4.2598    -26.8837
           7    4.1555      2.1570     -2.1570      1.9200     -1.9200      3.0198
           8   22.8027     -2.1570      2.1570     -1.9200      1.9200     -3.0198
           9   -2.1570     22.5560      4.4021      2.1570     -2.1570     -1.7134
          10    2.1570      4.4021     22.5560     -2.1570      2.1570      1.7134
          11   -1.9200      2.1570     -2.1570     22.8027      4.1555     -0.8115
          12    1.9200     -2.1570      2.1570      4.1555     22.8027      0.8115
```

This solution matrix results from sweeping all but the last row of the mixed model equations matrix. The final column contains a solution vector for the fixed and random effects. The first four rows correspond to fixed effects and the last eight to random effects.

```
                    The Mixed Procedure

               Type 3 Tests of Fixed Effects

                         Num     Den
           Effect         DF      DF    F Value    Pr > F

           a               2       2      28.00    0.0345
```

The A factor is significant at the 5% level.

```
                              The Mixed Procedure

                              Least Squares Means

                                      Standard
       Effect     a     Estimate         Error       DF     t Value     Pr > |t|

       a          1       212.82       27.6014        2        7.71       0.0164
       a          2       167.50       27.5463        2        6.08       0.0260
       a          3       159.61       27.6014        2        5.78       0.0286
```

The significance of A appears to be from the difference between its first level and its other two levels.

```
                                  StdErr
    Obs   a   b    y      Pred      Pred    DF   Alpha    Lower     Upper       Resid

      1   1   1  237   242.723   4.72563   10    0.05   232.193   253.252    -5.7228
      2   1   1  254   242.723   4.72563   10    0.05   232.193   253.252    11.2772
      3   1   1  246   242.723   4.72563   10    0.05   232.193   253.252     3.2772
      4   1   2  178   182.916   5.52589   10    0.05   170.603   195.228    -4.9159
      5   1   2  179   182.916   5.52589   10    0.05   170.603   195.228    -3.9159
      6   2   1  208   192.670   4.70076   10    0.05   182.196   203.144    15.3297
      7   2   1  178   192.670   4.70076   10    0.05   182.196   203.144   -14.6703
      8   2   1  187   192.670   4.70076   10    0.05   182.196   203.144    -5.6703
      9   2   2  146   142.330   4.70076   10    0.05   131.856   152.804     3.6703
     10   2   2  145   142.330   4.70076   10    0.05   131.856   152.804     2.6703
     11   2   2  141   142.330   4.70076   10    0.05   131.856   152.804    -1.3297
     12   3   1  186   185.687   5.52589   10    0.05   173.374   197.999     0.3134
     13   3   1  183   185.687   5.52589   10    0.05   173.374   197.999    -2.6866
     14   3   2  142   133.542   4.72563   10    0.05   123.013   144.072     8.4578
     15   3   2  125   133.542   4.72563   10    0.05   123.013   144.072    -8.5422
     16   3   2  136   133.542   4.72563   10    0.05   123.013   144.072     2.4578
```

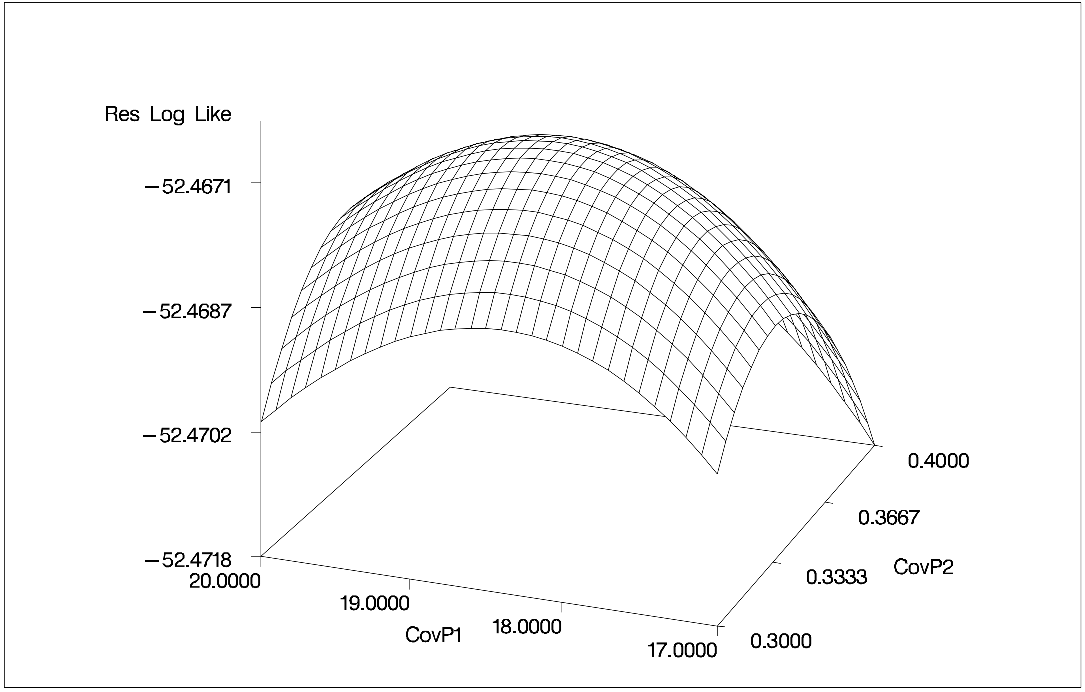
The preceding output lists the predicted values from the model. These values are the sum of the fixed effects estimates and the empirical best linear unbiased predictors (EBLUPs) of the random effects. It is often useful to plot predicted values and residuals to assess the adequacy of the model, using another SAS procedure to generate plots and diagnostic measures.

To plot the likelihood surface using the G3D procedure from SAS/GRAPH software, use the following source:

```
proc g3d data=parms;
   plot CovP1*CovP2 = ResLogLike
        / ctop=red cbottom=blue caxis=black;
run;
```

The results from this plot are shown in Output 46.3.2. The peak of the surface is the REML estimates for the B and A*B variance components.

Output 46.3.2. Plot of Likelihood Surface



Example 46.4. Known G and R

This animal breeding example from Henderson (1984, p. 53) considers multiple traits. The data are artificial and consist of measurements of two traits on three animals, but the second trait of the third animal is missing. Assuming an additive genetic model, you can use PROC MIXED to predict the breeding value of both traits on all three animals and also to predict the second trait of the third animal. The data are as follows:

```
data h;
  input Trait Animal Y;
  datalines;
1 1 6
1 2 8
1 3 7
2 1 9
2 2 5
2 3 .
;
```

Both $\mathbf{G}$ and $\mathbf{R}$ are known.

$$\mathbf{G} = \begin{bmatrix} 2 & 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & .5 & 1 & 2 & .5 \\ 1 & .5 & 2 & 1 & .5 & 2 \\ 2 & 1 & 1 & 3 & 1.5 & 1.5 \\ 1 & 2 & .5 & 1.5 & 3 & .75 \\ 1 & .5 & 2 & 1.5 & .75 & 3 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 4 & 0 & 0 & 1 \\ 1 & 0 & 0 & 5 & 0 & 0 \\ 0 & 1 & 0 & 0 & 5 & 0 \\ 0 & 0 & 1 & 0 & 0 & 5 \end{bmatrix}$$

In order to read $\mathbf{G}$ into PROC MIXED using the GDATA= option in the RANDOM statement, perform the following DATA step:

```
data g;
   input Row Col1-Col6;
   datalines;
1  2  1  1  2   1  1
2  1  2 .5  1   2  .5
3  1 .5  2  1  .5  2
4  2  1  1  3  1.5 1.5
5  1  2 .5 1.5  3  .75
6  1 .5  2 1.5 .75 3
;
```

The preceding data are in the dense representation for a GDATA= data set. You can also construct a data set with the sparse representation using Row, Col, and Value variables, although this would require 21 observations instead of 6 for this example.

The PROC MIXED code is as follows:

```
proc mixed data=h mmeq mmeqsol;
   class Trait Animal;
   model Y = Trait / noint s outp=predicted;
   random Trait*Animal / type=un gdata=g g gi s;
   repeated / type=un sub=Animal r ri;
   parms (4) (1) (5) / noiter;
run;

proc print data=predicted;
run;
```

The MMEQ and MMEQSOL options request the mixed model equations and their solution. The variables Trait and Animal are classification variables, and Trait defines the entire $\mathbf{X}$ matrix for the fixed-effects portion of the model, since the intercept is omitted with the NOINT option. The fixed-effects solution vector and predicted values are also requested using the S and OUTP= options, respectively.

The random effect Trait*Animal leads to a $\mathbf{Z}$ matrix with six columns, the first five corresponding to the identity matrix and the last consisting of 0s. An unstructured $\mathbf{G}$ matrix is specified using the TYPE=UN option, and it is read into PROC MIXED from a SAS data set using the GDATA=G specification. The G and GI options request the display of $\mathbf{G}$ and $\mathbf{G}^{-1}$, respectively. The S option requests that the random-effects solution vector be displayed.

Note that the preceding $\mathbf{R}$ matrix is block diagonal if the data are sorted by animals. The REPEATED statement exploits this fact by requesting $\mathbf{R}$ to have unstructured $2{\times}2$ blocks corresponding to animals, which are the subjects. The R and RI options request that the estimated $2{\times}2$ blocks for the first animal and its inverse be displayed. The PARMS statement lists the parameters of this $2{\times}2$ matrix. Note that the parameters from $\mathbf{G}$ are not specified in the PARMS statement because they have already been assigned using the GDATA= option in the RANDOM statement. The NOITER option prevents PROC MIXED from computing residual (restricted) maximum likelihood estimates; instead, the known values are used for inferences.

The results from this analysis are shown in Output 46.4.1.

Output 46.4.1. Known G and R

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.H
              Dependent Variable           Y
              Covariance Structure         Unstructured
              Subject Effect               Animal
              Estimation Method            REML
              Residual Variance Method     None
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Containment
```

The “Unstructured” covariance structure applies to both $\mathbf{G}$ and $\mathbf{R}$ here.

```
                              The Mixed Procedure

                            Class Level Information

                Class      Levels     Values

                Trait           2     1 2
                Animal          3     1 2 3
```

The levels of Trait and Animal have been specified correctly.

```
                              The Mixed Procedure

                                  Dimensions

                       Covariance Parameters            3
                       Columns in X                     2
                       Columns in Z                     6
                       Subjects                         1
                       Max Obs Per Subject              6


                            Number of Observations

                   Number of Observations Read              6
                   Number of Observations Used              5
                   Number of Observations Not Used          1
```

The three covariance parameters indicated here correspond to those from the **R** matrix. Those from **G** are considered fixed and known because of the GDATA= option.

```
                              The Mixed Procedure

                                Parameter Search

      CovP1        CovP2        CovP3             Res Log Like    -2 Res Log Like

     4.0000       1.0000       5.0000                  -7.3731            14.7463
```

The preceding table results from the PARMS statement.

```
                              The Mixed Procedure

                               Estimated R Matrix
                                  for Subject 1

                            Row         Col1         Col2

                              1       4.0000       1.0000
                              2       1.0000       5.0000
```

The block of **R** corresponding to the first animal is shown in the “Estimated R Matrix” table.

```
                              The Mixed Procedure

                             Estimated Inv(R) Matrix
                                  for Subject 1

                           Row         Col1         Col2

                             1       0.2632     -0.05263
                             2     -0.05263       0.2105
```

The inverse of the block of **R** corresponding to the first animal is shown in the preceding table.

```
                              The Mixed Procedure

                               Estimated G Matrix

 Row   Effect         Trait   Animal       Col1       Col2       Col3       Col4

   1   Trait*Animal   1       1          2.0000     1.0000     1.0000     2.0000
   2   Trait*Animal   1       2          1.0000     2.0000     0.5000     1.0000
   3   Trait*Animal   1       3          1.0000     0.5000     2.0000     1.0000
   4   Trait*Animal   2       1          2.0000     1.0000     1.0000     3.0000
   5   Trait*Animal   2       2          1.0000     2.0000     0.5000     1.5000
   6   Trait*Animal   2       3          1.0000     0.5000     2.0000     1.5000

                               Estimated G Matrix

                           Row        Col5         Col6

                             1      1.0000       1.0000
                             2      2.0000       0.5000
                             3      0.5000       2.0000
                             4      1.5000       1.5000
                             5      3.0000       0.7500
                             6      0.7500       3.0000
```

The preceding table lists the **G** matrix as specified in the GDATA= data set.

```
                              The Mixed Procedure

                            Estimated Inv(G) Matrix

 Row   Effect         Trait   Animal       Col1        Col2        Col3        Col4

   1   Trait*Animal   1       1          2.5000     -1.0000     -1.0000     -1.6667
   2   Trait*Animal   1       2         -1.0000      2.0000                  0.6667
   3   Trait*Animal   1       3         -1.0000                  2.0000      0.6667
   4   Trait*Animal   2       1         -1.6667      0.6667      0.6667      1.6667
   5   Trait*Animal   2       2          0.6667     -1.3333                 -0.6667
   6   Trait*Animal   2       3          0.6667                 -1.3333     -0.6667

                           Estimated Inv(G) Matrix

                          Row       Col5        Col6

                            1     0.6667      0.6667
                            2    -1.3333
                            3                -1.3333
                            4    -0.6667     -0.6667
                            5     1.3333
                            6                 1.3333
```

The preceding table lists $\mathbf{G}^{-1}$. The blank values correspond to zeros.

```
                              The Mixed Procedure

                         Covariance Parameter Estimates

                         Cov Parm    Subject    Estimate

                         UN(1,1)     Animal       4.0000
                         UN(2,1)     Animal       1.0000
                         UN(2,2)     Animal       5.0000
```

The parameters from $\mathbf{R}$ are listed again.

```
                              The Mixed Procedure

                                Fit Statistics

                      -2 Res Log Likelihood            14.7
                      AIC (smaller is better)          14.7
                      AICC (smaller is better)         14.7
                      BIC (smaller is better)          14.7
```

You can use this model-fitting information to compare this model with others.

```
                             The Mixed Procedure

                             Mixed Model Equations

Row    Effect         Trait   Animal       Col1       Col2       Col3       Col4

  1    Trait          1                  0.7763    -0.1053     0.2632     0.2632
  2    Trait          2                 -0.1053     0.4211   -0.05263   -0.05263
  3    Trait*Animal   1       1          0.2632   -0.05263     2.7632    -1.0000
  4    Trait*Animal   1       2          0.2632   -0.05263    -1.0000     2.2632
  5    Trait*Animal   1       3          0.2500               -1.0000
  6    Trait*Animal   2       1        -0.05263     0.2105    -1.7193     0.6667
  7    Trait*Animal   2       2        -0.05263     0.2105     0.6667    -1.3860
  8    Trait*Animal   2       3                                0.6667

                            Mixed Model Equations

        Row       Col5        Col6        Col7        Col8        Col9

          1     0.2500    -0.05263    -0.05263                  4.6974
          2                 0.2105      0.2105                  2.2105
          3    -1.0000     -1.7193      0.6667      0.6667      1.1053
          4                 0.6667     -1.3860                  1.8421
          5     2.2500      0.6667                 -1.3333      1.7500
          6     0.6667      1.8772     -0.6667     -0.6667      1.5789
          7                -0.6667      1.5439                  0.6316
          8    -1.3333     -0.6667                  1.3333
```

The coefficients of the mixed model equations agree with Henderson (1984, p. 55).

```
                             The Mixed Procedure

                        Mixed Model Equations Solution

Row    Effect         Trait   Animal       Col1       Col2       Col3       Col4

  1    Trait          1                  2.5508     1.5685    -1.3047    -1.1775
  2    Trait          2                  1.5685     4.5539    -1.4112    -1.3534
  3    Trait*Animal   1       1         -1.3047    -1.4112     1.8282     1.0652
  4    Trait*Animal   1       2         -1.1775    -1.3534     1.0652     1.7589
  5    Trait*Animal   1       3         -1.1701    -0.9410     1.0206     0.7085
  6    Trait*Animal   2       1         -1.3002    -2.1592     1.8010     1.0900
  7    Trait*Animal   2       2         -1.1821    -2.1055     1.0925     1.7341
  8    Trait*Animal   2       3         -1.1678    -1.3149     1.0070     0.7209

                        Mixed Model Equations Solution

        Row       Col5        Col6        Col7        Col8        Col9

          1    -1.1701     -1.3002     -1.1821     -1.1678      6.9909
          2    -0.9410     -2.1592     -2.1055     -1.3149      6.9959
          3     1.0206      1.8010      1.0925      1.0070     0.05450
          4     0.7085      1.0900      1.7341      0.7209    -0.04955
          5     1.7812      1.0095      0.7197      1.7756     0.02230
          6     1.0095      2.7518      1.6392      1.4849      0.2651
          7     0.7197      1.6392      2.6874      0.9930     -0.2601
          8     1.7756      1.4849      0.9930      2.7645      0.1276
```

The solution to the mixed model equations also matches that given by Henderson (1984, p. 55).

```
                                 The Mixed Procedure

                              Solution for Fixed Effects

                                           Standard
     Effect     Trait     Estimate            Error       DF     t Value     Pr > |t|

     Trait      1           6.9909           1.5971        3        4.38       0.0221
     Trait      2           6.9959           2.1340        3        3.28       0.0465
```

The estimates for the two traits are nearly identical, but the standard error of the second one is larger because of the missing observation.

```
                                 The Mixed Procedure

                             Solution for Random Effects

                                                   Std Err
Effect          Trait    Animal    Estimate           Pred      DF    t Value    Pr > |t|

Trait*Animal    1        1          0.05450         1.3521       0       0.04          .
Trait*Animal    1        2         -0.04955         1.3262       0      -0.04          .
Trait*Animal    1        3          0.02230         1.3346       0       0.02          .
Trait*Animal    2        1           0.2651         1.6589       0       0.16          .
Trait*Animal    2        2          -0.2601         1.6393       0      -0.16          .
Trait*Animal    2        3           0.1276         1.6627       0       0.08          .
```

The Estimate column lists the best linear unbiased predictions (BLUPs) of the breeding values of both traits for all three animals. The *p*-values are missing because the default containment method for computing degrees of freedom results in zero degrees of freedom for the random effects parameter tests.

```
                                 The Mixed Procedure

                             Type 3 Tests of Fixed Effects

                                Num      Den
                     Effect      DF       DF     F Value    Pr > F

                     Trait        2        3       10.59    0.0437
```

The two estimated traits are significantly different from zero at the 5% level.

```
                                          StdErr
 Obs  Trait  Animal  Y     Pred       Pred    DF  Alpha  Lower  Upper     Resid

   1    1       1    6  7.04542  1.33027   0   0.05     .      .    -1.04542
   2    1       2    8  6.94137  1.39806   0   0.05     .      .     1.05863
   3    1       3    7  7.01321  1.41129   0   0.05     .      .    -0.01321
   4    2       1    9  7.26094  1.72839   0   0.05     .      .     1.73906
   5    2       2    5  6.73576  1.74077   0   0.05     .      .    -1.73576
   6    2       3    .  7.12015  2.99088   0   0.05     .      .      .
```

The preceding table contains the predicted values of the observations based on the trait and breeding value estimates, that is, the fixed and random effects. The predicted values are not the predictions of future records in the sense that they do not contain a component corresponding to a new observational error. Refer to Henderson (1984) for information on predicting future records. The L95 and U95 columns usually contain confidence limits for the predicted values; they are missing here because the random-effects parameter degrees of freedom equals 0.

Example 46.5. Random Coefficients

This example comes from a pharmaceutical stability data simulation performed by Obenchain (1990). The observed responses are replicate assay results, expressed in percent of label claim, at various shelf ages, expressed in months. The desired mixed model involves three batches of product that differ randomly in intercept (initial potency) and slope (degradation rate). This type of model is also known as a hierarchical or multilevel model (Singer 1998; Sullivan, Dukes, and Losina 1999).

The SAS code is as follows:

```
data rc;
   input Batch Month @@;
   Monthc = Month;
   do i = 1 to 6;
      input Y @@;
      output;
   end;
   datalines;
1   0  101.2 103.3 103.3 102.1 104.4 102.4
1   1   98.8  99.4  99.7  99.5    .     .
1   3   98.4  99.0  97.3  99.8    .     .
1   6  101.5 100.2 101.7 102.7    .     .
1   9   96.3  97.2  97.2  96.3    .     .
1  12   97.3  97.9  96.8  97.7  97.7  96.7
2   0  102.6 102.7 102.4 102.1 102.9 102.6
2   1   99.1  99.0  99.9 100.6    .     .
2   3  105.7 103.3 103.4 104.0    .     .
2   6  101.3 101.5 100.9 101.4    .     .
2   9   94.1  96.5  97.2 95.6     .     .
2  12   93.1  92.8  95.4 92.2   92.2  93.0
```

```
3   0  105.1 103.9 106.1 104.1 103.7 104.6
3   1  102.2 102.0 100.8  99.8    .     .
3   3  101.2 101.8 100.8 102.6    .     .
3   6  101.1 102.0 100.1 100.2    .     .
3   9  100.9  99.5 102.2 100.8    .     .
3  12   97.8  98.3  96.9  98.4  96.9  96.5
;

proc mixed data=rc;
   class Batch;
   model Y = Month / s;
   random Int Month / type=un sub=Batch s;
run;
```

In the DATA step, Monthc is created as a duplicate of Month in order to enable both a continuous and classification version of the same variable. The variable Monthc is used in a subsequent analysis on page 2814.

In the PROC MIXED code, Batch is listed as the only classification variable. The fixed effect Month in the MODEL statement is not declared a classification variable; thus it models a linear trend in time. An intercept is included as a fixed effect by default, and the S option requests that the fixed-effects parameter estimates be produced.

The two RANDOM effects are Int and Month, modeling random intercepts and slopes, respectively. Note that Intercept and Month are used as both fixed and random effects. The TYPE=UN option in the RANDOM statement specifies an unstructured covariance matrix for the random intercept and slope effects. In mixed model notation, $\mathbf{G}$ is block diagonal with unstructured 2×2 blocks. Each block corresponds to a different level of Batch, which is the SUBJECT= effect. The unstructured type provides a mechanism for estimating the correlation between the random coefficients. The S option requests the production of the random-effects parameter estimates.

The results from this analysis are shown in Output 46.5.1.

Output 46.5.1. Random Coefficients Analysis

```
                          The Mixed Procedure

                           Model Information

              Data Set                        WORK.RC
              Dependent Variable              Y
              Covariance Structure            Unstructured
              Subject Effect                  Batch
              Estimation Method               REML
              Residual Variance Method        Profile
              Fixed Effects SE Method         Model-Based
              Degrees of Freedom Method       Containment
```

The “Unstructured” covariance structure applies to $\mathbf{G}$ here.

```
                         The Mixed Procedure

                       Class Level Information

            Class    Levels    Values

            Batch         3    1 2 3
```

Batch is the only classification variable in this analysis, and it has three levels.

```
                         The Mixed Procedure

                             Dimensions

                 Covariance Parameters             4
                 Columns in X                      2
                 Columns in Z Per Subject          2
                 Subjects                          3
                 Max Obs Per Subject              36


                       Number of Observations

             Number of Observations Read             108
             Number of Observations Used              84
             Number of Observations Not Used          24
```

The “Dimensions” table indicates that there are three subjects (corresponding to batches). The 24 observations not used correspond to the missing values of Y in the input data set.

```
                         The Mixed Procedure

                          Iteration History

        Iteration    Evaluations    -2 Res Log Like       Criterion

                0              1       367.02768461
                1              1       350.32813577      0.00000000


                       Convergence criteria met.
```

Only one iteration is required for convergence.

```
      The Mixed Procedure

 Covariance Parameter Estimates

Cov Parm     Subject    Estimate

UN(1,1)      Batch        0.9768
UN(2,1)      Batch       -0.1045
UN(2,2)      Batch       0.03717
Residual                  3.2932
```

The estimated elements of the unstructured 2×2 matrix comprising the blocks of $\mathbf{G}$ are listed in the Estimate column. Note that the random coefficients are negatively correlated.

```
         The Mixed Procedure

           Fit Statistics

-2 Res Log Likelihood           350.3
AIC (smaller is better)         358.3
AICC (smaller is better)        358.8
BIC (smaller is better)         354.7


   Null Model Likelihood Ratio Test

     DF    Chi-Square      Pr > ChiSq

      3         16.70          0.0008
```

The null model likelihood ratio test indicates a significant improvement over the null model consisting of no random effects and a homogeneous residual error.

```
                      The Mixed Procedure

                   Solution for Fixed Effects

                         Standard
Effect       Estimate       Error      DF    t Value    Pr > |t|

Intercept      102.70      0.6456       2     159.08      <.0001
Month         -0.5259      0.1194       2      -4.41      0.0478
```

The fixed effects estimates represent the estimated means for the random intercept and slope, respectively.

```
                            The Mixed Procedure

                        Solution for Random Effects

                                            Std Err
 Effect        Batch    Estimate       Pred      DF    t Value    Pr > |t|

 Intercept     1         -1.0010     0.6842      78      -1.46      0.1474
 Month         1          0.1287     0.1245      78       1.03      0.3047
 Intercept     2          0.3934     0.6842      78       0.58      0.5669
 Month         2         -0.2060     0.1245      78      -1.65      0.1021
 Intercept     3          0.6076     0.6842      78       0.89      0.3772
 Month         3         0.07731     0.1245      78       0.62      0.5365
```

The random effects estimates represent the estimated deviation from the mean intercept and slope for each batch. Therefore, the intercept for the first batch is close to $102.7 - 1 = 101.7$, while the intercepts for the other two batches are greater than 102.7. The second batch has a slope less than the mean slope of -0.526, while the other two batches have slopes larger than -0.526.

```
                            The Mixed Procedure

                       Type 3 Tests of Fixed Effects

                              Num      Den
              Effect          DF       DF     F Value    Pr > F

              Month            1        2       19.41    0.0478
```

The *F*-statistic in the “Type 3 Tests of Fixed Effects” table is the square of the *t*-statistic used in the test of Month in the preceding “Solution for Fixed Effects” table. Both statistics test the null hypothesis that the slope assigned to Month equals 0, and this hypothesis can barely be rejected at the 5% level.

It is also possible to fit a random coefficients model with error terms that follow a nested structure (Fuller and Battese 1973). The following SAS code represents one way of doing this:

```
proc mixed data=rc;
   class Batch Monthc;
   model Y = Month / s;
   random Int Month Monthc / sub=Batch s;
run;
```

The variable Monthc is added to the CLASS and RANDOM statements, and it models the nested errors. Note that Month and Monthc are continuous and classification versions of the same variable. Also, the TYPE=UN option is dropped from the RANDOM statement, resulting in the default variance components model instead of correlated random coefficients.

The results from this analysis are shown in Output 46.5.2.

Output 46.5.2. Random Coefficients with Nested Errors Analysis

```
                              The Mixed Procedure

                               Model Information

             Data Set                     WORK.RC
             Dependent Variable           Y
             Covariance Structure         Variance Components
             Subject Effect               Batch
             Estimation Method            REML
             Residual Variance Method     Profile
             Fixed Effects SE Method      Model-Based
             Degrees of Freedom Method    Containment


                            Class Level Information

               Class     Levels    Values

               Batch          3    1 2 3
               Monthc         6    0 1 3 6 9 12


                                  Dimensions

                      Covariance Parameters             4
                      Columns in X                      2
                      Columns in Z Per Subject          8
                      Subjects                          3
                      Max Obs Per Subject              36


                            Number of Observations

                  Number of Observations Read             108
                  Number of Observations Used              84
                  Number of Observations Not Used          24


                               Iteration History

          Iteration    Evaluations    -2 Res Log Like       Criterion

                  0              1       367.02768461
                  1              4       277.51945360      .
                  2              1       276.97551718      0.00104208
                  3              1       276.90304909      0.00003174
                  4              1       276.90100316      0.00000004
                  5              1       276.90100092      0.00000000


                           Convergence criteria met.


                        Covariance Parameter Estimates

                       Cov Parm      Subject    Estimate

                       Intercept     Batch             0
                       Month         Batch       0.01243
                       Monthc        Batch        3.7411
                       Residual                   0.7969
```

For this analysis, the Newton-Raphson algorithm requires five iterations and nine likelihood evaluations to achieve convergence. The missing value in the Criterion column in iteration 1 indicates that a boundary constraint has been dropped.

The estimate for the Intercept variance component equals 0. This occurs frequently in practice and indicates that the restricted likelihood is maximized by setting this variance component equal to 0. Whenever a zero variance component estimate occurs, the following note appears in the SAS log:

```
NOTE: Estimated G matrix is not positive definite.
```

The remaining variance component estimates are positive, and the estimate corresponding to the nested errors (MONTHC) is much larger than the other two.

```
                              The Mixed Procedure

                                Fit Statistics

                     -2 Res Log Likelihood           276.9
                     AIC (smaller is better)         282.9
                     AICC (smaller is better)        283.2
                     BIC (smaller is better)         280.2
```

A comparison of AIC and BIC for this model with those of the previous model favors the nested error model. Strictly speaking, a likelihood ratio test cannot be carried out between the two models because one is not contained in the other; however, a cautious comparison of likelihoods can be informative.

```
                              The Mixed Procedure

                           Solution for Fixed Effects

                                 Standard
        Effect       Estimate       Error      DF    t Value    Pr > |t|

        Intercept      102.56      0.7287       2     140.74      <.0001
        Month         -0.5003      0.1259       2      -3.97      0.0579
```

The better-fitting covariance model impacts the standard errors of the fixed effects parameter estimates more than the estimates themselves.

```
                            The Mixed Procedure

                       Solution for Random Effects

                                                Std Err
Effect       Batch    Monthc    Estimate        Pred     DF    t Value    Pr > |t|

Intercept    1                         0           .      .         .          .
Month        1                  -0.00028     0.09268     66      -0.00      0.9976
Monthc       1           0        0.2191      0.7896     66       0.28      0.7823
Monthc       1           1       -2.5690      0.7571     66      -3.39      0.0012
Monthc       1           3       -2.3067      0.6865     66      -3.36      0.0013
Monthc       1           6        1.8726      0.7328     66       2.56      0.0129
Monthc       1           9       -1.2350      0.9300     66      -1.33      0.1888
Monthc       1          12        0.7736      1.1992     66       0.65      0.5211
Intercept    2                         0           .      .         .          .
Month        2                  -0.07571     0.09268     66      -0.82      0.4169
Monthc       2           0      -0.00621      0.7896     66      -0.01      0.9938
Monthc       2           1       -2.2126      0.7571     66      -2.92      0.0048
Monthc       2           3        3.1063      0.6865     66       4.53      <.0001
Monthc       2           6        2.0649      0.7328     66       2.82      0.0064
Monthc       2           9       -1.4450      0.9300     66      -1.55      0.1250
Monthc       2          12       -2.4405      1.1992     66      -2.04      0.0459
Intercept    3                         0           .      .         .          .
Month        3                   0.07600     0.09268     66       0.82      0.4152
Monthc       3           0        1.9574      0.7896     66       2.48      0.0157
Monthc       3           1       -0.8850      0.7571     66      -1.17      0.2466
Monthc       3           3        0.3006      0.6865     66       0.44      0.6629
Monthc       3           6        0.7972      0.7328     66       1.09      0.2806
Monthc       3           9        2.0059      0.9300     66       2.16      0.0347
Monthc       3          12      0.002293      1.1992     66       0.00      0.9985
```

The random effects solution provides the empirical best linear unbiased predictions (EBLUPs) for the realizations of the random intercept, slope, and nested errors. You can use these values to compare batches and months.

```
                            The Mixed Procedure

                       Type 3 Tests of Fixed Effects

                          Num      Den
               Effect      DF       DF    F Value    Pr > F

               Month        1        2      15.78    0.0579
```

The test of Month is similar to that from the previous model, although it is no longer significant at the 5% level.

Example 46.6. Line-Source Sprinkler Irrigation

These data appear in Hanks et al. (1980), Johnson, Chaudhuri, and Kanemasu (1983), and Stroup (1989b). Three cultivars (Cult) of winter wheat are randomly assigned to rectangular plots within each of three blocks (Block). The nine plots are located side-by-side, and a line-source sprinkler is placed through the middle. Each plot is subdivided into twelve subplots, six to the north of the line-source, six to the south (Dir). The two plots closest to the line-source represent the maximum irrigation level

(Irrig=6), the two next-closest plots represent the next-highest level (Irrig=5), and so forth.

This example is a case where both $\mathbf{G}$ and $\mathbf{R}$ can be modeled. One of Stroup's models specifies a diagonal $\mathbf{G}$ containing the variance components for Block, Block*Dir, and Block*Irrig, and a Toeplitz $\mathbf{R}$ with four bands. The SAS code to fit this model and carry out some further analyses follows.

Caution: This analysis may require considerable CPU time.

```
data line;
   length Cult$ 8;
   input Block Cult$ @;
   row = _n_;
   do Sbplt=1 to 12;
     if Sbplt le 6 then do;
       Irrig = Sbplt;
       Dir = 'North';
     end;
     else do;
       Irrig = 13 - Sbplt;
       Dir = 'South';
     end;
     input Y @; output;
   end;
   datalines;
1 Luke     2.4 2.7 5.6 7.5 7.9 7.1 6.1 7.3 7.4 6.7 3.8 1.8
1 Nugaines 2.2 2.2 4.3 6.3 7.9 7.1 6.2 5.3 5.3 5.2 5.4 2.9
1 Bridger  2.9 3.2 5.1 6.9 6.1 7.5 5.6 6.5 6.6 5.3 4.1 3.1
2 Nugaines 2.4 2.2 4.0 5.8 6.1 6.2 7.0 6.4 6.7 6.4 3.7 2.2
2 Bridger  2.6 3.1 5.7 6.4 7.7 6.8 6.3 6.2 6.6 6.5 4.2 2.7
2 Luke     2.2 2.7 4.3 6.9 6.8 8.0 6.5 7.3 5.9 6.6 3.0 2.0
3 Nugaines 1.8 1.9 3.7 4.9 5.4 5.1 5.7 5.0 5.6 5.1 4.2 2.2
3 Luke     2.1 2.3 3.7 5.8 6.3 6.3 6.5 5.7 5.8 4.5 2.7 2.3
3 Bridger  2.7 2.8 4.0 5.0 5.2 5.2 5.9 6.1 6.0 4.3 3.1 3.1
;

proc mixed;
   class Block Cult Dir Irrig;
   model Y = Cult|Dir|Irrig@2;
   random Block Block*Dir Block*Irrig;
   repeated / type=toep(4) sub=Block*Cult r;
   lsmeans Cult|Irrig;
   estimate 'Bridger vs Luke' Cult 1 -1 0;
   estimate 'Linear Irrig' Irrig -5 -3 -1 1 3 5;
   estimate 'B vs L x Linear Irrig' Cult*Irrig
            -5 -3 -1 1 3 5 5 3 1 -1 -3 -5;
run;
```

The preceding code uses the bar operator (|) and the at sign (@) to specify all two-factor interactions between Cult, Dir, and Irrig as fixed effects.

The RANDOM statement sets up the **Z** and **G** matrices corresponding to the random effects Block, Block*Dir, and Block*Irrig.

In the REPEATED statement, the TYPE=TOEP(4) option sets up the blocks of the **R** matrix to be Toeplitz with four bands below and including the main diagonal. The subject effect is Block(Cult), and it produces nine 12×12 blocks. The R option requests that the first block of **R** be displayed.

Least-squares means (LSMEANS) are requested for Cult, Irrig, and Cult*Irrig, and a few ESTIMATE statements are specified to illustrate some linear combinations of the fixed effects.

The results from this analysis are shown in Output 46.6.1.

Output 46.6.1. Line-Source Sprinkler Irrigation Analysis

```
                              The Mixed Procedure

                               Model Information

              Data Set                     WORK.LINE
              Dependent Variable           Y
              Covariance Structures        Variance Components,
                                           Toeplitz
              Subject Effect               Block*Cult
              Estimation Method            REML
              Residual Variance Method     Profile
              Fixed Effects SE Method      Model-Based
              Degrees of Freedom Method    Containment
```

The Covariance Structures row reveals the two different structures assumed for **G** and **R**.

```
                              The Mixed Procedure

                            Class Level Information

                Class    Levels    Values

                Block         3    1 2 3
                Cult          3    Bridger Luke Nugaines
                Dir           2    North South
                Irrig         6    1 2 3 4 5 6
```

The levels of each class variable are listed as a single string in the Values column, regardless of whether the levels are numeric or character.

```
                              The Mixed Procedure

                                  Dimensions

                      Covariance Parameters             7
                      Columns in X                     48
                      Columns in Z                     27
                      Subjects                          1
                      Max Obs Per Subject             108


                            Number of Observations

                  Number of Observations Read             108
                  Number of Observations Used             108
                  Number of Observations Not Used           0
```

Even though there is a SUBJECT= effect in the REPEATED statement, the analysis considers all of the data to be from one subject because there is no corresponding SUBJECT= effect in the RANDOM statement.

```
                              The Mixed Procedure

                               Iteration History

          Iteration    Evaluations    -2 Res Log Like       Criterion

                  0              1       226.25427252
                  1              4       187.99336173        .
                  2              3       186.62579299      0.10431081
                  3              1       184.38218213      0.04807260
                  4              1       183.41836853      0.00886548
                  5              1       183.25111475      0.00075353
                  6              1       183.23809997      0.00000748
                  7              1       183.23797748      0.00000000


                            Convergence criteria met.
```

The Newton-Raphson algorithm converges successfully in seven iterations.

```
                           The Mixed Procedure

                     Estimated R Matrix for Subject 1

Row      Col1      Col2      Col3      Col4      Col5      Col6      Col7

  1    0.2850  0.007986  0.001452  -0.09253
  2  0.007986    0.2850  0.007986  0.001452  -0.09253
  3  0.001452  0.007986    0.2850  0.007986  0.001452  -0.09253
  4  -0.09253  0.001452  0.007986    0.2850  0.007986  0.001452  -0.09253
  5            -0.09253  0.001452  0.007986    0.2850  0.007986  0.001452
  6                      -0.09253  0.001452  0.007986    0.2850  0.007986
  7                                -0.09253  0.001452  0.007986    0.2850
  8                                          -0.09253  0.001452  0.007986
  9                                                    -0.09253  0.001452
 10                                                              -0.09253
 11
 12

                     Estimated R Matrix for Subject 1

      Row      Col8        Col9       Col10       Col11       Col12

        1
        2
        3
        4
        5  -0.09253
        6  0.001452    -0.09253
        7  0.007986    0.001452    -0.09253
        8    0.2850    0.007986    0.001452    -0.09253
        9  0.007986      0.2850    0.007986    0.001452    -0.09253
       10  0.001452    0.007986      0.2850    0.007986    0.001452
       11  -0.09253    0.001452    0.007986      0.2850    0.007986
       12              -0.09253    0.001452    0.007986      0.2850
```

The first block of the estimated **R** matrix has the TOEP(4) structure, and the observations that are three plots apart exhibit a negative correlation.

```
                           The Mixed Procedure

                      Covariance Parameter Estimates

                  Cov Parm        Subject       Estimate

                  Block                           0.2194
                  Block*Dir                      0.01768
                  Block*Irrig                    0.03539
                  TOEP(2)         Block*Cult    0.007986
                  TOEP(3)         Block*Cult    0.001452
                  TOEP(4)         Block*Cult    -0.09253
                  Residual                        0.2850
```

The preceding table lists the estimated covariance parameters from both **G** and **R**. The first three are the variance components making up the diagonal **G**, and the final four make up the Toeplitz structure in the blocks of **R**. The Residual row corresponds to the variance of the Toeplitz structure, and it was the parameter profiled out during the optimization process.

```
                              The Mixed Procedure

                                Fit Statistics

                     -2 Res Log Likelihood           183.2
                     AIC (smaller is better)         197.2
                     AICC (smaller is better)        198.8
                     BIC (smaller is better)         190.9
```

The “-2 Res Log Likelihood” value is the same as the final value listed in the “Iteration History” table.

```
                              The Mixed Procedure

                         Type 3 Tests of Fixed Effects

                               Num     Den
                Effect          DF      DF    F Value    Pr > F

                Cult             2      68       7.98    0.0008
                Dir              1       2       3.95    0.1852
                Cult*Dir         2      68       3.44    0.0379
                Irrig            5      10     102.60    <.0001
                Cult*Irrig      10      68       1.91    0.0580
                Dir*Irrig        5      68       6.12    <.0001
```

Every fixed effect except for Dir and Cult*Irrig is significant at the 5% level.

```
                              The Mixed Procedure

                                   Estimates

                                           Standard
  Label                       Estimate        Error       DF    t Value    Pr > |t|

  Bridger vs Luke             -0.03889      0.09524       68      -0.41      0.6843
  Linear Irrig                 30.6444       1.4412       10      21.26      <.0001
  B vs L x Linear Irrig        -9.8667       2.7400       68      -3.60      0.0006
```

The “Estimates” table lists the results from the various linear combinations of fixed effects specified in the ESTIMATE statements. Bridger is not significantly different from Luke, and Irrig possesses a strong linear component. This strength appears to be influencing the significance of the interaction.

```
                              The Mixed Procedure

                              Least Squares Means

                                                Standard
Effect       Cult       Irrig   Estimate        Error     DF   t Value   Pr > |t|

Cult         Bridger              5.0306       0.2874     68     17.51     <.0001
Cult         Luke                 5.0694       0.2874     68     17.64     <.0001
Cult         Nugaines             4.7222       0.2874     68     16.43     <.0001
Irrig                   1         2.4222       0.3220     10      7.52     <.0001
Irrig                   2         3.1833       0.3220     10      9.88     <.0001
Irrig                   3         5.0556       0.3220     10     15.70     <.0001
Irrig                   4         6.1889       0.3220     10     19.22     <.0001
Irrig                   5         6.4000       0.3140     10     20.38     <.0001
Irrig                   6         6.3944       0.3227     10     19.81     <.0001
Cult*Irrig   Bridger    1         2.8500       0.3679     68      7.75     <.0001
Cult*Irrig   Bridger    2         3.4167       0.3679     68      9.29     <.0001
Cult*Irrig   Bridger    3         5.1500       0.3679     68     14.00     <.0001
Cult*Irrig   Bridger    4         6.2500       0.3679     68     16.99     <.0001
Cult*Irrig   Bridger    5         6.3000       0.3463     68     18.19     <.0001
Cult*Irrig   Bridger    6         6.2167       0.3697     68     16.81     <.0001
Cult*Irrig   Luke       1         2.1333       0.3679     68      5.80     <.0001
Cult*Irrig   Luke       2         2.8667       0.3679     68      7.79     <.0001
Cult*Irrig   Luke       3         5.2333       0.3679     68     14.22     <.0001
Cult*Irrig   Luke       4         6.5500       0.3679     68     17.80     <.0001
Cult*Irrig   Luke       5         6.8833       0.3463     68     19.87     <.0001
Cult*Irrig   Luke       6         6.7500       0.3697     68     18.26     <.0001
Cult*Irrig   Nugaines   1         2.2833       0.3679     68      6.21     <.0001
Cult*Irrig   Nugaines   2         3.2667       0.3679     68      8.88     <.0001
Cult*Irrig   Nugaines   3         4.7833       0.3679     68     13.00     <.0001
Cult*Irrig   Nugaines   4         5.7667       0.3679     68     15.67     <.0001
Cult*Irrig   Nugaines   5         6.0167       0.3463     68     17.37     <.0001
Cult*Irrig   Nugaines   6         6.2167       0.3697     68     16.81     <.0001
```

The LS-means are useful in comparing the levels of the various fixed effects. For example, it appears that irrigation levels 5 and 6 have virtually the same effect.

An interesting exercise is to try fitting other variance-covariance models to these data and comparing them to this one using likelihood ratio tests, Akaike's Information Criterion, or Schwarz's Bayesian Information Criterion. In particular, some spatial models are worth investigating (Marx and Thompson 1987; Zimmerman and Harville 1991). The following is one example of spatial model code.

```
proc mixed;
   class Block Cult Dir Irrig;
   model Y = Cult|Dir|Irrig@2;
   repeated / type=sp(pow)(Row Sbplt)
      sub=intercept;
run;
```

The TYPE=SP(POW)(ROW SBPLT) option in the REPEATED statement requests the spatial power structure, with the two defining coordinate variables being Row and Sbplt. The SUB=INTERCEPT option indicates that the entire data set is to be considered as one subject, thereby modeling $\mathbf{R}$ as a dense 108×108 covariance matrix. Refer to Wolfinger (1993) for further discussion of this example and additional analyses.

Example 46.7. Influence in Heterogeneous Variance Model (Experimental)

In this example from Snedecor and Cochran (1976, p. 256) a one-way classification model with heterogeneous variances is fit. The data represent amounts of different types of fat absorbed by batches of doughnuts during cooking, measured in grams.

```
data absorb;
   input FatType Absorbed @@;
   datalines;
   1 164  1 172  1 168  1 177  1 156  1 195
   2 178  2 191  2 197  2 182  2 185  2 177
   3 175  3 193  3 178  3 171  3 163  3 176
   4 155  4 166  4 149  4 164  4 170  4 168
;
```

The statistical model for these data can be written as

$$\begin{aligned} Y_{ij} &= \mu + \tau_i + \epsilon_{ij} \\ i &= 1, \cdots, t = 4 \\ j &= 1, \cdots, r = 6 \\ \epsilon_{ij} &= (0, \sigma_i^2) \end{aligned}$$

where Y_{ij} is the amount of fat absorbed by the jth batch of the ith fat type, and τ_i denotes the fat-type effects. A quick glance at the data suggests that observations 6, 9, 14, and 21 might be influential on the analysis, because these are extreme observations for the respective fat types.

The following SAS statements fit this model and request influence diagnostics for the fixed effects and covariance parameters. The experimental ODS GRAPHICS statement requests plots of the influence diagnostics in addition to the tabular output. The ESTIMATE suboption requests plots of “leave-one-out” estimates for the fixed effects and group variances.

```
ods html;
ods graphics on;

proc mixed data=absorb asycov;
   class FatType;
   model Absorbed = FatType / s
                    influence(iter=10 estimates);
   repeated / group=FatType;
   ods output Influence=inf;
run;

ods graphics off;
ods html close;
```

The “Influence” table is output to the SAS data set inf so that parameter estimates can be printed subsequently. Results from this analysis are shown in Output 46.7.1.

Output 46.7.1. Heterogeneous Variance Analysis

```
                              The Mixed Procedure

                               Model Information

              Data Set                       WORK.ABSORB
              Dependent Variable             Absorbed
              Covariance Structure           Variance Components
              Group Effect                   FatType
              Estimation Method              REML
              Residual Variance Method       None
              Fixed Effects SE Method        Model-Based
              Degrees of Freedom Method      Between-Within


                          Covariance Parameter Estimates

                        Cov Parm     Group        Estimate

                        Residual     FatType 1      178.00
                        Residual     FatType 2     60.4000
                        Residual     FatType 3     97.6000
                        Residual     FatType 4     67.6000


                            Solution for Fixed Effects

                     Fat                    Standard
     Effect          Type     Estimate        Error       DF    t Value    Pr > |t|

     Intercept                  162.00       3.3566       20      48.26      <.0001
     FatType         1         10.0000       6.3979       20       1.56      0.1337
     FatType         2         23.0000       4.6188       20       4.98      <.0001
     FatType         3         14.0000       5.2472       20       2.67      0.0148
     FatType         4               0            .        .          .           .
```

The variances in the four groups are shown in the “Covariance Parameter Estimates” table. The estimated variance in the first group is two to three times larger than the variance in the other groups.

The fixed effects solutions correspond to estimates of the following parameters:

$$\begin{array}{rcl} \text{Intercept} & : & \mu + \tau_4 \\ \text{FatType 1} & : & \tau_1 - \tau_4 \\ \text{FatType 2} & : & \tau_2 - \tau_4 \\ \text{FatType 3} & : & \tau_3 - \tau_4 \\ \text{FatType 4} & : & 0 \end{array}$$

You can easily verify that these estimates are simple functions of the arithmetic means $\overline{y}_{i.}$ in the groups. For example, $\widehat{\mu + \tau_4} = \overline{y}_{4.} = 162.0$, $\widehat{\tau_1 - \tau_4} = \overline{y}_{1.} - \overline{y}_{4.} = 10.0$, and so forth. The covariance parameter estimates are the sample variances in the groups and are uncorrelated.

```
                              The Mixed Procedure

                    Asymptotic Covariance Matrix of Estimates

     Row    Cov Parm        CovP1       CovP2       CovP3       CovP4

       1    Residual        12674
       2    Residual                  1459.26
       3    Residual                              3810.30
       4    Residual                                          1827.90
```

The following statements print the “leave-one-out” estimates for fixed effects and covariance parameters that were written to the inf data set with the ESTIMATES suboption.

```
proc print data=inf label;
   var parm1-parm5 covp1-covp4;
run;
```

Output 46.7.2. Leave-One-Out Estimates

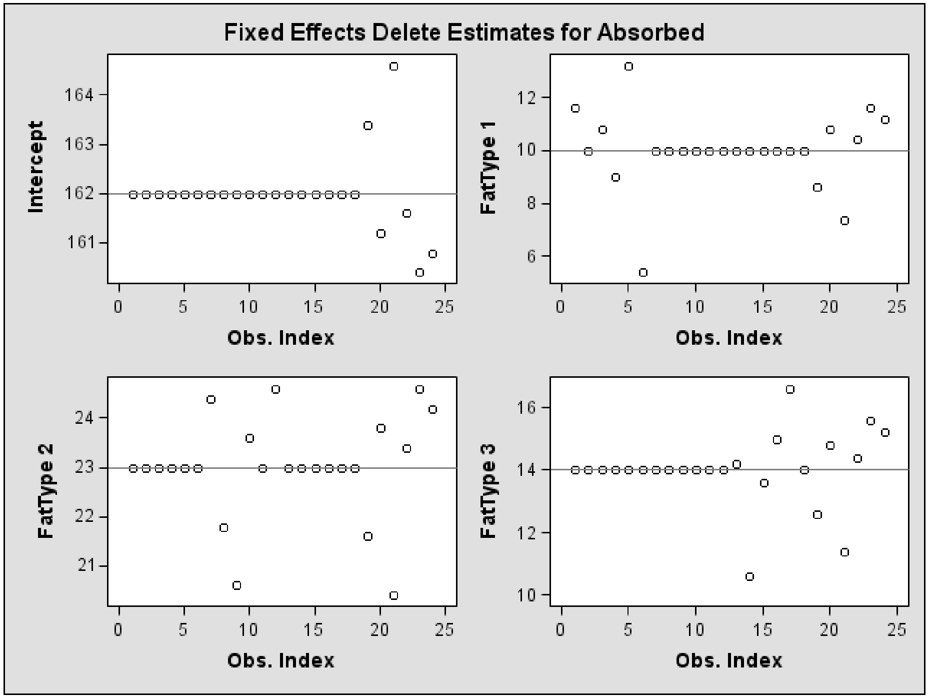
```
                                             Residual Residual Residual Residual
                Fat    Fat    Fat    Fat     FatType  FatType  FatType  FatType
Obs Intercept Type 1 Type 2 Type 3 Type 4       1        2        3        4

  1  162.00   11.600 23.000 14.000      0    203.30   60.400    97.60   67.600
  2  162.00   10.000 23.000 14.000      0    222.47   60.400    97.60   67.600
  3  162.00   10.800 23.000 14.000      0    217.68   60.400    97.60   67.600
  4  162.00    9.000 23.000 14.000      0    214.99   60.400    97.60   67.600
  5  162.00   13.200 23.000 14.000      0    145.70   60.400    97.60   67.600
  6  162.00    5.400 23.000 14.000      0     63.80   60.400    97.60   67.600
  7  162.00   10.000 24.400 14.000      0    178.00   60.795    97.60   67.600
  8  162.00   10.000 21.800 14.000      0    178.00   64.691    97.60   67.600
  9  162.00   10.000 20.600 14.000      0    178.00   32.296    97.60   67.600
 10  162.00   10.000 23.600 14.000      0    178.00   72.797    97.60   67.600
 11  162.00   10.000 23.000 14.000      0    178.00   75.490    97.60   67.600
 12  162.00   10.000 24.600 14.000      0    178.00   56.285    97.60   67.600
 13  162.00   10.000 23.000 14.200      0    178.00   60.400   121.68   67.600
 14  162.00   10.000 23.000 10.600      0    178.00   60.400    35.30   67.600
 15  162.00   10.000 23.000 13.600      0    178.00   60.400   120.79   67.600
 16  162.00   10.000 23.000 15.000      0    178.00   60.400   114.50   67.600
 17  162.00   10.000 23.000 16.600      0    178.00   60.400    71.30   67.600
 18  162.00   10.000 23.000 14.000      0    178.00   60.400   121.98   67.600
 19  163.40    8.600 21.600 12.600      0    178.00   60.400    97.60   69.799
 20  161.20   10.800 23.800 14.800      0    178.00   60.400    97.60   79.698
 21  164.60    7.400 20.400 11.400      0    178.00   60.400    97.60   33.800
 22  161.60   10.400 23.400 14.400      0    178.00   60.400    97.60   83.292
 23  160.40   11.600 24.600 15.600      0    178.00   60.400    97.60   65.299
 24  160.80   11.200 24.200 15.200      0    178.00   60.400    97.60   73.677
```

The estimate of the intercept is affected only when observations from the last group are removed. The estimate of the "FatType 1" effect reacts to removal of observations in the first and last group (Output 46.7.3).

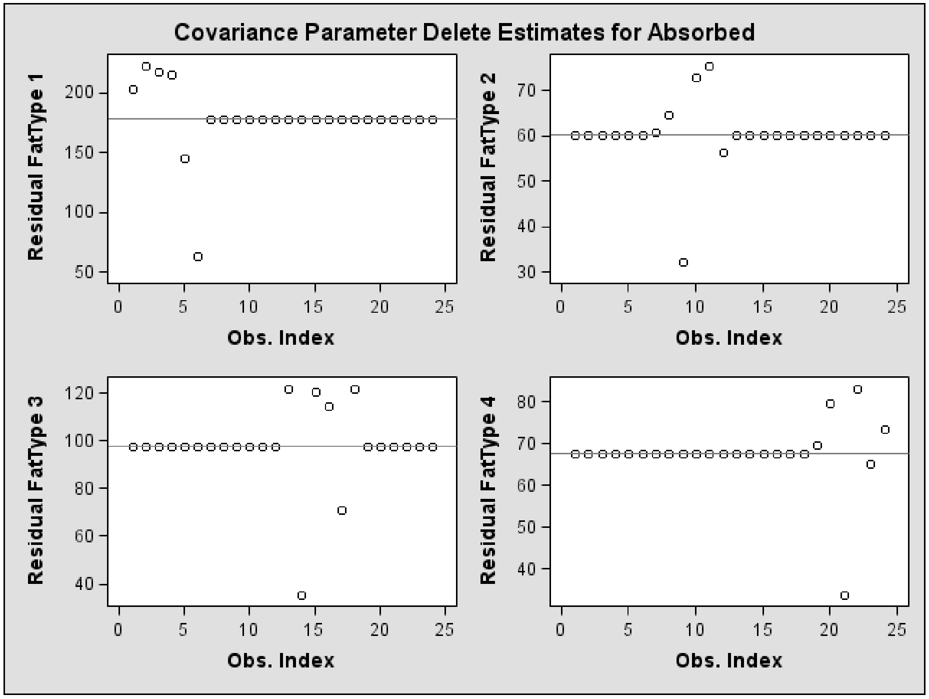
While observations can impact one or more fixed effects solutions in this model, they can only affect one covariance parameter, the variance in their group (Output 46.7.4). Observations 6, 9, 14, and 21, which are extreme in their group, reduce the group variance considerably.

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the "ODS Graphics" section on page 2757.

Output 46.7.3. Fixed Effects Delete Estimates (Experimental)



Output 46.7.4. Covariance Parameter Delete Estimates (Experimental)



Diagnostics related to residuals and predicted values are printed with the following statements.

```
proc print data=inf label;
   var observed predicted residual pressres
       student Rstudent;
run;
```

Observations 6, 9, 14, and 21 have large studentized residuals (Output 46.7.5). That the externally studentized residuals are much larger than the internally studentized residuals for these observations indicates that the variance estimate in the group shrinks when the observation is removed. Also important to note is that comparisons based on raw residuals in models with heterogeneous variance can be misleading. Observation 5, for example, has a larger residual but a smaller studentized residual than observation 21. The variance for the first fat type is much larger than the variance in the fourth group. A “large” residual is more “surprising” in the groups with small variance.

Output 46.7.5. Residual Diagnostics

```
                                                    Internally   Externally
       Observed   Predicted               PRESS    Studentized  Studentized
 Obs    Value       Mean      Residual   Residual    Residual     Residual

   1   164.000    172.000     -8.000     -9.600      -0.6569      -0.6146
   2   172.000    172.000      0.000      0.000       0.0000       0.0000
   3   168.000    172.000     -4.000     -4.800      -0.3284      -0.2970
   4   177.000    172.000      5.000      6.000       0.4105       0.3736
   5   156.000    172.000    -16.000    -19.200      -1.3137      -1.4521
   6   195.000    172.000     23.000     27.600       1.8885       3.1544
   7   178.000    185.000     -7.000     -8.400      -0.9867      -0.9835
   8   191.000    185.000      6.000      7.200       0.8457       0.8172
   9   197.000    185.000     12.000     14.400       1.6914       2.3131
  10   182.000    185.000     -3.000     -3.600      -0.4229      -0.3852
  11   185.000    185.000     -0.000     -0.000      -0.0000      -0.0000
  12   177.000    185.000     -8.000     -9.600      -1.1276      -1.1681
  13   175.000    176.000     -1.000     -1.200      -0.1109      -0.0993
  14   193.000    176.000     17.000     20.400       1.8850       3.1344
  15   178.000    176.000      2.000      2.400       0.2218       0.1993
  16   171.000    176.000     -5.000     -6.000      -0.5544      -0.5119
  17   163.000    176.000    -13.000    -15.600      -1.4415      -1.6865
  18   176.000    176.000     -0.000      0.000      -0.0000      -0.0000
  19   155.000    162.000     -7.000     -8.400      -0.9326      -0.9178
  20   166.000    162.000      4.000      4.800       0.5329       0.4908
  21   149.000    162.000    -13.000    -15.600      -1.7321      -2.4495
  22   164.000    162.000      2.000      2.400       0.2665       0.2401
  23   170.000    162.000      8.000      9.600       1.0659       1.0845
  24   168.000    162.000      6.000      7.200       0.7994       0.7657
```

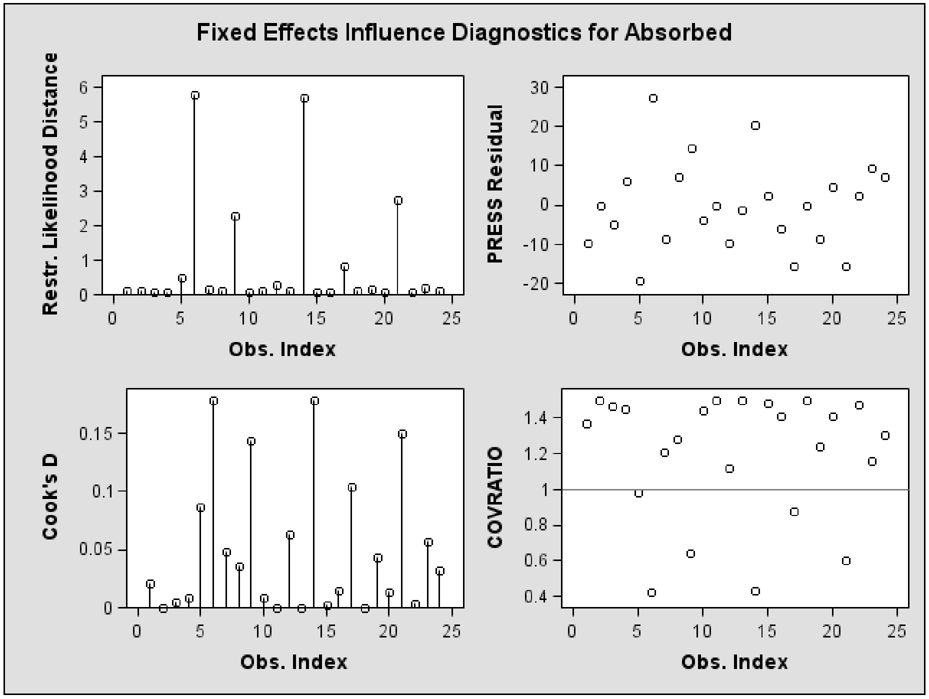
Diagnostics related to the fixed effects estimates, their precision, and the overall influence on the analysis (likelihood distance) are printed with the following statements.

```
proc print data=inf label;
   var leverage observed CookD DFFITS CovRatio RLD;
run;
```

Output 46.7.6. Restricted Likelihood Distance and Fixed Effects Diagnostics

```
                                                                       Restr.
                         Observed      Cook's                        Likelihood
 Obs      Leverage         Value          D       DFFITS    COVRATIO   Distance

   1      0.16667        164.000     0.02157   -0.27487     1.3706     0.1178
   2      0.16667        172.000     0.00000    0.00000     1.4998     0.1156
   3      0.16667        168.000     0.00539   -0.13282     1.4675     0.1124
   4      0.16667        177.000     0.00843    0.16706     1.4494     0.1117
   5      0.16667        156.000     0.08629   -0.64938     0.9822     0.5290
   6      0.16667        195.000     0.17831    1.41069     0.4301     5.8101
   7      0.16667        178.000     0.04868   -0.43982     1.2078     0.1935
   8      0.16667        191.000     0.03576    0.36546     1.2853     0.1451
   9      0.16667        197.000     0.14305    1.03446     0.6416     2.2909
  10      0.16667        182.000     0.00894   -0.17225     1.4463     0.1116
  11      0.16667        185.000     0.00000   -0.00000     1.4998     0.1156
  12      0.16667        177.000     0.06358   -0.52239     1.1183     0.2856
  13      0.16667        175.000     0.00061   -0.04441     1.4961     0.1151
  14      0.16667        193.000     0.17766    1.40175     0.4340     5.7044
  15      0.16667        178.000     0.00246    0.08915     1.4851     0.1139
  16      0.16667        171.000     0.01537   -0.22892     1.4078     0.1129
  17      0.16667        163.000     0.10389   -0.75423     0.8766     0.8433
  18      0.16667        176.000     0.00000    0.00000     1.4998     0.1156
  19      0.16667        155.000     0.04349   -0.41047     1.2390     0.1710
  20      0.16667        166.000     0.01420    0.21950     1.4148     0.1124
  21      0.16667        149.000     0.15000   -1.09545     0.6000     2.7343
  22      0.16667        164.000     0.00355    0.10736     1.4786     0.1133
  23      0.16667        170.000     0.05680    0.48500     1.1592     0.2383
  24      0.16667        168.000     0.03195    0.34245     1.3079     0.1353
```

Output 46.7.7. REML Distance and Fixed Effects Diagnostics (Experimental)



Scanning the restricted likelihood distances, observations 6, 9, 14, and 21 clearly displace the REML solution more than any other observations (Output 46.7.6, Output 46.7.7). These observations are also associated with large values for Cook's D and values of COVRATIO far less than one. The latter indicates that the fixed effects are estimated more precisely when these observations are removed from the analysis.

The same conclusions hold for the covariance parameter estimates.

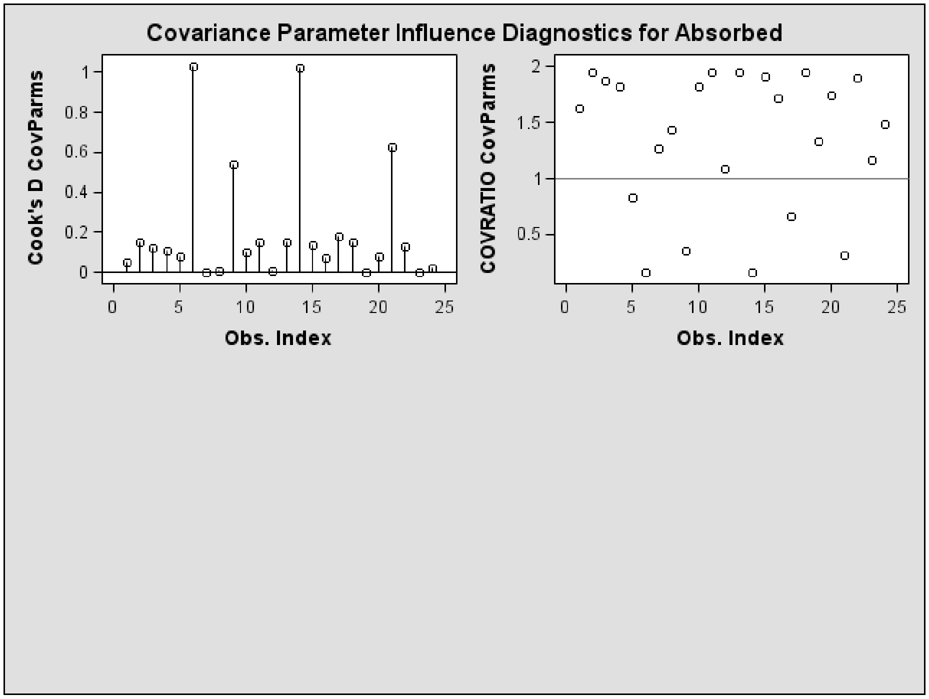
```
proc print data=inf label;
  var iter CookDCP CovRatioCP;
run;
```

Observations 6, 9, 14, and 21 change the estimates and their precision considerably (Output 46.7.8, Output 46.7.9). All iterative updates converged within at most four iterations.

Output 46.7.8. Covariance Parameter Diagnostics

```
                                   Cook's D    COVRATIO
               Obs    Iterations   CovParms    CovParms

                 1         3       0.05050      1.6306
                 2         3       0.15603      1.9520
                 3         3       0.12426      1.8692
                 4         3       0.10796      1.8233
                 5         4       0.08232      0.8375
                 6         4       1.02909      0.1606
                 7         1       0.00011      1.2662
                 8         2       0.01262      1.4335
                 9         3       0.54126      0.3573
                10         3       0.10531      1.8156
                11         3       0.15603      1.9520
                12         2       0.01160      1.0849
                13         3       0.15223      1.9425
                14         4       1.01865      0.1635
                15         3       0.14111      1.9141
                16         3       0.07494      1.7203
                17         3       0.18154      0.6671
                18         3       0.15603      1.9520
                19         2       0.00265      1.3326
                20         3       0.08008      1.7374
                21         1       0.62500      0.3125
                22         3       0.13472      1.8974
                23         2       0.00290      1.1663
                24         2       0.02020      1.4839
```

Output 46.7.9. Covariance Parameter Diagnostics (Experimental)



Example 46.8. Influence Analysis for Repeated Measures Data (Experimental)

This example revisits the repeated measures data analyzed in Example 46.2. Recall that the data consist of growth measurements at ages 8, 10, 12, and 14 for 11 girls and 16 boys. The model being fit contains fixed effects for Gender, Age and their interaction.

The earlier analysis of these data indicated some unusual observations in this data set. Because of the clustered data structure it is of interest to study the influence of clusters (children) on the analysis rather than the influence of individual observations. A cluster comprises the repeated measurements for each child.

The repeated measures are first modeled with an unstructured within-child variance-covariance matrix. A residual variance is not profiled in this model. A noniterative influence analysis will update the fixed effects only. The statements

```
proc mixed data=pr method=ml;
  class person gender;
  model y = gender age gender*age /
             influence(effect=person);
  repeated / type=un subject=person;
  ods select influence;
run;
```

request this noniterative maximum likelihood analysis and produce Output 46.8.1. Each observation in the "Influence Diagnostics" table represents the removal of four observations. The subjects 10, 15, and 24 have the greatest impact on the fixed effects (Cook's D), and subject 10 and 21 have large PRESS statistics. The twenty-first child has a large PRESS statistic, and its D statistic is not that extreme. This is an indication that the model fits rather poorly for this child, whether it is part of the data or not.

Output 46.8.1. Default Influence Statistics in Noniterative Analysis

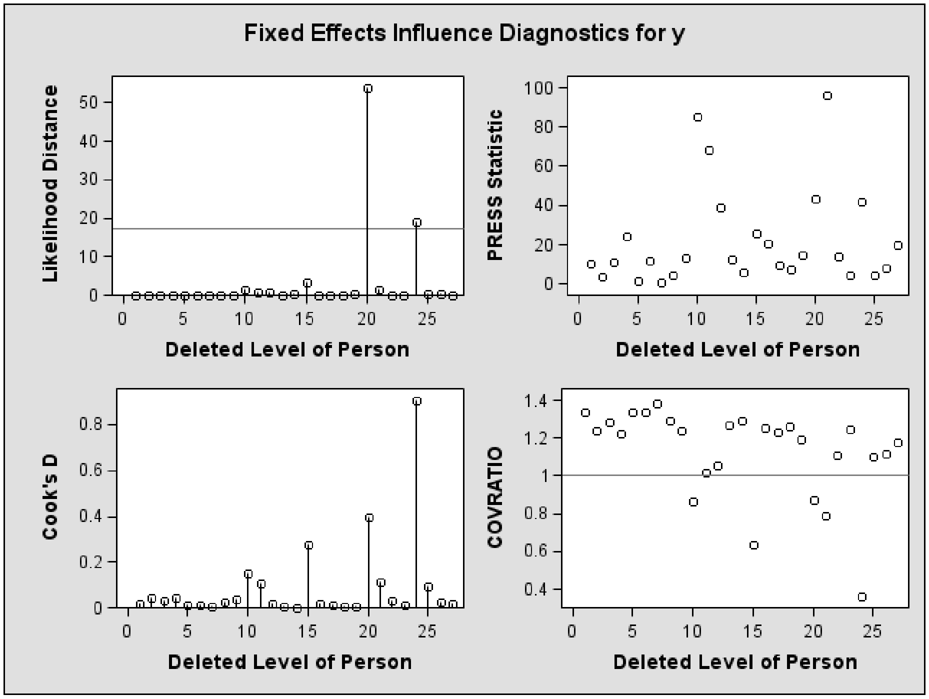
```
                         The Mixed Procedure

               Influence Diagnostics for Levels of Person

                      Number of
                    Observations          PRESS      Cook's
          Person      in Level        Statistic           D

             1             4            10.1716     0.01539
             2             4             3.8187     0.03988
             3             4            10.8448     0.02891
             4             4            24.0339     0.04515
             5             4             1.6900     0.01613
             6             4            11.8592     0.01634
             7             4             1.1887     0.00521
             8             4             4.6717     0.02742
             9             4            13.4244     0.03949
            10             4            85.1195     0.13848
            11             4            67.9397     0.09728
            12             4            40.6467     0.04438
            13             4            13.0304     0.00924
            14             4             6.1712     0.00411
            15             4            24.5702     0.12727
            16             4            20.5266     0.01026
            17             4             9.9917     0.01526
            18             4             7.9355     0.01070
            19             4            15.5955     0.01982
            20             4            42.6845     0.01973
            21             4            95.3282     0.10075
            22             4            13.9649     0.03778
            23             4             4.9656     0.01245
            24             4            37.2494     0.15094
            25             4             4.3756     0.03375
            26             4             8.1448     0.03470
            27             4            20.2913     0.02523
```

The previous analysis does not take into account the effect on the covariance parameters when a subject is removed from the analysis. If you also update the covariance parameters, the impact of observations on these can amplify or allay their effect on the fixed effects. To assess the overall influence of subjects on the analysis and to compute separate statistics for the fixed effects and covariance parameters, an iterative analysis is obtained by adding the INFLUENCE suboption ITER=.

```
ods html;
ods graphics on;

proc mixed data=pr method=ml;
  class person gender;
  model y = gender age gender*age /
                     influence(effect=person iter=5);
  repeated / type=un subject=person;
run;

ods graphics off;
ods html close;
```

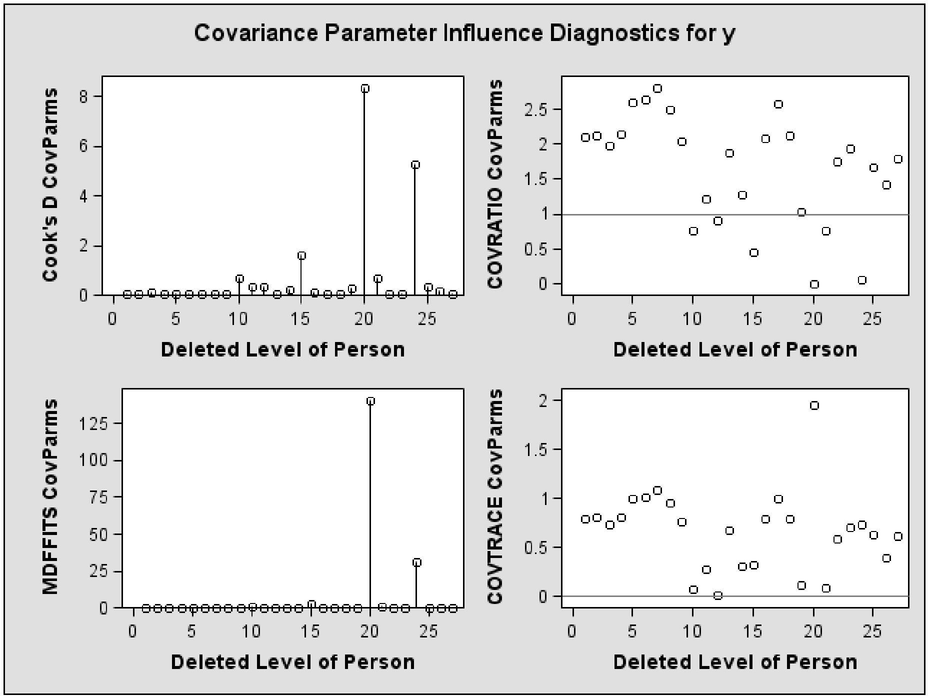
The number of additional iterations following removal of the observations for a particular subject is limited to five. Graphical displays of influence diagnostics are requested by specifying the experimental ODS GRAPHICS statement. The MIXED procedure produces a plot of influence statistics that pertain to overall influence and the fixed effects (Output 46.8.2) and a second panel that shows the influence on the covariance parameters (Output 46.8.3). For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the MIXED procedure, see the “ODS Graphics” section on page 2757.

Output 46.8.2. Overall and Fixed Effects Diagnostics (Experimental)



As judged by the restricted likelihood distance in Output 46.8.2, subjects 20 and 24 clearly have the most influence on the overall analysis. The reference line in this plot is the 75th percentile of a chi-square distribution with degrees of freedom equal to the number of parameters being updated (14). Comparing the D statistics for fixed effects and covariance parameters, subject 20 has a dramatic effect on the estimates of variances and covariances. This subject also affects the precision of the covariance parameter estimates more than any other subject in Output 46.8.3 (COVRATIO near 0, large COVTRACE, MDFFITS large compared to Cook's D).

Output 46.8.3. Covariance Parameter Diagnostics (Experimental)



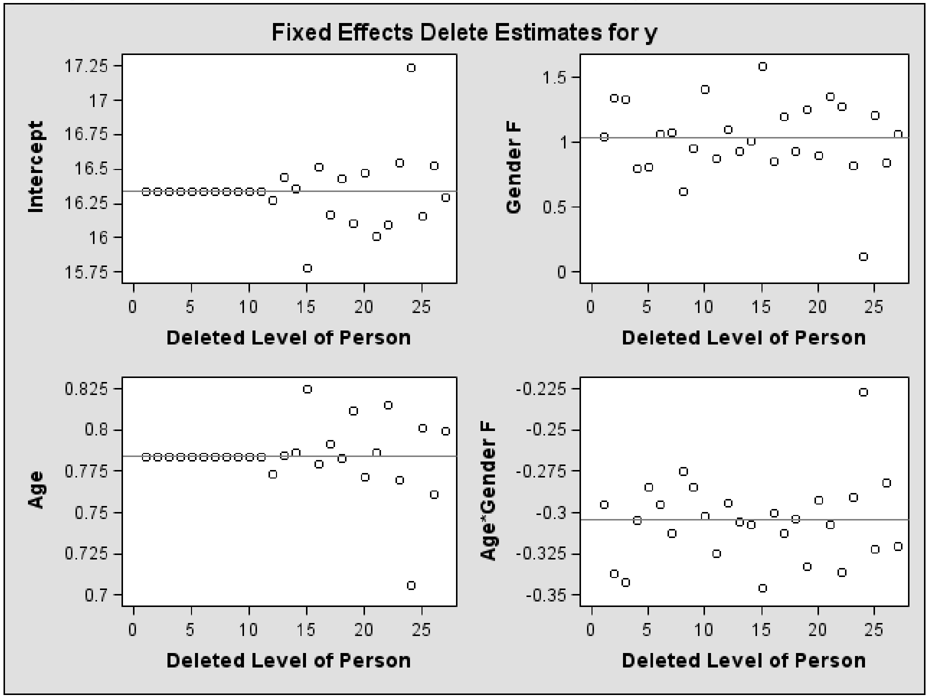
The child who exerts the greatest influence on the fixed effects is subject 24. Maybe surprisingly, this subject impacts the variance-covariance matrix of the fixed effects more than subject 20 (small COVRATIO in Output 46.8.3).

The final model investigated for these data is a random coefficient model as in Stram and Lee (1994) with random effects for the intercept and age effect. The following statements examine the estimates for fixed effects and the entries of the unstructured 2 × 2 variance matrix of the random coefficients graphically.

```
ods html;
ods graphics on;

proc mixed data=pr method=ml;
   class person gender;
   model y = gender age gender*age /
         influence(iter=5 effect=person est);
   random intercept age / type=un subject=person;
   ods select InfluenceEstParmPanel1
              InfluenceEstCovPPanel1;
run;

ods graphics off;
ods html close;
```

In Output 46.8.4 the graphs on the left-hand side of the panel represent the intercept and slope estimate for boys; the graphs on the right-hand side represent the difference in intercept and slope between boys and girls. Removing any one of the first eleven children, who are girls, does not alter the intercept or slope in the group of boys. The difference in these parameters between boys and girls is altered by the removal of any child. Subject 24 changes the fixed effects considerably, subject 20 much less so. This is consistent with findings based on the unstructured variance-covariance model (Output 46.8.2).

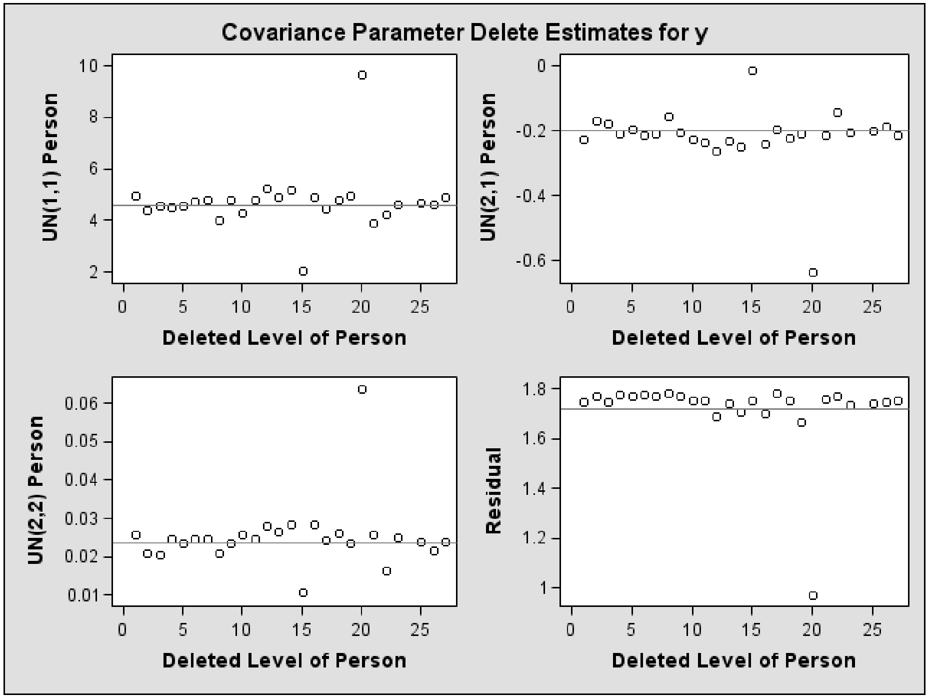
Output 46.8.4. Fixed Effects Delete Estimates (Experimental)



The covariance parameter estimates in Output 46.8.5 show several important features.

- The panels do not contain information on subject 24. Estimation of the **G** matrix following removal of that child did not yield a positive definite matrix. As a consequence, covariance parameter diagnostics are not produced for this subject.
- Subject 20 has great impact on the four covariance parameters. Removing this child from the analysis increases the variance of the random intercept and random slope and reduces the residual variance by almost 80%. The repeated measurements of this child exhibit an up-and-down behavior.
- The variance of the random intercept and slope are reduced when child 15 is removed from the analysis. This child's growth measurements oscillate about 27.0 from age 10 on.

Output 46.8.5. Covariance Parameter Delete Estimates (Experimental)



Examining observed and residual values by levels of classification variables is also a useful tool to diagnose the adequacy of the model and unusual observations. Box plots for effects in the model that comprise a single classification variable are requested with the experimental ODS GRAPHICS statement and the experimental BOXPLOT option of the PROC MIXED statement. Since this includes SUBJECT= and GROUP= effects, the statements

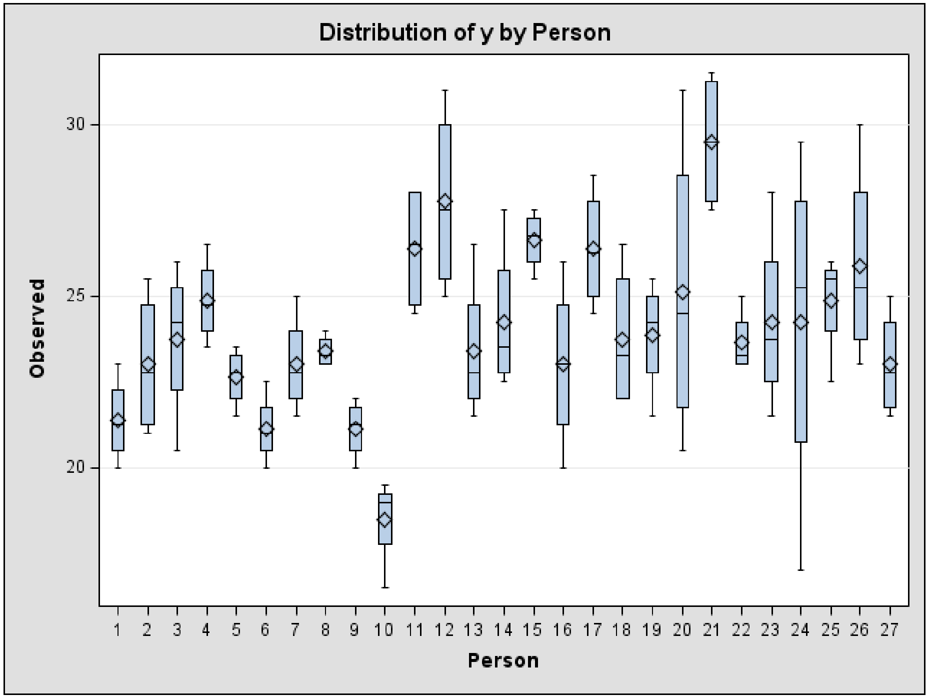
```
ods html;
ods graphics on;

proc mixed data=pr method=ml boxplot;
   class person gender;
   model y = gender age gender*age;
   random intercept age / type=un subject=person;
run;

ods graphics off;
ods html close;
```

produce six box plots in the following order: Y by Gender, residual by Gender, Y by Person, residual by Person, and conditional residuals by Gender and Person. The plots for the Person effect are shown in Output 46.8.6 –Output 46.8.8.

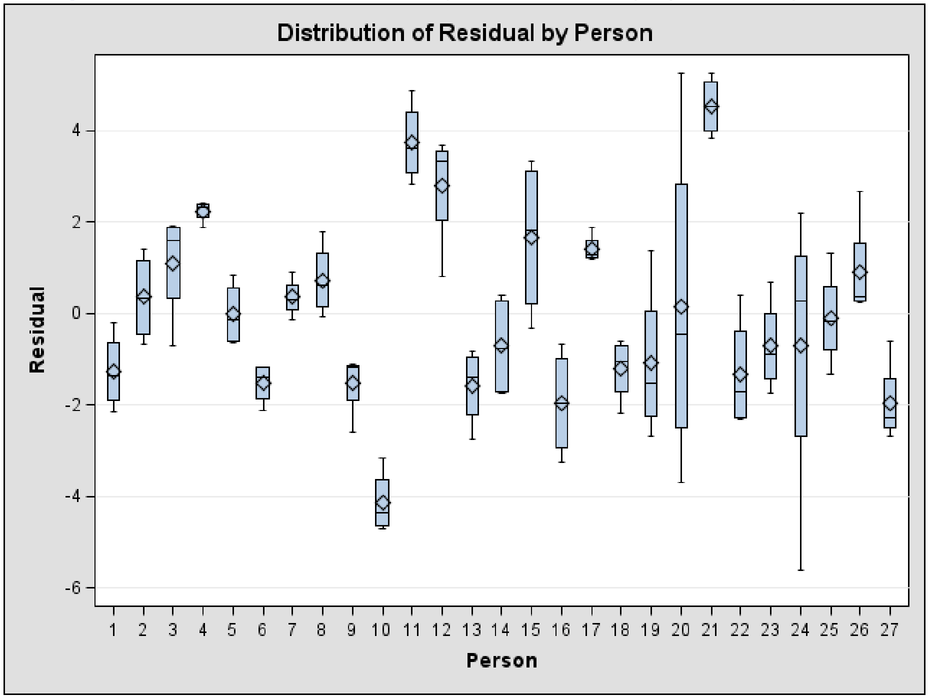
Box plots of the observed values show the variation within and between children clearly. The group of girls (subjects 1–11) is distinguishable from the group of boys by somewhat lesser average growth and lesser within-child variation (Output 46.8.6). After adjusting for overall (population-averaged) gender and age effects, the residual within-child variation is reduced but substantial differences in the means remain

(Output 46.8.7). If child-specific inferences are desired, a model accounting for only Gender, Age, and Gender*Age effects is not adequate for these data.

Output 46.8.6. Distribution of Observed Values (Experimental)

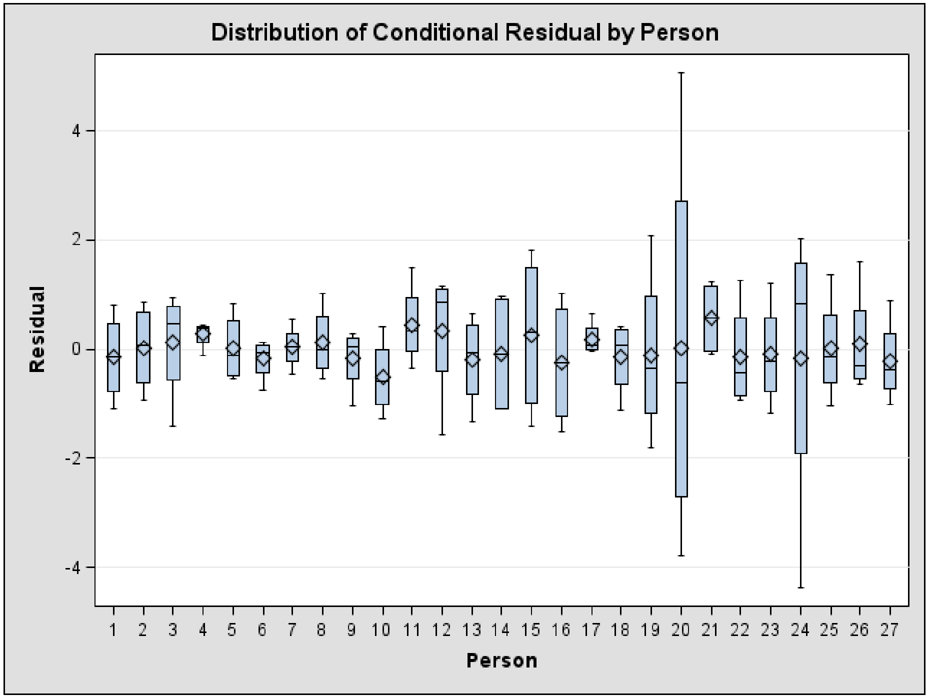


Output 46.8.7. Distribution of Marginal Residuals (Experimental)



The conditional residuals incorporate the EBLUPs for each child and enable you to examine whether the subject-specific model is adequate (Output 46.8.8). By using each child "as its own control" the residuals are now centered near zero. Subjects 20 and 24 stand out as unusual in all three sets of box plots.

Output 46.8.8. Distribution of Conditional Residuals (Experimental)



Example 46.9. Examining Individual Test Components

The LCOMPONENTS option of the MODEL statement enables you to perform single degree of freedom tests for individual rows of the **L** matrix. Such tests are useful to identify interaction patterns. In a balanced layout, Type 3 components of **L** associated with A*B interactions correspond to simple contrasts of cell mean differences.

The first example revisits the data from the split-plot design by Stroup (1989a) that was analyzed in Example 46.1 on page 2777. Recall that variables A and B represent the whole-plot and subplot factors, respectively.

```
proc mixed data=sp;
   class a b block ;
   model y = a b a*b  / LComponents e3;
   random block a*block;
run;
```

The MIXED procedure constructs a separate **L** matrix for each of the three fixed effects components. The matrices are displayed in Output 46.9.1. The test for fixed effects are shown in Output 46.9.2.

Output 46.9.1. Coefficients of Type 3 Estimable Functions

```
          The Mixed Procedure

       Type 3 Coefficients for A

Effect       A    B      Row1      Row2

Intercept
A            1              1
A            2                        1
A            3             -1        -1
B                 1
B                 2
A*B          1    1       0.5
A*B          1    2       0.5
A*B          2    1                 0.5
A*B          2    2                 0.5
A*B          3    1      -0.5      -0.5
A*B          3    2      -0.5      -0.5


       Type 3 Coefficients for B

     Effect       A    B      Row1

     Intercept
     A            1
     A            2
     A            3
     B                 1         1
     B                 2        -1
     A*B          1    1    0.3333
     A*B          1    2    -0.333
     A*B          2    1    0.3333
     A*B          2    2    -0.333
     A*B          3    1    0.3333
     A*B          3    2    -0.333


      Type 3 Coefficients for A*B

Effect       A    B      Row1      Row2

Intercept
A            1
A            2
A            3
B                 1
B                 2
A*B          1    1         1
A*B          1    2        -1
A*B          2    1                   1
A*B          2    2                  -1
A*B          3    1        -1        -1
A*B          3    2         1         1
```

Output 46.9.2. Type 3 Tests in Split-Plot Example

```
                             The Mixed Procedure

                         Type 3 Tests of Fixed Effects

                               Num      Den
                Effect          DF       DF    F Value    Pr > F

                A                2        6       4.07    0.0764
                B                1        9      19.39    0.0017
                A*B              2        9       4.02    0.0566
```

If $\mu_{i.}$ denotes a whole-plot main effect mean, $\mu_{.j}$ denotes a subplot main effect mean, and μ_{ij} denotes a cell mean, the five components shown in Output 46.9.3 correspond to tests of

- $H_0 : \mu_{1.} = \mu_{2.}$
- $H_0 : \mu_{2.} = \mu_{3.}$
- $H_0 : \mu_{.1} = \mu_{.2}$
- $H_0 : \mu_{11} - \mu_{12} = \mu_{31} - \mu_{32}$
- $H_0 : \mu_{21} - \mu_{22} = \mu_{31} - \mu_{32}$

Output 46.9.3. Type 3 LComponents Table

```
                             The Mixed Procedure

                 L Components of Type 3 Tests of Fixed Effects

                    L                 Standard
 Effect         Index    Estimate        Error      DF    t Value    Pr > |t|

 A                  1      7.1250       3.1672       6       2.25      0.0655
 A                  2      8.3750       3.1672       6       2.64      0.0383
 B                  1      5.5000       1.2491       9       4.40      0.0017
 A*B                1      7.7500       3.0596       9       2.53      0.0321
 A*B                2      7.2500       3.0596       9       2.37      0.0419
```

The first three components are comparisons of marginal means. The fourth component compares the effect of factor B at the first whole-plot level against the effect of B at the third whole-plot level. Finally, the last component tests whether the factor B effect changes between the second and third whole-plot level.

The Type 3 component tests can also be produced with corresponding ESTIMATE statements.

```
proc mixed data=sp;
   class a b block ;
   model y = a b a*b;
   random block a*block;
   estimate ’a    1’ a 1 0 -1;
   estimate ’a    2’ a 0 1 -1;
   estimate ’b    1’ b   1 -1;
   estimate ’a*b  1’ a*b 1 -1 0  0 -1 1;
   estimate ’a*b  2’ a*b 0  0 1 -1 -1 1;
   ods select estimates;
run;
```

```
                              The Mixed Procedure

                                   Estimates

                                 Standard
      Label      Estimate           Error      DF    t Value    Pr > |t|

      a    1       7.1250          3.1672       6       2.25      0.0655
      a    2       8.3750          3.1672       6       2.64      0.0383
      b    1       5.5000          1.2491       9       4.40      0.0017
      a*b  1       7.7500          3.0596       9       2.53      0.0321
      a*b  2       7.2500          3.0596       9       2.37      0.0419
```

A second useful application of the LCOMPONENT option is in polynomial models where Type 1 tests are often used to test the entry of model terms sequentially. The SOLUTION option of the MODEL statement displays the regression coefficients that correspond to a Type 3 analysis, that is,

- the coefficients represent the partial coefficients you would get by adding the regressor variable last in a model containing all other effects
- the tests are identical to those in the “Type 3 Tests of Fixed Effects” table

Consider the following data set and the fit of a third order polynomial regression model.

```
data polynomial;
  do x=1 to 20; input y@@; output; end;
  datalines;
1.092   1.758   1.997   3.154   3.880
3.810   4.921   4.573   6.029   6.032
6.291   7.151   7.154   6.469   7.137
6.374   5.860   4.866   4.155   2.711
;

proc mixed data=polynomial;
  model y = x x*x x*x*x / s lcomponents htype=1;
run;
```

The *t*-tests displayed in the “Solution for Fixed Effects” table are Type 3 tests, sometimes referred to as partial tests. They measure the contribution of a regressor in the presence of all other regressor variables in the model.

```
                              The Mixed Procedure

                           Solution for Fixed Effects

                                  Standard
      Effect       Estimate          Error      DF    t Value    Pr > |t|

      Intercept      0.7837         0.3545      16       2.21      0.0420
      x              0.3726         0.1426      16       2.61      0.0189
      x*x           0.04756        0.01558      16       3.05      0.0076
      x*x*x        -0.00306       0.000489      16      -6.27      <.0001
```

The Type 3 L components are identical to the tests in the “Solutions for Fixed Effects” table. The Type 1 table yields

- sequential (Type 1) tests of regression variables which test the significance of a regressor given all other variables preceding it in the model list
- the regression coefficients for sequential submodels

```
                              The Mixed Procedure

                  L Components of Type 1 Tests of Fixed Effects

                    L                  Standard
 Effect         Index    Estimate         Error      DF    t Value    Pr > |t|

 x                  1      0.1763       0.01259      16      14.01      <.0001
 x*x                1    -0.04886      0.002449      16     -19.95      <.0001
 x*x*x              1    -0.00306      0.000489      16      -6.27      <.0001


                  L Components of Type 3 Tests of Fixed Effects

                    L                  Standard
 Effect         Index    Estimate         Error      DF    t Value    Pr > |t|

 x                  1      0.3726        0.1426      16       2.61      0.0189
 x*x                1     0.04756       0.01558      16       3.05      0.0076
 x*x*x              1    -0.00306      0.000489      16      -6.27      <.0001
```

The estimate of 0.1763 is the regression coefficient in a simple linear regression of Y on X. The estimate of -0.04886 is the partial coefficient for the quadratic term when it is added to a model containing only a linear component. Similarly, the value -0.00306 is the partial coefficient for the cubic term when it is added to a model containing a linear and quadratic component. The last Type 1 component is always identical to the corresponding Type 3 component.

References

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC–19, 716–723.

Allen, D.M. (1974), "The Relationship between Variable Selection and Data Augmentation and a Method of Prediction," *Technometrics*, 16, 125–127.

Bates, D.M. and Watts, D.G. (1988), *Nonlinear Regression Analysis and Its Applications*, New York: John Wiley & Sons, Inc.

Beckman, R.J., Nachtsheim, C.J., and Cook, D.R. (1987), "Diagnostics for Mixed-Model Analysis of Variance," *Technometrics*, 29, 413–426

Belsley, D.A., Kuh, E., and Welsch, R.E. (1980), *Regression Diagnostics; Identifying Influential Data and Sources of Collinearity*, New York: John Wiley & Sons.

Box, G.E.P. and Tiao, G.C. (1973), *Bayesian Inference in Statistical Analysis*, Wiley Classics Library Edition Published 1992, New York: John Wiley & Sons, Inc.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.

Brown, H. and Prescott, R. (1999), *Applied Mixed Models in Medicine*, New York: John Wiley & Sons, Inc.

Burdick, R.K. and Graybill, F.A. (1992), *Confidence Intervals on Variance Components*, New York: Marcel Dekker.

Burnham, K.P. and Anderson, D.R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Carlin, B.P. and Louis, T.A. (1996), *Bayes and Empirical Bayes Methods for Data Analysis*, London: Chapman and Hall.

Carroll, R.J. and Ruppert, D. (1988), *Transformation and Weighting in Regression*, London: Chapman and Hall.

Christensen, R., Pearson, L.M., and Johnson, W. (1992), "Case-deletion Diagnostics for Mixed Models," *Technometrics*, 34, 38–45.

Cook, R.D. (1977), "Detection of Influential Observations in Linear Regression," *Technometrics*, 19, 15–18.

Cook, R.D. (1979), "Influential Observations in Linear Regression," *Journal of the American Statistical Association*, 74, 169–174.

Cook, R.D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman and Hall.

Cressie, N. (1991), *Statistics for Spatial Data*, New York: John Wiley & Sons, Inc.

Crowder, M.J. and Hand, D.J. (1990), *Analysis of Repeated Measures*, New York: Chapman and Hall.

Dempster, A.P., Laird, N.M., and Rubin, D.B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, Ser. B., 39, 1–38.

Diggle, P.J. (1988), "An Approach to the Analysis of Repeated Measurements," *Biometrics*, 44, 959–971.

Diggle, P.J., Liang, K.Y., and Zeger, S.L. (1994), *Analysis of Longitudinal Data*, Oxford: Clarendon Press.

Edwards, D. and Berry, J.J. (1987), "The Efficiency of Simulation-based Multiple Comparisons," *Biometrics*, 43, 913–928.

Everitt, B.S. (1995), "The Analysis of Repeated Measures: A Practical Review with Examples," *The Statistician*, 44, 113–135.

Fai, A.H.T. and Cornelius, P.L. (1996), "Approximate *F*-tests of Multiple Degree of Freedom Hypotheses in Generalized Least Squares Analyses of Unbalanced Split-plot Experiments," *Journal of Statistical Computation and Simulation*, 54, 363–378.

Federer, W.T. and Wolfinger, R.D. (1998), "SAS Code for Recovering Intereffect Information in Experiments with Incomplete Block and Lattice Rectangle Designs," *Agronomy Journal*, 90, 545–551.

Fuller, W.A. (1976), *Introduction to Statistical Time Series*, New York: John Wiley & Sons, Inc.

Fuller, W.A. and Battese, G.E. (1973), "Transformations for Estimation of Linear Models with Nested Error Structure," *Journal of the American Statistical Association*, 68, 626–632.

Galecki, A.T. (1994), "General Class of Covariance Structures for Two or More Repeated Factors in Longitudinal Data Analysis," *Communications in Statistics–Theory and Methods*, 23(11), 3105–3119.

Gelfand, A.E., Hills, S.E., Racine-Poon, A., and Smith, A.F.M. (1990), "Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling," *Journal of the American Statistical Association*, 85, 972–985.

Ghosh, M. (1992), Discussion of Schervish, M., "Bayesian Analysis of Linear Models," *Bayesian Statistics 4*, eds. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, Oxford: University Press, 432–433.

Giesbrecht, F.G. (1989), "A General Structure for the Class of Mixed Linear Models," *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 183–201.

Giesbrecht, F.G. and Burns, J.C. (1985), "Two-Stage Analysis Based on a Mixed Model: Large-sample Asymptotic Theory and Small-Sample Simulation Results," *Biometrics*, 41, 477–486.

Golub, G.H. and Van Loan, C.F. (1989), *Matrix Computations*, Second Edition, Baltimore: Johns Hopkins University Press.

Goodnight, J.H. (1978), SAS Technical Report R-101, *Tests of Hypotheses in Fixed-Effects Linear Models*, Cary, NC: SAS Institute Inc.

Goodnight, J.H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.

Goodnight, J.H. and Hemmerle, W.J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Gregoire, T.G., Schabenberger, O., and Barrett, J.P. (1995), "Linear Modelling of Irregularly Spaced, Unbalanced, Longitudinal Data from Permanent Plot Measurements," *Canadian Journal of Forest Research*, 25, 137–156.

Handcock, M.S. and Stein, M.L. (1993), "A Bayesian Analysis of Kriging," *Technometrics,* 35(4), 403–410

Handcock, M.S. and Wallis, J.R. (1994), "An Approach to Statistical Spatial-temporal Modeling of Meteorological Fields (with Discussion)," *Journal of the American Statistical Association*, 89, 368–390.

Hanks, R.J., Sisson, D.V., Hurst, R.L, and Hubbard K.G. (1980), "Statistical Analysis of Results from Irrigation Experiments Using the Line-Source Sprinkler System," *Soil Science Society American Journal*, 44, 886–888.

Hannan, E.J. and Quinn, B.G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society,* Series B, 41, 190–195.

Hartley, H.O. and Rao, J.N.K. (1967), "Maximum-Likelihood Estimation for the Mixed Analysis of Variance Model," *Biometrika*, 54, 93–108.

Harville, D.A. (1977), "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems," *Journal of the American Statistical Association*, 72, 320–338.

Harville, D.A. (1988), "Mixed-Model Methodology: Theoretical Justifications and Future Directions," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 41–49.

Harville, D.A. (1990), "BLUP (Best Linear Unbiased Prediction), and Beyond," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, Springer-Verlag, 239–276.

Harville, D.A. and Jeske, D.R. (1992), "Mean Squared Error of Estimation or Prediction Under a General Linear Model," *Journal of the American Statistical Association*, 87, 724–731.

Helms, R.W. and Edwards, L. (1991), "Analysis of Irregular and/or Incomplete from Crossover Designs Using Mixed Models with Linear Covariance Structure," *Proceedings of the Biopharmaceutical Section*, American Statistical Association, Atlanta.

Hemmerle, W.J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics,* 15, 819–831.

Henderson, C.R. (1984), *Applications of Linear Models in Animal Breeding*, University of Guelph.

Henderson, C.R. (1990), "Statistical Method in Animal Improvement: Historical Overview," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, New York: Springer-Verlag, 1–14.

Hsu, J.C. (1992), "The Factor Analytic Approach to Simultaneous Inference in the General Linear Model," *Journal of Computational and Graphical Statistics*, 1, 151–168.

Huber, P.J. (1967), "The Behavior of Maximum Likelihood Estimates Under Nonstandard Conditions," *Proc. Fifth Berkeley Symp. Math. Statist. Prob.*, 1, 221–233.

Hurtado, G.I.H. (1993), *Detection of Influential Observations in Linear Mixed Models*, Ph.D. dissertation, Department of Statistics, North Carolina State University, Raleigh, NC

Hurvich, C.M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Huynh, H. and Feldt, L.S. (1970), "Conditions under which Mean Square Ratios in Repeated Measurements Designs have Exact *F*-Distributions," *Journal of the American Statistical Association*, 65, 1582–1589.

Jennrich, R.I. and Schluchter, M.D. (1986), "Unbalanced Repeated-Measures Models with Structured Covariance Matrices," *Biometrics*, 42, 805–820.

Johnson, D.E., Chaudhuri, U.N., and Kanemasu, E.T. (1983), "Statistical Analysis of Line-Source Sprinkler Irrigation Experiments and Other Nonrandomized Experiments Using Multivariate Methods," *Soil Science Society American Journal*, 47, 309–312.

Jones, R.H. and Boadi-Boateng, F. (1991), "Unequally Spaced Longitudinal Data with AR(1) Serial Correlation," *Biometrics*, 47, 161–175.

Kackar, R.N. and Harville, D.A. (1984), "Approximations for Standard Errors of Estimators of Fixed and Random Effects in Mixed Linear Models," *Journal of the American Statistical Association*, 79, 853–862.

Kass, R.E. and Steffey, D. (1989), "Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models)," *Journal of the American Statistical Association*, 84, 717–726.

Kenward, M.G. (1987), "A Method for Comparing Profiles of Repeated Measurements," *Applied Statistics*, 36, 296–308.

Kenward, M.G. and Roger, J.H. (1997), "Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood," *Biometrics*, 53, 983–997.

Keselman, H.J., Algina, J., Kowalchuk, R.K., and Wolfinger, R.D. (1998), "A Comparison of Two Approaches for Selecting Covariance Structures in the Analysis of Repeated Measures," *Communications in Statistics–Computation and Simulation*, 27(3), 591–604.

Keselman, H.J., Algina, J., Kowalchuk, R.K., and Wolfinger, R.D. (1999). "A Comparison of Recent Approaches to the Analysis of Repeated Measurements," *British Journal of Mathematical and Statistical Psychology,* 52, 63–78.

Kramer, C.Y. (1956), "Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications," *Biometrics*, 12, 309–310.

Laird, N.M. and Ware, J.H. (1982), "Random-Effects Models for Longitudinal Data," *Biometrics*, 38, 963–974.

Laird, N.M., Lange, N., and Stram, D. (1987), "Maximum Likelihood Computations with Repeated Measures: Application of the EM Algorithm," *Journal of the American Statistical Association*, 82, 97–105.

LaMotte, L.R. (1973), "Quadratic Estimation of Variance Components," *Biometrics*, 29, 311–330.

Liang, K.Y. and Zeger, S.L. (1986), "Longitudinal Data Analysis using Generalized Linear Models," *Biometrika*, 73, 13–22.

Lindsey, J.K. (1993), *Models for Repeated Measurements*, Oxford: Clarendon Press.

Lindstrom, M.J. and Bates, D.M. (1988), "Newton-Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-Measures Data," *Journal of the American Statistical Association*, 83, 1014–1022.

Littell, R.C., Milliken, G.A., Stroup, W.W., and Wolfinger, R.D. (1996), *SAS System for Mixed Models,* Cary, NC: SAS Institute Inc.

Little, R.J.A. (1995), "Modeling the Drop-Out Mechanism in Repeated-Measures Studies," *Journal of the American Statistical Association*, 90, 1112–1121.

Louis, T.A. (1988), "General Methods for Analyzing Repeated Measures," *Statistics in Medicine*, 7, 29–45.

Macchiavelli, R.E. and Arnold, S.F. (1994), "Variable Order Ante-Dependence Models," *Communications in Statistics–Theory and Methods*, 23(9), 2683–2699.

Marx, D. and Thompson, K. (1987), "Practical Aspects of Agricultural Kriging," Bulletin 903, Arkansas Agricultural Experiment Station, Fayetteville.

Matérn, B. (1986), *Spatial Variation, Second ed.*, Lecture Notes in Statistics, New York: Springer-Verlag.

McKeon, J.J. (1974), "F Approximations to the Distribution of Hotelling's T_0^2," *Biometrika*, 61, 381–383.

McLean, R.A. and Sanders, W.L. (1988), "Approximating Degrees of Freedom for Standard Errors in Mixed Linear Models," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 50–59.

McLean, R.A., Sanders, W.L., and Stroup, W.W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Milliken, G.A. and Johnson, D.E. (1992), *Analysis of Messy Data, Volume 1: Designed Experiments,* New York: Chapman and Hall.

Murray, D.M. (1998), *Design and Analysis of Group-Randomized Trials,* New York: Oxford University Press.

Myers, R.H. (1990), *Classical and Modern Regression With Applications. 2nd. ed.*, Belmont, CA: PWS-Kent.

Obenchain, R.L. (1990), *STABLSIM.EXE*, Version 9010, Eli Lilly and Company, Indianapolis, Indiana, unpublished C code.

Patel, H.I. (1991), "Analysis of Incomplete Data from a Clinical Trial with Repeated Measurements," *Biometrika*, 78, 609–619.

Patterson, H.D. and Thompson, R. (1971), "Recovery of Inter-Block Information when Block Sizes are Unequal," *Biometrika*, 58, 545–554.

Pillai, K.C. and Samson, P. (1959), "On Hotelling's Generalization of T^2," *Biometrika*, 46, 160–168.

Pothoff, R.F. and Roy, S.N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Prasad, N.G.N. and Rao, J.N.K. (1990), "The Estimation of Mean Squared Error of Small-Area Estimators," *Journal of the American Statistical Association*, 85, 163–171.

Rao, C.R. (1972), "Estimation of Variance and Covariance Components in Linear Models," *Journal of the American Statistical Association*, 67, 112–115.

Ripley, B.D. (1987), *Stochastic Simulation*, New York: John Wiley & Sons, Inc.

Robinson, G.K. (1991), "That BLUP is a Good Thing: The Estimation of Random Effects," *Statistical Science*, 6, 15–51.

Rubin, D.B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592.

Sacks, J., Welch, W.J., Mitchell, T.J. and Wynn, H.P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science* 4, 409–435.

Schluchter, M.D. and Elashoff, J.D. (1990), "Small-Sample Adjustments to Tests with Unbalanced Repeated Measures Assuming Several Covariance Structures," *Journal of Statistical Computation and Simulation*, 37, 69–87.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Schervish, M.J. (1992), "Bayesian Analysis of Linear Models," *Bayesian Statistics 4*, eds. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, Oxford: University Press, 419–434, (with discussion).

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Searle, S.R. (1982), *Matrix Algebra Useful for Statisticians*, New York: John Wiley & Sons, Inc.

Searle, S. R. (1988), "Mixed Models and Unbalanced Data: Wherefrom, Whereat, and Whereto?" *Communications in Statistics–Theory and Methods*, 17(4), 935–968.

Searle, S. R., Casella, G., and McCulloch, C.E. (1992), *Variance Components*, New York: John Wiley & Sons, Inc.

Self, S.G. and Liang, K.Y. (1987), "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions," *Journal of the American Statistical Association*, 82, 605–610.

Serfling, R.J. (1980), *Approximation Theorems of Mathematical Statistics*, New York: John Wiley & Sons, Inc.

Singer, Judith D. (1998), "Using SAS PROC MIXED to Fit Multilevel Models, Hierarchical Models, and Individual Growth Models," *Journal of Educational and Behavioral Statistics*, 23(4), 323–355.

Smith, A.F.M. and Gelfand, A.E. (1992), "Bayesian Statistics without Tears: a Sampling-Resampling Perspective," *American Statistician*, 46, 84–88.

Snedecor, G.W. and Cochran, W.G. (1976), *Statistical Methods, 6th ed.*, Ames, IA: Iowa State University Press.

Snedecor, G.W. and Cochran, W.G. (1980), *Statistical Methods*, Ames: The Iowa State University Press.

Steel, R.G.D., Torrie, J.H., and Dickey D. (1997), *Principles and Procedures of Statistics: A Biometrical Approach, Third Edition,* New York: McGraw-Hill, Inc.

Stram, D.O. and Lee, J.W. (1994), "Variance Components Testing in the Longitudinal Mixed Effects Model," *Biometrics*, 50, 1171–1177.

Stroup, W.W. (1989a), "Predictable Functions and Prediction Space in the Mixed Model Procedure," in *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 39–48.

Stroup, W.W. (1989b), "Use of Mixed Model Procedure to Analyze Spatially Correlated Data: An Example Applied to a Line-Source Sprinkler Irrigation Experiment," *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 104–122.

Sullivan, L.M., Dukes, K.A., and Losina, E. (1999), "An Introduction to Hierarchical Linear Modelling," *Statistics in Medicine*, 18, 855–888.

Swallow, W.H. and Monahan, J.F. (1984), "Monte Carlo Comparison of ANOVA, MIVQUE, REML, and ML Estimators of Variance Components," *Technometrics*, 28, 47–57.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions" (with discussion), *Annals of Statistics*, 22, 1701–1762.

Verbeke, G. and Molenberghs, G., eds. (1997), *Linear Mixed Models in Practice: A SAS-Oriented Approach*, New York: Springer.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data,* New York: Springer.

Westfall, P.J. and Young, S.S. (1993), *Resampling-based Multiple Testing*, New York: John Wiley & Sons, Inc.

Westfall, P.H., Tobias, R.D., Rom, D., Wolfinger, R.D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests using the SAS System*, SAS Institute Inc., Cary, NC.

White, H. (1980), “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity,” *Econometrica*, 48, 817–838.

Whittle, P. (1954), “On stationary processes in the plane,” *Biometrika*, 41, 434–449.

Winer, B.J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill, Inc.

Wolfinger, R.D. (1993), “Covariance Structure Selection in General Mixed Models,” *Communications in Statistics, Simulation and Computation*, 22(4), 1079–1106.

Wolfinger, R.D. (1996), “Heterogeneous Variance- Covariance Structures for Repeated Measures,” *Journal of Agricultural, Biological, and Environmental Statistics*, 1, 205-230.

Wolfinger, R.D. (1997), “An Example of Using Mixed Models and PROC MIXED for Longitudinal Data,” *Journal of Biopharmaceutical Statistics*, 7(4), 481–500.

Wolfinger, R.D. and Chang, M. (1995), “Comparing the SAS GLM and MIXED Procedures for Repeated Measures,” *Proceedings of the Twentieth Annual SAS Users Group Conference*.

Wolfinger, R.D., Tobias, R.D., and Sall, J. (1991), “Mixed Models: A Future Direction,” *Proceedings of the Sixteenth Annual SAS Users Group Conference*, 1380–1388.

Wolfinger, R.D., Tobias, R.D., and Sall, J. (1994), “Computing Gaussian Likelihoods and their Derivatives for General Linear Mixed Models,” *SIAM Journal on Scientific Computing*, 15(6), 1294–1310.

Wright, P.S. (1994), “Adjusted F Tests for Repeated Measures with the MIXED Procedure,” 328 SMC-Statistics Department, University of Tennessee.

Zimmerman, D.L. and Harville, D.A. (1991), “A Random Field Approach to the Analysis of Field-Plot Experiments and Other Spatial Experiments,” *Biometrics*, 47, 223–239.

Chapter 47
The MODECLUS Procedure

Chapter Contents

Chapter 47
The MODECLUS Procedure

Overview

The MODECLUS procedure clusters observations in a SAS data set using any of several algorithms based on nonparametric density estimates. The data can be numeric coordinates or distances. PROC MODECLUS can perform approximate significance tests for the number of clusters and can hierarchically join nonsignificant clusters. The significance tests are empirically validated by simulations with sample sizes ranging from 20 to 2000.

PROC MODECLUS produces output data sets containing density estimates and cluster membership, various cluster statistics including approximate p-values, and a summary of the number of clusters generated by various algorithms, smoothing parameters, and significance levels.

Most clustering methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least-squares criterion (Sarle 1982), such as k-means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage (see Chapter 23, "The CLUSTER Procedure,") is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation (Silverman 1986, pp. 130–146; Scott 1992, pp. 125–190) such as density linkage (see Chapter 23, "The CLUSTER Procedure," Wong and Lane 1983, and Wong and Schaack 1982). The biases of many commonly used clustering methods are discussed in Chapter 7, "Introduction to Clustering Procedures."

PROC MODECLUS implements several clustering methods using nonparametric density estimation. Such clustering methods are referred to hereafter as *nonparametric clustering methods*. The methods in PROC MODECLUS are related to, but not identical to, methods developed by Gitman (1973), Huizinga (1978), Koontz and Fukunaga (1972a, 1972b), Koontz, Narendra, and Fukunaga (1976), Mizoguchi and Shimura (1980), and Wong and Lane (1983). Details of the algorithms are provided in the section "Clustering Methods" on page 2874.

For nonparametric clustering methods, a cluster is loosely defined as a region surrounding a local maximum of the probability density function (see the section "Significance Tests" on page 2876 for a more rigorous definition). Given a sufficiently large sample, nonparametric clustering methods are capable of detecting clusters of unequal size and dispersion and with highly irregular shapes. Nonparametric methods can also obtain good results for compact clusters of equal size and dispersion, but they naturally require larger sample sizes for good recovery than clustering methods that are biased toward finding such “nice” clusters.

For coordinate data, nonparametric clustering methods are less sensitive to changes in scale of the variables or to affine transformations of the variables than are most other commonly used clustering methods. Nevertheless, it is necessary to consider questions of scaling and transformation, since variables with large variances tend to have more effect on the resulting clusters than those with small variances. If two or more variables are not measured in comparable units, some type of standardization or scaling is necessary; otherwise, the distances used by the procedure may be based on inappropriate apples-and-oranges computations. For variables with comparable units of measurement, standardization or scaling may still be desirable if the scale estimates of the variables are not related to their expected importance for defining clusters. If you want two variables to have equal importance in the analysis, they should have roughly equal scale estimates. If you want one variable to have more effect than another, the former should be scaled to have a greater scale estimate than the latter. The STD option in the PROC MODECLUS statement scales all variables to equal variance. However, the variance is not necessarily the most appropriate scale estimate for cluster analysis. In particular, outliers should be removed before using PROC MODECLUS with the STD option. A variety of scale estimators including robust estimators are provided in the STDIZE procedure (for detailed information, see Chapter 66, “The STDIZE Procedure,”). Additionally, the ACECLUS procedure provides another way to transform the variables to try to improve the separation of clusters.

Since clusters are defined in terms of local maxima of the probability density function, nonlinear transformations of the data may change the number of population clusters. The variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required. Ordinal or ranked data are generally inappropriate, since monotone transformations can produce any arbitrary number of modes.

Unlike the methods in the CLUSTER procedure, the methods in the MODECLUS procedure are not inherently hierarchical. However, PROC MODECLUS can do approximate nonparametric significance tests for the number of clusters by obtaining an approximate p-value for each cluster, and it can hierarchically join nonsignificant clusters.

Another important difference between the MODECLUS procedure and many other clustering methods is that you do not tell PROC MODECLUS how many clusters you want. Instead, you specify a *smoothing parameter* (see the section “Density Estimation” on page 2870) and, optionally, a significance level, and PROC MODECLUS determines the number of clusters. You can specify a list of smoothing parameters, and PROC MODECLUS performs a separate cluster analysis for each value in the list.

Getting Started

This section illustrates how PROC MODECLUS can be used to examine the clusters of data in the following artificial data set.

```
data example;
   input x y @@;
   datalines;
18 18  20 22  21 20  12 23  17 12  23 25  25 20  16 27
20 13  28 22  80 20  75 19  77 23  81 26  55 21  64 24
72 26  70 35  75 30  78 42  18 52  27 57  41 61  48 64
59 72  69 72  80 80  31 53  51 69  72 81
;
```

It is a good practice to plot the data to check for obvious clusters or pathologies prior to the analysis. The interactive graphics of the SAS/INSIGHT product are effective for visualizing clusters. In this example, with only two variables and a small sample size, the GPLOT procedure is adequate. The following statements produce Figure 47.1:

```
axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 80 by 20);
axis2 minor=none;
proc gplot;
   plot y*x /frame cframe=ligr vaxis=axis1 haxis=axis2;
run;
```

The plot suggests three clusters. Of these clusters, the one in the lower left corner is the most compact, while the lower right cluster is more dispersed.

The upper cluster is elongated and would be difficult for most clustering algorithms to identify as a single cluster. The plot also suggests that a Euclidean distance of 10 or 20 is a good initial guess for the neighborhood size in density estimation and clustering.

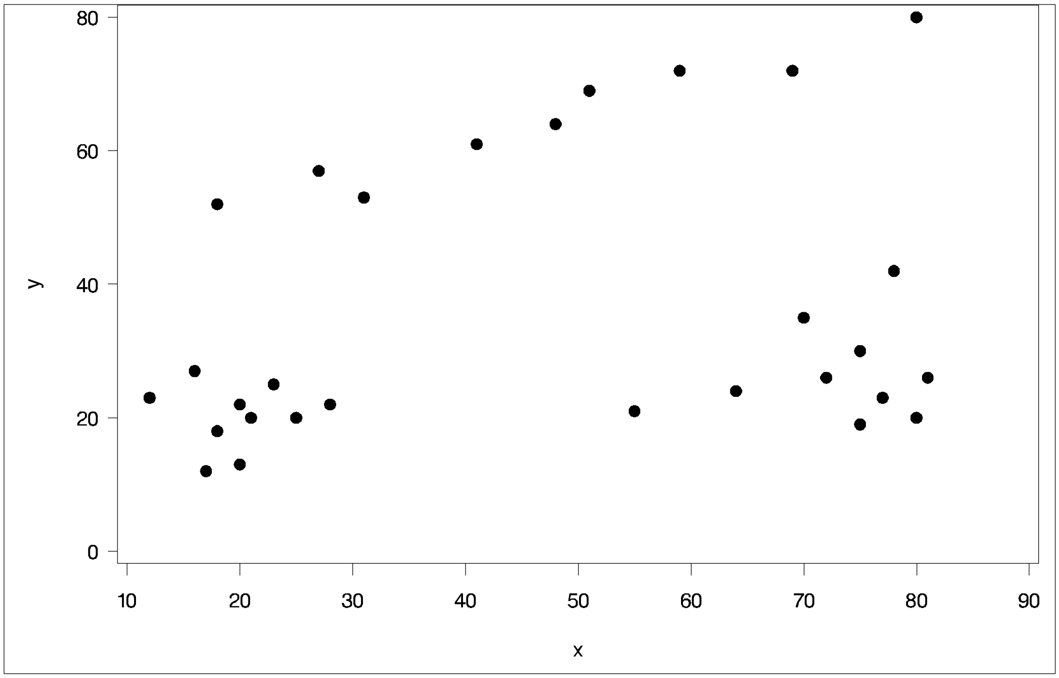


Figure 47.1. Scatter Plot of Data

To obtain a cluster analysis, you must specify the METHOD= option; for most purposes, METHOD=1 is recommended. The cluster analysis can be performed with a list of radii (R=10 15 35), as illustrated in the following PROC MODECLUS step. An output data set containing the cluster membership is created with the OUT= option and then used by PROC GPLOT to display the membership. The following statements produce Figure 47.2 through Figure 47.5:

```
proc modeclus data=example method=1 r=10 15 35 out=out;
run;
```

For each cluster solution, PROC MODECLUS produces a table of cluster statistics including the cluster number, the number of observations in the cluster, the maximum estimated density within the cluster, the number of observations in the cluster having a neighbor that belongs to a different cluster, and the estimated saddle density of the cluster. The results are displayed in Figure 47.2, Figure 47.3, and Figure 47.4 for three different radii. A smaller radius (R=10) yields a larger number of clusters (6), as displayed in Figure 47.1; a larger radius (R=35) includes all observations in a single cluster, as displayed in Figure 47.5. Note that all clusters in these three figures are “isolated” since their corresponding boundary frequencies are all 0s. Therefore, all the estimated saddle densities are missing.

```
                     The MODECLUS Procedure
                          R=10  METHOD=1

                      Cluster Statistics
                            Maximum                   Estimated
                          Estimated      Boundary        Saddle
Cluster     Frequency       Density     Frequency       Density
---------------------------------------------------------------
1                  10    0.00106103             0             .
2                   9    0.00084883             0             .
3                   7    0.00031831             0             .
4                   2    0.00021221             0             .
5                   1     0.0001061             0             .
6                   1     0.0001061             0             .
```

Figure 47.2. Results from PROC MODECLUS for METHOD=1 and R=10

```
                     The MODECLUS Procedure
                          R=15  METHOD=1

                      Cluster Statistics
                            Maximum                   Estimated
                          Estimated      Boundary        Saddle
Cluster     Frequency       Density     Frequency       Density
---------------------------------------------------------------
1                  10    0.00047157             0             .
2                  10    0.00042441             0             .
3                  10    0.00023579             0             .
```

Figure 47.3. Results from PROC MODECLUS for METHOD=1 and R=15

```
                     The MODECLUS Procedure
                          R=35  METHOD=1

                      Cluster Statistics
                            Maximum                   Estimated
                          Estimated      Boundary        Saddle
Cluster     Frequency       Density     Frequency       Density
---------------------------------------------------------------
1                  30    0.00012126             0             .
```

Figure 47.4. Results from PROC MODECLUS for METHOD=1 and R=35

A table summarizing each cluster solution is then produced, as displayed in Figure 47.5.

```
                            The MODECLUS Procedure

                                Cluster Summary
                                           Frequency of
                                Number of  Unclassified
                           R     Clusters       Objects
                      -----------------------------------
                          10            6             0
                          15            3             0
                          35            1             0
```

Figure 47.5. Summary Table

The OUT= data set contains a complete copy of the input data set for each cluster solution. Using a BY statement in the following PROC GPLOT step, you can examine the differences in cluster memberships for each radius. The following statements produce Figure 47.6 through Figure 47.8:

```
symbol1 v='1' font=swiss c=white; symbol2 v='2' font=swiss c=yellow;
symbol3 v='3' font=swiss c=cyan;  symbol4 v='4' font=swiss c=green;
symbol5 v='5' font=swiss c=orange;symbol6 v='6' font=swiss c=blue;
symbol7 v='7' font=swiss c=black;
proc gplot data=out;
   plot y*x=cluster /frame cframe=ligr nolegend vaxis=axis1
        haxis=axis2;
by _r_;
run;
```

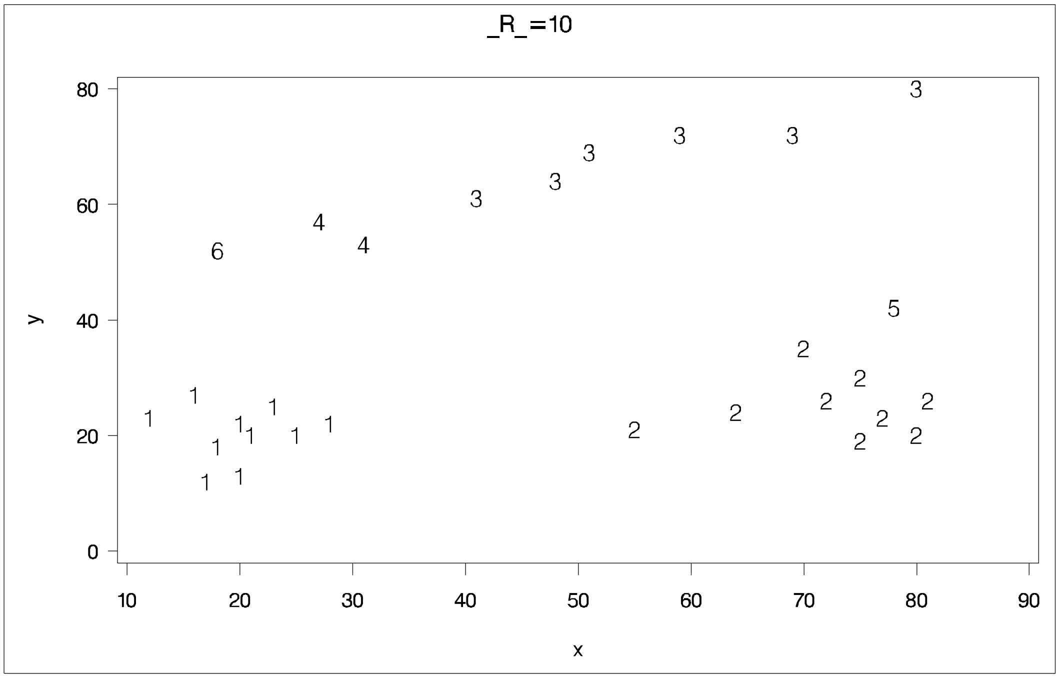


Figure 47.6. Scatter Plots of Cluster Memberships with _R_=10

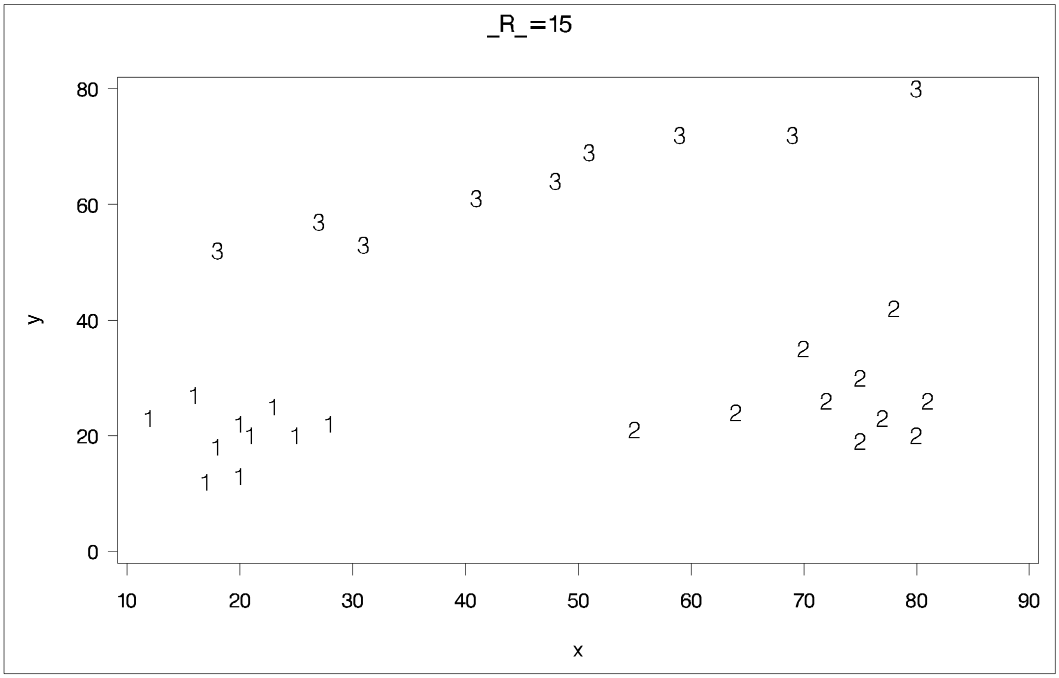


Figure 47.7. Scatter Plots of Cluster Memberships with _R_=15

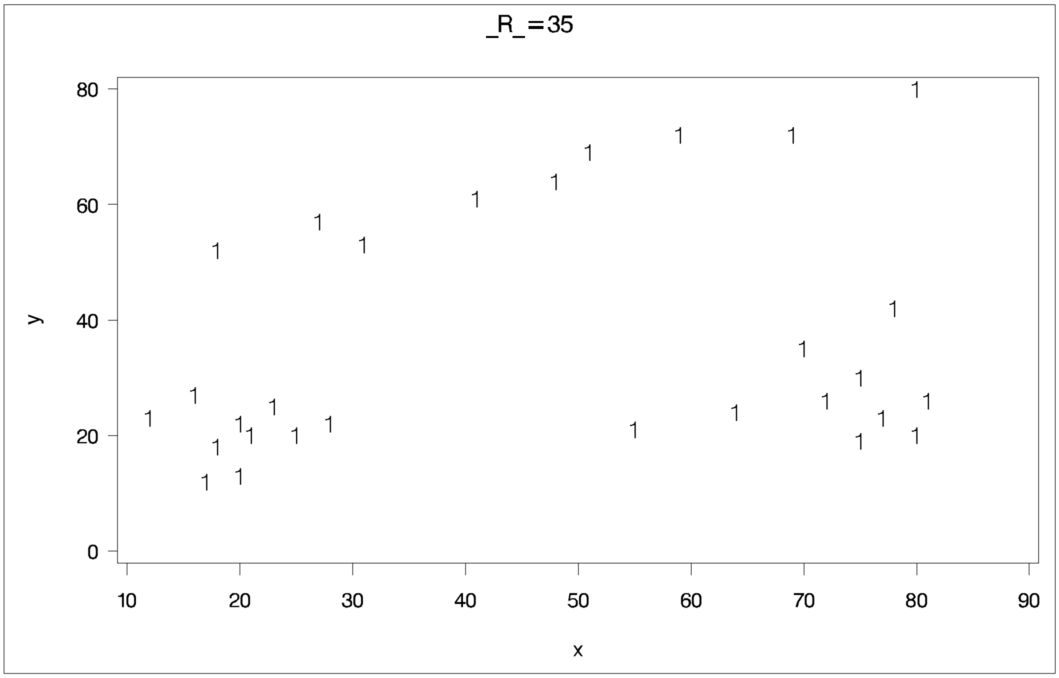


Figure 47.8. Scatter Plots of Cluster Memberships with _R_=35

Syntax

The following statements invoke the MODECLUS procedure:

PROC MODECLUS < *options* > **;**
  BY *variables* **;**
  FREQ | **FREQUENCY** *variable* **;**
  ID *variable* **;**
  VAR | **VARIABLES** *variables* **;**

The PROC MODECLUS statement is required. All other statements are optional.

PROC MODECLUS Statement

PROC MODECLUS < *options* > **;**

The PROC MODECLUS statement invokes the procedure.

Options available in the PROC MODECLUS statement are classified by function in Table 47.1. The corresponding default value for each option, if applicable, is also listed in this table.

Table 47.1. Functional Summary

Description	Option	Default Value
Data Sets		
specify input data set name	DATA=	_LAST_
specify output data set name for observations	OUT=	
specify output data set name for clusters	OUTC=	
specify output data set name for cluster solutions	OUTS=	
Variables in Output Data Sets		
specify variable in the OUT= and OUTCLUS= data sets identifying clusters	CLUSTER=	CLUSTER
specify variable in the OUT= data set containing density estimates	DENSITY=	DENSITY
specify length of variables in the output data sets	OUTLENGTH=	8
Results and Data Processing before Clustering *		
request simple statistics	SIMPLE	
standardize the variables to mean 0 and standard deviation 1	STANDARD	
Smoothing Parameters		
specify number of neighbors to use for kth-nearest-neighbor density estimation	DK=	

*for coordinate data only

Table 47.1. (continued)

Description	Option	Default Value
specify number of neighbors to use for clustering	CK=	
specify number of neighbors to use for kth-nearest-neighbor density estimation and clustering	K=	
specify radius of the sphere of support for uniform-kernel density estimation	DR=	
specify radius of the neighborhood for clustering	CR=	
specify radius of the sphere of support for uniform-kernel density estimation and the neighborhood clustering	R=	
Density Estimation Options		
specify number of times the density estimates are to be cascaded	CASCADE=	0
compute the likelihood cross-validation criterion	CROSS or CROSSLIST	
specify dimensionality to be used when computing density estimates	DIMENSION=	nvar* or 1 †
use arithmetic means for cascading density estimates	AM	
use harmonic means for cascading density estimates	HM	
use sums for cascading density estimates	SUM	
Clustering Methods Options		
dissolve clusters with *n* or fewer members	DOCK	
stop the analysis after obtaining a solution with either no cluster or a single cluster	EARLY	
request that nonsignificant clusters be hierarchically joined.	JOIN(=)	
specify maximum number of clusters to be obtained with METHOD=6	MAXCLUSTERS=	no limit
specify clustering method to use	METHOD=	
specify minimum members for either cluster to be designated a modal cluster when two clusters are joined using METHOD=5	MODE=	the value of K ‡ or 2§
specify power of the density used with METHOD=6	POWER=	2
specify approximate significance tests for the number of clusters	TEST	

*number of variables, for coordinate data
†for distance data
‡when K= is specified
§when K= is not specified

Table 47.1. (continued)

Description	Option	Default Value
specify assignment threshold used with METHOD=6	THRESHOLD=	0.5
Miscellaneous Options		
produce all optional output	ALL	
display the density and cluster membership of observations with neighbors belonging to a different cluster	BOUNDARY	
retain the neighbor lists for each observation in memory	CORE	
display the estimated cross-validated log density of each observation	CROSSLIST	
display the estimated density and cluster membership of each observation	LIST	
display estimates of local dimensionality and write them to the OUT=data set	LOCAL	
display the neighbors of each observation	NEIGHBOR	
suppress the display of the output	NOPRINT	
suppress the display of the summary of the number of clusters, number of unassigned observations, and maximum p-value for each analysis	NOSUMMARY	
suppress the display of statistics for each cluster	SHORT	
trace the cluster assignments for the METHOD=6 algorithm	TRACE	

You can specify at least one of the following options for smoothing parameters for density estimation: DK=, K=, DR=, or R=. To obtain a cluster analysis, you can specify the METHOD= option and at least one of the following smoothing parameters for clustering: CK=, K=, CR=, or R=. If you want significance tests for the number of clusters, you should specify either the DR= or R= option. If none of the smoothing parameters is specified, the MODECLUS procedure provides a default value for the R= option. See the section "Density Estimation" on page 2870 for the formula of a reasonable first guess for R= and a discussion of smoothing parameters.

You can specify lists of values for the DK=, CK=, K=, DR=, CR=, and R= options. Numbers in the lists can be separated by blanks or commas. You can include in the lists one or more items of the form *start* TO *stop* BY *increment*. Each list can contain either one value or the same number of values as in every other list that contains more than one value. If a list has only one value, that value is used in combination with all the values in longer lists. If two or more lists have more than one value, then one analysis is done using the first value in each list, another analysis is done using the second value in each list, and so on.

You can specify the following options in the PROC MODECLUS statement.

ALL
produces all optional output.

AM
specifies arithmetic means for cascading density estimates. See the description of the CASCADE= option.

BOUNDARY
displays the density and cluster membership of observations with neighbors belonging to a different cluster.

CASCADE=*n*
CASC=*n*
specifies the number of times the density estimates are to be cascaded (see the section "Density Estimation" on page 2870). The default value 0 performs no cascading.

You can specify a list of values for the CASCADE= option. Each value in the list is combined with each combination of smoothing parameters to produce a separate analysis.

CK=*n*
specifies the number of neighbors to use for clustering. The number of neighbors should be at least 2 but less than the number of observations. See the section "Density Estimation" on page 2870.

CLUSTER=*name*
provides a name for the variable in the OUT= and OUTCLUS= data sets identifying clusters. The default name is CLUSTER.

CORE
keeps the neighbor lists for each observation in the computer memory to make small problems run faster.

CR=*n*
specifies the radius of the neighborhood for clustering. See the section "Density Estimation" on page 2870.

CROSS
computes the likelihood cross-validation criterion (Silverman 1986, pp. 52–55). This option appears to be of limited usefulness (see the section "Density Estimation" on page 2870).

CROSSLIST
displays the cross-validated log density of each observation.

DATA=*SAS-data-set*
specifies the input data set containing observations to be clustered. If you omit the DATA= option, the most recently created SAS data set is used.

If the data set is TYPE=DISTANCE, the data are interpreted as a distance matrix. The number of variables must equal the number of observations in the data set or in each BY group. The distances are assumed to be Euclidean, but the procedure accepts other types of distances or dissimilarities. Unlike the CLUSTER procedure, PROC MODECLUS uses the entire distance matrix, not just the lower triangle; the distances are not required to be symmetric. The neighbors of a given observation are determined solely from the distances in that observation. Missing values are considered infinite. Various distance measures can be computed from coordinate data using the DISTANCE procedure (for detailed information, see Chapter 26, “The DISTANCE Procedure,”).

If the data set is not TYPE=DISTANCE, the data are interpreted as coordinates in a Euclidean space, and Euclidean distances are computed. The variables can be discrete or continuous and should be at the interval level of measurement.

DENSITY=*name*

provides a name for the variable in the OUT= data set containing density estimates. The default name is DENSITY.

DIMENSION=*n*
DIM=*n*

specifies the dimensionality to be used when computing density estimates. The default is the number of variables if the data are coordinates; the default is 1 if the data are distances.

DK=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation. The number of neighbors should be at least 2 but less than the number of observations. See the section “Density Estimation” on page 2870.

DOCK=*n*

dissolves clusters with *n* or fewer members by making the members unassigned.

DR=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation. See the section “Density Estimation” on page 2870.

EARLY

stops the cluster analysis after obtaining either a solution with no cluster or a solution with one cluster to which all observations are assigned. The smoothing parameters should be specified in increasing order. This can reduce the computer time required for the analysis but may occasionally miss some multiple-cluster solutions.

HM

uses harmonic means for cascading density estimates. See the description of the CASCADE= option.

JOIN<=*p*>

requests that nonsignificant clusters be hierarchically joined. The JOIN option implies the TEST option. After each solution is obtained, the cluster with the largest approximate p-value is either joined to a neighboring cluster or, if there is no neighboring cluster, dissolved by making all of its members unassigned. After two clusters are joined, an analysis of the remaining clusters is displayed.

If you do not specify a p-value with the JOIN= option, joining continues until only one cluster remains, and the results are written to the output data sets after each analysis. If you specify a p-value with the JOIN= option, joining continues until the greatest approximate p-value is less than the value given in the JOIN= option, and only if there is more than one cluster are the results for that analysis written to the output data sets.

Any value of p less than 1E−8 is set to 1E−8.

K=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation and clustering. The number of neighbors should be at least 2 but less than the number of observations. Specifying K=*n* is equivalent to specifying both DK=*n* and CK=*n*. See the section "Density Estimation" on page 2870.

LIST

displays the estimated density and cluster membership of each observation.

LOCAL

requests estimates of local dimensionality (Tukey and Tukey 1981, pp. 236–237).

MAXCLUSTERS=*n*
MAXC=*n*

specifies the maximum number of clusters to be obtained with the METHOD=6 option. By default, there is no fixed limit.

METHOD=*n*
MET=*n*
M=*n*

specifies what clustering method to use. Since these methods do not have widely recognized names, the methods are indicated by numbers from 0 to 6. The methods are described in the section "Clustering Methods" on page 2874. For most purposes, METHOD=1 is recommended, although METHOD=6 may occasionally produce better results in return for considerably greater computer time and space requirements. METHOD=1 is not good for discrete coordinate data with only a few equally spaced values. In this case, METHOD=6 or METHOD=3 works better. METHOD=4 or METHOD=5 is less desirable than other methods when there are ties, since a general characteristic of agglomerative hierarchical clustering methods is that the results are indeterminate in the presence of ties.

You must specify the METHOD= option to obtain a cluster analysis.

You can specify a list of values for the METHOD= option. Each value in the list is combined with each combination of smoothing and cascading parameters to produce a separate cluster analysis.

MODE=*n*

specifies that when two clusters are joined using the METHOD=5 option (no other methods are affected by the MODE= option), each must have at least *n* members for either cluster to be designated a modal cluster. In any case, each cluster must also have a maximum density greater than the fusion density for either cluster to be designated a modal cluster. If you specify the K= option, the default value of the MODE= option is the same as the value of the K= option because the use of kth-nearest-neighbor density estimation limits the resolution that can be obtained for clusters with fewer than k members. If you do not specify the K= option, the default is MODE=2. If you specify MODE=0, the default value is used instead of 0. If you specify a FREQ statement, the MODE= value is compared to the number of observations in each cluster, not to the sum of the frequencies.

NEIGHBOR

displays the neighbors of each observation in a table called "Nearest Neighbor List."

NOPRINT

suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

NOSUMMARY

suppresses the display of the summary of the number of clusters, number of unassigned observations, and maximum p-value for each analysis.

OUT=*SAS-data-set*

specifies the output data set containing the input data plus density estimates, cluster membership, and variables identifying the type of solution. There is an output observation corresponding to each input observation for each solution. Therefore, the OUT= data set can be very large.

OUTCLUS=*SAS-data-set*
OUTC=*SAS-data-set*

specifies the output data set containing an observation corresponding to each cluster in each solution. The variables identify the solution and contain statistics describing the clusters.

OUTSUM=*SAS-data-set*
OUTS=*SAS-data-set*

specifies the output data set containing an observation corresponding to each cluster solution giving the number of clusters and the number of unclassified observations for that solution.

OUTLENGTH=*n*
OUTL=*n*

specifies the length of those output variables that are not copied from the input data set but are created by PROC MODECLUS.

The OUTLENGTH= option applies only to the following variables that appear in all of the output data sets:_K_, _DK_, _CK_, _R_, _DR_, _CR_, _CASCAD_, _METHOD_, _NJOIN_, and _LOCAL_.

The minimum value is 2 or 3, depending on the operating system. The maximum value is 8. The default value is 8.

POWER=*n*
POW=*n*

specifies the power of the density used with the METHOD=6 option. The default value is 2.

R=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation and the neighborhood for clustering. Specifying R=*n* is equivalent to specifying both DR=*n* and CR=*n*. See the section "Density Estimation" on page 2870.

SHORT

suppresses the display of statistics for each cluster.

SIMPLE
S

displays means, standard deviations, skewness, kurtosis, and a coefficient of bimodality. The SIMPLE option applies only to coordinate data.

STANDARD
STD

standardizes the variables to mean 0 and standard deviation 1. The STANDARD option applies only to coordinate data.

SUM

uses sums for cascading density estimates. See the description of the CASCADE= option on page 2865.

TEST

performs approximate significance tests for the number of clusters. The R= or DR= option must also be specified with a nonzero value to obtain significance tests.

The significance tests performed by PROC MODECLUS are valid only for simple random samples, and they require at least 20 observations per cluster to have enough power to be of any use. See the section "Significance Tests" on page 2876.

THRESHOLD=*n*
THR=*n*

specifies the assignment threshold used with the METHOD=6 option. The default is 0.5.

TRACE

traces the cluster assignments for the METHOD=6 algorithm.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MODECLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MODECLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ | **FREQUENCY** *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC MODECLUS then treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value.

ID Statement

ID *variable* **;**

The values of the ID variable identify observations in the displayed results and in the OUT= data set. If you omit the ID statement, each observation is identified by its observation number, and a variable called _OBS_ is written to the OUT= data set containing the original observation numbers.

VAR Statement

VAR | **VARIABLES** *variables* **;**

The VAR statement specifies numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not specified in other statements are used.

Details

Density Estimation

Refer to Silverman (1986) or Scott (1992) for an introduction to nonparametric density estimation.

PROC MODECLUS uses (hyper)spherical uniform kernels of fixed or variable radius. The density estimate at a point is computed by dividing the number of observations within a sphere centered at the point by the product of the sample size and the volume of the sphere. The size of the sphere is determined by the smoothing parameters that you are required to specify.

For fixed-radius kernels, specify the radius as a Euclidean distance with either the DR= or R= option. For variable-radius kernels, specify the number of neighbors desired within the sphere with either the DK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation at which the density is being estimated. If you specify both the DR= or R= option and the DK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers.

It is convenient to refer to the sphere of support of the kernel at observation x_i as the *neighborhood* of x_i. The observations within the neighborhood of x_i are the *neighbors* of x_i. In some contexts, x_i is considered a neighbor of itself, but in other contexts it is not. The following notation is used in this chapter.

x_i	the ith observation
d(x,y)	the distance between points x and y
n	the total number of observations in the sample
n_i	the number of observations within the neighborhood of x_i including x_i itself
n_i^-	the number of observations within the neighborhood of x_i not including x_i itself
N_i	the set of indices of neighbors of x_i including i
N_i^-	the set of indices of neighbors of x_i not including i
v_i	the volume of the neighborhood of x_i
$\hat{f}_i$	the estimated density at x_i
$\hat{f}_i^-$	the cross-validated density estimate at x_i
C_k	the set of indices of observations assigned to cluster k
v	the number of variables or the dimensionality
s_l	standard deviation of the *l*th variable

The estimated density at x_i is

$$\hat{f}_i = \frac{n_i}{nv_i}$$

that is, the number of neighbors of x_i divided by the product of the sample size and the volume of the neighborhood at x_i.

The density estimates provided by uniform kernels are not quite as good as those provided by some other types of kernels, but they are quite satisfactory for clustering. The significance tests for the number of clusters require the use of fixed-size uniform kernels.

There is no simple answer to the question of which smoothing parameter to use (Silverman 1986, pp. 43–61, 84–88, 98–99). It is usually necessary to try several different smoothing parameters. A reasonable first guess for the K= option is in the range of 0.1 to 1 times $n^{4/(v+4)}$, smaller values being suitable for higher dimensionalities. A reasonable first guess for the R= option in many coordinate data sets is given by

$$\left[\frac{2^{v+2}(v+2)\Gamma(.5v+1)}{nv^2}\right]^{1/(v+4)}\sqrt{\sum_{l=1}^{v} s_l^2}$$

which can be computed in a DATA step using the GAMMA function for Γ. The MODECLUS procedure also provides this first guess as a default smoothing parameter if none of the options (DR=, CR=, R=, DK=, CK=, and K=) is specified. This formula is derived under the assumption that the data are sampled from a multivariate normal distribution and, therefore, tend to be too large (oversmooth) if the true distribution is multimodal. Robust estimates of the standard deviations may be preferable if there are outliers. If the data are distances, the factor $\sqrt{\sum {s_l}^2}$ can be replaced by an average root-mean-square Euclidean distance divided by $\sqrt{2}$. To prevent outliers from appearing as separate clusters, you can also specify K=2 or CK=2 or, more generally, K=*m* or CK=*m*, $m \geq 2$, which in most cases forces clusters to have at least *m* members.

If the variables all have unit variance (for example, if you specify the STD option), you can use Table 47.2 to obtain an initial guess for the R= option.

Table 47.2. Reasonable First Guess for R= for Standardized Data

Number	Number of Variables									
of Obs	1	2	3	4	5	6	7	8	9	10
20	1.01	1.36	1.77	2.23	2.73	3.25	3.81	4.38	4.98	5.60
35	0.91	1.24	1.64	2.08	2.56	3.08	3.62	4.18	4.77	5.38
50	0.84	1.17	1.56	1.99	2.46	2.97	3.50	4.06	4.64	5.24
75	0.78	1.09	1.47	1.89	2.35	2.85	3.38	3.93	4.50	5.09
100	0.73	1.04	1.41	1.82	2.28	2.77	3.29	3.83	4.40	4.99
150	0.68	0.97	1.33	1.73	2.18	2.66	3.17	3.71	4.27	4.85
200	0.64	0.93	1.28	1.67	2.11	2.58	3.09	3.62	4.17	4.75
350	0.57	0.85	1.18	1.56	1.98	2.44	2.93	3.45	4.00	4.56
500	0.53	0.80	1.12	1.49	1.91	2.36	2.84	3.35	3.89	4.45
750	0.49	0.74	1.06	1.42	1.82	2.26	2.74	3.24	3.77	4.32
1000	0.46	0.71	1.01	1.37	1.77	2.20	2.67	3.16	3.69	4.23
1500	0.43	0.66	0.96	1.30	1.69	2.11	2.57	3.06	3.57	4.11
2000	0.40	0.63	0.92	1.25	1.63	2.05	2.50	2.99	3.49	4.03

One data-based method for choosing the smoothing parameter is likelihood cross validation (Silverman 1986, pp. 52–55). The cross-validated density estimate at an observation is obtained by omitting the observation from the computations.

$$\hat{f}_i^- = \frac{n_i^-}{nv_i}$$

The (log) likelihood cross-validation criterion is then computed as

$$\sum_{i=1}^{n} \log \hat{f}_i^-$$

The suggested smoothing parameter is the one that maximizes this criterion. With fixed-radius kernels, likelihood cross validation oversmooths long-tailed distributions; for purposes of clustering, it tends to undersmooth short-tailed distributions. With k-nearest-neighbor density estimation, likelihood cross validation is useless because it almost always indicates k=2.

Cascaded density estimates are obtained by computing initial kernel density estimates and then, at each observation, taking the arithmetic mean, harmonic mean, or sum of the initial density estimates of the observations within the neighborhood. The cascaded density estimates can, in turn, be cascaded, and so on. Let ${}_k\hat{f}_i$ be the density estimate at x_i cascaded k times. For all types of cascading, ${}_0\hat{f}_i = \hat{f}_i$. If the cascading is done by arithmetic means, then, for $k \geq 0$,

$${}_{k+1}\hat{f}_i = \sum_{j \in N_i} {}_k\hat{f}_j / n_i$$

For harmonic means,

$${}_{k+1}\hat{f}_i = \left(\sum_{j \in N_i} {}_k\hat{f}_j^{-1} / n_i \right)^{-1}$$

and for sums,

$${}_{k+1}\hat{f}_i = \left(\sum_{j \in N_i} {}_k\hat{f}_j^{k+1} \right)^{\frac{1}{k+2}}$$

To avoid cluttering formulas, the symbol $\hat{f}_i$ is used from now on to denote the density estimate at x_i whether cascaded or not, since the clustering methods and significance tests do not depend on the degree of cascading.

Cascading increases the smoothness of the estimates with less computation than would be required by increasing the smoothing parameters to yield a comparable degree of smoothness. For population densities with bounded support and discontinuities at the boundaries, cascading improves estimates near the boundaries. Cascaded estimates, especially using sums, may be more sensitive to the local covariance structure of the distribution than are the uncascaded kernel estimates. Cascading seems to be useful for detecting very nonspherical clusters. Cascading was suggested by Tukey and Tukey (1981, p. 237). Additional research into the properties of cascaded density estimates is needed.

Clustering Methods

The number of clusters is a function of the smoothing parameters. The number of clusters tends to decrease as the smoothing parameters increase, but the relationship is not strictly monotonic. Generally, you should specify several different values of the smoothing parameters to see how the number of clusters varies.

The clustering methods used by PROC MODECLUS use spherical clustering neighborhoods of fixed or variable radius that are similar to the spherical kernels used for density estimation. For fixed-radius neighborhoods, specify the radius as a Euclidean distance with either the CR= or R= option. For variable-radius neighborhoods, specify the number of neighbors desired within the sphere with either the CK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation for which the neighborhood is being determined. However, in the following descriptions of clustering methods, an observation is not considered to be one of its own neighbors. If you specify both the CR= or R= option and the CK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers. In this section, the symbols N_i, N_i^-, n_i, and n_i^- refer to clustering neighborhoods, not density estimation neighborhoods.

METHOD=0

Begin with each observation in a separate cluster. For each observation and each of its neighbors, join the cluster to which the observation belongs with the cluster to which the neighbor belongs. This method does not use density estimates. With a fixed clustering radius, the clusters are those obtained by cutting the single linkage tree at the specified radius (see Chapter 23, “The CLUSTER Procedure,”).

METHOD=1

Begin with each observation in a separate cluster. For each observation, find the nearest neighbor with a greater estimated density. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Next, consider each observation with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor. Join the cluster containing the observation with (1) each cluster containing a neighbor of the observation such that the maximum density estimate in the cluster equals the density estimate at the observation and (2) the cluster containing the nearest neighbor of the observation such that the maximum density estimate in the cluster exceeds the density estimate at the observation.

This method is similar to the classification or assignment stage of algorithms described by Gitman (1973) and Huizinga (1978).

METHOD=2

Begin with each observation in a separate cluster. For each observation, find the neighbor with the greatest estimated density exceeding the estimated density of the observation. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1.

This method is similar to the first stage of an algorithm proposed by Mizoguchi and Shimura (1980).

METHOD=3

Begin with each observation in a separate cluster. For each observation, find the neighbor with greater estimated density such that the slope of the line connecting the point on the estimated density surface at the observation with the point on the estimated density surface at the neighbor is a maximum. That is, for observation x_i, find a neighbor x_j such that $(\hat{f}_j - \hat{f}_i)/\mathrm{d}(x_j, x_i)$ is a maximum. If this slope is positive, join the cluster to which observation x_i belongs with the cluster to which the specified neighbor x_j belongs. This method was invented by Koontz, Narendra, and Fukunaga (1976).

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1. The algorithm suggested for this situation by Koontz, Narendra, and

Fukunaga (1976) may fail for flat areas in the estimated density that contain four or more observations.

METHOD=4

This method is equivalent to the first stage of two-stage density linkage (see Chapter 23, “The CLUSTER Procedure,”) without the use of the MODE=option.

METHOD=5

This method is equivalent to the first stage of two-stage density linkage (see Chapter 23, “The CLUSTER Procedure,”) with the use of the MODE=option.

METHOD=6

Begin with all observations unassigned.

Step 1: Form a list of seeds, each seed being a single observation such that the estimated density of the observation is not less than the estimated density of any of its neighbors. If you specify the MAXCLUSTERS=n option, retain only the n seeds with the greatest estimated densities.

Step 2: Consider each seed in decreasing order of estimated density.

1. If the current seed has already been assigned, proceed to the next seed. Otherwise, form a new cluster consisting of the current seed.
2. Add to the cluster any unassigned seed that is a neighbor of a member of the cluster or that shares a neighbor with a member of the cluster; repeat until no unassigned seed satisfies these conditions.
3. Add to the cluster all neighbors of seeds that belong to the cluster.
4. Consider each unassigned observation. Compute the ratio of the sum of the $p-1$ powers of the estimated density of the neighbors that belong to the current cluster to the sum of the $p-1$ powers of the estimated density of all of its neighbors, where p is specified by the POWER= option and is 2 by default. Let x_i be the current observation, and let k be the index of the current cluster. Then this ratio is

$$r_{ik} = \frac{\sum_{j \in N_i \cap C_k} \hat{f}_j^{p-1}}{\sum_{j \in N_i} \hat{f}_j^{p-1}}$$

(The sum of the $p-1$ powers of the estimated density of the neighbors of an observation is an estimate of the integral of the pth power of the density over the neighborhood.) If r_{ik} exceeds the maximum of 0.5 and the value of the THRESHOLD= option, add the observation x_i to the current cluster k. Repeat until no more observations can be added to the current cluster.

Step 3: (This step is performed only if the value of the THRESHOLD= option is less than 0.5.) Form a list of unassigned observations in decreasing order of estimated density. Repeat the following actions until the list is empty.

1. Remove the first observation from the list, for example, observation x_i.
2. For each cluster k, compute r_{ik}.
3. If the maximum over clusters of r_{ik} exceeds the value of the THRESHOLD= option, assign observation x_i to the corresponding cluster and insert all observations of which the current observation is a neighbor into the list, keeping the list in decreasing order of estimated density.

METHOD=6 is related to a method invented by Koontz and Fukunaga (1972a) and discussed by Koontz and Fukunaga (1972b).

Significance Tests

Significance tests require that a fixed-radius kernel be specified for density estimation via the DR= or R= option. You can also specify the DK= or K= option, but only the fixed radius is used for the significance tests.

The purpose of the significance tests is as follows: given a simple random sample of objects from a population, obtain an estimate of the number of clusters in the population such that the probability in repeated sampling that the estimate exceeds the true number of clusters is not much greater than α, 1%$\leq$ α $\leq$ 10%. In other words, a sequence of null hypotheses of the form

$H_0^{(i)}$: The number of population clusters is *i* or less

where $i = 1, 2, \cdots, n$, is tested against the alternatives such as

$H_a^{(i)}$: The number of population clusters exceeds *i*

with a maximum experimentwise error rate of approximately α. The tests protect you from overestimating the number of population clusters. It is impossible to protect against underestimating the number of population clusters without introducing much stronger assumptions than are used here, since the number of population clusters could conceivably exceed the sample size.

The method for conducting significance tests is as follows:

1. Estimate densities using fixed-radius uniform kernels.
2. Obtain preliminary clusters by a "valley-seeking" method. Other clustering methods could be used but would yield less power.
3. Compute an approximate p-value for each cluster by comparing the estimated maximum density in the cluster with the estimated maximum density on the cluster boundary.
4. Repeatedly join the least significant cluster with a neighboring cluster until all remaining clusters are significant.
5. Estimate the number of population clusters as the number of significant sample clusters.
6. The preceding steps can be repeated for any number of different radii, and the estimate of the number of population clusters can be taken to be the maximum number of significant sample clusters for any radius.

This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the p-values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from 20 to 2000 indicate that the p-values are almost always conservative. The only case discovered so far in which the p-values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

To make inferences regarding population clusters, it is first necessary to define what is meant by a cluster. For clustering methods using nonparametric density estimation, a cluster is usually loosely defined as a region surrounding a local maximum of the probability density function or a maximal connected set of local maxima. This definition may not be satisfactory for very rough densities with many local maxima. It is not applicable at all to discrete distributions for which the density does not exist. As another example in which this definition is not intuitively reasonable, consider a uniform distribution in two dimensions with support in the shape of a figure eight (including the interior). This density might be considered to contain two clusters even though it does not have two distinct modes.

These difficulties can be avoided by defining clusters in terms of the local maxima of a smoothed probability density or mass function. For example, define the neighborhood distribution function (NDF) with radius r at a point x as the probability that a randomly selected point will lie within a radius r of x, that is, the probability integral over a hypersphere of radius r centered at x:

$$s(x) \quad = \quad P(\mathrm{d}(x,X) <= r)$$

where X is the random variable being sampled, r is a user-specified radius, and d(x,y) is the distance between points x and y.

The NDF exists for all probability distributions. You can select the radius according to the degree of resolution required. The minimum-variance unbiased estimate of the NDF at a point x is proportional to the uniform-kernel density estimate with corresponding support.

You can define a *modal region* as a maximal connected set of local maxima of the NDF. A cluster is a connected set containing exactly one modal region. This definition seems to give intuitively reasonable results in most cases. An exception is a uniform density on the perimeter of a square. The NDF has four local maxima. There are eight local maxima along the perimeter, but running PROC MODECLUS with the R= option would yield four clusters since the two local maxima at each corner are separated by a distance equal to the radius. While this density does indeed have four distinctive features (the corners), it is not obvious that each corner should be considered a cluster.

The number of population clusters depends on the radius of the NDF. The significance tests in PROC MODECLUS protect against overestimating the number of clusters at any specified radius. It is often useful to look at the clustering results across a range of radii. A plot of the number of sample clusters as a function of the radius is a useful descriptive display, especially for high-dimensional data (Wong and Schaack 1982).

If a population has two clusters, it must have two modal regions. If there are two modal regions, there must be a "valley" between them. It seems intuitively desirable that the boundary between the two clusters should follow the bottom of this valley. All the clustering methods in PROC MODECLUS are designed to locate the estimated cluster boundaries in this way, although methods 1 and 6 seem to be much more successful at this than the others. Regardless of the precise location of the cluster boundary, it is clear that the maximum of the NDF along the boundary between two clusters must be strictly less than the value of the NDF in either modal region; otherwise, there would be only a single modal region; according to Hartigan and Hartigan (1985), there must be a "dip" between the two modes. PROC MODECLUS assesses the significance of a sample cluster by comparing the NDF in the modal region with the maximum of the NDF along the cluster boundary. If the NDF has second-order derivatives in the region of interest and if the boundary between the two clusters is indeed at the bottom of the valley, then the maximum value of the NDF along the boundary occurs at a saddle point. Hence, this test is called a *saddle test*. This term is intended to describe any test for clusters that compares modal densities with saddle densities, not just the test currently implemented in the MODECLUS procedure.

The obvious estimate of the maximum NDF in a sample cluster is the maximum estimated NDF at an observation in the cluster. Let $m(k)$ be the index of the observation for which the maximum is attained in cluster k.

Estimating the maximum NDF on the cluster boundary is more complicated. One approach is to take the maximum NDF estimate at an observation in the cluster that has a neighbor belonging to another cluster. This method yields excessively large estimates when the neighborhood is large. Another approach is to try to choose an object closer to the boundary by taking the observation with the maximum sum of estimated densities of neighbors belonging to a different cluster. After some experimentation, it is found that a combination of these two methods works well. Let B_k be the set of indices of observations in cluster k that have neighbors belonging to a different cluster, and compute

$$\text{max}_{i \in B_k} \left(0.2 \hat{f}_i n_i + \sum_{j \in N_i - C_k} \hat{f}_j \right)$$

Let $s(k)$ be the index of the observation for which the maximum is attained.

Using the notation $\#(S)$ for the cardinality of set S, let

$$\begin{aligned}
n_{ij}^{-} &= \#(N_i^{-} \cap N_j^{-}) \\
c_m(k) &= n_{m(k)}^{-} - n_{m(k)s(k)}^{-} \\
c_s(k) &= n_{s(k)}^{-} - n_{m(k)s(k)}^{-} \;\; \text{if } B_k \neq \emptyset, \\
&= 0 \;\; \text{otherwise} \\
q_k &= 1/2 \;\; \text{if } B_k \neq \emptyset, \\
&= 2/3 \;\; \text{otherwise} \\
z_k &= \frac{c_m(k) - q_k(c_m(k) + c_s(k)) - 1/2}{\sqrt{q_k(1-q_k)(c_m(k) + c_s(k))}}
\end{aligned}$$

$$u = \left\lceil (.2 + .05\sqrt{n}) \sum_{i:n_i>1} \frac{1}{n_i+1} \right\rceil$$

Let $R(u)$ be a random variable distributed as the range of a random sample of u observations from a standard normal distribution. Then the approximate p-value p_k for cluster k is

$$p_k = Pr(z_k > R(u)/\sqrt{2})$$

If points $m(k)$ and $s(k)$ are fixed a priori, z_k would be the usual approximately normal test statistic for comparing two binomial random variables. In fact, $m(k)$ and $s(k)$ are selected in such a way that $c_m(k)$ tends to be large and $c_s(k)$ tends to be small. For this reason, and because there may be a large number of clusters, each with its own z_k to be tested, each z_k is referred to the distribution of $R(u)$ instead of a standard normal distribution. If the tests are conducted for only one radius and if u is chosen equal to n, then the p-values are very conservative because (1) you are not making all possible pairwise comparisons of observations in the sample and (2) n_i^- and n_j^- are positively correlated if the neighborhoods overlap. In the formula for u, the summation overcorrects somewhat for the conservativeness due to correlated n_i^-'s. The factor $.2+.05\sqrt{n}$ is empirically estimated from simulation results to adjust for the use of more than one radius.

If the JOIN option is specified, the least significant cluster (the cluster with the smallest z_k) is either dissolved or joined with a neighboring cluster. If no members of the cluster have neighbors belonging to a different cluster, all members of the cluster are unassigned. Otherwise, the cluster is joined to the neighboring cluster such that the sum of density estimates of neighbors of the estimated saddle point belonging to it is a maximum. Joining clusters increases the power of the saddle test. For example, consider a population with two well-separated clusters. Suppose that, for a certain radius, each population cluster is divided into two sample clusters. None of the four sample clusters is likely to be significant, but after the two sample clusters corresponding to each population cluster are joined, the remaining two clusters may be highly significant.

The saddle test implemented in PROC MODECLUS has been evaluated by simulation from known distributions. Some results are given in the following three tables. In Table 47.3, samples of 20 to 2000 observations are generated from a one-dimensional uniform distribution. For sample sizes of 1000 or less, 2000 samples are generated and analyzed by PROC MODECLUS. For a sample size of 2000, only 1000 samples are generated. The analysis is done with at least 20 different values of the R= option spread across the range of radii most likely to yield significant results. The six central columns of the table give the observed error rates at the nominal error rates (α) at the head of each column. The standard errors of the observed error rates are given at the bottom of the table. The observed error rates are conservative for $\alpha \leq 5\%$, but they increase with α and become slightly liberal for sample sizes in the middle of the range tested.

Table 47.3. Observed Error Rates (%) for Uniform Distribution

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
20	0.00	0.00	0.00	0.60	11.65	27.05	2000
50	0.35	0.70	4.50	10.95	20.55	29.80	2000
100	0.35	0.85	3.90	11.05	18.95	28.05	2000
200	0.30	1.35	4.00	10.50	18.60	27.05	2000
500	0.45	1.05	4.35	9.80	16.55	23.55	2000
1000	0.70	1.30	4.65	9.55	15.45	19.95	2000
2000	0.40	1.10	3.00	7.40	11.50	16.70	1000
Standard Error	0.22	0.31	0.49	0.67	0.80	0.89	2000
	0.31	0.44	0.69	0.95	1.13	1.26	1000

All unimodal distributions other than the uniform that have been tested, including normal, Cauchy, and exponential distributions and uniform mixtures, have produced much more conservative results. Table 47.4 displays results from a unimodal mixture of two normal distributions with equal variances and equal sampling probabilities and with means separated by two standard deviations. Any greater separation would produce a bimodal distribution. The observed error rates are quite conservative.

Table 47.4. Observed Error Rates (%) for Normal Mixture with 2σ Separation

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
100	0.0	0.0	0.0	1.0	2.0	4.0	200
200	0.0	0.0	0.0	2.0	3.0	3.0	200
500	0.0	0.0	0.5	0.5	0.5	0.5	200

All distributions in two or more dimensions that have been tested yield extremely conservative results. For example, a uniform distribution on a circle yields observed error rates that are never more than one-tenth of the nominal error rates for sample sizes up to 1000. This conservatism is due to the fact that, as the dimensionality increases, more and more of the probability lies in the tails of the distribution (Silverman 1986, p. 92), and the saddle test used by PROC MODECLUS is more conservative for distributions with pronounced tails. This applies even to a uniform distribution on a hypersphere because, although the density has no tails, the NDF does.

Since the formulas for the significance tests do not involve the dimensionality, no problems are created when the data are linearly dependent. Simulations of data in nonlinear subspaces (the circumference of a circle or surface of a sphere) have also yielded conservative results.

Table 47.5 displays results in terms of power for identifying two clusters in samples from a bimodal mixture of two normal distributions with equal variances and equal sampling probabilities separated by four standard deviations. In this simulation, PROC MODECLUS never indicated more than two significant clusters.

Table 47.5. Power (%) for Normal Mixture with 4σ Separation

Sample	Nominal Type 1 Error Rate						Number of
Size	1	2	5	10	15	20	Simulations
20	0.0	0.0	0.0	2.0	37.5	68.5	200
35	0.0	13.5	38.5	48.5	64.0	75.5	200
50	17.5	26.0	51.5	67.0	78.5	84.0	200
75	25.5	36.0	58.5	77.5	85.5	89.5	200
100	40.0	54.5	72.5	84.5	91.5	92.5	200
150	70.5	80.0	92.0	97.0	100.0	100.0	200
200	89.0	96.0	99.5	100.0	100.0	100.0	200

The saddle test is not as efficient as excess-mass tests for multimodality (Müller and Sawitzki 1991, Polonik 1993). However, there is not yet a general approximation for the distribution of excess-mass statistics to circumvent the need for simulations to do significance tests. Refer to Minnotte (1992) for a review of tests for multimodality.

Computational Resources

The MODECLUS procedure stores coordinate data in memory if there is enough space. For distance data, only one observation at a time is in memory.

PROC MODECLUS constructs lists of the neighbors of each observation. The total space required is $12\sum n_i$ bytes, where n_i is based on the largest neighborhood required by any analysis. The lists are stored in a SAS utility data set unless you specify the CORE option. You may get an error message from the SAS System or from the operating system if there is not enough disk space for the utility data set. Clustering method 6 requires a second list that is always stored in memory.

For coordinate data, the time required to construct the neighbor lists is roughly proportional to $v(\log n)(\sum n_i)\log(\sum n_i/n)$. For distance data, the time is roughly proportional to $n^2\log(\sum n_i/n)$.

The time required for density estimation is proportional to $\sum n_i$ and is usually small compared to the time required for constructing the neighbor lists.

Clustering methods 0 through 3 are quite efficient, requiring time proportional to $\sum n_i$. Methods 4 and 5 are slower, requiring time roughly proportional to $(\sum n_i)\log(\sum n_i)$. Method 6 can also be slow, but the time requirements depend very much on the data and the particular options specified. Methods 4, 5, and 6 also require more memory than the other methods.

The time required for significance tests is roughly proportional to $g\sum n_i$, where *g* is the number of clusters.

PROC MODECLUS can process data sets of several thousand observations if you specify reasonable smoothing parameters. Very small smoothing values produce many clusters, whereas very large values produce many neighbors; either case can require excessive time or space.

Missing Values

If the data are coordinates, observations with missing values are excluded from the analysis.

If the data are distances, missing values are treated as infinite. The neighbors of each observation are determined solely by the distances in that observation. The distances are not required to be symmetric, and there is no check for symmetry; the neighbors of each observation are determined only from the distances in that observation. This treatment of missing values is quite different from that of the CLUSTER procedure, which ignores the upper triangle of the distance matrix.

Output Data Sets

The OUT= data set contains one complete copy of the input data set for each cluster solution. There are additional variables identifying each solution and giving information about individual observations. Solutions with only one remaining cluster when JOIN=*p* is specified are omitted from the OUT= data set (see the description of the JOIN= option on page 2866). The OUT= data set can be extremely large, so it may be advisable to specify the DROP= data set option to exclude unnecessary variables.

The OUTCLUS= or OUTC= data set contains one observation for each cluster in each cluster solution. The variables identify the solution and provide statistics describing the cluster.

The OUTSUM= or OUTS= data set contains one observation for each cluster solution. The variables identify the solution and provide information about the solution as a whole.

The following variables can appear in all of the output data sets:

- _K_, which is the value of the K= option for the current solution. This variable appears only if you specify the K= option.
- _DK_, which is the value of the DK= option for the current solution. This variable appears only if you specify the DK= option.
- _CK_, which is the value of the CK= option for the current solution. This variable appears only if you specify the CK= option.
- _R_, which is the value of the R= option for the current solution. This variable appears only if you specify the R= option.
- _DR_, which is the value of the DR= option for the current solution. This variable appears only if you specify the DR= option.
- _CR_, which is the value of the CR= option for the current solution. This variable appears only if you specify the CR= option.
- _CASCAD_, which is the number of times the density estimates have been cascaded for the current solution. This variable appears only if you specify the CASCADE= option.
- _METHOD_, which is the value of the METHOD= option for the current solution. This variable appears only if you specify the METHOD= option.

- _NJOIN_, which is the number of clusters that are joined or dissolved in the current solution. This variable appears only if you specify the JOIN option.
- _LOCAL_, which is the local dimensionality estimate of the observation. This variable appears only if you specify the LOCAL option.

The OUT= data set contains the following variables:

- the variables from the input data set
- _OBS_, which is the observation number from the input data set. This variable appears only if you omit the ID statement.
- DENSITY, which is the estimated density at the observation. This variable can be renamed by the DENSITY= option.
- CLUSTER, which is the number of the cluster to which the observation is assigned. This variable can be renamed by the CLUSTER= option.

The OUTC= data set contains the following variables:

- the BY variables, if any
- _NCLUS_, which is the number of clusters in the solution
- CLUSTER, which is the number of the current cluster
- _FREQ_, which is the number of observations in the cluster
- _MODE_, which is the maximum estimated density in the cluster
- _BFREQ_, which is the number of observations in the cluster with neighbors belonging to a different cluster
- _SADDLE_, which is the estimated saddle density for the cluster
- _MC_, which is the number of observations within the fixed-radius density-estimation neighborhood of the modal observation. This variable appears only if you specify the TEST or JOIN option.
- _SC_, which is the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation. This variable appears only if you specify the TEST or JOIN option.
- _OC_, which is the number of observations within the overlap of the two previous neighborhoods. This variable appears only if you specify the TEST or JOIN option.
- _Z_, which is the approximate *z* statistic for the cluster. This variable appears only if you specify the TEST or JOIN option.
- _P_, which is the approximate p-value for the cluster. This variable appears only if you specify the TEST or JOIN option.

The OUTS= data set contains the following variables:

- the BY variables, if any
- _NCLUS_, which is the number of clusters in the solution
- _UNCL_, which is the number of unclassified observations
- _CROSS_, which is the likelihood cross-validation criterion if you specify the CROSS or CROSSLIST option

Displayed Output

If you specify the SIMPLE option and the data are coordinates, PROC MODECLUS displays the following simple descriptive statistics for each variable:

- the MEAN
- the standard deviation, STD DEV
- the SKEWNESS
- the KURTOSIS
- a coefficient of BIMODALITY (see Chapter 23, "The CLUSTER Procedure,")

If you specify the NEIGHBOR option, PROC MODECLUS displays a list of the neighbors of each observation. The table contains

- the observation number or ID value of the observation
- the observation number or ID value of each of its neighbors
- the distance to each neighbor

If you specify the CROSSLIST option, PROC MODECLUS produces a table of information regarding cross validation of the density estimates. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the number of neighbors
- the estimated log density
- the estimated cross-validated log density

If you specify the LOCAL option, PROC MODECLUS produces a table of information regarding estimates of local dimensionality. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the estimated local dimensionality

If you specify the LIST option, PROC MODECLUS produces a table listing the observations within each cluster. The table can include

- the cluster number
- the observation number or ID value of the observation
- the estimated density
- the sum of the density estimates of observations within the neighborhood that belong to the same cluster
- the sum of the density estimates of observations within the neighborhood that belong to a different cluster
- the sum of the density estimates of all the observations within the neighborhood
- the ratio of the sum of the density estimates for the same cluster to the sum of all the density estimates in the neighborhood

If you specify the LIST option and there are unassigned objects, PROC MODECLUS produces a table listing those observations. The table includes

- the observation number or ID value of the observation
- the estimated density
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you specify the BOUNDARY option, PROC MODECLUS produces a table listing the observations in each cluster that have a neighbor belonging to a different cluster. The table includes

- the observation number or ID value of the observation
- the estimated density
- the cluster number
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you do not specify the SHORT option, PROC MODECLUS produces a table of cluster statistics including

- the cluster number

- the cluster frequency (the number of observations in the cluster)
- the maximum estimated density within the cluster
- the number of observations in the cluster having a neighbor that belongs to a different cluster
- the estimated saddle density of the cluster

If you specify the TEST or JOIN option, the table of cluster statistics includes the following items pertaining to the saddle test:

- the number of observations within the fixed-radius density-estimation neighborhood of the modal observation
- the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation
- the number of observations within the overlap of the two preceding neighborhoods
- the z statistic for comparing the preceding counts
- the approximate p-value

If you do not specify the NOSUMMARY option, PROC MODECLUS produces a table summarizing each cluster solution containing the following items:

- the smoothing parameters and cascade value
- the number of clusters
- the frequency of unclassified objects
- the likelihood cross-validation criterion if you specify the CROSS or CROSSLIST option

If you specify the JOIN option, the summary table also includes

- the number of clusters joined
- the maximum p-value of any cluster in the solution

If you specify the TRACE option, PROC MODECLUS produces a table for each cluster solution that lists each observation along with its cluster membership as it is reassigned from the “Old” cluster to the “New” cluster. This reassignment is described in **Step 1** through **Step 3** of the section “METHOD=6” on page 2875. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the estimated density

- the “Old” cluster membership. 0 represents an unassigned observation and -1 represents a seed.
- the “New” cluster membership
- “Ratio,” which is documented in the section “METHOD=6” on page 2875. The following character values can also be displayed:

“M”	means the observation is a mode
“S”	means the observation is a seed
“N”	means the neighbor of a mode or seed, for which the ratio is not computed

ODS Table Names

PROC MODECLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 47.6. ODS Tables Produced in PROC MODECLUS

ODS Table Name	Description	Statement	Option
BoundaryFreq	Boundary objects information	PROC	BOUNDARY (or ALL)
ClusterList	Cluster listing, cluster id, freq., density etc.	PROC	LIST (or ALL)
ClusterStats	Cluster statistics	PROC	default
	Cluster statistics, significance test statistics	PROC	TEST or JOIN (or ALL)
ClusterSummary	Cluster summary	PROC	default
	Cluster summary, crossvalidation criterion	PROC	CROSS or CROSSLIST (or ALL)
	Cluster summary, clusters joined information	PROC	JOIN (or ALL)
CrossList	Cross-validated log density	PROC	CROSSLIST
ListLocal	Local dimensionality estimates	PROC	LOCAL
Neighbor	Nearest neighbor list	PROC	NEIGHBOR (or ALL)
SimpleStatistics	Simple statistics	PROC	SIMPLE (or ALL)
Trace	Trace of clustering algorithm (METHOD= 6 only)	PROC	TRACE (or ALL) with METHOD= 6
UnassignObjects	Information on unassigned objects	PROC	LIST (or ALL)

Examples

Example 47.1. Cluster Analysis of Samples from Univariate Distributions

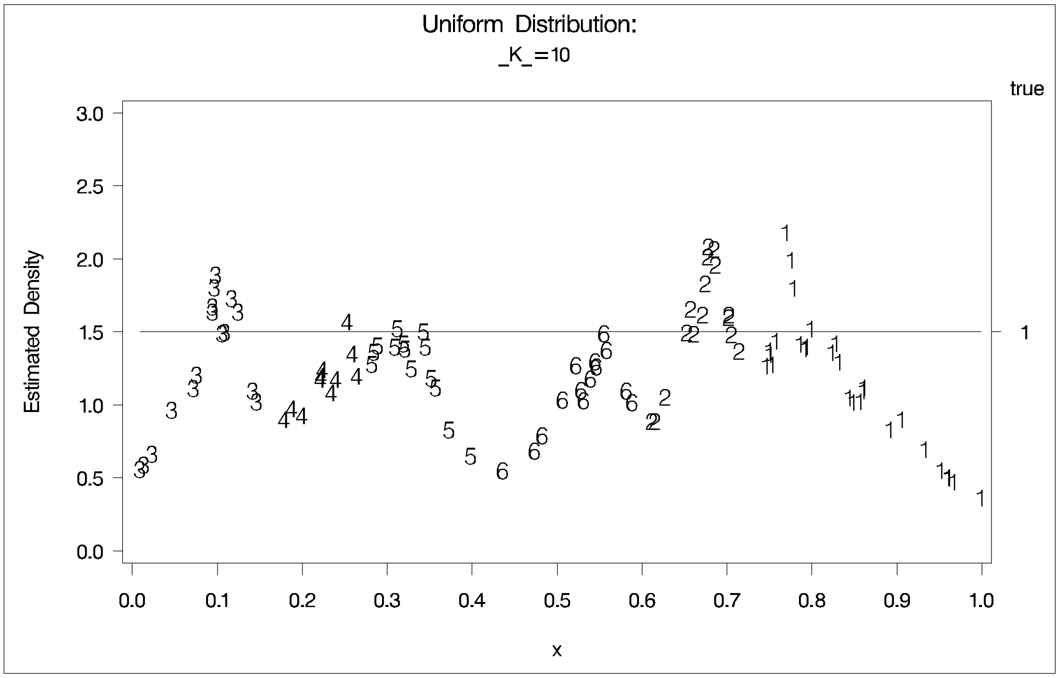
This example uses pseudo-random samples from a uniform distribution, an exponential distribution, and a bimodal mixture of two normal distributions. Results are presented in Output 47.1.1 through Output 47.1.3 as plots displaying both the true density and the estimated density, as well as cluster membership.

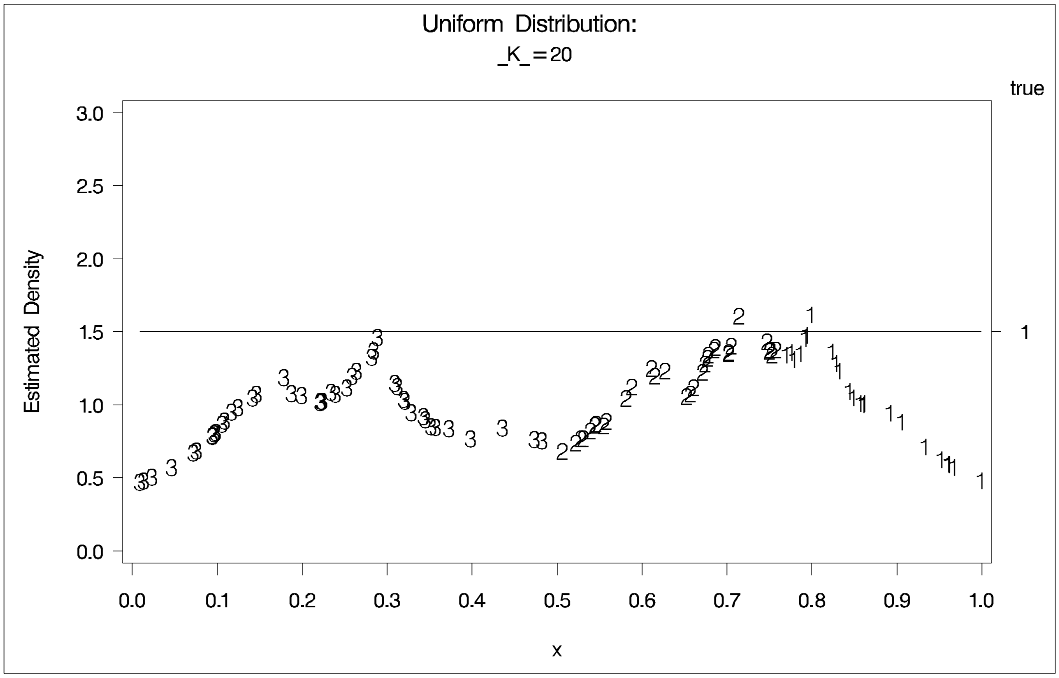
The following statements produce Output 47.1.1:

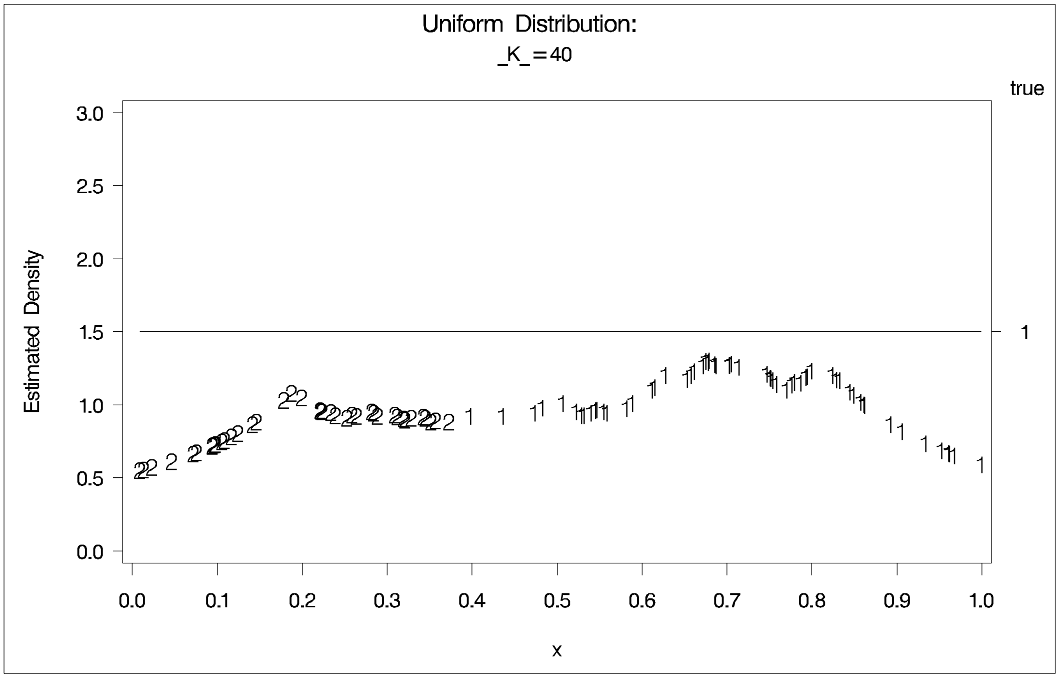
```
options noovp ps=28 ls=95;
title 'Modeclus Example with Univariate Distributions';
title2 'Uniform Distribution';

data uniform;
   drop n;
   true=1;
   do n=1 to 100;
      x=ranuni(123);
      output;
   end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 3 by 0.5);
axis2 minor=none;
symbol9 v=none i=splines;

proc modeclus data=uniform m=1 k=10 20 40 60 out=out short;
   var x;

proc gplot data=out;
   plot density*x=cluster /frame cframe=ligr
                           vzero nolegend
                           vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _K_;
run;

proc modeclus data=uniform m=1 r=.05 .10 .20 .30
              out=out short;
   var x;

axis1 label=(angle=90 rotate=0)
      minor=none order=(0 to 2 by 0.5);
proc gplot data=out;
   plot density*x=cluster /frame cframe=ligr
                           vzero nolegend
                           vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _R_;
run;
```

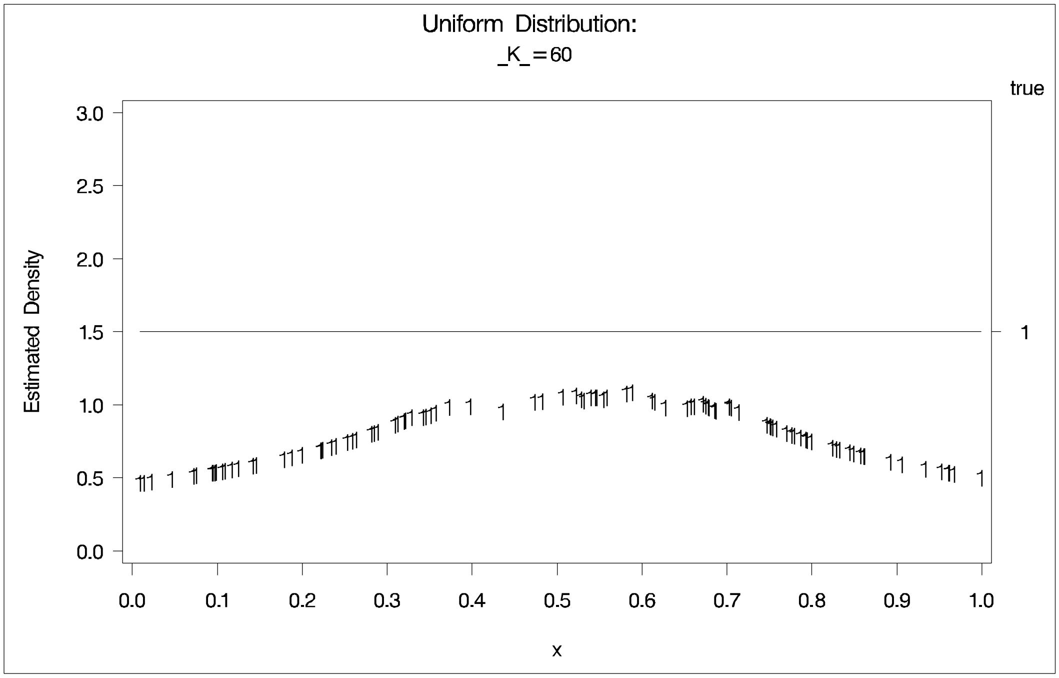
Output 47.1.1. Cluster Analysis of Sample from a Uniform Distribution

```
                 Modeclus Example with Univariate Distributions
                              Uniform Distribution

                             The MODECLUS Procedure

                                Cluster Summary
                                            Frequency of
                               Number of    Unclassified
                       K        Clusters         Objects
                  ------------------------------------
                      10               6               0
                      20               3               0
                      40               2               0
                      60               1               0
```

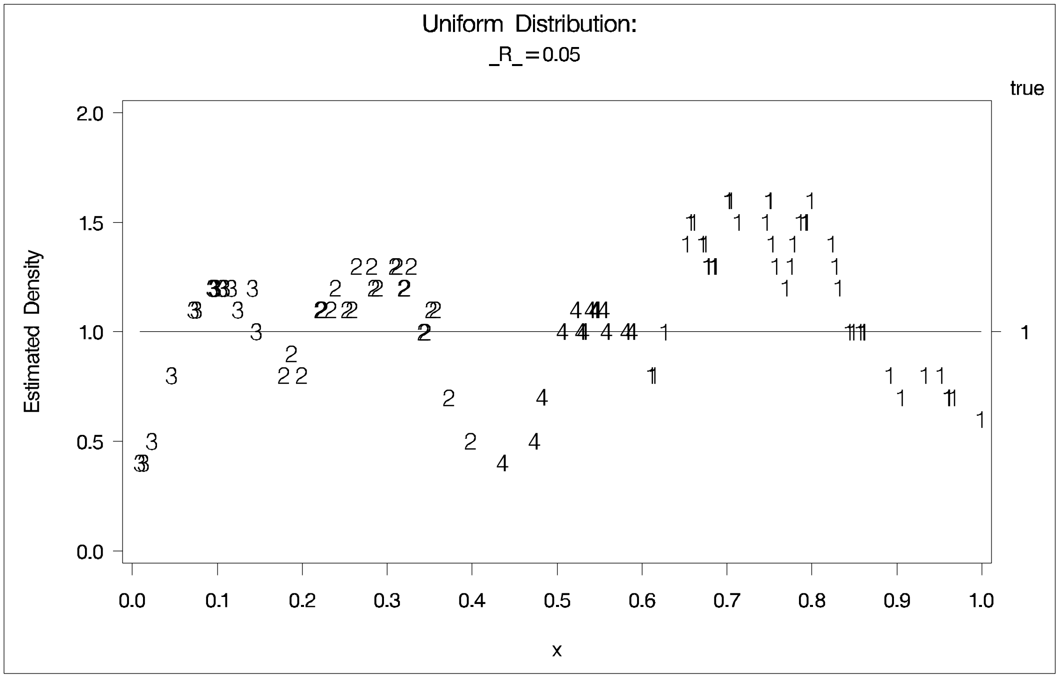


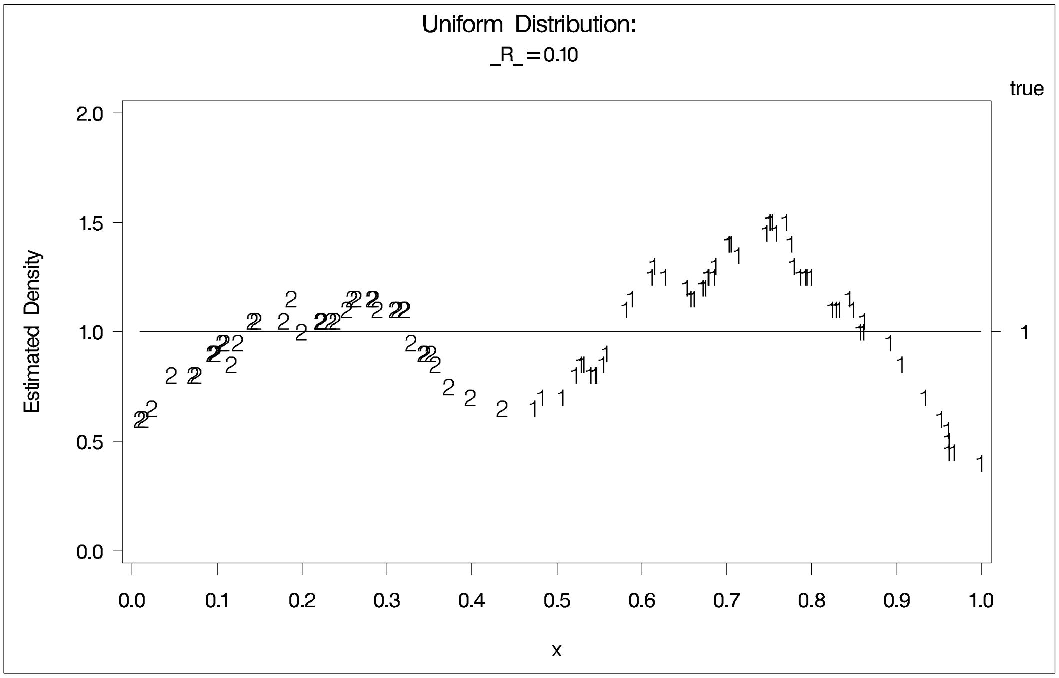

Uniform Distribution:
_K_=20
Estimated Density
x
true
1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

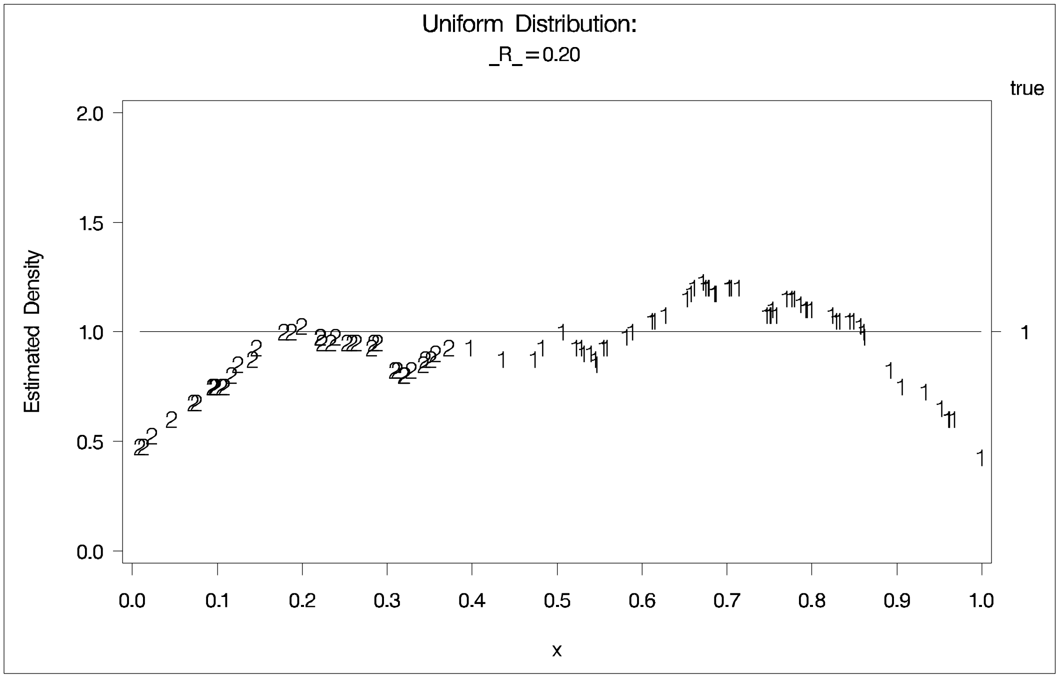

Uniform Distribution:
_K_=40
Estimated Density
x
true
1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

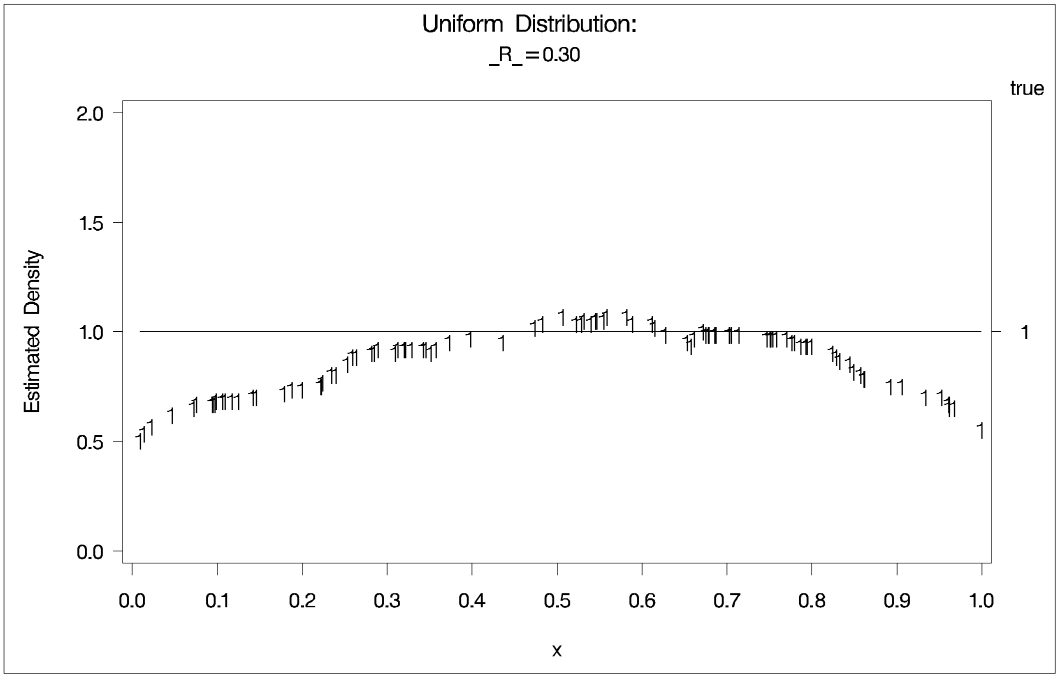


```
                  Modeclus Example with Univariate Distributions
                               Uniform Distribution

                              The MODECLUS Procedure

                                  Cluster Summary
                                              Frequency of
                                 Number of    Unclassified
                          R       Clusters         Objects
                   -----------------------------------
                       0.05              4               0
                        0.1              2               0
                        0.2              2               0
                        0.3              1               0
```


Uniform Distribution:
_R_=0.05
true
Estimated Density
2.0
1.5
1.0
0.5
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
x
1


Uniform Distribution:
_R_=0.10
true
Estimated Density
2.0
1.5
1.0
0.5
0.0
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
x
1





The following statements produce Output 47.1.2:

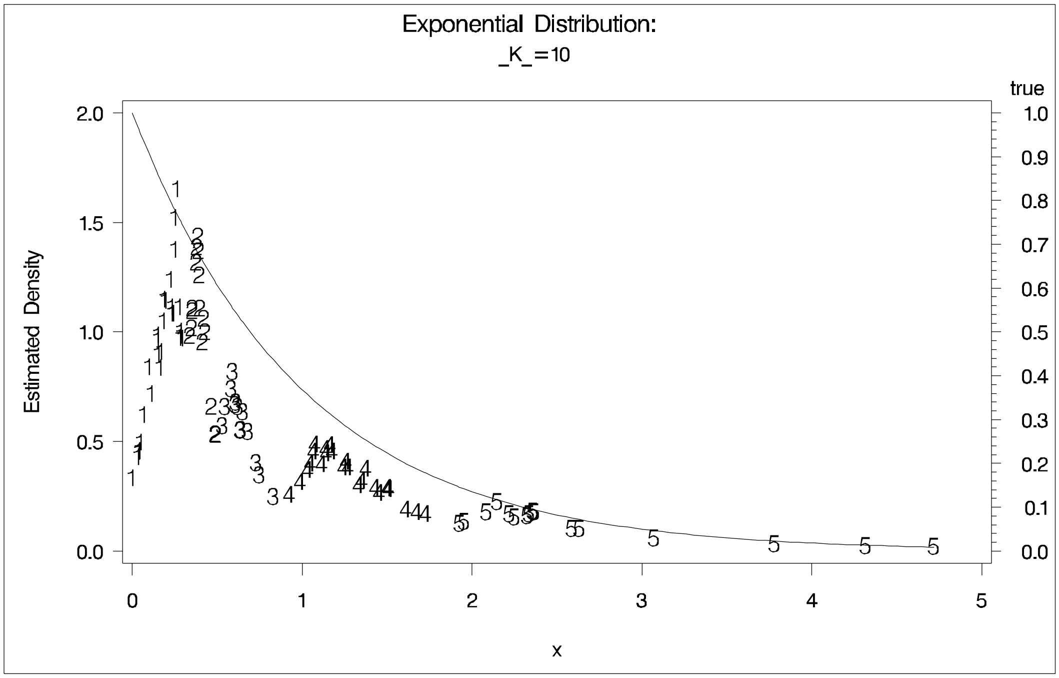
```
title2 ’Exponential Distribution’;
data expon;
   drop n;
   do n=1 to 100;
```

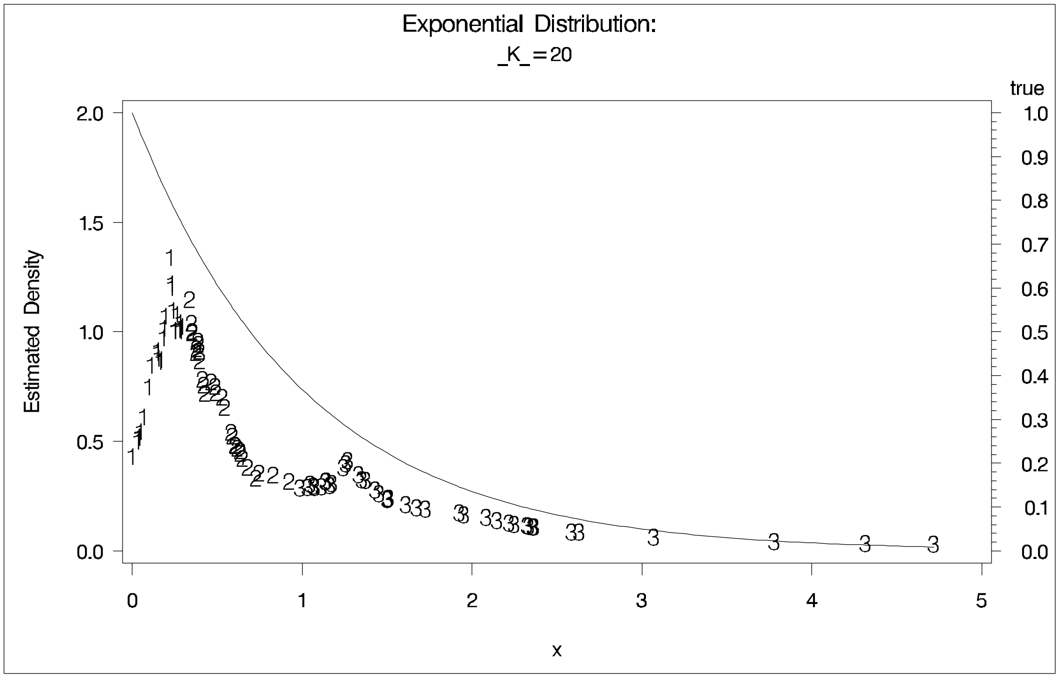
```
      x=ranexp(123);
      true=exp(-x);
      output;
   end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(0 to 2 by 0.5);
axis2 minor=none;

proc modeclus data=expon m=1 k=10 20 40 out=out short;
   var x;

proc gplot;
   plot density*x=cluster /frame cframe=ligr
                          vzero nolegend
                          vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _K_;
run;
/*********************************************/

proc modeclus data=expon m=1 r=.20 .40 .80 out=out short;
   var x;

axis1 label=(angle=90 rotate=0)
      minor=none order=(0 to 1 by 0.5);
proc gplot;
   plot density*x=cluster /frame cframe=ligr
                          vzero nolegend
                          vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _R_;
run;
/*********************************************/

title3 'Different Density-Estimation and Clustering Windows';
proc modeclus data=expon m=1 r=.20 ck=10 20 40
              out=out short;
   var x;

proc gplot;
   plot density*x=cluster /frame cframe=ligr
                          vzero nolegend
                          vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _CK_;
run;
/*********************************************/

title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=expon m=1 r=.20 cascade=1 2 4 am out=out short;
   var x;

proc gplot;
```

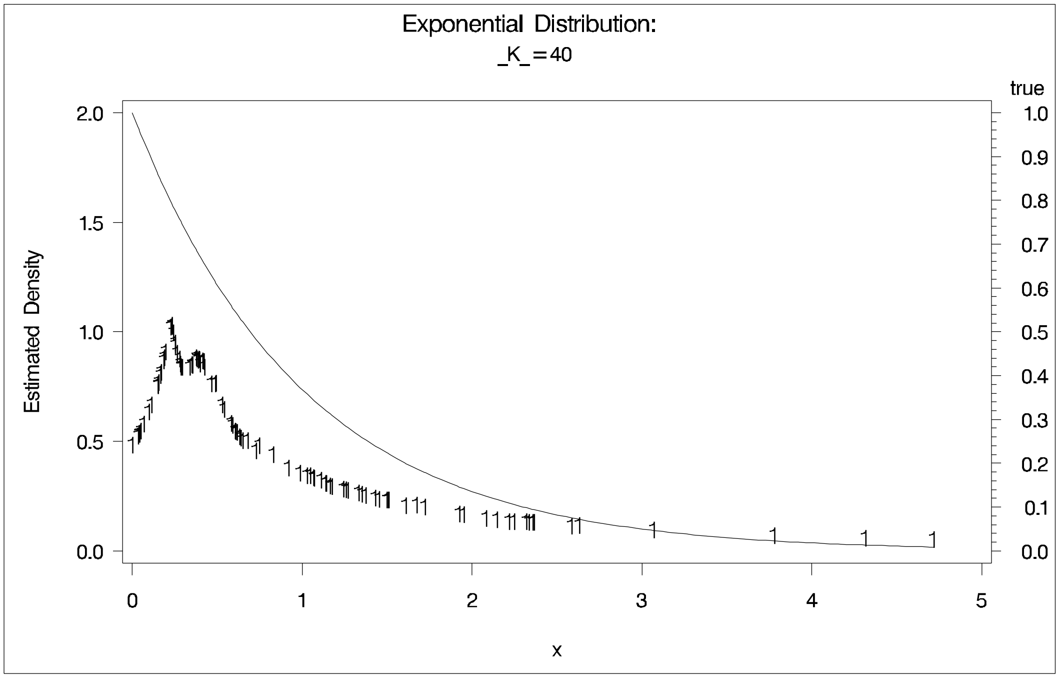
```
   plot density*x=cluster /frame cframe=ligr
                           vzero nolegend
                           vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _R_ _CASCAD_;
run;
```

Output 47.1.2. Cluster Analysis of Sample from an Exponential Distribution

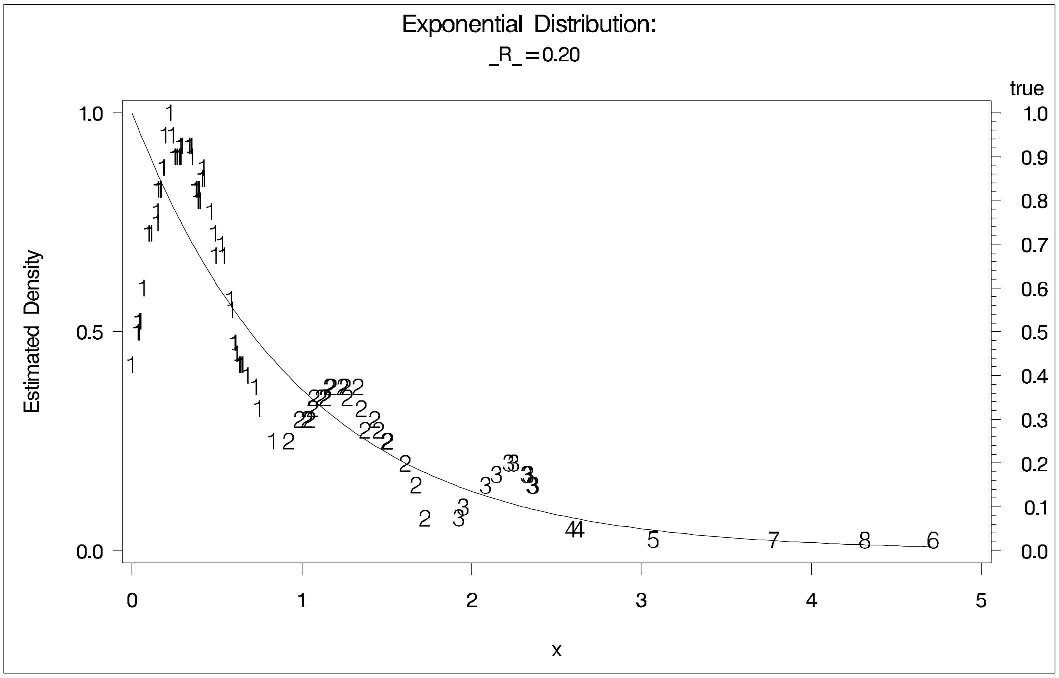
```
Modeclus Example with Univariate Distributions
           Exponential Distribution

            The MODECLUS Procedure

                Cluster Summary
                             Frequency of
                Number of    Unclassified
           K     Clusters         Objects
     ------------------------------------
          10            5               0
          20            3               0
          40            1               0
```

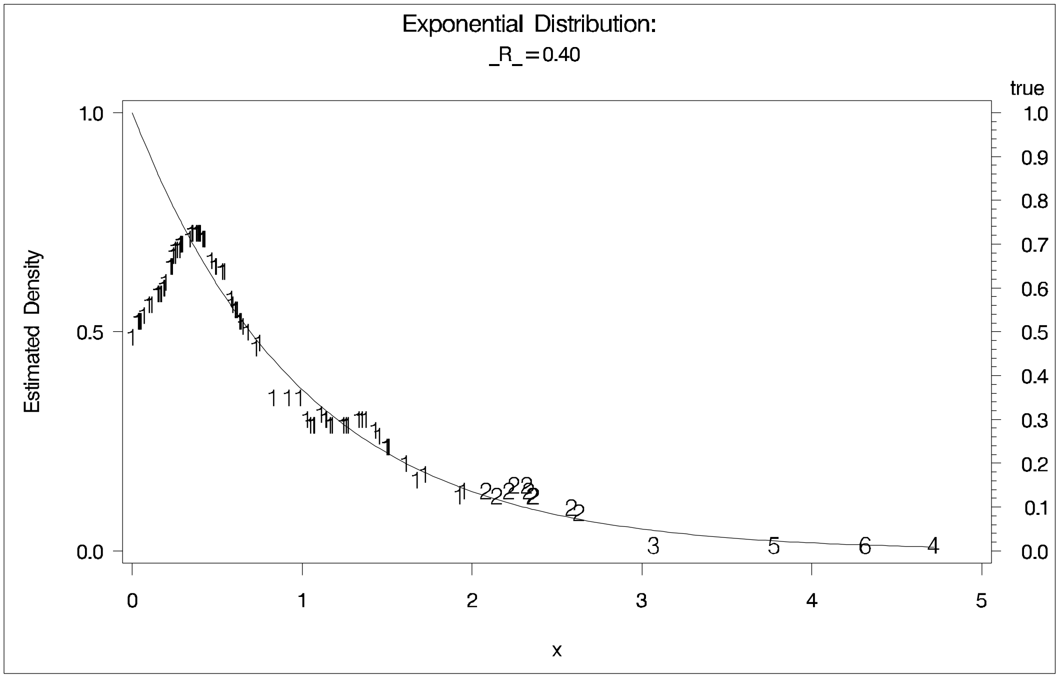


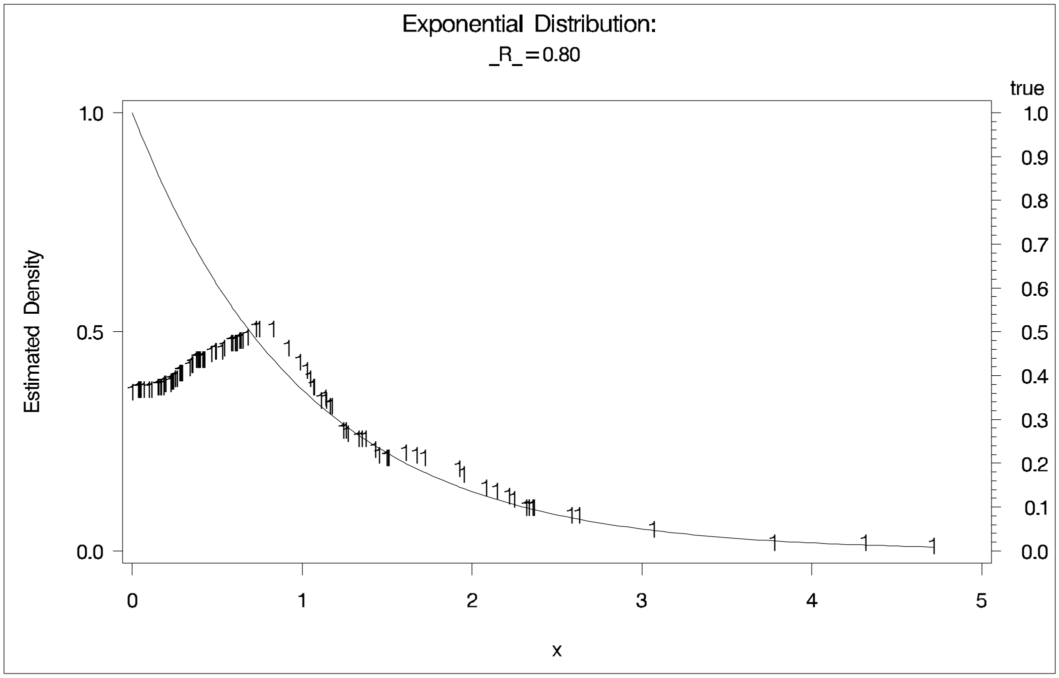

Exponential Distribution:
_K_=20
Estimated Density
true
2.0
1.5
1.0
0.5
0.0
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x


Exponential Distribution:
_K_=40
Estimated Density
true
2.0
1.5
1.0
0.5
0.0
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x

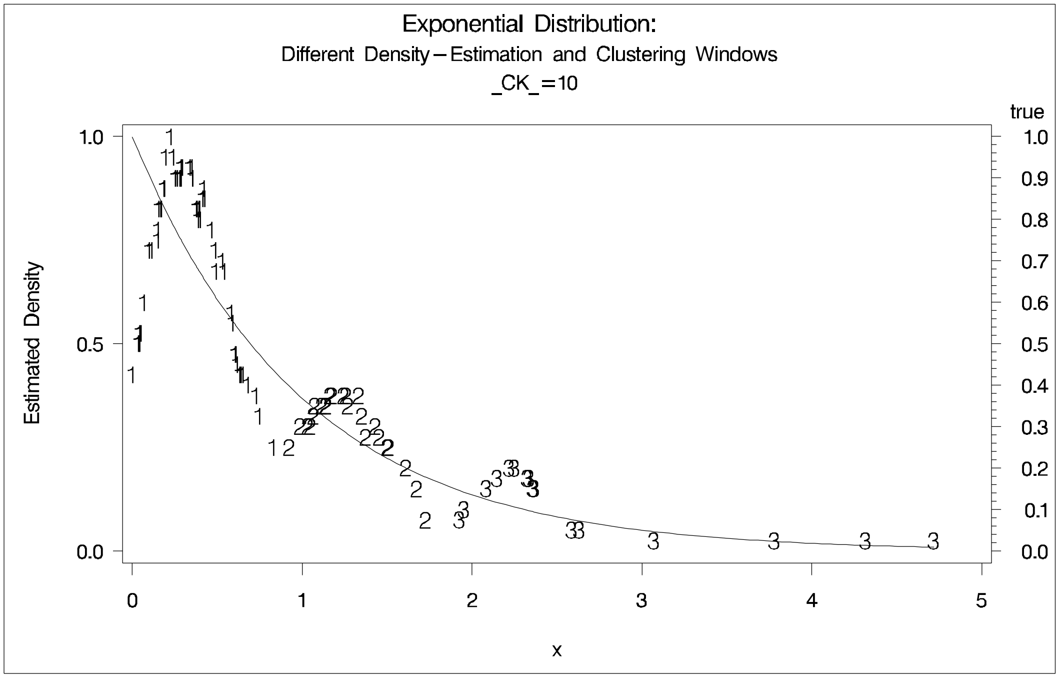
```
                    Modeclus Example with Univariate Distributions
                               Exponential Distribution

                                The MODECLUS Procedure

                                    Cluster Summary
                                                 Frequency of
                                    Number of    Unclassified
                             R       Clusters         Objects
                        ------------------------------------
                           0.2              8               0
                           0.4              6               0
                           0.8              1               0
```

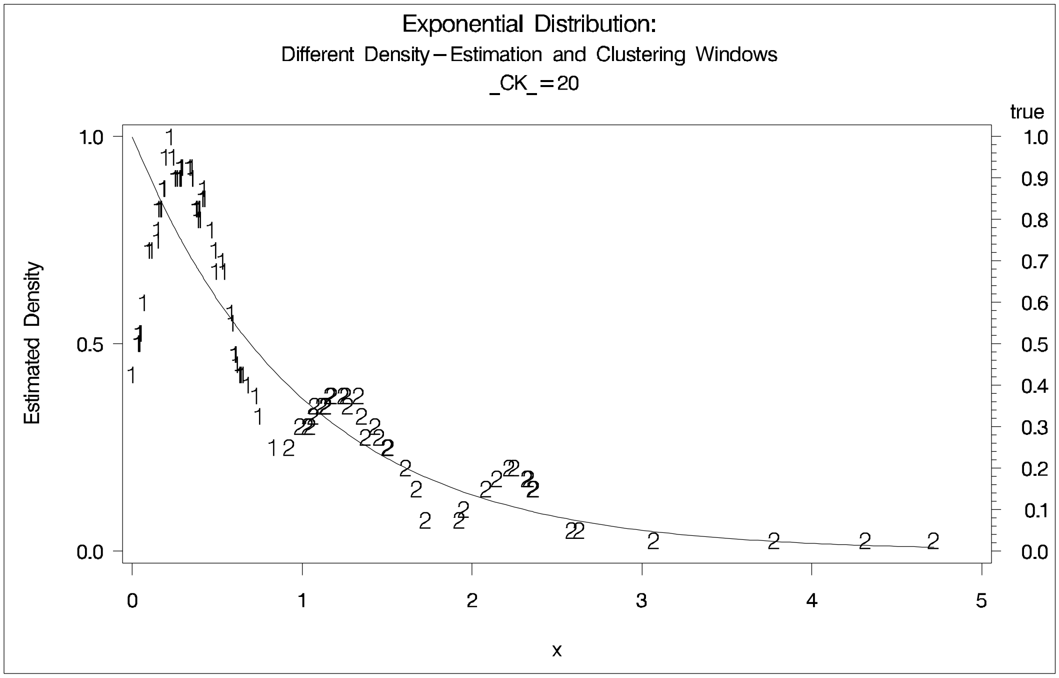


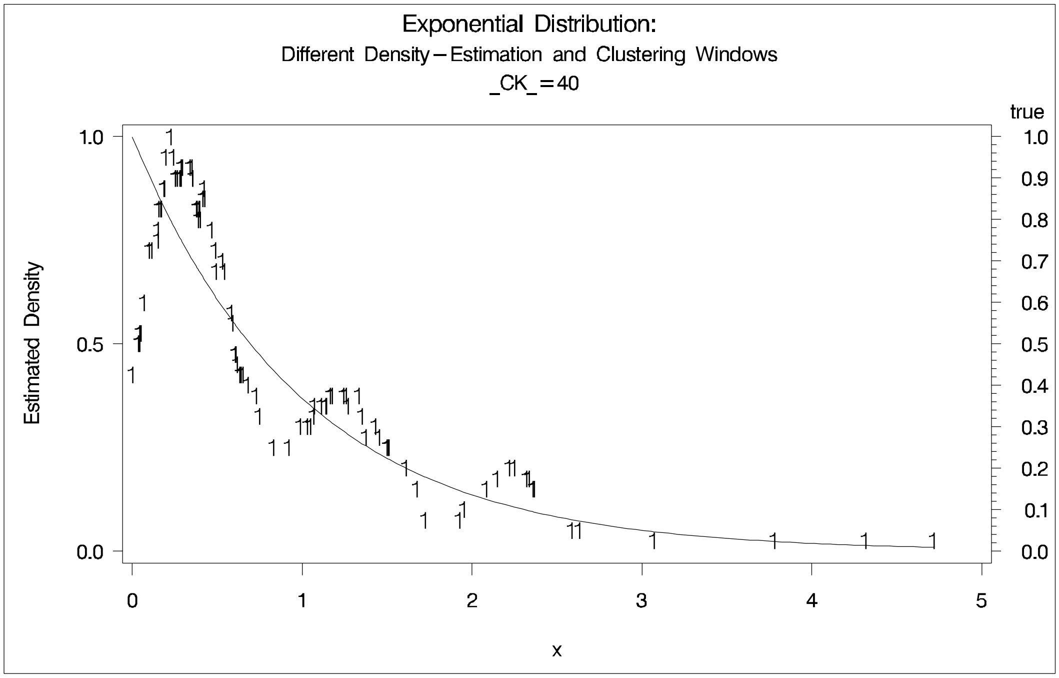

Exponential Distribution:
_R_=0.40
Estimated Density
1.0
0.5
0.0
true
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x


Exponential Distribution:
_R_=0.80
Estimated Density
1.0
0.5
0.0
true
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x

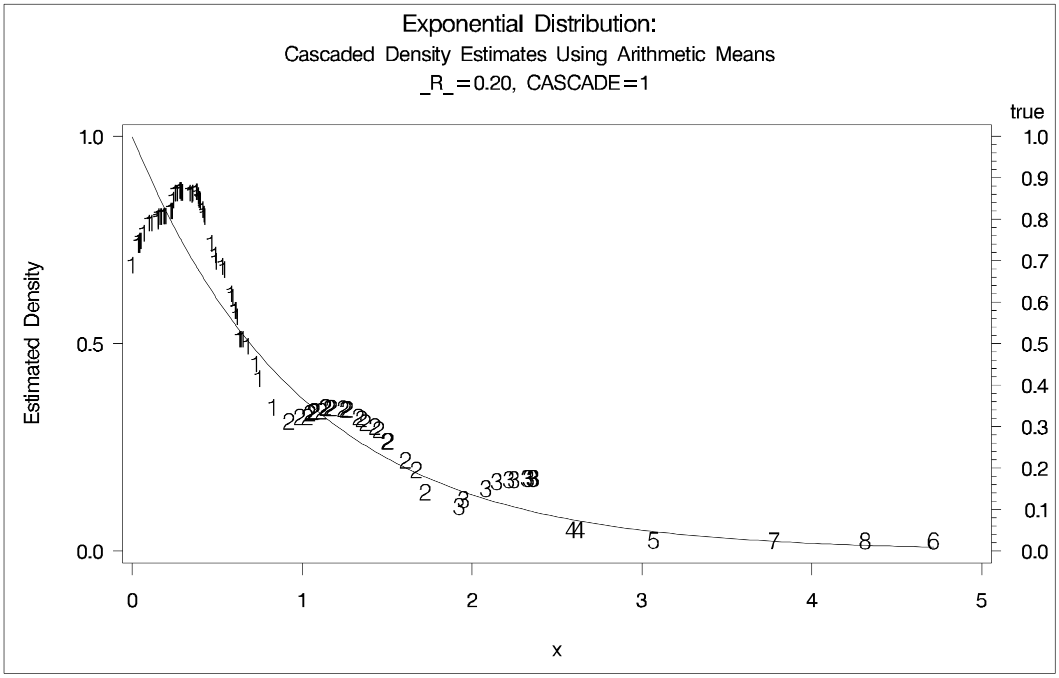
```
                    Modeclus Example with
      Different Density-Estimation and Clustering Windows

                    The MODECLUS Procedure

                       Cluster Summary
                                            Frequency of
                                Number of   Unclassified
         R          CK          Clusters         Objects
      ----------------------------------------------------
       0.2          10                 3               0
       0.2          20                 2               0
       0.2          40                 1               0
```

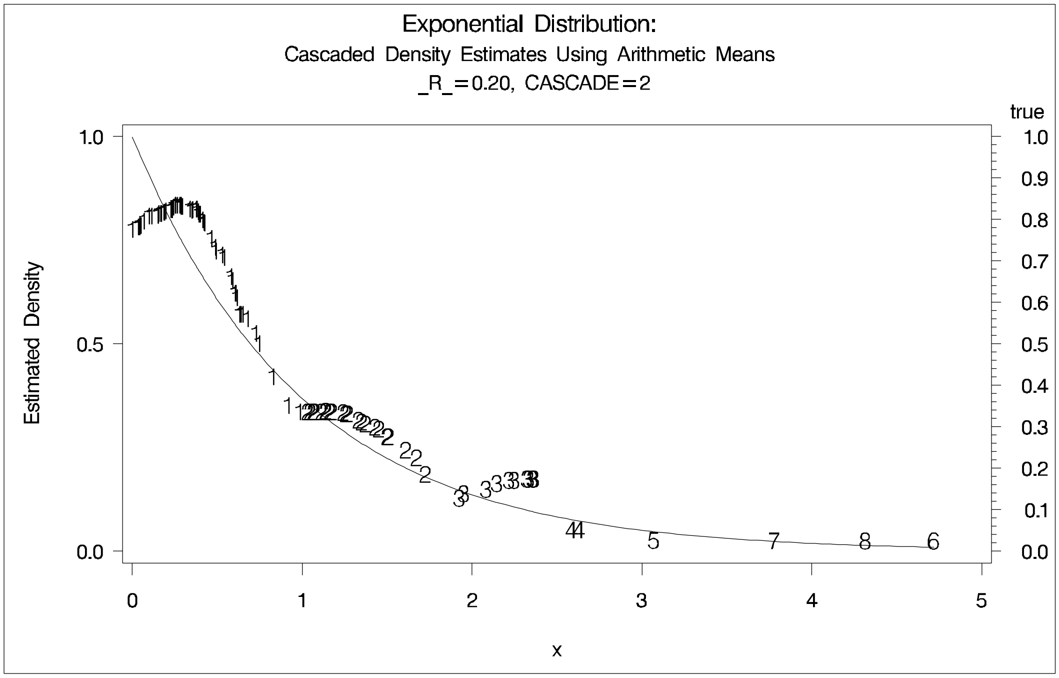


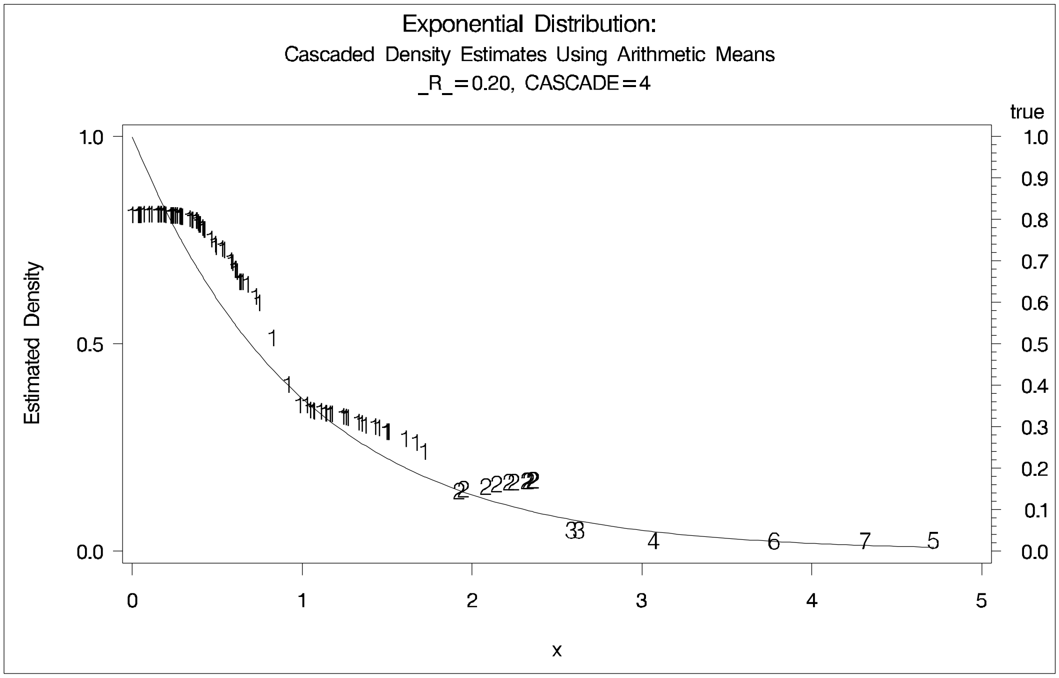

Exponential Distribution:
Different Density−Estimation and Clustering Windows
_CK_=20
Estimated Density
true
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x


Exponential Distribution:
Different Density−Estimation and Clustering Windows
_CK_=40
Estimated Density
true
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5
x

```
                         Modeclus Example with
                Cascaded Density Estimates Using Arithmetic Means

                         The MODECLUS Procedure

                            Cluster Summary
                                                  Frequency of
                                     Number of    Unclassified
           R     Cascade     Clusters         Objects
     -----------------------------------------------
         0.2           1            8               0
         0.2           2            8               0
         0.2           4            7               0
```





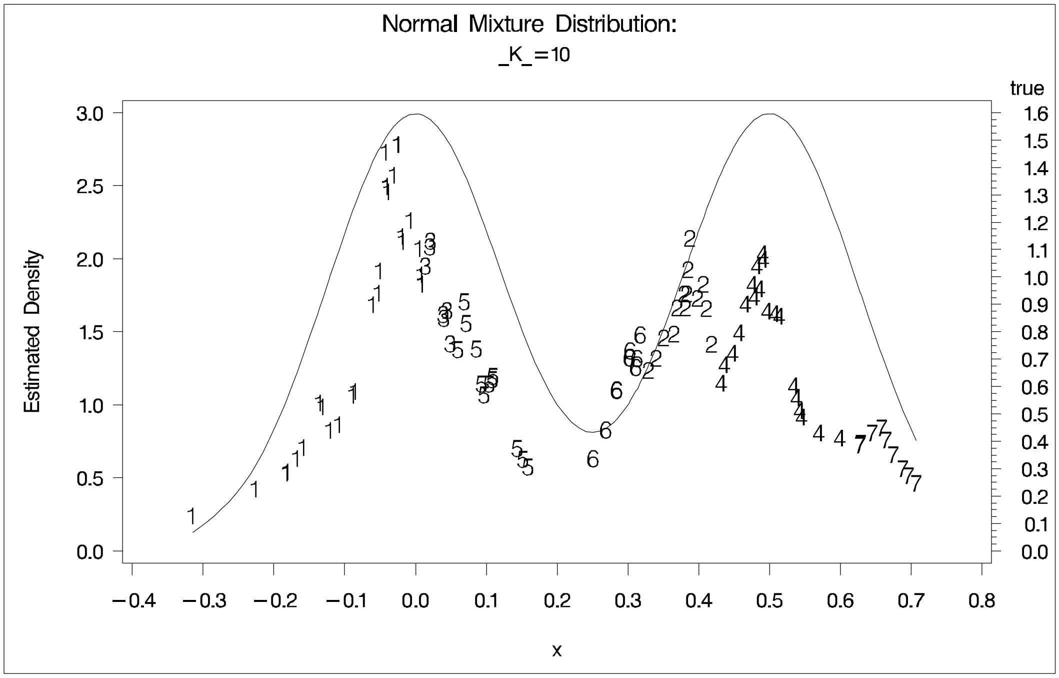


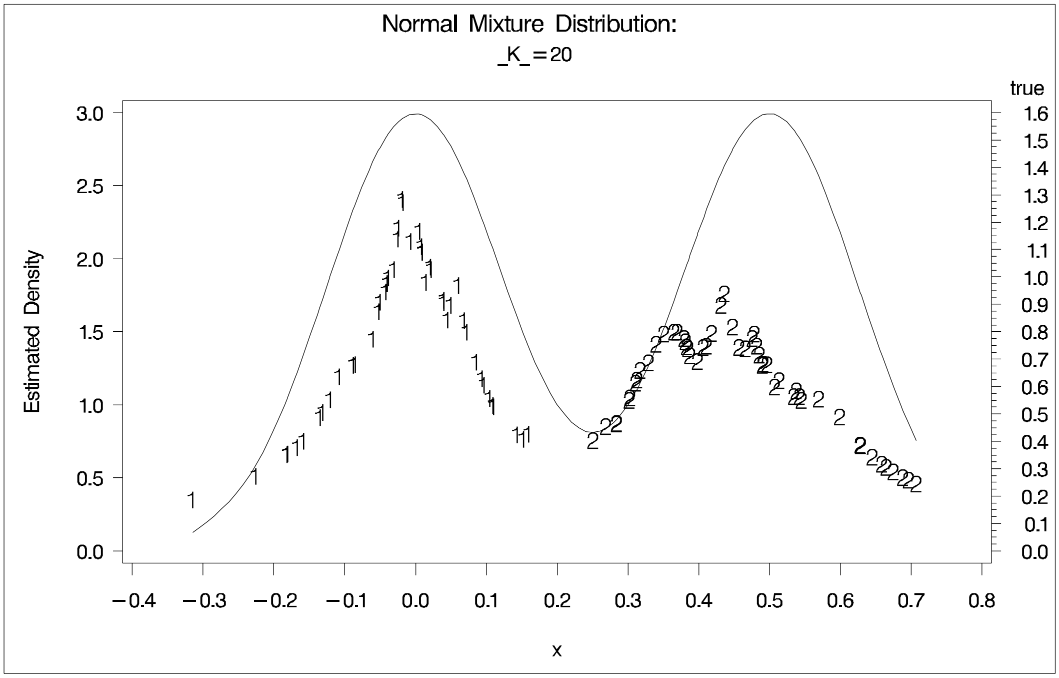
The following statements produce Output 47.1.3:

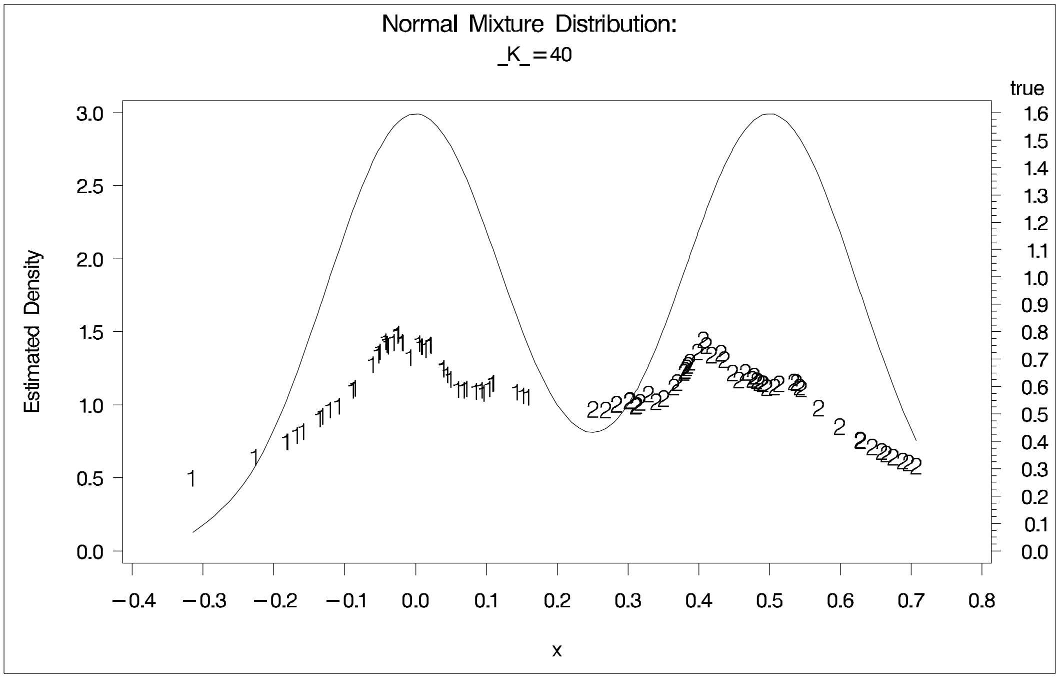
```
title2 'Normal Mixture Distribution';
data normix;
   drop n sigma;
   sigma=.125;
   do n=1 to 100;
      x=rannor(456)*sigma+mod(n,2)/2;
      true=exp(-.5*(x/sigma)**2)+exp(-.5*((x-.5)/sigma)**2);
      true=.5*true/(sigma*sqrt(2*3.1415926536));
      output;
   end;

axis1 label=(angle=90 rotate=0) minor=none order=(0 to 3 by 0.5);
axis2 minor=none;

proc modeclus data=normix m=1 k=10 20 40 60 out=out short;
   var x;

proc gplot;
   plot density*x=cluster /frame cframe=ligr
                           vzero nolegend
                           vaxis=axis1 haxis=axis2;
   plot2 true*x=9;
   by _K_;
run;
/*********************************************/

proc modeclus data=normix m=1 r=.05 .10 .20 .30 out=out short;
   var x;

proc gplot;
   plot density*x=cluster /frame cframe=ligr
                           vzero nolegend
                           vaxis=axis1 haxis=axis2 ;
   plot2 true*x=9;
   by _R_;
run;
/*********************************************/

title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=normix m=1 r=.05 cascade=1 2 4 am out=out short;
   var x;

axis1 label=(angle=90 rotate=0)
      minor=none order=(0 to 2 by 0.5);
proc gplot;
   plot density*x=cluster /frame cframe=ligr
                           vzero  nolegend
                           vaxis=axis1 haxis=axis2 ;
   plot2 true*x=9;
   by _R_ _CASCAD_;
run;
```

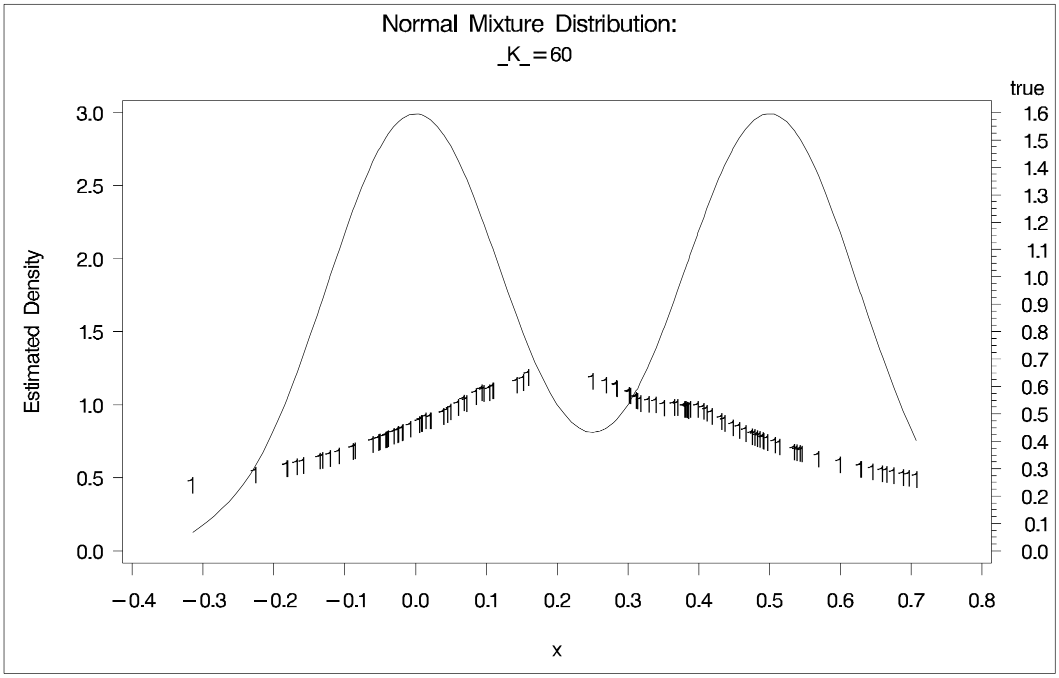
Output 47.1.3. Cluster Analysis of Sample from a Bimodal Mixture of Two Normal Distributions

```
                 Modeclus Example with
               Normal Mixture Distribution

                 The MODECLUS Procedure

                     Cluster Summary
                                 Frequency of
                     Number of   Unclassified
             K        Clusters        Objects
       -----------------------------------
            10               7              0
            20               2              0
            40               2              0
            60               1              0
```

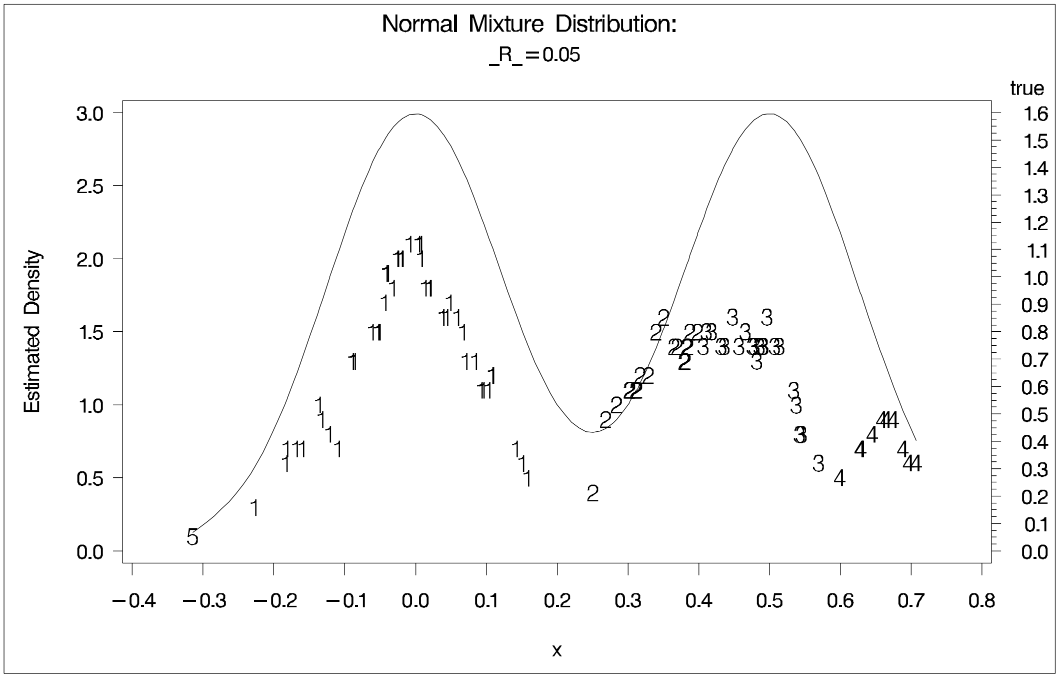


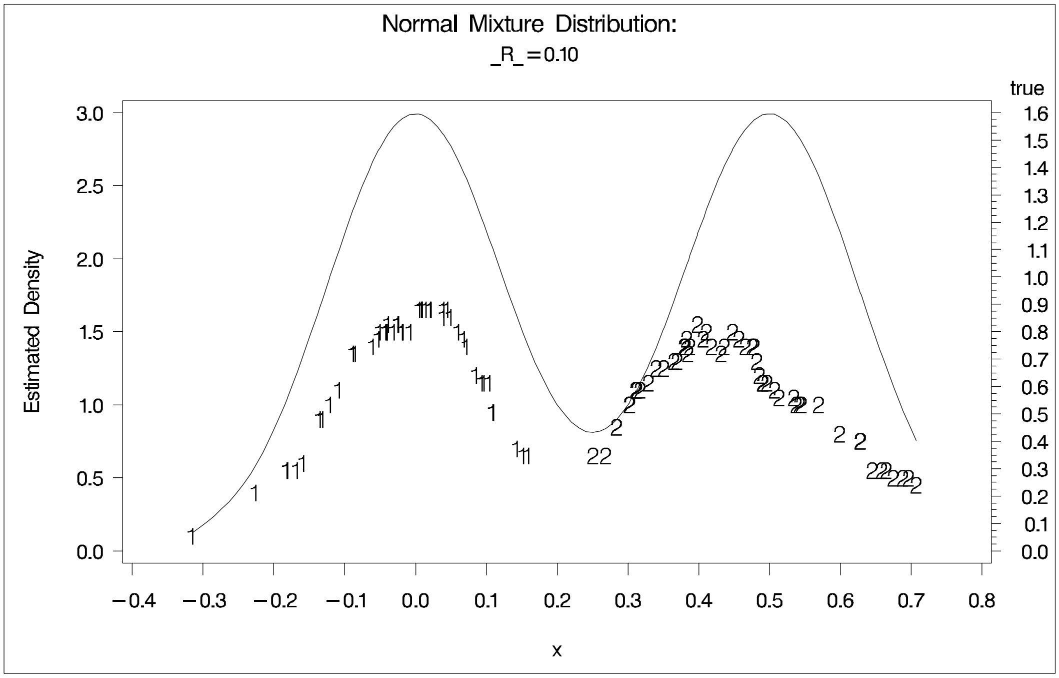

Normal Mixture Distribution:
_K_=20
Estimated Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
true
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
x

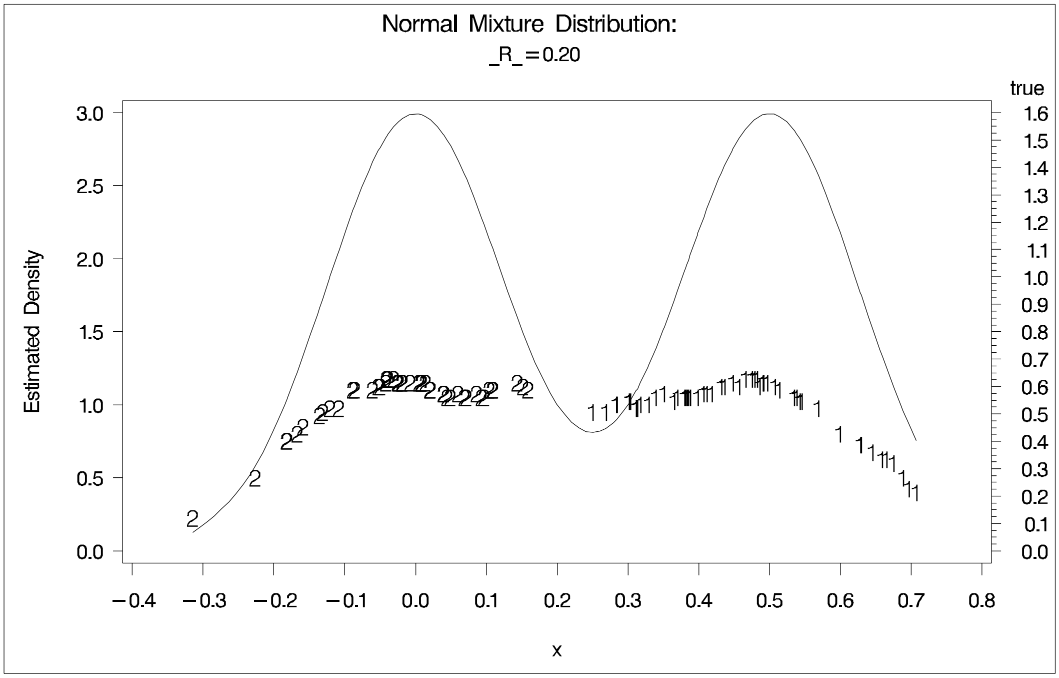

Normal Mixture Distribution:
_K_=40
Estimated Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
true
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4 −0.3 −0.2 −0.1 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8
x

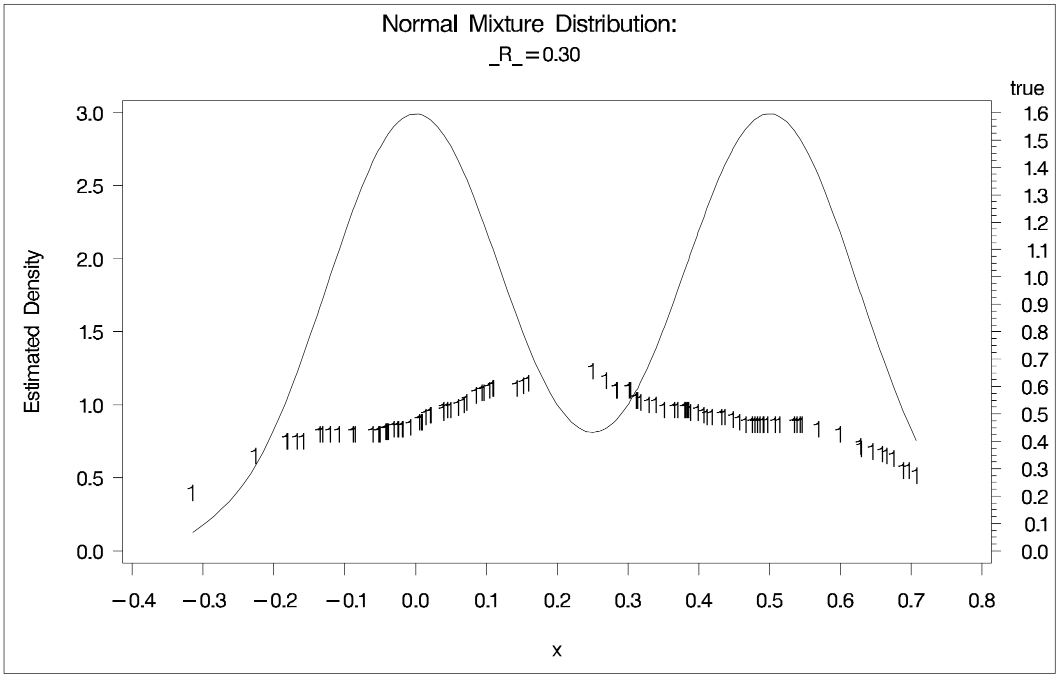


```
                     Modeclus Example with
                  Normal Mixture Distribution

                     The MODECLUS Procedure

                         Cluster Summary
                                     Frequency of
                         Number of   Unclassified
                  R       Clusters        Objects
              ------------------------------------
               0.05              5              0
                0.1              2              0
                0.2              2              0
                0.3              1              0
```

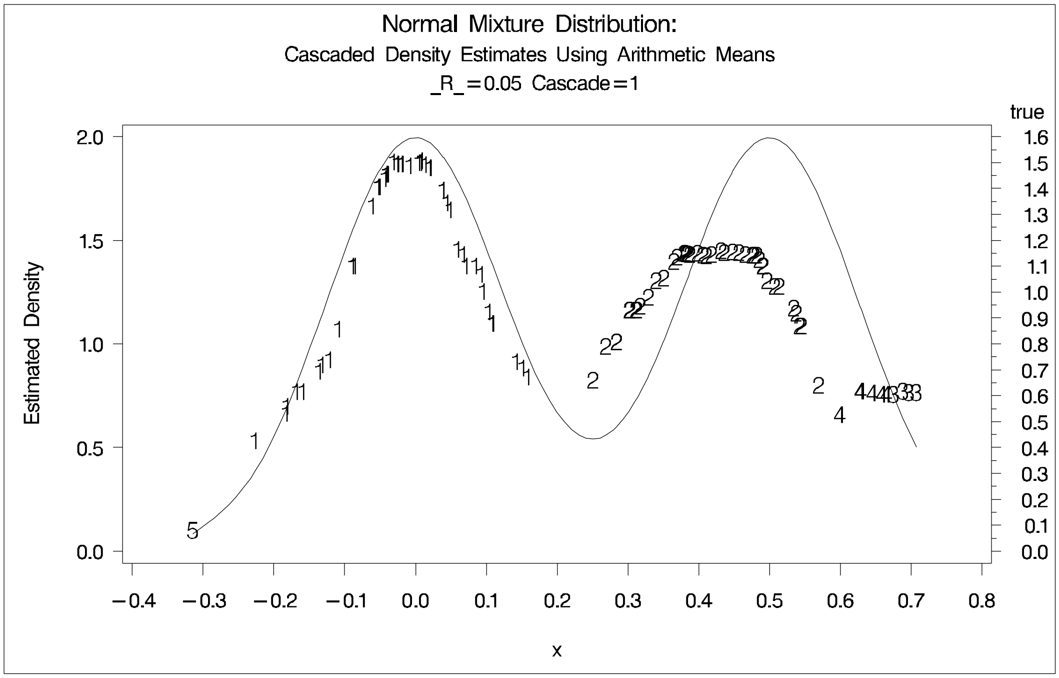

Normal Mixture Distribution:
_R_=0.05
Estimated Density
true
x
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

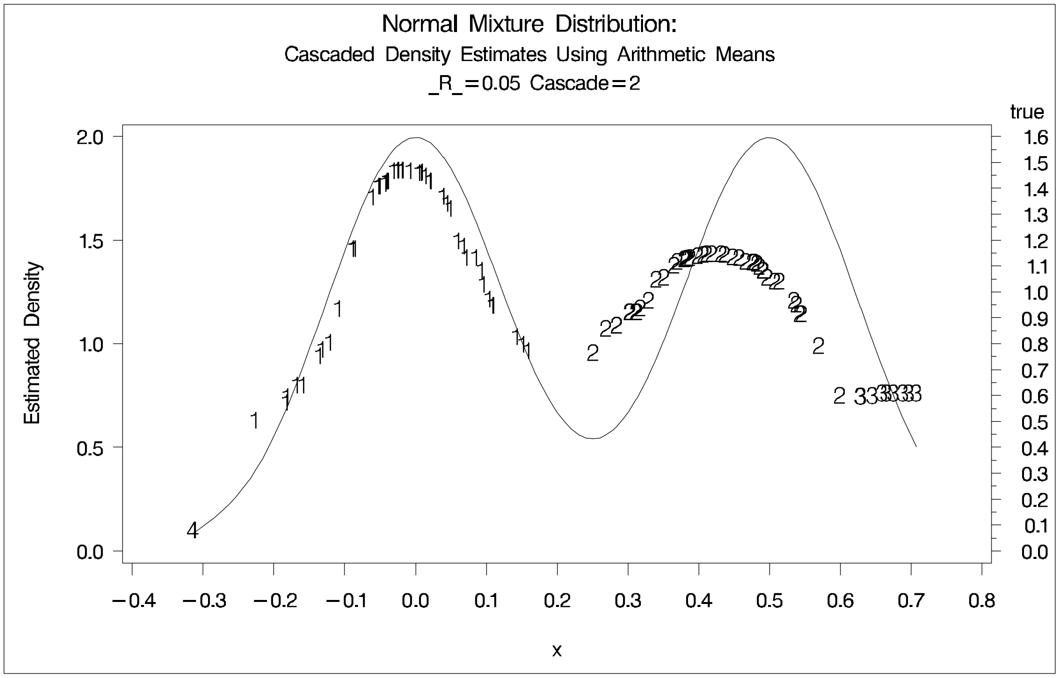

Normal Mixture Distribution:
_R_=0.10
Estimated Density
true
x
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8

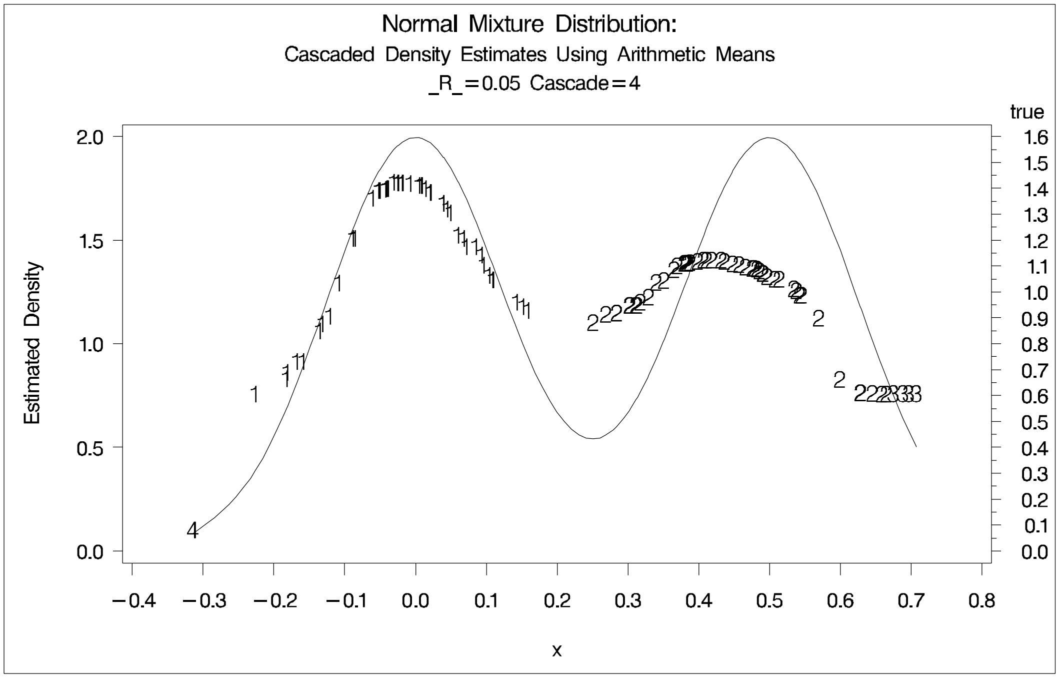

Normal Mixture Distribution:
_R_=0.20
Estimated Density
true
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x


Normal Mixture Distribution:
_R_=0.30
Estimated Density
true
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x

```
                      Modeclus Example with
                   Normal Mixture Distribution
         Cascaded Density Estimates Using Arithmetic Means

                      The MODECLUS Procedure

                         Cluster Summary
                                               Frequency of
                                   Number of   Unclassified
           R    Cascade     Clusters        Objects
        ----------------------------------------------
        0.05          1            5              0
        0.05          2            4              0
        0.05          4            4              0
```




Normal Mixture Distribution:
Cascaded Density Estimates Using Arithmetic Means
_R_=0.05 Cascade=2
Estimated Density
2.0
1.5
1.0
0.5
0.0
true
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x


Normal Mixture Distribution:
Cascaded Density Estimates Using Arithmetic Means
_R_=0.05 Cascade=4
Estimated Density
2.0
1.5
1.0
0.5
0.0
true
1.6
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
x

Example 47.2. Cluster Analysis of Flying Mileages between Ten American Cities

This example uses distance data and illustrates the use of the TRANSPOSE procedure and the DATA step to fill in the upper triangle of the distance matrix. The results are displayed in Output 47.2.1 through Output 47.2.2.

The following statements produce Output 47.2.1:

```
title ’Modeclus Analysis of 10 American Cities’;
title2 ’Based on Flying Mileages’;
options ls=90;

data mileages(type=distance);
   input (ATLANTA CHICAGO DENVER HOUSTON LOSANGELES
   MIAMI NEWYORK SANFRAN SEATTLE WASHDC) (5.)
   @53 CITY $15.;
   datalines;
   0                                                ATLANTA
 587    0                                           CHICAGO
1212  920    0                                      DENVER
 701  940  879    0                                 HOUSTON
1936 1745  831 1374    0                            LOS ANGELES
 604 1188 1726  968 2339    0                       MIAMI
 748  713 1631 1420 2451 1092    0                  NEW YORK
2139 1858  949 1645  347 2594 2571    0             SAN FRANCISCO
2182 1737 1021 1891  959 2734 2408  678    0        SEATTLE
 543  597 1494 1220 2300  923  205 2442 2329    0   WASHINGTON D.C.
;

*-----Fill in Upper Triangle of Distance Matrix---------------;
proc transpose out=tran;
   copy CITY;
data mileages(type=distance);
   merge mileages tran;
   array var ATLANTA--WASHDC;
   array col col1-col10;
   drop col1-col10 _name_;
   do over var;
      var=sum(var,col);
   end;

*-----Clustering with K-Nearest-Neighbor Density Estimates-----;
proc modeclus data=mileages all m=1 k=3;
   id CITY;
run;
```

Output 47.2.1. Clustering with K-Nearest-Neighbor Density Estimates

```
                     Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure

                             Nearest Neighbor List
               CITY              Neighbor                  Distance
               --------------------------------------------------
               ATLANTA           WASHINGTON D.C.     543.0000000
                                 CHICAGO             587.0000000
               --------------------------------------------------
               CHICAGO           ATLANTA             587.0000000
                                 WASHINGTON D.C.     597.0000000
               --------------------------------------------------
               DENVER            LOS ANGELES         831.0000000
                                 HOUSTON             879.0000000
               --------------------------------------------------
               HOUSTON           ATLANTA             701.0000000
                                 DENVER              879.0000000
               --------------------------------------------------
               LOS ANGELES       SAN FRANCISCO       347.0000000
                                 DENVER              831.0000000
               --------------------------------------------------
               MIAMI             ATLANTA             604.0000000
                                 WASHINGTON D.C.     923.0000000
               --------------------------------------------------
               NEW YORK          WASHINGTON D.C.     205.0000000
                                 CHICAGO             713.0000000
               --------------------------------------------------
               SAN FRANCISCO     LOS ANGELES         347.0000000
                                 SEATTLE             678.0000000
               --------------------------------------------------
               SEATTLE           SAN FRANCISCO       678.0000000
                                 LOS ANGELES         959.0000000
               --------------------------------------------------
               WASHINGTON D.C.   NEW YORK            205.0000000
                                 ATLANTA             543.0000000
```

```
                     Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure
                                K=3  METHOD=1

                Boundary Objects         -Cluster Proportions-
          CITY                     Density    Cluster        1        2

          DENVER             0.0001706485          2     0.486    0.514
          HOUSTON            0.0001706485          1     0.600    0.400


                               Cluster Statistics
                                     Maximum                    Estimated
                                   Estimated      Boundary         Saddle
         Cluster     Frequency       Density     Frequency        Density
         ---------------------------------------------------------------
         1                   6    0.00027624             1     0.00017065
         2                   4    0.00022124             1     0.00017065
```

```
      Modeclus Analysis of 10 American Cities
             Based on Flying Mileages

              The MODECLUS Procedure

                  Cluster Summary
                               Frequency of
                  Number of    Unclassified
             K     Clusters         Objects
       ------------------------------------
             3            2               0
```

The following statements produce Output 47.2.2:

```
*------Clustering with Uniform Kernel Density Estimates--------;
proc modeclus data=mileages all m=1 r=600 800;
   id CITY;
run;
```

Output 47.2.2. Clustering with Uniform Kernel Density Estimates

```
      Modeclus Analysis of 10 American Cities
             Based on Flying Mileages

              The MODECLUS Procedure

              Nearest Neighbor List
CITY               Neighbor                 Distance
--------------------------------------------------
ATLANTA            WASHINGTON D.C.     543.0000000
                   CHICAGO             587.0000000
                   MIAMI               604.0000000
                   HOUSTON             701.0000000
                   NEW YORK            748.0000000
--------------------------------------------------
CHICAGO            ATLANTA             587.0000000
                   WASHINGTON D.C.     597.0000000
                   NEW YORK            713.0000000
--------------------------------------------------
HOUSTON            ATLANTA             701.0000000
--------------------------------------------------
LOS ANGELES        SAN FRANCISCO       347.0000000
--------------------------------------------------
MIAMI              ATLANTA             604.0000000
--------------------------------------------------
NEW YORK           WASHINGTON D.C.     205.0000000
                   CHICAGO             713.0000000
                   ATLANTA             748.0000000
--------------------------------------------------
SAN FRANCISCO      LOS ANGELES         347.0000000
                   SEATTLE             678.0000000
--------------------------------------------------
SEATTLE            SAN FRANCISCO       678.0000000
--------------------------------------------------
WASHINGTON D.C.    NEW YORK            205.0000000
                   ATLANTA             543.0000000
                   CHICAGO             597.0000000
```

```
                     Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure
                                R=600  METHOD=1

                               No Boundary Objects


                               Cluster Statistics
                                     Maximum                        Estimated
                                   Estimated      Boundary             Saddle
         Cluster      Frequency      Density     Frequency            Density
         ----------------------------------------------------------------
         1                    4   0.00033333             0                  .
         2                    2   0.00016667             0                  .
         3                    1   0.00008333             0                  .
         4                    1   0.00008333             0                  .
         5                    1   0.00008333             0                  .
         6                    1   0.00008333             0                  .



                     Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure
                                R=800  METHOD=1

                               No Boundary Objects


                               Cluster Statistics
                                     Maximum                        Estimated
                                   Estimated      Boundary             Saddle
         Cluster      Frequency      Density     Frequency            Density
         ----------------------------------------------------------------
         1                    6     0.000375             0                  .
         2                    3    0.0001875             0                  .
         3                    1    0.0000625             0                  .
```

```
                     Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure

                                 Cluster Summary
                                             Frequency of
                                 Number of   Unclassified
                           R      Clusters        Objects
                      ------------------------------------
                         600             6              0
                         800             3              0
```

The following statements produce Output 47.2.3:

```
*------Uniform Kernel Density Estimates, Clustering
          Neighborhoods extended to nearest neighbor-------------;
proc modeclus data=mileages list m=1 ck=2 r=600 800;
   id CITY;
run;
```

Output 47.2.3. Uniform Kernel Density Estimates, Clustering Neighborhoods Extended to Nearest Neighbor

```
                      Modeclus Analysis of 10 American Cities
                             Based on Flying Mileages

                              The MODECLUS Procedure
                                CK=2  R=600  METHOD=1

                                Cluster Statistics
                                      Maximum                       Estimated
                                    Estimated      Boundary            Saddle
        Cluster     Frequency         Density     Frequency           Density
        ---------------------------------------------------------------------
        1                   6      0.00033333             0                 .
        2                   4      0.00016667             0                 .
```

```
                      Modeclus Analysis of 10 American Cities
                             Based on Flying Mileages

                              The MODECLUS Procedure
                                CK=2  R=800  METHOD=1

                                Cluster Statistics
                                      Maximum                       Estimated
                                    Estimated      Boundary            Saddle
        Cluster     Frequency         Density     Frequency           Density
        ---------------------------------------------------------------------
        1                   6        0.000375             0                 .
        2                   4       0.0001875             0                 .
```

```
                      Modeclus Analysis of 10 American Cities
                             Based on Flying Mileages

                              The MODECLUS Procedure

                                  Cluster Summary
                                                   Frequency of
                                       Number of   Unclassified
                      R          CK     Clusters        Objects
                ----------------------------------------------
                    600           2            2              0
                    800           2            2              0
```

Example 47.3. Cluster Analysis with Significance Tests

This example uses artificial data containing two clusters. One cluster is from a circular bivariate normal distribution. The other is a ring-shaped cluster that completely surrounds the first cluster. Without significance tests, the ring is divided into several sample clusters for any degree of smoothing that yields reasonable density estimates. The JOIN= option puts the ring back together. Output 47.3.1 displays a short summary generated from the first PROC MODECLUS statement. Output 47.3.2 contains a series of tables produced from the second PROC MODECLUS statement. The lack of p-value in the JOIN= option makes joining continue until only one cluster remains (see the description of the JOIN= option on page 2866). The cluster memberships are then plotted as displayed in Output 47.3.3.

```
title  'Modeclus Analysis with the JOIN= option';
title2 'A Normal Cluster Surrounded by a Ring Cluster';
```

```
options ls=120 ps=38;

data circle; keep x y;
   c=1;
   do n=1 to 30;
      x=rannor(5);
      y=rannor(5);
      output;
   end;

   c=2;
   do n=1 to 300;
      x=rannor(5);
      y=rannor(5);
      z=rannor(5)+8;
      l=z/sqrt(x**2+y**2);
      x=x*l;
      y=y*l;
      output;
   end;

axis1 label=(angle=90 rotate=0) minor=none
      order=(-10 to 10 by 5);
axis2 minor=none order=(-15 to 15 by 5);

proc modeclus data=circle m=1 r=1 to 3.5 by .25 join=20 short;
proc modeclus data=circle m=1 r=2.5 join out=out;

proc gplot data=out;
   plot y*x=cluster/frame cframe=ligr
                    vzero nolegend
                    vaxis=axis1 haxis=axis2 ;
   by _NJOIN_;
run;
```

Output 47.3.1. Significance Tests with the JOIN=20 and SHORT Options

```
                         Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure

                                     Cluster Summary
                         Number of                                  Frequency of
                          Clusters      Maximum     Number of       Unclassified
                    R       Joined      P-value      Clusters            Objects
          ------------------------------------------------------------------------
                    1           36       0.9339             1                301
                 1.25           20       0.7131             1                301
                  1.5           10       0.3296             1                300
                 1.75            5       0.1990             2                  0
                    2            5       0.0683             2                  0
                 2.25            3       0.0504             2                  0
                  2.5            4       0.0301             2                  0
                 2.75            3       0.0585             2                  0
                    3            5       0.0003             1                  0
                 3.25            4       0.1923             2                  0
                  3.5            4       0.0000             1                  0
```

Output 47.3.2. Significance Tests with the JOIN Option

```
                         Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                     R=2.5  METHOD=1

                      Cluster Statistics
                            Maximum                         Estimated    -------------Saddle Test: Version 92.7------------
                          Estimated       Boundary             Saddle        Mode     Saddle    Overlap                   Approx
Cluster       Frequency     Density      Frequency            Density       Count      Count      Count            Z     P-value
-------------------------------------------------------------------------------------------------------------------------------
1                   103  0.00617328             22         0.00308664          39         19          0        2.495      0.5055
2                    71  0.00571029             20          0.0043213          36         27          9        1.193       0.999
3                    53  0.00509296             18         0.00401263          32         25         10        0.986      0.9999
4                    45  0.00478429             19         0.00354964          30         22         14        1.429      0.9924
5                    30  0.00462996              0                  .          29          0          .        3.611      0.0301
6                    28  0.00370397             17         0.00354964          23         22          9        0.000           1
```

```
                         Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                     R=2.5  METHOD=1

                      Cluster Statistics
                            Maximum                         Estimated    -------------Saddle Test: Version 92.7------------
                          Estimated       Boundary             Saddle        Mode     Saddle    Overlap                   Approx
Cluster       Frequency     Density      Frequency            Density       Count      Count      Count            Z     P-value
-------------------------------------------------------------------------------------------------------------------------------
1                   103  0.00617328             22         0.00308664          39         19          0        2.495      0.5055
2                    71  0.00571029             20          0.0043213          36         27          9        1.193       0.999
3                    53  0.00509296             18         0.00401263          32         25         10        0.986      0.9999
4                    73  0.00478429             13         0.00293231          30         18          0        1.588      0.9778
5                    30  0.00462996              0                  .          29          0          .        3.611      0.0301
```

```
                        Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                    R=2.5  METHOD=1

                        Cluster Statistics
                              Maximum                  Estimated    -------------Saddle Test: Version 92.7------------
                            Estimated      Boundary       Saddle      Mode    Saddle    Overlap                 Approx
Cluster     Frequency         Density     Frequency      Density     Count     Count      Count            Z   P-value
----------------------------------------------------------------------------------------------------------------------
1                 156      0.00617328            17   0.00246931        39        15          0        3.130    0.1318
2                  71      0.00571029            20    0.0043213        36        27          9        1.193     0.999
3                  73      0.00478429            13   0.00293231        30        18          0        1.588    0.9778
4                  30      0.00462996             0            .        29         0          .        3.611    0.0301
```

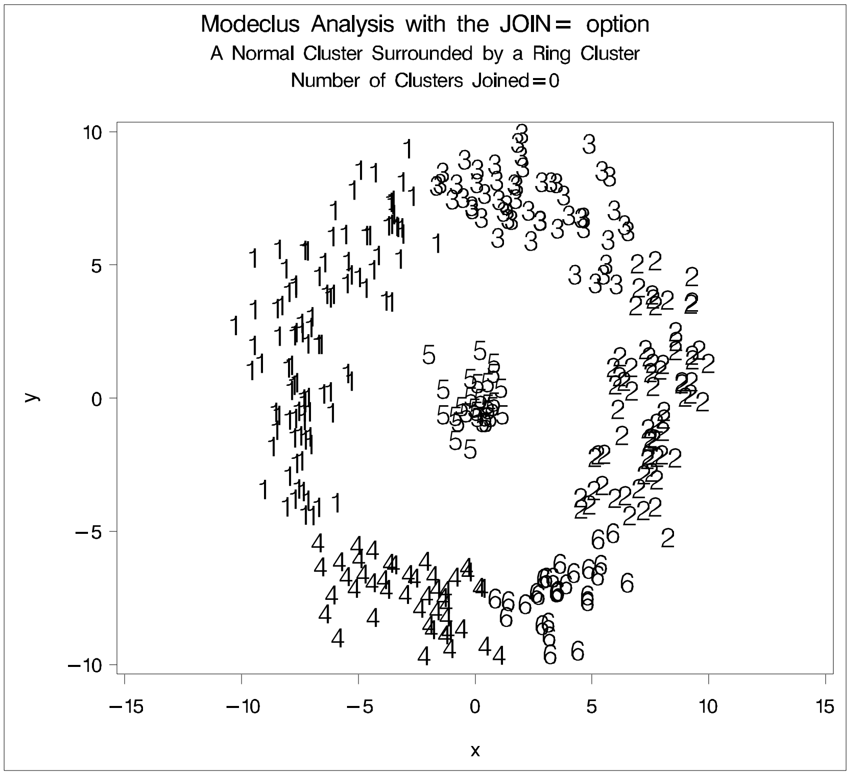
```
                        Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                    R=2.5  METHOD=1

                        Cluster Statistics
                              Maximum                  Estimated    -------------Saddle Test: Version 92.7------------
                            Estimated      Boundary       Saddle      Mode    Saddle    Overlap                 Approx
Cluster     Frequency         Density     Frequency      Density     Count     Count      Count            Z   P-value
----------------------------------------------------------------------------------------------------------------------
1                 156      0.00617328            17   0.00246931        39        15          0        3.130    0.1318
2                 144      0.00571029            14   0.00293231        36        18          0        2.313    0.6447
3                  30      0.00462996             0            .        29         0          .        3.611    0.0301
```

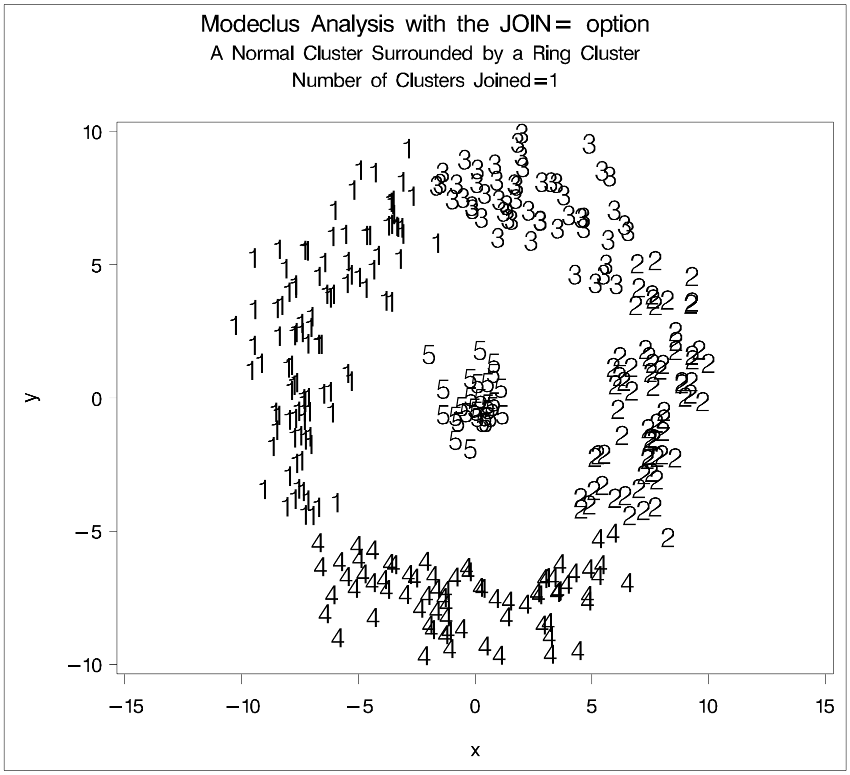
```
                        Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                    R=2.5  METHOD=1

                        Cluster Statistics
                              Maximum                  Estimated    -------------Saddle Test: Version 92.7------------
                            Estimated      Boundary       Saddle      Mode    Saddle    Overlap                 Approx
Cluster     Frequency         Density     Frequency      Density     Count     Count      Count            Z   P-value
----------------------------------------------------------------------------------------------------------------------
1                 300      0.00617328             0            .        39         0          .        4.246    0.0026
2                  30      0.00462996             0            .        29         0          .        3.611    0.0301
```

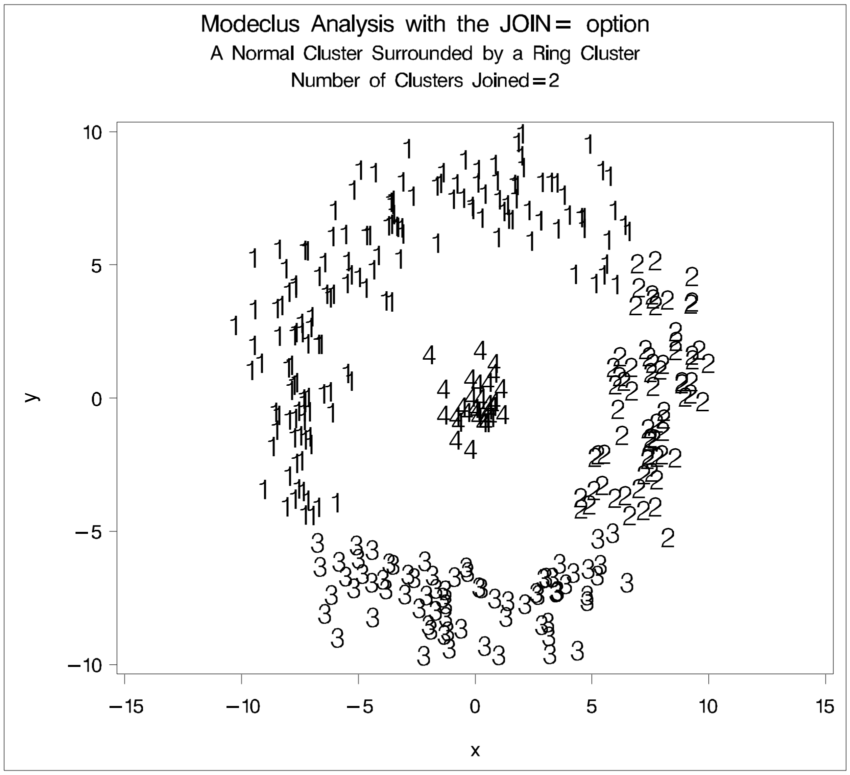
```
                        Modeclus Analysis with the JOIN= option
                      A Normal Cluster Surrounded by a Ring Cluster

                                 The MODECLUS Procedure
                                    R=2.5  METHOD=1

                        Cluster Statistics
                              Maximum                  Estimated    -------------Saddle Test: Version 92.7------------
                            Estimated      Boundary       Saddle      Mode    Saddle    Overlap                 Approx
Cluster     Frequency         Density     Frequency      Density     Count     Count      Count            Z   P-value
----------------------------------------------------------------------------------------------------------------------
1                 300      0.00617328             0            .        39         0          .        4.246    0.0026
```

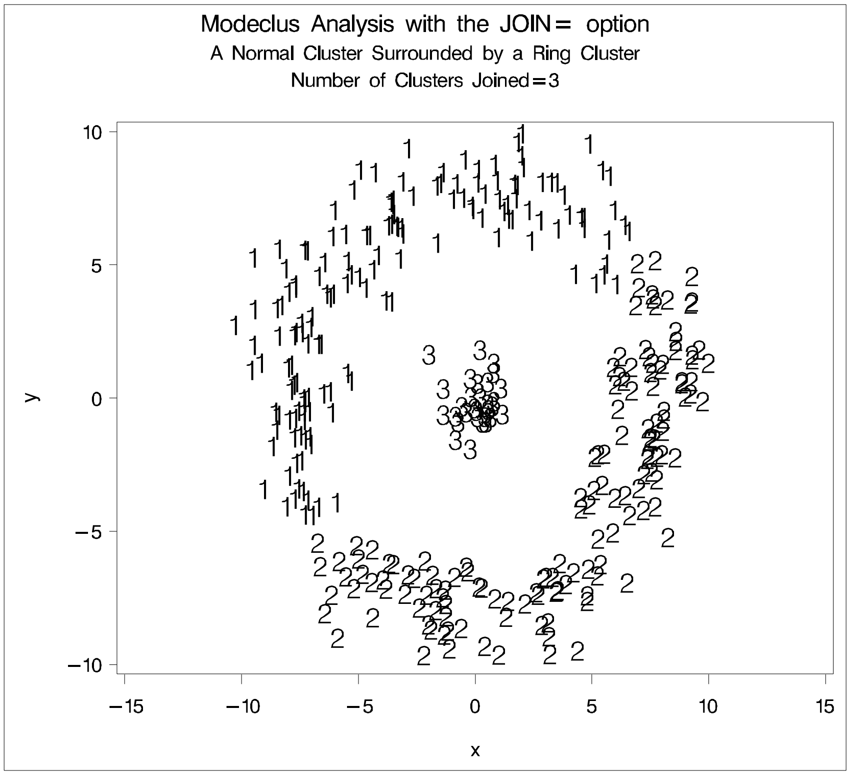
```
                       Modeclus Analysis with the JOIN= option
                     A Normal Cluster Surrounded by a Ring Cluster

                               The MODECLUS Procedure

                                   Cluster Summary
                      Number of                                Frequency of
                       Clusters     Maximum     Number of     Unclassified
                R        Joined     P-value      Clusters          Objects
         -------------------------------------------------------------
              2.5             0      1.0000             6                0
              2.5             1      0.9999             5                0
              2.5             2      0.9990             4                0
              2.5             3      0.6447             3                0
              2.5             4      0.0301             2                0
              2.5             5      0.0026             1               30
```

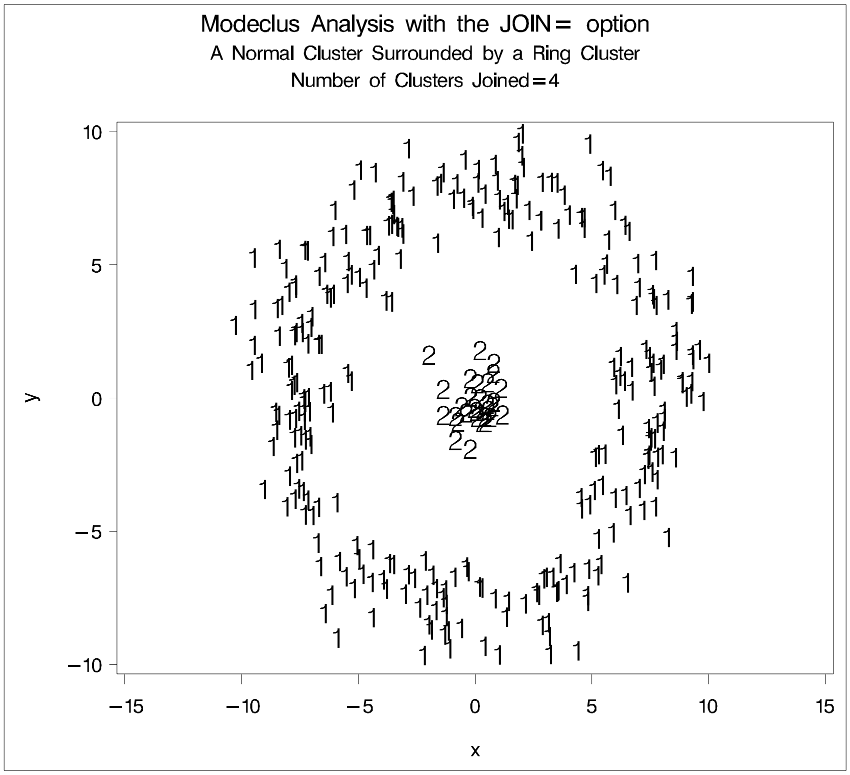
Output 47.3.3. Scatter Plots of Cluster Memberships by _NJOIN_

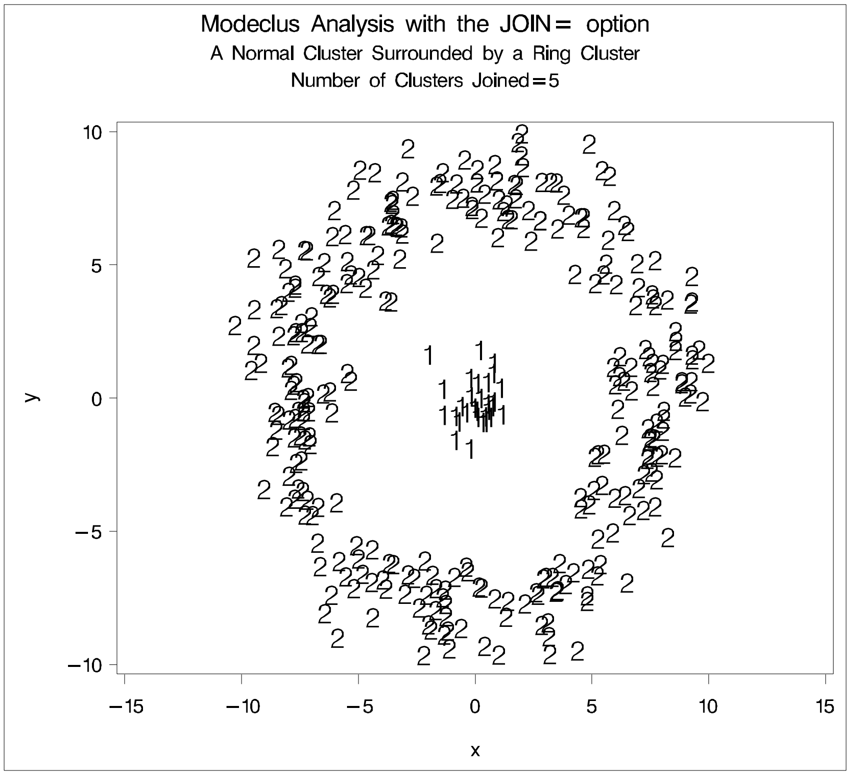



Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster
Number of Clusters Joined=1
y
x
10
5
0
−5
−10
−15
−10
−5
0
5
10
15


Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster
Number of Clusters Joined=2
y
x
10
5
0
−5
−10
−15
−10
−5
0
5
10
15


Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster
Number of Clusters Joined=3
y
x
10
5
0
−5
−10
−15
−10
−5
0
5
10
15
1
2
3


Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster
Number of Clusters Joined=4
y
x
10
5
0
−5
−10
−15
−10
−5
0
5
10
15
1
2



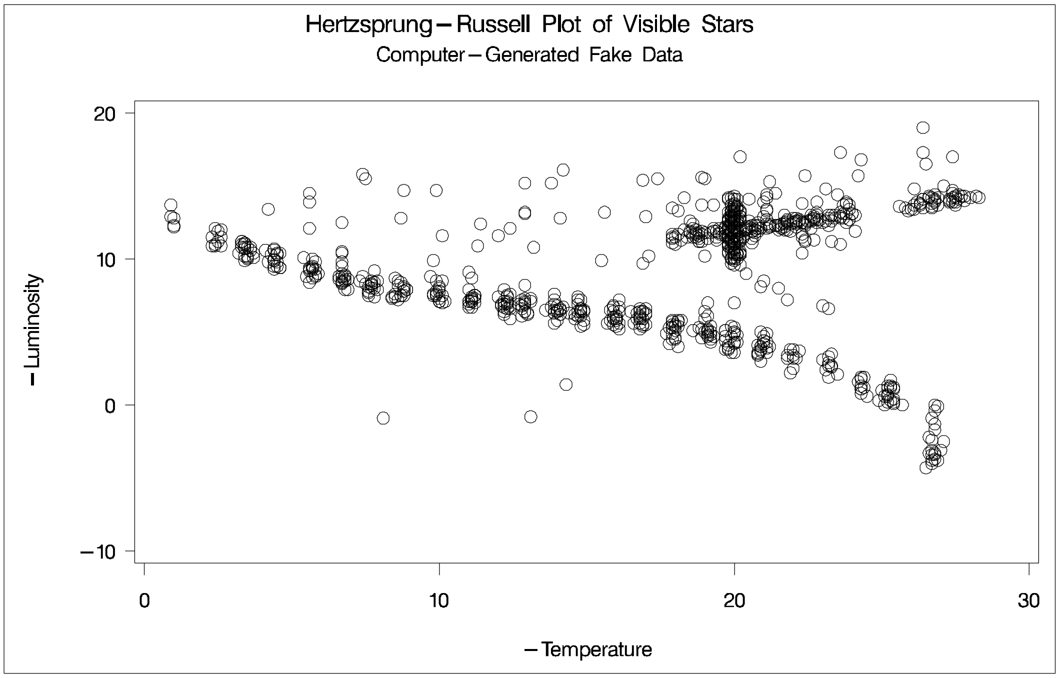
Example 47.4. Cluster Analysis: Hertzsprung-Russell Plot

This example uses computer-generated data to mimic a Hertzsprung-Russell plot (Struve and Zebergs 1962, p. 259) of the temperature and luminosity of stars. The data are plotted and displayed in Output 47.4.1; see “Example 4 from Proc Modeclus” in the SAS/STAT Sample Program Library for the complete data set. It appears that there are two main groups of stars and a collection of isolated stars. The long straggling group of points appearing diagonally across the figure represents the main group of stars; the more compact group in the top right-hand corner contains giant stars. The JOIN= option is specified at a 0.05 significance level with various smoothing parameters. The CK=5 option is specified in order to prevent the numerous outliers from forming separate clusters. The results from PROC MODECLUS is displayed in Output 47.4.2. The cluster memberships are then plotted by PROC GPLOT, as displayed in Output 47.4.3.

Notice in Output 47.4.3 that the graphic output from PROC GPLOT when _R_ = 2.5 is not available because only one cluster remains after joining at a 5% significance level, and the results are not written to the OUT= data set. See the description of the JOIN= option on page 2866 for more information.

```
title 'Hertzsprung-Russell Plot of Visible Stars';
title2 'Computer-Generated Fake Data';
data hr;
   input x y @@;
   label x='-Temperature'
         y='-Luminosity';
   datalines;
 1.0  12.8   0.9  13.7   0.9  12.9   1.0  12.3   1.0  12.2
 2.6  10.9   2.4  10.9   2.5  11.2   2.3  11.5   2.6  12.0
 2.4  12.1   2.3  10.9   2.6  11.5   2.5  11.9   2.4  11.0
 3.4  11.1   3.3  11.2   3.4  11.1   3.4   9.9   3.2  10.4

                   ... 150 lines omitted ...

18.5  12.6  14.2  16.1  23.2   6.6  11.4  12.4  20.4  11.7
20.9   8.1  18.9  13.7  16.9   9.7  15.5   9.9  18.3  14.2
19.3  13.7  17.0  12.9  10.1  11.6  17.9  13.5  14.3   1.4
13.1  -0.8   8.1  -0.9  20.0   7.0  21.0   8.5  15.6  13.2
;

symbol1 value=circle c=white;
symbol2 value=plus c=yellow;
symbol3 value=triangle c=cyan;
legend1 frame cframe=ligr cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

proc gplot;
  plot y*x/legend=legend1 frame cframe=ligr vzero
           vaxis=axis1 haxis=axis2 ;

proc modeclus data=hr m=1 r=1 1.5 2 2.5 ck=5
              join=.05 short out=out;
run;

title2 'MODECLUS Analysis';
proc gplot;
   plot y*x=cluster/frame cframe=ligr
                    vzero legend=legend1
                    vaxis=axis1 haxis=axis2;
   by _R_;
run;
```

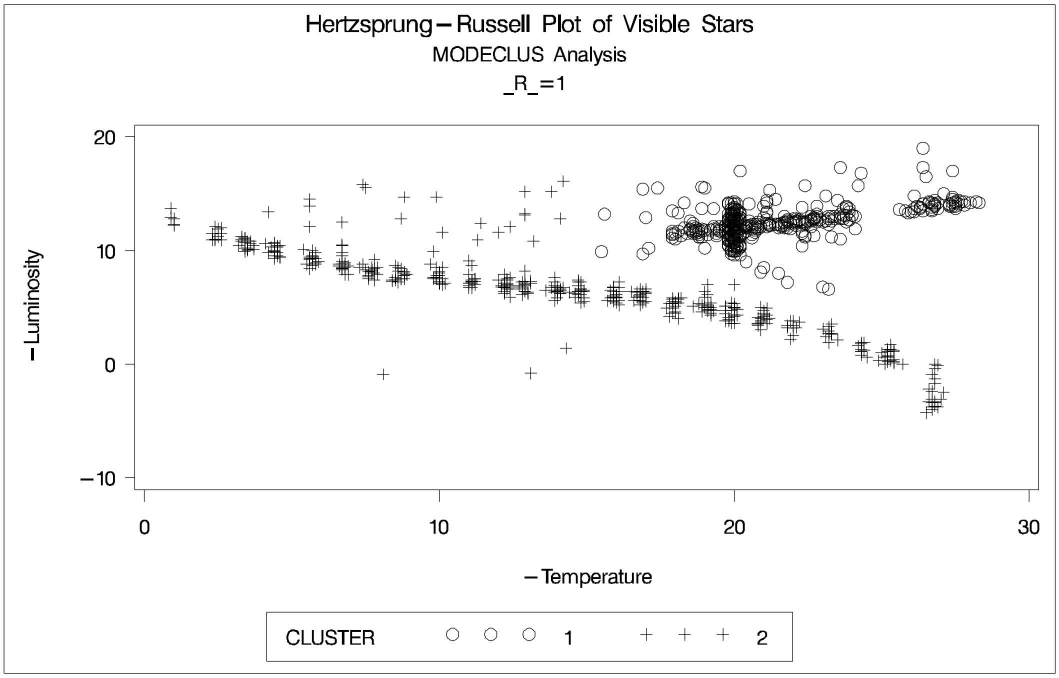
Output 47.4.1. Scatter Plot of Data

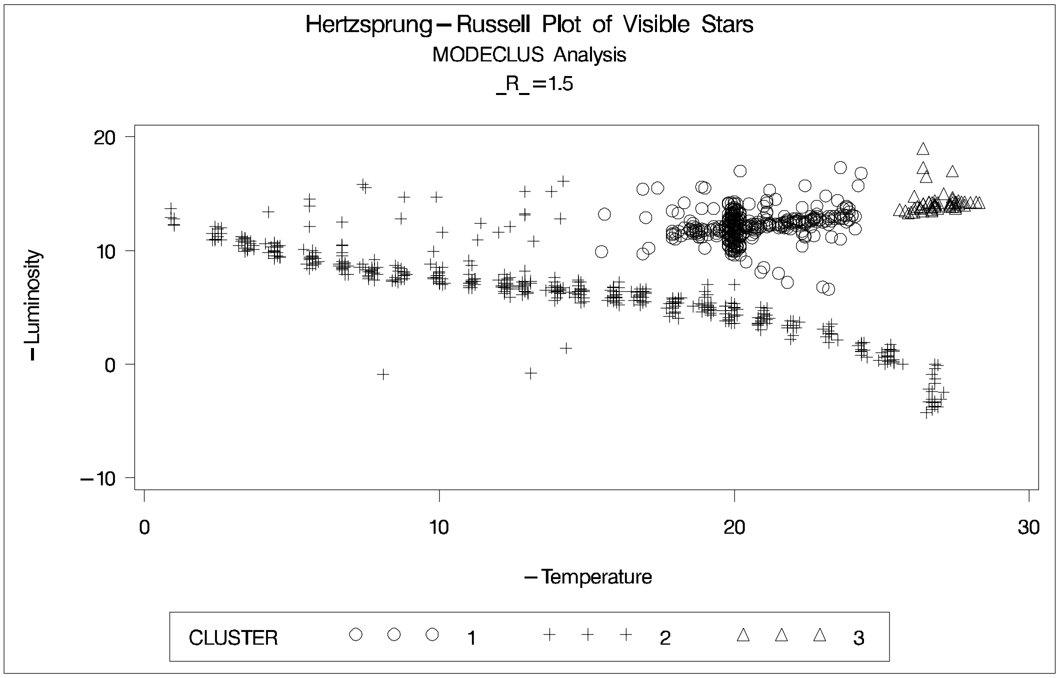


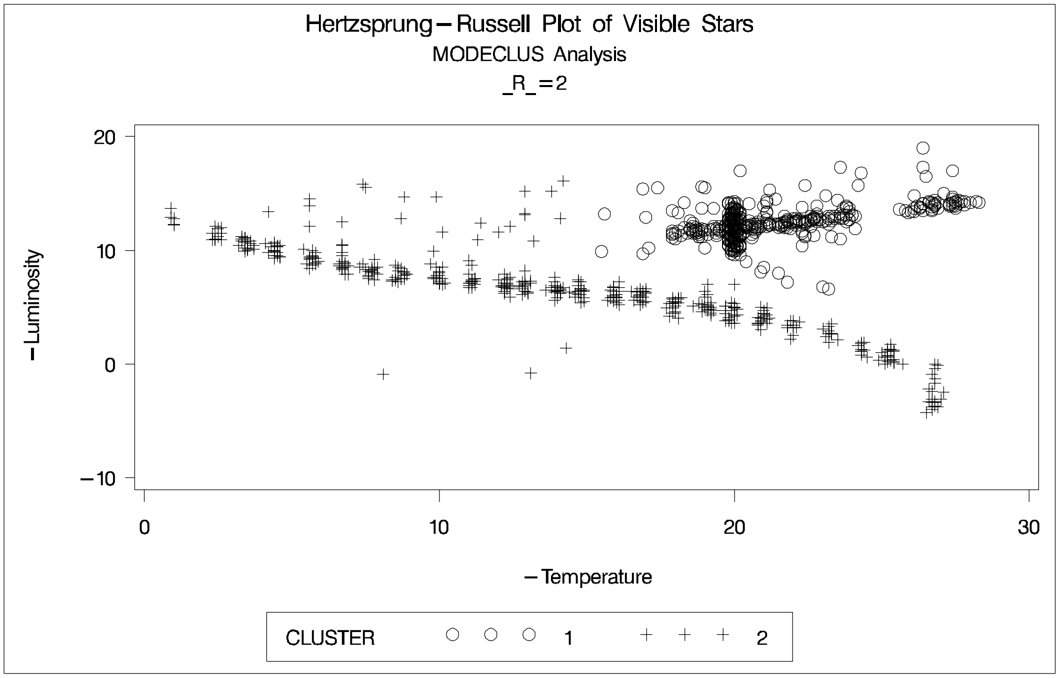
Output 47.4.2. Results from PROC MODECLUS

```
              Hertzsprung-Russell Plot of Visible Stars
                     Computer-Generated Fake Data

                        The MODECLUS Procedure

                           Cluster Summary
                      Number of                            Frequency of
                       Clusters    Maximum    Number of    Unclassified
      R         CK       Joined    P-value     Clusters         Objects
-----------------------------------------------------------------------
      1          5           14     0.0001            2               0
    1.5          5            6     0.0000            3               0
      2          5            4     0.0000            2               0
    2.5          5            2     0.0000            1               0
```

Output 47.4.3. Scatter Plots of Cluster Memberships by _R_







Example 47.5. Using the TRACE Option when METHOD=6

To illustrate how the TRACE option can help you to understand the clustering process when METHOD=6 is specified, the following data set is created with 12 observations.

```
data test;
   input x@@;
   datalines;
1 2 3 4 5 7.5 9 11.5 13 14.5 15 16
;
```

The first five observations seem to be close to each other, and the last five observations seem to be close to each other. Observation 6 is separated from the first five observations with a (Euclidean) distance of 2.5, and the same distance separates observation 7 from the last five observations. Observations 6 and 7 differ by 1.5.

Suppose METHOD=6 with a radius=2.5 is chosen for the cluster analysis. You can specify the TRACE option to understand how each observation is assigned.

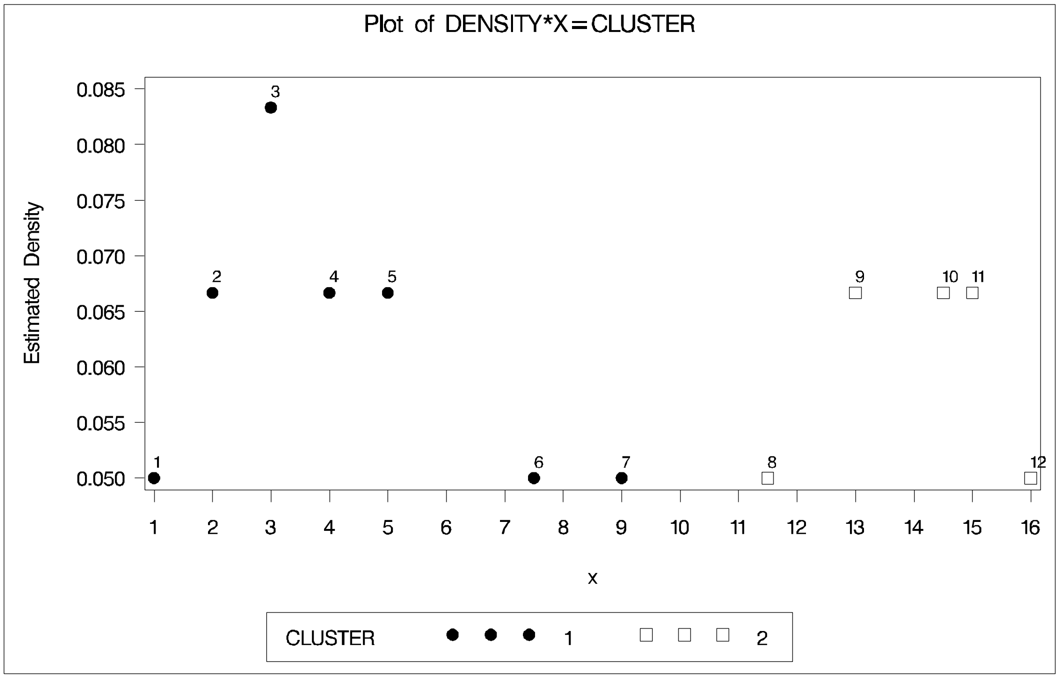
The following statements produce Output 47.5.1 and Output 47.5.2:

```
/*-- METHOD=6 with TRACE and THRESHOLD=0.5 (default) --*/
proc modeclus method=6 r=2.5 trace short out=out;
   var x;
run;

data markobs;
   drop _r_ _method_ _obs_ density cluster;
   length function style $8 text $ 2;
   retain xsys ’2’ ysys ’2’ hsys ’1’ when ’a’;
   set out;
   /* create the text for obs */
   function=’label’; size=4;
   style=’swiss’;
   text=left(put(_obs_,2.));
   position=’3’;
   x=x; y=density;
   output;
run;
legend1 frame cframe=ligr cborder=black
       position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
title ’Plot of DENSITY*X=CLUSTER’;
proc gplot data=out;
   plot density*x=cluster/ annotate=markobs
                           frame cframe=ligr
                           legend=legend1
                           vaxis=axis1 haxis=axis2;
run;
```

Output 47.5.1. Partial Output of METHOD=6 with TRACE and Default THRESHOLD=

```
                  The MODECLUS Procedure
                    R=2.5  METHOD=6

              Trace of Clustering Algorithm
                                 Cluster
Obs           Density      Old      New      Ratio
---------------------------------------------------
  3         0.0833333      -1        1       M
  2         0.0666667       0        1       N
  4         0.0666667       0        1       N
  5         0.0666667       0        1       N
  1         0.0500000       0        1       N
  6         0.0500000       0        1       0.571
  7         0.0500000      -1        1       0.500
  9         0.0666667      -1        2       M
  8         0.0500000       0        2       N
 10         0.0666667      -1        2       S
 12         0.0500000       0        2       N
 11         0.0666667      -1        2       S
```

Output 47.5.2. Density Plot



Notice that in Output 47.5.1, observation 7 is originally a seed (indicated by a value of -1 in the “Old” column) and then assigned to cluster 1. This is because the ratio of observation 7 to cluster 1 is 0.5 and is not less than the default value of THRESHOLD= (0.5).

If the value of the THRESHOLD= option is increased to 0.55, observation 7 should be excluded from cluster 1 and the cluster membership of observation 7 is changed.

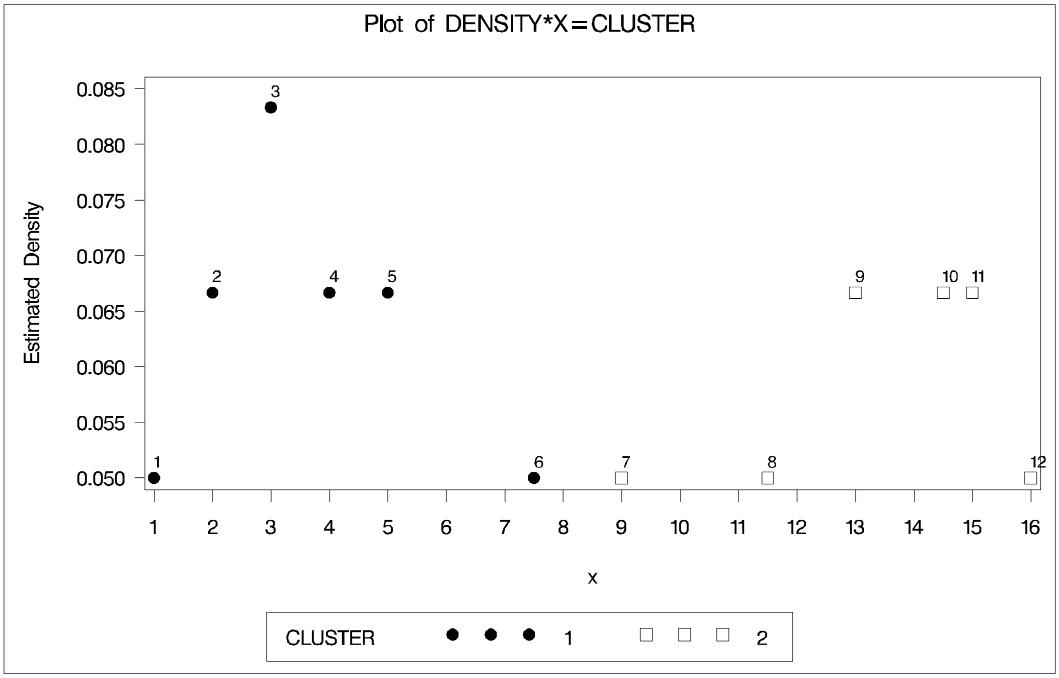
The following statements produce Output 47.5.3 and Output 47.5.4:

```
/*-- METHOD=6 with TRACE and THRESHOLD=0.55 --*/
proc modeclus method=6 r=2.5 trace threshold=0.55 short
              out=out;
   var x;
run;

     . . .   (the Data Step and the PROC GPLOT statement
              are omitted because they are the same as the
              previous job)
```

Output 47.5.3. Partial Output of METHOD=6 with TRACE and THRESHOLD=.55

```
                         The MODECLUS Procedure
                           R=2.5   METHOD=6

                     Trace of Clustering Algorithm
                                        Cluster
          Obs            Density      Old      New      Ratio
          -----------------------------------------------
            3          0.0833333       -1        1        M
            2          0.0666667        0        1        N
            4          0.0666667        0        1        N
            5          0.0666667        0        1        N
            1          0.0500000        0        1        N
            6          0.0500000        0        1      0.571
            9          0.0666667       -1        2        M
            8          0.0500000        0        2        N
           10          0.0666667       -1        2        S
           12          0.0500000        0        2        N
           11          0.0666667       -1        2        S
            7          0.0500000       -1        2        S
```

Output 47.5.4. Density Plot



In Output 47.5.3, observation 7 is a seed that is excluded by cluster 1 because its ratio to cluster 1 is less than 0.55. Being a neighbor of a member (observation 8) of cluster 2, observation 7 eventually joins cluster 2 even though it remains a “SEED.” (See **Step 2.2** in the section “METHOD=6” on page 2875.)

References

Barnett, V., ed. (1981), *Interpreting Multivariate Data*, New York: John Wiley & Sons, Inc.

Gitman, I. (1973), “An Algorithm for Nonsupervised Pattern Classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 66–74.

Hartigan, J.A. and Hartigan, P.M. (1985), “The Dip Test of Unimodality,” *Annals of Statistics*, 13, 70–84.

Huizinga, D.H. (1978), “A Natural or Mode Seeking Cluster Analysis Algorithm,” Technical Report 78-1, Behavioral Research Institute, 2305 Canyon Blvd., Boulder, Colorado 80302.

Koontz, W.L.G. and Fukunaga, K. (1972a), “A Nonparametric Valley-Seeking Technique for Cluster Analysis,” *IEEE Transactions on Computers*, C-21, 171–178.

Koontz, W.L.G. and Fukunaga, K. (1972b), “Asymptotic Analysis of a Nonparametric Clustering Technique,” *IEEE Transactions on Computers*, C-21, 967–974.

Koontz, W.L.G., Narendra, P.M., and Fukunaga, K. (1976), “A Graph-Theoretic Approach to Nonparametric Cluster Analysis,” *IEEE Transactions on Computers*, C-25, 936–944.

Minnotte, M.C. (1992), “A Test of Mode Existence with Applications to Multimodality,” Ph.D. thesis, Rice University, Department of Statistics.

Mizoguchi, R. and Shimura, M. (1980), “A Nonparametric Algorithm for Detecting Clusters Using Hierarchical Structure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, 292–300.

Müller, D.W. and Sawitzki, G. (1991), “Excess Mass Estimates and Tests for Multimodality,” JASA 86, 738–746.

Polonik, W. (1993), “Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach,” Technical Report, Beitraege zur Statistik Nr. 7, Universitaet Heidelberg.

Sarle, W.S. (1982), “Cluster Analysis by Least Squares,” *SAS Users Group International Conference Proceedings: SUGI 7*, 651–653.

Scott, D.W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons, Inc.

Silverman, B.W. (1986), *Density Estimation*, New York: Chapman and Hall.

Struve, O. and Zebergs, V. (1962), *Astronomy of the Twentieth Century*, New York: Macmillan.

Tukey, P.A. and Tukey, J.W. (1981), “Data-Driven View Selection; Agglomeration and Sharpening,” in Barnett (1981).

Wong, M.A. and Lane, T. (1983), “A kth Nearest Neighbor Clustering Procedure,” *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M.A. and Schaack, C. (1982), “Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations,” *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 48
The MULTTEST Procedure

Chapter Contents

Chapter 48
The MULTTEST Procedure

Overview

The MULTTEST procedure addresses the multiple testing problem. This problem arises when you perform many hypothesis tests on the same data set. Carrying out multiple tests is often reasonable because of the cost of obtaining data, the discovery of new aspects of the data, and the many alternative statistical methods. However, a negative feature of multiple testing is the greatly increased probability of declaring false significances.

For example, suppose you carry out 10 hypothesis tests at the 5% level, and you assume that the distributions of the p-values from these tests are uniform and independent. Then, the probability of declaring a particular test significant under its null hypothesis is 0.05, but the probability of declaring at least 1 of the 10 tests significant is 0.401. If you perform 20 hypothesis tests, the latter probability increases to 0.642. These high chances illustrate the danger in multiple testing.

PROC MULTTEST approaches the multiple testing problem by adjusting the p-values from a family of hypothesis tests. An adjusted p-value is defined as the smallest significance level for which the given hypothesis would be rejected, when the entire family of tests is considered. The decision rule is to reject the null hypothesis when the adjusted p-value is less then α; in most cases, this procedure controls the *familywise error rate* at or below the α level. PROC MULTTEST offers the following p-value adjustments:

- Bonferroni
- Sidak
- Stepdown methods
- Hochberg
- Hommel
- Fisher Combination
- False Discovery Rate
- Bootstrap
- Permutation

The Bonferroni and Sidak adjustments are simple functions of the raw p-values. They are computationally quick, but they can be too conservative. Stepdown methods remove some conservativeness, as do the step-up methods of Hochberg (1988). The bootstrap and permutation adjustments resample the data with and without replacement, respectively, to approximate the distribution of the minimum p-value of all tests. This distribution is then used to adjust the individual raw p-values. The bootstrap and permutation methods are computationally intensive but appealing in that, unlike the other methods, correlations and distributional characteristics are incorporated into the adjustments (Westfall and Young 1989, 1993; Westfall *et al.* 1999).

PROC MULTTEST handles data arising from a multivariate one-way ANOVA model, possibly stratified, with continuous and discrete response variables; it can also accept raw p-values as input data. You can perform a *t*-test for the mean for continuous data and the following statistical tests for discrete data:

- Cochran-Armitage (CA) linear trend test
- Freeman-Tukey (FT) double arcsine test
- Peto (PETO) mortality-prevalence (log-rank) test
- Fisher (FISHER) exact test

The CA and PETO tests have exact versions that use permutation distributions and asymptotic versions that use an optional continuity correction. Also, with the exception of the FISHER test, you can use a stratification variable to construct Mantel-Haenszel type tests. All of the previously mentioned tests can be one- or two-sided.

As in the GLM procedure, you can specify linear contrasts that compare means or proportions of the treated groups. The output contains summary statistics and regular and multiplicity-adjusted p-values. You can create output data sets containing raw and adjusted p-values, test statistics and other intermediate calculations, permutation distributions, and resampling information.

Getting Started

Drug Example

Suppose you conduct a small study to test the effect of a drug on 15 subjects. You randomly divide the subjects into three balanced groups receiving 0 mg, 1 mg, and 2 mg of the drug, respectively. You carry out the experiment and record the presence or absence of 10 side effects for each subject. Your data set is as follows:

```
data Drug;
   input Dose$ SideEff1-SideEff10;
   datalines;
0MG  0   0   1   0   0   1   0   0   0   0
0MG  0   0   0   0   0   0   0   0   0   1
0MG  0   0   0   0   0   0   0   0   1   0
0MG  0   0   0   0   0   0   0   0   0   0
0MG  0   1   0   0   0   0   0   0   0   0
1MG  1   0   0   1   0   1   0   0   1   0
1MG  0   0   0   1   1   0   0   1   0   1
1MG  0   1   0   0   0   0   1   0   0   0
1MG  0   0   1   0   0   0   0   0   0   1
1MG  1   0   1   0   0   0   0   1   0   0
```

```
2MG  0   1   1   1   0   1   1   1   0   1
2MG  1   1   1   1   1   1   0   1   1   0
2MG  1   0   0   1   0   1   1   0   1   0
2MG  0   1   1   1   1   0   1   1   1   1
2MG  1   0   1   0   1   1   1   0   0   1
;
```

The increasing incidence of 1s for higher dosages in the preceding data set provides an initial visual indication that the drug has an effect. To explore this statistically, you decide to perform an analysis in which the possibility of side effects increases linearly with drug level. You can analyze the data for each side effect separately, but you are concerned that, with so many tests, there may be a high probability of incorrectly declaring some drug effects significant. You want to correct for this multiplicity problem in a way that accounts for the discreteness of the data and for the correlations between observations on the same unit.

PROC MULTTEST addresses these concerns by processing all of the data simultaneously and adjusting the p-values. The following statements perform a typical analysis:

```
proc multtest bootstrap nsample=20000 seed=41287 notables pvals;
   class Dose;
   test ca(SideEff1-SideEff10);
   contrast ’Trend’ 0 1 2;
run;
```

This analysis uses the BOOTSTRAP option to adjust the p-values. The NSAMPLE= option requests 20,000 samples for the bootstrap analysis, and the starting seed for the random number generator is 41287. The NOTABLES option suppresses the display of summary statistics for each side effect and drug level combination.

The CLASS statement is used to specify the grouping variable, Dose. The CA(SIDEEFF1-SIDEEFF10) specification in the TEST statement requests a Cochran-Armitage linear trend test for all 10 characteristics. The CONTRAST statement gives the coefficients for the linear trend test.

The results from this analysis are as follows.

```
                            The Multtest Procedure

                              Model Information

         Test for discrete variables                 Cochran-Armitage
         Z-score approximation used                  Everywhere
         Continuity correction                       0
         Tails for discrete tests                    Two-tailed
         Strata weights                              None
         P-value adjustment                          Bootstrap
         Number of resamples                         20000
         Seed                                        41287
```

Figure 48.1. Output Summary for the MULTTEST Procedure

Figure 48.1 describes the statistical tests performed by PROC MULTTEST. For this example, PROC MULTTEST carries out a two-tailed Cochran-Armitage linear trend test with no continuity correction or strata adjustment. This test is performed on the raw data and on 20,000 bootstrap samples.

```
                          The Multtest Procedure

                          Contrast Coefficients

                                         Dose

             Contrast             0MG              1MG              2MG

             Trend                  0                1                2
```

Figure 48.2. Coefficients Used in the MULTTEST Procedure

Figure 48.2 displays the coefficients for the Cochran-Armitage test. They are 0, 1, and 2, as specified in the CONTRAST statement.

```
                          The Multtest Procedure

                                p-Values

                Variable     Contrast           Raw     Bootstrap

                SideEff1     Trend           0.0519        0.3471
                SideEff2     Trend           0.1949        0.8388
                SideEff3     Trend           0.0662        0.5232
                SideEff4     Trend           0.0126        0.0937
                SideEff5     Trend           0.0382        0.2438
                SideEff6     Trend           0.0614        0.4455
                SideEff7     Trend           0.0095        0.0540
                SideEff8     Trend           0.0519        0.3471
                SideEff9     Trend           0.1949        0.8388
                SideEff10    Trend           0.2123        0.9002
```

Figure 48.3. Summary of *p*-values for the MULTTEST Procedure

Figure 48.3 lists the p-values for the drug example. The Raw column lists the p-values for the Cochran-Armitage test on the original data, and the Bootstrap column provides the bootstrap adjustment of the raw p-values.

Note that the raw p-values lead you to reject the null hypothesis of no linear trend for 3 of the 10 characteristics at the 5% level and for 7 of the 10 characteristics at the 10% level. The bootstrap p-values, however, lead to this conclusion for 0 of the 10 characteristics at the 5% level and only 2 of the 10 characteristics at the 10% level. The bootstrap adjustment gives the probability of observing a p-value as extreme as each given p-value, considering all ten tests simultaneously. This adjustment incorporates the correlation of the raw p-values, the discreteness of the data, and the multiple testing problem. Failure to account for these issues can certainly lead to misleading inferences for these data.

Syntax

The following statements are available in PROC MULTTEST.

PROC MULTTEST < *options* > **;**
BY *variables* **;**
CLASS *variable* **;**
CONTRAST *'label' values* **;**
FREQ *variable* **;**
STRATA *variable* **;**
TEST *name (variables* < */ options* >*)* **;**

Items within angle brackets (< >) are optional, and statements following the PROC MULTTEST statement can appear in any order. The CLASS and TEST statements are required. The syntax of each statement is described in the following section in alphabetical order after the description of the PROC MULTTEST statement.

PROC MULTTEST Statement

PROC MULTTEST < *options* > **;**

You can specify the following options in the PROC MULTTEST statement.

BONFERRONI
BON

specifies that the Bonferroni adjustments (number of tests × p-value) be computed for each test. These adjustments can be extremely conservative and should be viewed with caution. When exact tests are specified via the PERMUTATION= option in the TEST statement, the actual permutation distributions are used, resulting in a much less conservative version of this procedure (Westfall and Wolfinger 1997).

BOOTSTRAP
BOOT

specifies that the p-values be adjusted using the bootstrap method to resample vectors (Westfall and Young 1993). Resampling is performed with replacement and independently within levels of the STRATA variable. Continuous variables are mean-centered by default prior to resampling. The BOOTSTRAP option is not allowed with the PETO test for theoretical reasons.

If the PERMUTATION= suboption is used with the CA test on the TEST statement, the exact permutation distribution is recomputed for each bootstrap sample. **Caution:** This can be very time-consuming. It is preferable to use permutation resampling when permutation base tests are used.

CENTER

requests that continuous variables be mean-centered prior to resampling. The default action is to mean-center for bootstrap resampling and not to mean-center for permutation resampling.

DATA=*SAS-data-set*

names the input SAS data set to be used by PROC MULTTEST. The default is to use the most recently created data set. The DATA= and PDATA= options cannot both be specified.

FDR

requests adjusted p-values using the method of Benjamini and Hochberg (1995). These p-values do not control the familywise error rate, but they do control the false discovery rate in some cases.

FISHER_C

requests adjusted p-values using Fisher's combination method.

HOC

requests adjusted p-values using Hochberg's (1988) step-up Bonferroni method.

HOMMEL
HOM

requests adjusted p-values using Hommel's (1988) method.

HOLM

is an alias for the STEPBON adjustment.

NOCENTER

requests that continuous variables not be mean-centered prior to resampling. The default action is to mean-center for bootstrap resampling and not to mean-center for permutation resampling.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "The Output Delivery System," for more information.

NOTABLES

suppresses display of the "Discrete Variable Tabulations" and "Continuous Variable Tabulations" tables.

NOZEROS

suppresses display of tables having zero occurrences for all CLASS levels.

NSAMPLE= *number*
N= *number*

specifies the number of resamples for use with the BOOTSTRAP and PERMUTATION options; it is assumed to be 20,000 by default. Large values of *number* (20,000 or more) are usually recommended for accuracy, but long execution times may result, particularly with large data sets.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the CLASS variable. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use CONTRAST statements.

When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values. In order to revert to the previous method, you can specify this format explicitly for the CLASS variable. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and required you to use an explicit format or to specify ORDER=INTERNAL to get the more natural ordering.

The following table shows how PROC MULTTEST interprets values of the ORDER= option.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUT=*SAS-data-set*

names the output SAS data set containing variable names, contrast names, intermediate calculations, and all associated p-values.

OUTPERM=*SAS-data-set*

names the output SAS data set containing entire permutation distributions (upper-tail probabilities) for all tests when the PERMUTATION= option is used. **Caution:** This data set can be very large.

OUTSAMP=*SAS-data-set*

names the output SAS data set containing information from the resampled data sets when resampling is performed. **Caution:** This data set can be very large.

PDATA=*SAS-data-set*

names an input SAS data set containing the variable raw_p with observations that consist of raw p-values. The MULTTEST procedure adjusts the collection of raw p-values for multiplicity. Resampling-based adjustments are not permitted with this type of data input. The PDATA= and DATA= options cannot both be specified.

PERMUTATION
PERM

specifies adjusted p-values in identical fashion as the BOOTSTRAP option, with the exception that PROC MULTTEST resamples without replacement rather than with replacement. Resampling is performed independently within levels of the STRATA variable. Continuous variables are not mean-centered prior to resampling. The PERMUTATION option is not allowed with the PETO test for theoretical reasons.

PVALS

requests that a summary table of raw and adjusted p-values be included.

SEED= *number*
S= *number*

specifies the initial seed for the random number generator used for resampling. The value for *number* must be an integer. If you do not specify a seed, or if you specify a value less than or equal to zero, then PROC MULTTEST uses the time of day from the computer's clock to generate an initial seed. For more details about seed values, refer to *SAS Language Reference: Concepts*.

SIDAK
SID

specifies that the Sidak adjustments be computed for each test. These adjustments take the form

$$1 - (1 - p)^n$$

where p is the raw p-value and n is the number of tests. These are slightly less conservative than the Bonferroni adjustments, but they still should be viewed with caution. When exact tests are specified via the PERMUTATION= option in the TEST statement, the actual permutation distributions are used, resulting in a much less conservative version of this procedure (Westfall and Wolfinger 1997).

STEPBON

requests adjusted p-values using the stepdown Bonferroni method of Holm (1979).

STEPBOOT

requests that adjusted p-values be computed using bootstrap resampling as described under the BOOTSTRAP option, but in stepdown fashion.

STEPPERM

requests that adjusted p-values be computed using permutation resampling as described under the PERMUTATION option, but in stepdown fashion.

STEPSID

requests adjusted p-values using the Sidak method as described in the SIDAK option, but in stepdown fashion.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MULTTEST to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MIXED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

Since sorting the data changes the order in which PROC MULTTEST reads observations, this can affect the sorting order for the levels of the CLASS variable if you have specified ORDER=DATA in the PROC MULTTEST statement. This, in turn, affects specifications in the CONTRAST statements.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable***;**

The CLASS statement is required. It declares a single variable (character or numeric) used to identify the groups for the analysis. For example, if the variable Treatment defines different levels of a treatment, then the statement is

```
class Treatment;
```

The CLASS variable can be either character or numeric. By default, its levels are determined from entire formatted values. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC MULTTEST statement. You need to be aware of the order when using the CONTRAST statement, and you should check the “Contrast Coefficients” table to verify that it is suitable.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE specifies that class levels should be determined using only no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

The order of the CLASS levels used by PROC MULTTEST correspond to their formatted values; this order can be changed with the ORDER= option in the PROC MULTTEST statement.

CONTRAST Statement

CONTRAST *’label’ values* **;**

This statement is used to identify tests between the levels of the CLASS variable; in particular, it is used to specify the coefficients for the trend tests. The *label* is a string naming the contrast; it contains a maximum of 21 characters. The *values* are scoring coefficients across the CLASS variable levels.

You can specify multiple CONTRAST statements, thereby specifying multiple contrasts for each variable. Multiplicity adjustments are computed for all contrasts and all variables simultaneously. The coefficients are applied in the order of the CLASS variables; this order can be changed with the ORDER= option in the PROC MULTTEST statement. For example, consider a four-group experiment with CLASS variable levels A1, A2, B1, and B2 denoting two levels of two treatments. The following statements produce three linear trend tests for each variable identified in the TEST statement. PROC MULTTEST computes the multiplicity adjustments over the entire collection of tests, which is three times the number of variables.

```
contrast ’a vs b’    -1 -1  1  1;
contrast ’a linear’  -1  1  0  0;
contrast ’b linear’   0  0 -1  1;
```

As another example, consider an animal carcinogenicity experiment with dose levels 0, 4, 8, 16, and 50. You might consider trend tests defined using the following statement:

```
contrast ’arithmetic trend’ 0 4 8 16 50;
```

This statement produces a trend test using the indicated scoring coefficients. Multiplicity-adjusted p-values are then computed over the collection of variables identified in the TEST statement. Refer to Lagakos and Louis (1985) for guidelines on the selection of contrast-scoring values.

When a Fisher test is specified in the TEST statement, the CONTRAST statement coefficients are used to group the CLASS variable's levels. Groups with a -1 contrast coefficient are combined and compared with groups with a 1 contrast coefficient for each test, and groups with a 0 coefficient are not included in the contrast. For example, the statements

```
contrast 'c vs all' 1 -1 -1 -1;
contrast 'c vs t1'  1 -1  0  0;
contrast 'c vs t3'  1  0  0 -1;
```

compute Fisher exact tests for (a) control versus the combined treatment groups, (b) control versus the first treatment group, and (c) control versus the third treatment group. Multiplicity adjustments are then computed over the entire collection of tests and variables. Only -1, 1, and 0 are acceptable CONTRAST coefficients when the Fisher test is specified; PROC MULTTEST ignores the CONTRAST statement if any other coefficients appear.

If you specify the FISHER test and no CONTRAST statements, then all contrasts of control versus treatment are automatically generated, with the first level of the CLASS variable deemed to be the control. In this case, the control level is assigned the value 1 in each contrast and the other treatment levels are assigned -1. You should therefore use the LOWERTAILED option to test for higher success rates in the treatment groups.

For tests other than FISHER, CONTRAST values are 0,1,2,... by default. For *t*-tests for the mean using continuous data (and for the FT tests), the contrast coefficients are centered to have mean 0. The resulting centered scoring coefficients are then applied to the sample means (or to the double-arcsine-transformed proportions in the case of the FT tests).

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

STRATA Statement

STRATA *variable* **;**

The STRATA statement identifies a single variable to use as a stratification variable in the analysis. This yields tests similar to those discussed in Mantel and Haenszel (1959) and Hoel and Walburg (1972) for binary data and pooled-means tests for continuous data. For example, when you test for prevalence in a carcinogenicity study, it is common to stratify on intervals of the time at death; the first level of the stratification variable may represent weeks $0-52$, the second weeks $53-80$, and so on. In multicenter clinical studies, each level of the stratification variable may represent a particular center.

The following option is available in the STRATA statement after a slash (/):

WEIGHT=

specifies the type of strata weighting to use when computing the Freeman-Tukey and *t*-tests for the mean. Valid values are SAMPLESIZE, HARMONIC, and EQUAL. SAMPLESIZE requests weights proportional to the within-stratum sample sizes, and is the default method. HARMONIC sets up weights equal to the harmonic mean of the non-missing within-stratum CLASS sizes, and is similar to a Type 2 analysis in PROC GLM. EQUAL specifies equal weights, and is similar to a Type 3 analysis in PROC GLM.

TEST Statement

TEST *name (variables < / options >)* **;**

The TEST statement is required. It identifies statistical tests to be performed and the discrete and continuous variables to be tested. The following tests are permitted as *name* in the TEST statement.

CA — requests the Cochran-Armitage linear trend tests for group comparisons. The test variables should take the value 0 for a failure and 1 for a success. The PERMUTATION= option can be used to request an exact permutation test; otherwise, a Z-score approximation is used. The CONTINUITY= option can be used to specify a continuity correction for the Z-score approximation.

FISHER — requests Fisher exact tests for comparing two treatment groups. The test variables should take the value 0 for a failure and 1 for a success.

FT — requests Z-score CA tests based upon the Freeman-Tukey double arcsine transformation of the frequencies. The test variables should take the value 0 for a failure and 1 for a success.

MEAN — requests the t-test for the mean. The test variables can take on any numeric values.

PETO — requests the Peto mortality-prevalence test. The test variables should take the value 0 for a nonoccurrence, 1 for an incidental occurrence, and 2 for a fatal occurrence. The TIME= option should be used with the PETO test to specify a variable giving the age at death. The CONTINUITY= option can be used to specify a continuity correction for the test.

If the value of a TEST variable is invalid, the observation is not used in the analysis. You can specify two tests only if one of them is MEAN. For example, the following statement is valid

```
test ca(d1-d2) mean(c1-c2);
```

but the statement

```
test ca(d1-d2) ft(d1-d2);
```

is invalid.

You can specify the following options in the TEST statement (some apply to only one test).

BINOMIAL

specifies that the binomial variance estimate be used for CA and PETO tests in their asymptotic normal approximations. The default is to use the hypergeometric variance.

CONTINUITY= *number*
C= *number*

specifies *number* as a particular continuity correction for the Z-score approximation in the CA and PETO tests. The default is 0.

LOWERTAILED
LOWER

is used to make all tests lower-tailed. All tests are two-tailed by default.

PERMUTATION= *number*
PERM= *number*

specifies that p-values for the CA and PETO tests be computed using exact permutation distributions when marginal success or failure totals within a stratum are *number* or less. For values greater than *number* (or when the PERMUTATION= option is omitted), PROC MULTTEST uses standard normal approximations with a continuity correction chosen to approximate the permutation distribution. PROC MULTTEST computes the appropriate convolution distributions when you use the STRATA statement along with the PERMUTATION= option.

TIME= *variable*

identifies the PETO test variable containing the age at death, which is assumed to be integer valued. If the TIME= option is omitted, all ages are assumed to equal 1.

UPPERTAILED
UPPER

is used to make all tests upper-tailed. All tests are two-tailed by default.

Details

Statistical Tests

The following section discusses the statistical tests performed in the MULTTEST procedure. For continuous data, a t-test for the mean is available. For discrete variables, available tests are the Cochran-Armitage (CA) linear trend test, the Freeman-Tukey (FT) double arcsine test, the Peto mortality-prevalence test, and the Fisher exact test.

Throughout this section, the discrete and continuous variables are denoted by S_{vgsr} and X_{vgsr}, respectively, where v is the variable, g is the treatment group, s is the stratum, and r is the replication. A plus sign $(+)$ subscript denotes summation over an index. Note that the tests are invariant to the location and scale of the contrast coefficients t_g.

Cochran-Armitage Linear Trend Test

The Cochran-Armitage linear trend test (Cochran 1954; Armitage 1955; Agresti 1990) is implemented using a Z-score approximation, an exact permutation distribution, or a combination of both.

Z-Score Approximation

Let m_{vgs} denote the sample size for a binary variable v within group g and stratum s. The pooled probability estimate for variable v and stratum s is

$$p_{vs} = \frac{S_{v+s+}}{m_{v+s}}$$

The expected value (under constant within-stratum treatment probabilities) for variable v, group g, and stratum s is

$$E_{vgs} = m_{vgs} p_{vs}$$

The test statistic for variable v has numerator

$$N_v = \sum_s \sum_g t_g (S_{vgs+} - E_{vgs})$$

where t_g denotes the contrast trend coefficients specified by the CONTRAST statement. The binomial variance estimate for this statistic is

$$V_v = \sum_s p_{vs}(1 - p_{vs}) \sum_g m_{vgs}(t_g - \bar{t}_{vs})^2$$

where

$$\bar{t}_{vs} = \sum_g \frac{m_{vgs} t_g}{m_{v+s}}$$

The hypergeometric variance estimate (the default) is

$$V_v = \sum_s \{m_{v+s}/(m_{v+s}-1)\} p_{vs}(1-p_{vs}) \sum_g m_{vgs}(t_g - \bar{t}_{vs})^2$$

For any strata s with $m_{v+s} \leq 1$, the contribution to the variance is taken to be zero.

PROC MULTTEST computes the Z-score statistic

$$Z_v = \frac{N_v}{\sqrt{V_v}}$$

The p-value for this statistic comes from the standard normal distribution. Whenever a 0 is computed for the denominator, the p-value is set to 1. This p-value approximates the probability obtained from the exact permutation distribution, discussed in the following text.

The Z-score statistic can be continuity-corrected to better approximate the permutation distribution. With continuity correction c, the upper-tailed p-value is computed from

$$Z_v = \frac{N_v - c}{\sqrt{V_v}}$$

For two-tailed, noncontinuity-corrected tests, PROC MULTTEST reports the p-value as $2\min(p, 1-p)$, where p is the upper-tailed p-value. The same formula holds for the continuity-corrected test, with the exception that when the noncontinuity-corrected Z and the continuity-corrected Z have opposite signs, the two-tailed p-value is 1.

When the PERMUTATION= option is specified and no STRATA variable is specified, PROC MULTTEST uses a continuity correction selected to optimally approximate the upper-tail probability of permutation distributions with smaller marginal totals (Westfall and Lin 1988). Otherwise, the continuity correction is specified using the CONTINUITY= option in the TEST statement.

The CA Z-score statistic is the Hoel-Walburg (Mantel-Haenszel) statistic reported by Dinse (1985).

Exact Permutation Test

When you use the PERMUTATION= option for CA in the TEST statement, PROC MULTTEST computes the exact permutation distribution of the trend score

$$T_v = \sum_s \sum_g t_g S_{vgs+}$$

and then compares the observed value of this trend with the permutation distribution to obtain the p-value

$$p_v = \Pr(X \geq \text{ observed } T_v)$$

where X is a random variable from the permutation distribution and where upper-tailed tests are requested. This probability can be viewed as a binomial probability, where the within-stratum probabilities are constant and where the probability is conditional with respect to the marginal totals S_{v+s+}. It also can be considered a rerandomization probability.

Because the computations can be quite time-consuming with large data sets, specifying the PERMUTATION=*number* option in the TEST statement limits the situations where PROC MULTTEST computes the exact permutation distribution. When marginal total success or total failure frequencies exceed *number* for a particular stratum, the permutation distribution is approximated using a continuity-corrected normal distribution. You should be cautious in using the PERMUTATION= option in conjunction with bootstrap resampling because the permutation distribution is recomputed for each bootstrap sample. This recomputation is not necessary with permutation resampling.

The permutation distribution is computed in two steps:

1. The permutation distributions of the trend scores are computed within each stratum.
2. The distributions are convolved to obtain the distribution of the total trend.

As long as the total success or failure frequency does not exceed *number* for any stratum, the computed distributions are exact. In other words, if $S_{v+s+} \leq$ *number* or $(m_{v+s} - S_{v+s+}) \leq$ *number* for all s, then the permutation trend distribution for variable v is computed exactly.

In step 1, the distribution of the within-stratum trend

$$\sum_{g} t_g S_{vgs+}$$

is computed using the multivariate hypergeometric distribution of the S_{vgs+}, provided *number* is not exceeded. This distribution can be written as

$$\Pr(S_{v1s+}, S_{v2s+}, \ldots, S_{vGs+}) = \prod_{g=1}^{G} \frac{\begin{pmatrix} m_{vgs} \\ S_{vgs+} \end{pmatrix}}{\begin{pmatrix} m_{v+s} \\ S_{v+s+} \end{pmatrix}}$$

The distribution of the within-stratum trend is then computed by summing these probabilities over appropriate configurations. For further information on this technique, refer to Bickis and Krewski (1986) and Westfall and Lin (1988). In step 2, the exact convolution distribution is obtained for the trend statistic summed over all strata having totals that meet the threshold criterion. This distribution is obtained by applying the fast Fourier transform to the exact within-stratum distributions. A description of this general method can be found in Pagano and Tritchler (1983) and Good (1987).

The convolution distribution of the overall trend is then computed by convolving the exact distribution with the distribution of the continuity-corrected standard normal approximation. To be more specific, let S_1 denote the subset of stratum indices that satisfy the threshold criterion, and let S_2 denote the subset of indices that do not satisfy the criterion. Let T_{v1} denote the combined trend statistic from the set S_1, which has an exact distribution obtained using Fourier analysis as previously outlined, and let T_{v1} denote the combined trend statistic from the set S_2. Then the distribution of the overall trend $T_v = T_{v1} + T_{v2}$ is obtained by convolving the analytic distribution of T_{v1} with the continuity-corrected normal approximation for T_{v2}. Using the notation from the "Z-Score Approximation" section on page 2948, this convolution can be written as

$$\begin{aligned}\Pr(T_{v1} + T_{v2} \geq u) &= \sum_{u1} \Pr(T_{v1} + T_{v2} \geq u \mid T_{v1} = u1) \Pr(T_{v1} = u1) \\ &\approx \sum_{u1} \Pr(Z \geq z) \Pr(T_{v1} = u1)\end{aligned}$$

where Z is a standard normal random variable, and

$$z = \frac{1}{\sqrt{V_v}} \left(u - u1 - \sum_{S_2} p_{vs} \sum_{g} t_g m_{vgs} - c \right)$$

In this expression, the summation of s in V_v is over S_2, and c is the continuity correction discussed under the Z-score approximation.

When a two-tailed test is requested, the expected trend

$$E_v = \sum_{s} \sum_{g} t_g E_{vgs}$$

is computed, and the two-tailed p-value is reported as the permutation tail probability for the observed trend T_v plus the permutation tail probability for $2E_v - T_v$, the reflected trend.

Freeman-Tukey Double Arcsine Test

For this test, the contrast trend coefficients $t_1, \ldots, t_G$ are centered to the values $c_1, \ldots, c_G$, where $c_g = t_g - \bar{t}$, $\bar{t} = \sum_g t_g / G$, and G is the number of groups. The numerator of this test statistic is

$$N_v = \sum_{s} w_{vs} \sum_{g} c_g f(S_{vgs+}, m_{vgs})$$

where the weights w_{vs} take on three different types of values depending upon your specification of the WEIGHT= option in the STRATA statement. The default value is the within-strata sample size m_{v+s}, ensuring comparability with the ordinary CA trend statistic. WEIGHT=HARMONIC sets w_{vs} equal to the harmonic mean

$$\left[\left(\sum_g \frac{1}{m_{vgs}}\right)/G^*\right]^{-1}$$

where G^* is the number of non-missing groups and the summation is over only the non-missing elements. The harmonic means analysis places more weight on the smaller sample sizes than does the default sample size method, and is similar to a Type 2 analysis in PROC GLM. WEIGHT=EQUAL sets $w_{vs} = 1$ for all v and s, and is similar to a Type 3 analysis in PROC GLM.

The function $f(r, n)$ is the double arcsine transformation:

$$f(r,n) = \arcsin\left(\sqrt{\frac{r}{n+1}}\right) + \arcsin\left(\sqrt{\frac{r+1}{n+1}}\right)$$

The variance estimate is

$$V_v = \sum_s w_{vs}^2 \sum_g \frac{c_g^2}{m_{vgs} + \frac{1}{2}}$$

and the test statistic is

$$Z_v = \frac{N_v}{\sqrt{V_v}}$$

The Freeman-Tukey transformation and its variance are described by Freeman and Tukey (1950) and Miller (1978). Since its variance is not weighted by the pooled probabilities, as is the CA test, the FT test can be more useful than the CA test for tests involving only a subset of the groups.

Peto Mortality-Prevalence Trend Test

The Peto test is a modified Cochran-Armitage procedure incorporating mortality and prevalence information. It represents a special case in PROC MULTTEST because the data structure requirements are different, and the resampling methods used for adjusting p-values are not valid. The TIME= option variable is required to specify “death” times or, more generally, time of occurrence. In addition, the test variables must assume one of the following three values.

- 0 = no occurrence
- 1 = incidental occurrence
- 2 = fatal occurrence

Use the TIME= option variable to define the mortality strata, and use the STRATA statement variable to define the prevalence strata.

The Peto test is computed like two Cochran-Armitage Z-score approximations, one for prevalence and one for mortality.

In the following notation, the subscript v represents the variable, g represents the treatment group, s represents the stratum, and t represents the time. Recall that a plus sign $(+)$ in a subscript location denotes summation over that subscript.

Let S^P_{vgs} be the number of incidental occurrences, and let m^P_{vgs} be the total sample size for variable v in group g, stratum s, excluding fatal tumors.

Let S^F_{vgt} be the number of fatal occurrences in time period t, and let m^F_{vgt} be the number alive at the end of time $t-1$.

The pooled probability estimates are

$$\begin{aligned} p^P_{vs} &= \frac{S^P_{v+s}}{m^P_{v+s}} \\ p^F_{vt} &= \frac{S^F_{v+t}}{m^F_{v+t}} \end{aligned}$$

The expected values are

$$\begin{aligned} E^P_{vgs} &= m^P_{vgs} p^P_{vs} \\ E^F_{vgt} &= m^F_{vgt} p^F_{vt} \end{aligned}$$

Define the numerator terms:

$$\begin{aligned} N^P_v &= \sum_s \sum_g t_g \left(S^P_{vgs} - E^P_{vgs}\right) \\ N^F_v &= \sum_t \sum_g t_g \left(S^F_{vgt} - E^F_{vgt}\right) \end{aligned}$$

where t_g denotes a contrast trend coefficient. Define the denominator variance terms (using the binomial variance):

$$\begin{aligned} V^P_v &= \sum_s p^P_{vs}\left(1 - p^P_{vs}\right) \left[\left(\sum_g m^P_{vgs} {t_g}^2 \right) - \frac{1}{m^P_{v+s}} \left(\sum_g m^P_{vgs} t_g \right)^2 \right] \\ V^F_v &= \sum_s p^F_{vt}\left(1 - p^F_{vt}\right) \left[\left(\sum_g m^F_{vgt} {t_g}^2 \right) - \frac{1}{m^F_{v+t}} \left(\sum_g m^F_{vgt} t_g \right)^2 \right] \end{aligned}$$

The hypergeometric variances (the default) are calculated by weighting the within-strata variances as discussed in the "Z-Score Approximation" section on page 2948.

The Peto statistic is computed as

$$Z_v = \frac{N_v^P + N_v^F - c}{\sqrt{V_v^P + V_v^F}}$$

where c is a continuity correction. The p-value is determined from the standard normal distribution unless the PERMUTATION=*number* option is used. When you use the PERMUTATION= option for PETO in the TEST statement, PROC MULTTEST computes the "discrete approximation" permutation distribution described by Mantel (1980) and Soper and Tonkonoh (1993). Specifically, the permutation distribution of

$$\sum_s \sum_g t_g S_{vgs}^P + \sum_t \sum_g t_g S_{vgt}^F$$

is computed, assuming that $\{\sum_g t_g S_{vgs}^P\}$ and $\{\sum_g t_g S_{vgt}^F\}$ are independent over all s and t. The p-values are exact under this independence assumption. However, the independence assumption is valid only asymptotically, which is why these p-values are called "approximate."

An exact permutation distribution is available only under the assumption of equal risk of censoring in all treatment groups; even then, computing this distribution can be cumbersome. Soper and Tonkonoh (1993) describe situations where the discrete approximation distribution closely fits the exact permutation distribution.

Fisher Exact Test

The CONTRAST statement in PROC MULTTEST enables you to compute Fisher exact tests for two-group comparisons. No stratification variable is allowed for this test. Note, however, that the FISHER exact test is a special case of the exact permutation tests performed by PROC MULTTEST and that these permutation tests allow a stratification variable. Recall that contrast coefficients can be -1, 0, or 1 for the Fisher test. The frequencies and sample sizes of the groups scored as -1 are combined, as are the frequencies and sample sizes of the groups scored as 1. Groups scored as 0 are excluded. The -1 group is then compared with the 1 group using the Fisher exact test.

Letting x and m denote the frequency and sample size of the 1 group, and y and n denote those of the -1 group, the p-value is calculated as

$$\Pr(X \geq x \mid X + Y = x + y) = \sum_{i=x}^{m} \frac{\binom{m}{i}\binom{n}{x+y-i}}{\binom{m+n}{x+y}}$$

where X and Y are independent binomially distributed random variables with sample sizes m and n and common probability parameters. The hypergeometric distribution is used to determine the stated probability; Yates (1984) discusses this technique.

PROC MULTTEST computes the two-tailed p-values by adding probabilities from both tails of the hypergeometric distribution. The first tail is from the observed x and y, and the other tail is chosen so that the resulting probability is as large as possible without exceeding the probability from the first tail.

t-Test for the Mean

For continuous variables, PROC MULTTEST automatically centers the contrast trend coefficients, as in the Freeman-Tukey test. These centered coefficients c_g are then used to form a t-statistic contrasting the within-group means. Let n_{vgs} denote the sample size within group g and stratum s; it depends on variable v only when there are missing values. Define

$$\bar{X}_{vgs+} = \frac{1}{n_{vgs}} \sum_r X_{vgsr}$$

as the sample mean within a group-and-stratum combination, and define

$$s_v^2 = \frac{\sum_s \sum_g \sum_r \left(X_{vgsr} - \bar{X}_{vgs+}\right)^2}{\sum_s \sum_g (n_{vgs} - 1)}$$

as the pooled sample variance. Assume constant variance for all group-and-stratum combinations. Then the t-statistic for the mean is

$$M_v = \frac{\sum_s w_{vs} \sum_g c_g \bar{X}_{vgs+}}{\sqrt{s_v^2 \left(\sum_s w_{vs}^2 \sum_g \frac{c_g^2}{n_{vgs}} \right)}}$$

where the weights w_{vs} are determined as in the Freeman-Tukey test with n_{vgs} replacing m_{vgs}.

Let μ_{vgs} denote the treatment means. Then under the null hypothesis that

$$\sum_s w_{vs} \sum_g c_g \mu_{vgs} = 0$$

and assuming normality, independence, and homoscedasticity, M_v follows a t-distribution with $\sum_s \sum_g (n_{vgs} - 1)$ degrees of freedom.

Whenever a denominator of 0 is computed, the p-value is set to 1. When missing data force $n_{vgs} = 0$, then the contribution to the denominator of the pooled variance is 0 and not -1. This is also true for degrees of freedom.

p-Value Adjustments

PROC MULTTEST offers p-value adjustments using Bonferroni, Sidak, Bootstrap resampling, and Permutation resampling, all with single-step or stepdown versions. In addition, Hochberg's (1988) and Benjamini and Hochberg's (1995) step-up methods are offered, as are Hommel's (1988) and Fisher's combination method. The Bonferroni and Sidak methods are calculated from the permutation distributions when exact permutation tests are used with CA or PETO tests.

All methods but the resampling methods are calculated using simple functions of the raw p-values or marginal permutation distributions; the permutation and bootstrap adjustments require the raw data. Because the resampling techniques incorporate distributional and correlational structures, they tend to be less conservative than the other methods.

When a resampling (bootstrap or permutation) method is used with only one test, the adjusted p-value is the bootstrap or permutation p-value for that test, with no adjustment for multiplicity, as described by Westfall and Soper (1994).

Bonferroni

Suppose that PROC MULTTEST performs R statistical tests, yielding p-values $p_1, \ldots, p_R$. Then the Bonferroni p-value for test r is simply Rp_r. If the adjusted p-value exceeds 1, it is set to 1.

If the unadjusted p-values are computed using exact permutation distributions, then the Bonferroni adjustment for p_r is $p_1^* + \cdots + p_R^*$, where p_j^* is the largest p-value from the permutation distribution of test j satisfying $p_j^* \leq p_r$, or 0 if all permutational p-values of test j are greater than p_r. These adjustments are much less conservative than the ordinary Bonferroni adjustments because they incorporate the discrete distributional characteristics. However, they remain conservative in that they do not incorporate correlation structures between multiple contrasts and multiple variables (Westfall and Wolfinger 1997).

Sidak

A technique slightly less conservative than Bonferroni is the Sidak p-value (Sidak 1967), which is $1 - (1 - p_r)^R$. It is exact when all of the p-values are uniformly distributed and independent, and it is conservative when the test statistics satisfy the positive orthant dependence condition (Holland and Copenhaver 1987).

If the unadjusted p-values are computed using exact permutation distributions, then the Sidak adjustment for p_r is $1 - (1 - p_1^*) \cdots (1 - p_R^*)$, where the p_j^* are as described previously. These adjustments are less conservative than the corresponding Bonferroni adjustments, but they do not incorporate correlation structures between multiple contrasts and multiple variables (Westfall and Wolfinger 1997).

Bootstrap

The bootstrap method creates pseudo-data sets by sampling observations with replacement from each within-stratum pool of observations. An entire data set is thus created, and p-values for all tests are computed on this pseudo-data set. A counter records whether the minimum p-value from the pseudo-data set is less than or equal to the actual p-value for each base test. (If there are R tests, then there are R such counters.) This process is repeated a large number of times, and the proportion of resampled data sets where the minimum pseudo-p-value is less than or equal to an actual p-value is the adjusted p-value reported by PROC MULTTEST. The algorithms are described by Westfall and Young (1993).

In the case of continuous data, the pooling of the groups is not likely to recreate the shape of the null hypothesis distribution, since the pooled data are likely to be multimodal. For this reason, PROC MULTTEST automatically mean-centers all continuous variables prior to resampling. Such mean-centering is akin to resampling residuals in a regression analysis, as discussed by Freedman (1981). You can specify the NOCENTER option if you do not want to center the data. (In most situations, it does not seem to make much difference whether or not you center the data.)

The bootstrap method explicitly incorporates all sources of correlation, from both the multiple contrasts and the multivariate structure. The adjusted p-values incorporate all correlations and distributional characteristics.

Permutation

The permutation-style adjusted p-values are computed in identical fashion as the bootstrap adjusted p-values, with the exception that the within-stratum resampling is performed without replacement instead of with replacement. This produces a rerandomization analysis such as in Brown and Fears (1981) and Heyse and Rom (1988). In the spirit of rerandomization analyses, the continuous variables are not centered prior to resampling. This default can be overridden by using the CENTER option.

The permutation method explicitly incorporates all sources of correlation, from both the multiple contrasts and the multivariate structure. The adjusted p-values incorporate all correlations and distributional characteristics.

Stepdown Methods

Stepdown testing is available for the Bonferroni, Sidak, bootstrap, and permutation methods. The benefit of using stepdown methods is that the tests are made more powerful (smaller adjusted p-values) while, in most cases, maintaining strong control of the familywise error rate. The stepdown method was pioneered by Holm (1979) and further developed by Shaffer (1986), Holland and Copenhaver (1987), and Hochberg and Tamhane (1987).

Suppose the base test p-values are ordered as $p_1 < p_2 < \cdots < p_R$. The Bonferroni stepdown p-values $s_1, \ldots, s_R$ are obtained from

$$\begin{aligned} s_1 &= Rp_1 \\ s_2 &= \max\left(s_1, (R-1)p_2\right) \\ s_3 &= \max\left(s_2, (R-2)p_3\right) \\ &\vdots \end{aligned}$$

As always, if any adjusted p-value exceeds 1, it is set to 1. The Sidak stepdown p-values are determined similarly:

$$\begin{aligned} s_1 &= 1-(1-p_1)^R \\ s_2 &= \max\left(s_1, 1-(1-p_2)^{R-1}\right) \\ s_3 &= \max\left(s_2, 1-(1-p_3)^{R-2}\right) \\ &\vdots \end{aligned}$$

Stepdown Bonferroni adjustments using exact tests are defined as

$$\begin{aligned} s_1 &= p_1^* + \cdots + p_R^* \\ s_2 &= \max\left(s_1, p_2^* + \cdots + p_R^*, \right) \\ s_3 &= \max\left(s_2, p_3^* + \cdots + p_R^*, \right) \\ &\vdots \end{aligned}$$

where the p_j^* are defined as before. Note that p_j^* is taken from the permutation distribution corresponding to the jth smallest unadjusted p-value. Also, any s_j greater than 1.0 is truncated to 1.0.

Stepdown Sidak adjustments for exact tests are defined analogously by substituting $1-(1-p_j^*)\cdots(1-p_R^*)$ for $p_j^* + \cdots + p_R^*$.

The resampling-style stepdown method is analogous to the preceding stepdown methods; the most extreme p-value is adjusted according to all R tests, the second-most extreme p-value is adjusted according to $(R-1)$ tests, and so on. The difference is that all correlational and distributional characteristics are incorporated when you use resampling methods. More specifically, assuming the same ordering of p-values as discussed previously, the resampling-style stepdown adjusted p-value for test r is the probability that the minimum pseudo-p-value of tests $r, \ldots, R$ is less than or equal to p_r.

This probability is evaluated using Monte Carlo, as are the previously described resampling-style adjusted p-values. In fact, the computations for stepdown adjusted p-values are essentially no more time-consuming than the computations for the non-stepdown adjusted p-values. After Monte Carlo, the stepdown adjusted p-values are corrected to ensure monotonicity; this correction leaves the first adjusted p-values alone, then corrects the remaining ones as needed. The stepdown method approximately controls the familywise error rate, and it is described in more detail by Westfall and Young (1993), Westfall *et al.* (1999), and Westfall and Wolfinger (2000).

Hochberg

Assuming p-values are independent and uniformly distributed under their respective null hypotheses, Hochberg (1988) demonstrated that Holm's stepdown adjustments control the familywise error rate even when calculated in *step-up* fashion. Since the adjusted p-values are uniformly smaller for Hochberg's method than for Holm's method, the Hochberg method is more powerful. However, this improved power comes at the cost of having to make the assumption of independence.

The Hochberg adjusted p-values are defined in reverse order as the stepdown Bonferroni:

$$\begin{aligned} s_R &= p_R \\ s_{(R-1)} &= \min\left(s_R, 2p_{(R-1)}\right) \\ s_{(R-2)} &= \min\left(s_{(R-1)}, 3p_{(R-2)}\right) \\ &\vdots \end{aligned}$$

Hommel

Hommel's (1988) method is a closed testing procedure based on Simes' (1986) test. The Simes p-value for a joint test of any set of S hypotheses with p-values $p_1 \leq p2 \leq \ldots \leq p_S$ is $\min((S/1)p_1, (S/2)p_2, \ldots, (S/S)p_S)$. The Hommel adjusted p-value for test j is the maximum of all such Simes p-values, taken over all joint tests that include j as one of their components.

Hochberg adjusted p-values are always as large or larger than Hommel adjusted p-values. Sarker and Chang (1997) showed that Simes' method is valid under independent or positively dependent p-values, so Hommel's and Hochberg's methods also are valid in such cases by the closure principle.

Fisher Combination

The FISHER_C option requests adjusted p-values using closed tests, based on the idea of Fisher's combination test. The Fisher combination test for a joint test of any set of S hypotheses with p-values uses the chi-square statistic $\chi^2 = -2\sum \log(p_i)$, with $2S$ degrees of freedom. The FISHER_C adjusted p-value for test j is the maximum of all p-values for the combination tests, taken over all joint tests that include j as one of their components. Independence of p-values is required for the validity of this method.

False Discovery Rate

The FDR option requests p-values that control the "false discovery rate," described by Benjamini and Hochberg (1995). These adjustments are potentially much less conservative than the Hochberg adjustments; however, they do not necessarily control the familywise error rate. Furthermore, they are guaranteed to control the false discovery rate only with independent p-values that are uniformly distributed under their respective null hypotheses.

The FDR adjusted p-values are defined in step-up fashion, like the Hochberg adjustments, but with less conservative multipliers:

$$\begin{aligned} s_R &= p_R \\ s_{(R-1)} &= \min\left(s_R, [R/(R-1)]p_{(R-1)}\right) \\ s_{(R-2)} &= \min\left(s_{(R-1)}, [R/(R-2)]p_{(R-2)}\right) \\ &\vdots \end{aligned}$$

Missing Values

If a CLASS or STRATA variable has a missing value, then PROC MULTTEST removes that observation from the analysis.

When there are missing values for test variables, the within group-and-stratum sample sizes may differ from variable to variable. In most cases this is not a problem; however, it is possible for all data to be missing for a particular group within a particular stratum. For continuous variables and Freeman-Tukey tests, PROC MULTTEST recenters the contrast trend coefficients within strata where all data for a particular group are missing. The Cochran-Armitage and Peto tests are unaffected by this situation.

PROC MULTTEST uses missing values for resampling if they exist in the original data set. If all variables have missing values for any observation, then PROC MULTTEST removes it prior to resampling. Otherwise, PROC MULTTEST treats all missing values as ordinary observations in the resampling. This means that different resampled data sets can have different group sizes. In some cases it means that a resampled data set can have all missing values for a particular variable in a particular group/stratum combination, even when values exist for that combination in the original data. For this reason, PROC MULTTEST recomputes all quantities within each pseudo-data set, including such items as centered scoring coefficients and degrees of freedom for p-values.

While PROC MULTTEST does provide analyses in missing value cases, you should not feel that it completely solves the missing value problem. If you are concerned about the adverse effects of missing data on a particular analysis, you should consider using imputation and sensitivity analyses to assess the effects of the missing data.

Computational Resources

PROC MULTTEST keeps all of the data in memory to expedite resampling. A large portion of the memory requirement is thus 8*NOBS*NVAR bytes, where NOBS is the number of observations in the data set, and NVAR is the number of variables analyzed, including CLASS, FREQ, and STRATA variables.

If you specify PERMUTATION=*number* (for exact permutation distributions), then PROC MULTTEST requires additional memory. This requirement is approximately 4*NTEST*NSTRATA*CMAX*number**(*number*+1) bytes, where NTEST is the number of contrasts, NSTRATA is the number of STRATA levels, and CMAX is the maximum contrast coefficient.

The execution time is linear in the number of resamples; that is, 10,000 resamples will take 10 times longer than 1,000 resamples.

Output Data Sets

OUT= Data Set

The OUT= data set contains contrast names (_test_), variable names (_var_), the contrast label (_contrast_), raw p-values (raw_p), and all requested adjusted p-values (bon_p, sid_p, stpbon_p, stpsid_p, boot_p, perm_p, stpbootp, stppermp, hoc_p, or fdr_p).

If a resampling-based adjusted p-value is requested, then the simulation standard error is included as either sim_se or stpsimse, depending upon whether single-step or stepdown adjustments are requested. The simulation standard errors are used to bound the true resampling-based adjusted p-value. For example, if the resampling-based estimate is 0.0312 and the simulation standard error is 0.00123, then a 95% confidence interval for the true adjusted p-value is $0.0312 \pm 1.96(0.00123)$, or 0.0288 to 0.0336.

Intermediate statistics used to calculate the p-values are also written to the OUT= data set. The statistics are separated by the _strat_ level. When _strat_ is reported as missing, then the statistics refer to the pooled analysis over all _strat_ levels. The p-values are provided only for the pooled analyses and are therefore reported as missing for the strata-specific statistics.

For the PETO test, an additional variable, _tstrat_, is included to indicate whether the stratum is an incidental occurrence stratum (_tstrat_=0) or a fatal occurrence stratum (_tstrat_=1).

The statistic _value_ is the per-strata contribution to the numerator of the overall test statistic. In the case of the MEAN test, this is the contrast function of the sample means multiplied by the total number of observations within the stratum. For the FT test, _value_ is the contrast function of the double-arcsine transformed proportions, again multiplied by the total number of observations within the stratum. For the CA and PETO tests, _value_ is the observed value of the trend statistic within that stratum.

When either PETO or CA is requested, the variable _exp_ is included; this variable contains the expected value of the trend statistic for the given stratum.

The statistic _se_ is the square root of the variance of the per-strata _value_ value for any of the tests.

For MEAN tests, the variable _nval_ is included. When reported with an individual stratum level (that is, when the _strat_ value is nonmissing), the value _nval_ refers to the within-stratum sample size. For the combined analysis (that is, the value of the _strat_ is missing), the value _nval_ contains degrees of freedom of the t-distribution used to compute the unadjusted p-value.

When the FISHER test is requested, the OUT= data set contains variables _xval_, _mval_, _yval_, and _nval_, which define observations and sample sizes in the two groups defined by the CONTRAST statement.

For example, the OUT= data set from the drug example in the "Getting Started" section on page 2936 is displayed in Figure 48.4.

```
Obs _test_   _var_    _contrast_ _value_ _exp_   _se_    raw_p   boot_p     sim_se

  1   CA   SideEff1    Trend         8      5   1.54303 0.05187 0.34705 .003366053
  2   CA   SideEff2    Trend         7      5   1.54303 0.19492 0.83880 .002600140
  3   CA   SideEff3    Trend        10      7   1.63299 0.06619 0.52315 .003531742
  4   CA   SideEff4    Trend        10      6   1.60357 0.01262 0.09370 .002060586
  5   CA   SideEff5    Trend         7      4   1.44749 0.03821 0.24380 .003036129
  6   CA   SideEff6    Trend         9      6   1.60357 0.06137 0.44545 .003514430
  7   CA   SideEff7    Trend         9      5   1.54303 0.00953 0.05400 .001598186
  8   CA   SideEff8    Trend         8      5   1.54303 0.05187 0.34705 .003366053
  9   CA   SideEff9    Trend         7      5   1.54303 0.19492 0.83880 .002600140
 10   CA   SideEff10   Trend         8      6   1.60357 0.21232 0.90020 .002119433
```

Figure 48.4. Output Data for the MULTTEST Procedure

OUTPERM= Data Set

The OUTPERM= data set contains contrast names (_contrast_), variable names (_var_), and the associated permutation distributions (_value_ and upper_p). PROC MULTTEST computes the permutation distributions when you use the PERMUTATION= option with the CA or Peto tests. The _value_ variable represents the support of the distributions, and upper_p represents their cumulative upper-tail probabilities. The size of this data set depends on the number of variables and the support of their permutation distributions. For information on how this distribution is computed, see the "Exact Permutation Test" section on page 2949. For an illustration, see Example 48.1 on page 2964.

OUTSAMP= Data Set

The OUTSAMP= data set contains the data sets used in the resampling analysis, if such an analysis is requested. The variable _sample_ indicates the number of the resampled data set. This variable ranges from 1 to NSAMPLE. For each value of the _sample_ variable, an entire resampled data set is included, with _strat_, _class_, and all other variables in the original data set. The values of the original variables are mean-centered for the mean test, if requested. The variable _obs_ indicates the observation's position in the original data set.

Each new data set is randomly drawn from the original data set, either with (bootstrap) or without (permutation) replacement. The size of this data set is, thus, the number of observations in the original data set times the number of samples.

Displayed Output

The output produced by PROC MULTTEST is divided into several tables:

- The “Model Information” table provides a list of the options and settings used for that particular invocation of the procedure. Included in this list are the following items:
  - statistical tests
  - support of the exact permutation distribution for the CA and PETO tests
  - continuity corrections used for the CA test
  - test tails
  - strata adjustment
  - p-value adjustments
  - centering of continuous variables
  - number of samples and seed
- The “Contrast Coefficients” table lists the coefficients used in constructing the statistical tests. These coefficients are either specified in CONTRAST statements or generated by default. The coefficients apply to the levels of the CLASS statement variable.
- The “Variable Tabulations” tables provide summary statistics for each variable listed in the TEST statement. Included for discrete variables are the count, sample size, and percentage of occurrences. For continuous variables, the mean, sample standard deviation, and sample size are displayed. All of the previously mentioned statistics are computed for distinct combinations of the CLASS and STRATA statement variables.
- The “p-Values” table is a collection of the raw and adjusted p-values from this run of PROC MULTTEST. The p-values are listed by variable and test.

ODS Table Names

PROC MULTTEST assigns a name to each table it creates, and you must use this name to reference the table when using the Table Delivery System (ODS). These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 48.1. ODS Tables Created by the MULTTEST Procedure

ODS Table Name	Description	Statement
Continuous	Continuous variable tabulations	TEST with MEAN
Contrasts	Contrast coefficients	default
Discrete	Discrete variable tabulations	TEST with CA, FT, PETO, or FISHER
ModelInfo	Model information	default
pValues	p-values from the tests	default

Examples

Example 48.1. Cochran-Armitage Test with Permutation Resampling

This example, from Keith Soper at Merck, illustrates the exact permutation Cochran-Armitage test carried out on permutation resamples. In the following data set, each observation represents an animal. The binary variables S1 and S2 indicate two tumor types, with 0s indicating no tumor (failure) and 1 indicating a tumor (success); note that they have perfect negative association. The grouping variable is Dose.

```
data a;
   input S1 S2 Dose @@;
   datalines;
0 1 1   1 0 1   0 1 1
0 1 1   0 1 1   1 0 1
1 0 2   1 0 2   0 1 2
1 0 2   0 1 2   1 0 2
1 0 3   1 0 3   1 0 3
0 1 3   0 1 3   1 0 3
;
proc multtest data=a permutation nsample=10000
              seed=36607 outperm=pmt pvals;
   test ca(S1 S2 / permutation=10 uppertailed);
   class Dose;
   contrast 'CA Linear Trend' 0 1 2;
run;
proc print data=pmt;
run;
```

The PROC MULTTEST statement requests 10,000 permutation resamples. The OUTPERM=PMT option creates an output SAS data set for the exact permutation distribution computed for the CA test.

The TEST statement specifies an upper-tailed Cochran-Armitage linear trend test for S1 and S2. The cutoff for exact permutation calculations is 10, as specified with the PERMUTATION= option in the TEST statement. Since S1 and S2 have ten and eight successes, respectively, PROC MULTTEST uses exact permutation distributions to compute the p-values for both variables.

The groups for the CA test are the levels of Dose from the CLASS statement. The trend coefficients applied to these groups are 0, 1, and 2, respectively, as specified in the CONTRAST statement.

Finally, PROC PRINT displays the SAS data set containing the permutation distributions.

The results from this analysis are listed in Output 48.1.1 through Output 48.1.5.

Output 48.1.1. Cochran-Armitage Test with Permutation Resampling

```
                              The Multtest Procedure

                                Model Information

         Test for discrete variables                 Cochran-Armitage
         Exact permutation distribution used         Everywhere
         Tails for discrete tests                    Upper-tailed
         Strata weights                              None
         P-value adjustment                          Permutation
         Number of resamples                         10000
         Seed                                        36607
```

You should check the preceding table to verify that the analysis specifications are correct.

Output 48.1.2. Contrast Coefficients

```
                              The Multtest Procedure

                              Contrast Coefficients

                                                Dose

         Contrast                      1               2               3

         CA Linear Trend               0               1               2
```

The preceding table lists the label and coefficients from the CONTRAST statement.

Output 48.1.3. Summary Statistics

```
                              The Multtest Procedure

                          Discrete Variable Tabulations

                 Variable    Dose     Count    NumObs    Percent

                 S1          1            2         6      33.33
                 S1          2            4         6      66.67
                 S1          3            4         6      66.67
                 S2          1            4         6      66.67
                 S2          2            2         6      33.33
                 S2          3            2         6      33.33
```

The preceding table contains summary statistics for the two test variables, S1 and S2. The Count column lists the number of successes for each level of the class variable, Dose. The NumObs column is the sample size, and the Percent column is the percentage of successes in the sample.

Output 48.1.4. Resulting *p*-Values

```
                              The Multtest Procedure

                                     p-Values

             Variable    Contrast                Raw    Permutation

             S1          CA Linear Trend      0.1993         0.4058
             S2          CA Linear Trend      0.9220         1.0000
```

The Raw column in the preceding “p-Values” table contains the p-values from the CA test, and the Permutation column contains the permutation-adjusted p-values.

This table shows that, for S1, the adjusted p-value is almost twice the raw p-value. In fact, from theoretical considerations, the permutation-adjusted p-value for S1 should be $2 \times 0.1993 = 0.3986$; the difference is due to resampling error. For S2, the raw p-value is 0.9220, and the adjusted p-value equals 1, as you would expect from theoretical considerations. The permutation p-values for S1 and S2 also happen to be the Bonferroni-adjusted p-values for this example.

Output 48.1.5. Exact Permutation Distribution

```
          Obs      _contrast_       _var_    _value_    upper_p

            1    CA Linear Trend     S1          0      1.00000
            2    CA Linear Trend     S1          1      1.00000
            3    CA Linear Trend     S1          2      1.00000
            4    CA Linear Trend     S1          3      1.00000
            5    CA Linear Trend     S1          4      1.00000
            6    CA Linear Trend     S1          5      0.99966
            7    CA Linear Trend     S1          6      0.99609
            8    CA Linear Trend     S1          7      0.97827
            9    CA Linear Trend     S1          8      0.92205
           10    CA Linear Trend     S1          9      0.80070
           11    CA Linear Trend     S1         10      0.61011
           12    CA Linear Trend     S1         11      0.38989
           13    CA Linear Trend     S1         12      0.19930
           14    CA Linear Trend     S1         13      0.07795
           15    CA Linear Trend     S1         14      0.02173
           16    CA Linear Trend     S1         15      0.00391
           17    CA Linear Trend     S1         16      0.00034
           18    CA Linear Trend     S1         17      0.00000
           19    CA Linear Trend     S1         18      0.00000
           20    CA Linear Trend     S1         19      0.00000
           21    CA Linear Trend     S1         20      0.00000
           22    CA Linear Trend     S2          0      1.00000
           23    CA Linear Trend     S2          1      1.00000
           24    CA Linear Trend     S2          2      1.00000
           25    CA Linear Trend     S2          3      0.99966
           26    CA Linear Trend     S2          4      0.99609
           27    CA Linear Trend     S2          5      0.97827
           28    CA Linear Trend     S2          6      0.92205
           29    CA Linear Trend     S2          7      0.80070
           30    CA Linear Trend     S2          8      0.61011
           31    CA Linear Trend     S2          9      0.38989
           32    CA Linear Trend     S2         10      0.19930
           33    CA Linear Trend     S2         11      0.07795
           34    CA Linear Trend     S2         12      0.02173
           35    CA Linear Trend     S2         13      0.00391
           36    CA Linear Trend     S2         14      0.00034
           37    CA Linear Trend     S2         15      0.00000
           38    CA Linear Trend     S2         16      0.00000
```

The preceding table lists the OUTPERM= data set, which contains the exact permutation distributions for S1 and S2 in terms of cumulative probabilities.

Example 48.2. Freeman-Tukey and *t*-Tests with Bootstrap Resampling

The data for the following example are the same as for Example 48.1, except that a continuous variable T, which indicates the time of death of the animal, has been added.

```
data a;
   input S1 S2 T Dose @@;
   datalines;
0 1 104 1   1 0  80 1   0 1 104 1
0 1 104 1   0 1 100 1   1 0 104 1
1 0  85 2   1 0  60 2   0 1  89 2
1 0  96 2   0 1  96 2   1 0  99 2
1 0  60 3   1 0  50 3   1 0  80 3
0 1  98 3   0 1  99 3   1 0  50 3
;
proc multtest data=a bootstrap nsample=10000
              pvals seed=37081 outsamp=res;
   test ft(S1 S2 / lowertailed) mean(T / lowertailed);
   class Dose;
   contrast 'Linear Trend' 0 1 2;
run;
proc print data=res(obs=36);
run;
```

The BOOTSTRAP option in the PROC MULTTEST statement requests bootstrap resampling, and NSAMPLE=10000 requests 10,000 bootstrap samples. The seed for the random number generation is 37081. The OUTSAMP=RES option creates an output SAS data set containing the 10,000 bootstrap samples.

The TEST statement specifies the Freeman-Tukey test for S1 and S2 and specifies the *t*-test for T. Both tests are lower-tailed. The grouping variable in the CLASS statement is Dose, and the coefficients across the levels of Dose are 0, 1, and 2, as specified in the CONTRAST statement. PROC PRINT displays the first 36 observations of the Res data set containing the bootstrap samples.

The results from this analysis are listed in Output 48.2.1 through Output 48.2.5.

Output 48.2.1. FT and *t*-tests with Bootstrap Resampling

```
                              The Multtest Procedure

                                Model Information

         Test for discrete variables                   Freeman-Tukey
         Test for continuous variables                 Mean t-test
         Tails for discrete tests                      Lower-tailed
         Tails for continuous tests                    Lower-tailed
         Strata weights                                None
         P-value adjustment                            Bootstrap
         Center continuous variables                   Yes
         Number of resamples                           10000
         Seed                                          37081
```

The information in the preceding table corresponds to the specifications in the invocation of PROC MULTTEST.

Output 48.2.2. Contrast Coefficients

```
                              The Multtest Procedure

                              Contrast Coefficients

                                                Dose

           Contrast                    1                2                3

           Linear Trend                0                1                2
```

The preceding table shows the coefficients from the CONTRAST statement, and they model a linear trend.

Output 48.2.3. Summary Statistics

```
                              The Multtest Procedure

                          Discrete Variable Tabulations

                 Variable     Dose      Count     NumObs     Percent

                 S1           1             2          6       33.33
                 S1           2             4          6       66.67
                 S1           3             4          6       66.67
                 S2           1             4          6       66.67
                 S2           2             2          6       33.33
                 S2           3             2          6       33.33


                         Continuous Variable Tabulations

                                                                 Standard
              Variable     Dose     NumObs          Mean        Deviation

              T            1             6       99.3333           9.6056
              T            2             6       87.5000          14.4326
              T            3             6       72.8333          22.7017
```

The summary statistics in the preceding table for S1 and S2 are the same as those from Example 48.1. The variables S1 and S2 are discrete, and T is a continuous variable. The mean, standard deviation, and sample size for each level of Dose is listed in the table for T. The p-values for S1 and S2 are from the Freeman-Tukey test, and the p-values for T are from the t-test.

Output 48.2.4. *p*-Values

```
                              The Multtest Procedure

                                     p-Values

               Variable     Contrast               Raw     Bootstrap

               S1           Linear Trend        0.8547        1.0000
               S2           Linear Trend        0.1453        0.4471
               T            Linear Trend        0.0070        0.0253
```

The p-values are listed in the preceding table. The Raw column contains the results from the tests on the original data, and the Bootstrap column contains the bootstrap resampled adjustment to raw_p. Note that the adjusted p-values are larger than the raw p-values for all three variables. The adjusted p-values more accurately reflect the correlation of the raw p-values, the small size of the data, and the multiple testing.

Output 48.2.5. Resampling Data Set

```
           Obs    _sample_    _class_    _obs_    S1    S2           T

             1        1          1         11      0     1      8.5000
             2        1          1         16      0     1     25.1667
             3        1          1         16      0     1     25.1667
             4        1          1         14      1     0    -22.8333
             5        1          1         18      1     0    -22.8333
             6        1          1         14      1     0    -22.8333
             7        1          2          4      0     1      4.6667
             8        1          2         12      1     0     11.5000
             9        1          2          8      1     0    -27.5000
            10        1          2          7      1     0     -2.5000
            11        1          2          3      0     1      4.6667
            12        1          2         12      1     0     11.5000
            13        1          3         13      1     0    -12.8333
            14        1          3          5      0     1      0.6667
            15        1          3          8      1     0    -27.5000
            16        1          3          5      0     1      0.6667
            17        1          3         13      1     0    -12.8333
            18        1          3          6      1     0      4.6667
            19        2          1          8      1     0    -27.5000
            20        2          1          3      0     1      4.6667
            21        2          1          9      0     1      1.5000
            22        2          1         13      1     0    -12.8333
            23        2          1         14      1     0    -22.8333
            24        2          1         12      1     0     11.5000
            25        2          2         14      1     0    -22.8333
            26        2          2         18      1     0    -22.8333
            27        2          2         15      1     0      7.1667
            28        2          2          6      1     0      4.6667
            29        2          2         13      1     0    -12.8333
            30        2          2          1      0     1      4.6667
            31        2          3          7      1     0     -2.5000
            32        2          3          7      1     0     -2.5000
            33        2          3          6      1     0      4.6667
            34        2          3         13      1     0    -12.8333
            35        2          3          4      0     1      4.6667
            36        2          3          6      1     0      4.6667
```

The preceding table lists the first 36 observations of the SAS data set resulting from the OUTSAMP=RES option in the PROC MULTTEST statement. The entire data set has 180,000 observations, which is 10,000 times the number of observations in the data set. The _sample_ variable is the sample indicator and _class_ indicates the resampling group, that is, the level of the CLASS variable Dose assigned to the new observation. The number of the observation in the original data set is represented by _obs_. Also listed are the values of the original test variables, S1 and S2, and the mean-centered values of T.

Example 48.3. Peto Mortality-Prevalence Test

This example illustrates the use of the Peto mortality-prevalence test. The test is a combination of analyses about the prevalence of incidental tumors in the population and mortality due to fatal tumors.

In the data set, each observation represents an animal. The variables S1–S3 are three tumor types, with a value of 0 indicating no tumor, 1 indicating an incidental (nonlethal) tumor, and 2 indicating a lethal tumor. The time variable T indicates the time of death of the animal, a strata variable B is constructed from T, and the grouping variable Dose is drug dosage.

```
data a;
   input S1-S3 T Dose @@;
   if T<=90 then B=1; else B=2;
   datalines;
0 0 0 104 0   2 0 1  80 0   0 0 1 104 0
0 0 0 104 0   0 2 0 100 0   1 0 0 104 0
2 0 0  85 1   2 1 0  60 1   0 1 0  89 1
2 0 1  96 1   0 0 0  96 1   2 0 1  99 1
2 1 1  60 2   2 0 0  50 2   2 0 1  80 2
0 0 2  98 2   0 0 1  99 2   2 1 1  50 2
;
proc multtest data=a notables out=p stepsid;
   test peto(S1-S3 / permutation=20 time=T uppertailed);
   class Dose;
   strata B;
   contrast 'mort-prev' 0  1  2;
run;
proc print data=p;
run;
```

The NOTABLES option in the PROC MULTTEST statement suppresses the display of the summary statistics for each variable. The OUT=P option creates an output SAS data set containing all p-values and intermediate statistics. The STEPSID option is used to adjust the p-values.

The TEST statement specifies an upper-tailed Peto test for S1–S3. The mortality strata are defined by TIME=T, the death times. The CLASS statement contains the grouping variable Dose. The prevalence strata are defined by the STRATA statement as the blocking variable B. The CONTRAST statement lists the default linear trend coefficients. PROC PRINT displays the requested p-value data set.

The results from this analysis are listed in Output 48.3.1 through Output 48.3.4.

Output 48.3.1. Peto Test

```
                         The Multtest Procedure

                           Model Information

        Test for discrete variables                 Peto
        Exact permutation distribution used         Everywhere
        Tails for discrete tests                    Upper-tailed
        Strata weights                              Sample size
        P-value adjustment                          Stepdown Sidak
```

The preceding information corresponds to the PROC MULTTEST invocation. In this case the totals for all prevalence and fatality strata are less than 20, so exact permutation tests are used everywhere, and the STEPSID adjustments are computed from these permutation distributions.

Output 48.3.2. Contrast Coefficients

```
                             The Multtest Procedure

                              Contrast Coefficients

                                          Dose

                Contrast          0              1              2

                mort-prev         0              1              2
```

The contrast trend coefficients are listed in the preceding table. They happen to be the same as the levels of the Dose variable.

Output 48.3.3. *p*-Values

```
                             The Multtest Procedure

                                    p-Values

                                                                Stepdown
                    Variable    Contrast             Raw           Sidak

                    S1          mort-prev         0.0681          0.0814
                    S2          mort-prev         0.5000          0.5000
                    S3          mort-prev         0.0363          0.0781
```

In the preceding "*p*-Values" table, the *p*-values for the Peto tests are listed in the Raw column, and the stepdown Sidak adjusted *p*-values are in the Stepdown Sidak column.

Significant *p*-values support the claim that higher dosage levels promote higher mortality and prevalence. The raw Peto test is significant at the 5% level for S3, but the adjusted S3 test is no longer significant at 5%. The raw and adjusted *p*-values for S2 are the same because of the stepdown technique.

Output 48.3.4. OUT= Data Set

```
Obs  _test_  _var_  _contrast_  _strat_  _tstrat_  _value_   _exp_      _se_     raw_p    stpsid_p

  1   PETO    S1    mort-prev     1          0         0    0.00000  0.00000   .          .
  2   PETO    S1    mort-prev     2          0         0    0.62500  0.85696   .          .
  3   PETO    S1    mort-prev     50         1         4    2.00000  1.12022   .          .
  4   PETO    S1    mort-prev     60         1         3    1.75000  1.06654   .          .
  5   PETO    S1    mort-prev     80         1         2    1.57143  1.04978   .          .
  6   PETO    S1    mort-prev     85         1         1    0.75000  0.72169   .          .
  7   PETO    S1    mort-prev     96         1         1    0.70000  0.78102   .          .
  8   PETO    S1    mort-prev     98         1         0    0.00000  0.00000   .          .
  9   PETO    S1    mort-prev     99         1         1    0.42857  0.72843   .          .
 10   PETO    S1    mort-prev     100        1         0    0.00000  0.00000   .          .
 11   PETO    S2    mort-prev     1          0         6    5.50000  1.05221   .          .
 12   PETO    S2    mort-prev     2          0         0    0.00000  0.00000   .          .
 13   PETO    S2    mort-prev     50         1         0    0.00000  0.00000   .          .
 14   PETO    S2    mort-prev     60         1         0    0.00000  0.00000   .          .
 15   PETO    S2    mort-prev     80         1         0    0.00000  0.00000   .          .
 16   PETO    S2    mort-prev     85         1         0    0.00000  0.00000   .          .
 17   PETO    S2    mort-prev     96         1         0    0.00000  0.00000   .          .
 18   PETO    S2    mort-prev     98         1         0    0.00000  0.00000   .          .
 19   PETO    S2    mort-prev     99         1         0    0.00000  0.00000   .          .
 20   PETO    S2    mort-prev     100        1         0    0.00000  0.00000   .          .
 21   PETO    S3    mort-prev     1          0         6    5.50000  1.05221   .          .
 22   PETO    S3    mort-prev     2          0         4    2.22222  1.08298   .          .
 23   PETO    S3    mort-prev     50         1         0    0.00000  0.00000   .          .
 24   PETO    S3    mort-prev     60         1         0    0.00000  0.00000   .          .
 25   PETO    S3    mort-prev     80         1         0    0.00000  0.00000   .          .
 26   PETO    S3    mort-prev     85         1         0    0.00000  0.00000   .          .
 27   PETO    S3    mort-prev     96         1         0    0.00000  0.00000   .          .
 28   PETO    S3    mort-prev     98         1         2    0.62500  0.85696   .          .
 29   PETO    S3    mort-prev     99         1         0    0.00000  0.00000   .          .
 30   PETO    S3    mort-prev     100        1         0    0.00000  0.00000   .          .
 31   PETO    S1    mort-prev     .          .        12    7.82500  2.42699  0.06808   0.08140
 32   PETO    S2    mort-prev     .          .         6    5.50000  1.05221  0.50000   0.50000
 33   PETO    S3    mort-prev     .          .        12    8.34722  1.73619  0.03627   0.07811
```

The preceding table lists the OUT= data set. The first 30 observations correspond to intermediate statistics used to compute the Peto p-values. The _test_ variable lists the name of the test, the _var_ variable lists the name of the TEST variables, and the _contrast_ variable lists the CONTRAST label. The _strat_ variable lists the level of the STRATA variable, and the _tstrat_ variable indicates whether or not the stratum corresponds to values of the TIME= variable. The _value_ variable is the observed contrast for a stratum and the _exp_ variable is its expected value. The variable _se_ contains the square root of the variance terms summed to form the denominator of the Peto statistics.

The final three observations correspond to the three Peto tests, with their p-values listed under the raw_p variable. The stpsid_p variable contains the stepdown Sidak adjusted p-values.

Example 48.4. Fisher Test with Permutation Resampling

These data, from Brown and Fears (1981), are the results from an 80-week carcinogenesis bioassay with female mice. Six tissue sites are examined at necropsy; 1 indicates the presence of a tumor and 0 the absence. A frequency variable Freq is included. A control and four different doses of a drug (in parts per milliliter) make up the levels of the grouping variable Dose.

```
data a;
   input Liver Lung Lymph Cardio Pitui Ovary Freq Dose$ @@;
   datalines;
1 0 0 0 0 0 8  CTRL   0 1 0 0 0 0 7  CTRL   0 0 1 0 0 0 6  CTRL
0 0 0 1 0 0 1  CTRL   0 0 0 0 0 1 2  CTRL   1 1 0 0 0 0 4  CTRL
1 0 1 0 0 0 1  CTRL   1 0 0 0 0 1 1  CTRL   0 1 1 0 0 0 1  CTRL
0 0 0 0 0 0 18 CTRL
1 0 0 0 0 0 9  4PPM   0 1 0 0 0 0 4  4PPM   0 0 1 0 0 0 7  4PPM
0 0 0 1 0 0 1  4PPM   0 0 0 0 1 0 2  4PPM   0 0 0 0 0 1 1  4PPM
1 1 0 0 0 0 4  4PPM   1 0 1 0 0 0 3  4PPM   1 0 0 0 1 0 1  4PPM
0 1 1 0 0 0 1  4PPM   0 1 0 1 0 0 1  4PPM   1 0 1 1 0 0 1  4PPM
0 0 0 0 0 0 15 4PPM
1 0 0 0 0 0 8  8PPM   0 1 0 0 0 0 3  8PPM   0 0 1 0 0 0 6  8PPM
0 0 0 1 0 0 3  8PPM   1 1 0 0 0 0 1  8PPM   1 0 1 0 0 0 2  8PPM
1 0 0 1 0 0 1  8PPM   1 0 0 0 1 0 1  8PPM   1 1 0 1 0 0 2  8PPM
1 1 0 0 0 1 2  8PPM   0 0 0 0 0 0 19 8PPM
1 0 0 0 0 0 4  16PPM  0 1 0 0 0 0 2  16PPM  0 0 1 0 0 0 9  16PPM
0 0 0 0 1 0 1  16PPM  0 0 0 0 0 1 1  16PPM  1 1 0 0 0 0 4  16PPM
1 0 1 0 0 0 1  16PPM  0 1 1 0 0 0 1  16PPM  0 1 0 1 0 0 1  16PPM
0 1 0 0 0 1 1  16PPM  0 0 1 1 0 0 1  16PPM  0 0 1 0 1 0 1  16PPM
1 1 1 0 0 0 2  16PPM  0 0 0 0 0 0 14 16PPM
1 0 0 0 0 0 8  50PPM  0 1 0 0 0 0 4  50PPM  0 0 1 0 0 0 8  50PPM
0 0 0 1 0 0 1  50PPM  0 0 0 0 0 1 4  50PPM  1 1 0 0 0 0 3  50PPM
1 0 1 0 0 0 1  50PPM  0 1 1 0 0 0 1  50PPM  0 1 0 0 1 1 1  50PPM
0 0 0 0 0 0 19 50PPM
;
proc multtest data=a order=data notables out=p
              permutation nsample=1000 seed=764511;
   test fisher(Liver Lung Lymph Cardio Pitui Ovary /
               lowertailed);
   class Dose;
   freq Freq;
run;
proc print data=p;
run;
```

In the PROC MULTTEST statement, the ORDER=DATA option is required to keep the levels of Dose in the order in which they appear in the data set. Without this option, the levels are sorted by their formatted value, resulting in an alphabetic ordering. The NOTABLES option suppresses the display of summary statistics, and the OUT=P option requests an output data set containing *p*-values. The PERMUTATION option specifies permutation resampling, NSAMPLE=1000 requests 1000 samples, and SEED=764511 provides a starting value for the random number generator. You should specify a seed if you need to duplicate resampling results.

To test for higher rates of tumor occurrence in the treatment groups compared to the control group, the LOWERTAILED option is specified in the TEST statement to produce a lower-tailed Fisher exact test for the six tissue sites. The Fisher test is appropriate for comparing a treatment and a control, but multiple testing can be a problem. Brown and Fears (1981) use a multivariate permutation to evaluate the entire collection of tests. PROC MULTTEST adjusts the p-values by simulation.

The treatments make up the levels of the grouping variable Dose, listed in the CLASS statement. Since no CONTRAST statement is specified, PROC MULTTEST uses the default pairwise contrasts with the first level of Dose. The FREQ statement is used since this is summary data containing frequency counts of occurrences.

The results from this analysis are listed in Output 48.4.1 through Output 48.4.4.

Output 48.4.1. Fisher Test with Permutation Resampling

```
                                    The Multtest Procedure

                                      Model Information

                Test for discrete variables                   Fisher
                Tails for discrete tests                      Lower-tailed
                Strata weights                                None
                P-value adjustment                            Permutation
                Number of resamples                           1000
                Seed                                          764511
```

The preceding table lists the PROC MULTTEST specifications.

Output 48.4.2. Default Contrast Coefficients

```
                                    The Multtest Procedure

                                    Contrast Coefficients

                                                         Dose

Contrast                   CTRL            4PPM            8PPM           16PPM           50PPM

CTRL vs. 4PPM                 1              -1               0               0               0
CTRL vs. 8PPM                 1               0              -1               0               0
CTRL vs. 16PPM                1               0               0              -1               0
CTRL vs. 50PPM                1               0               0               0              -1
```

The preceding table lists the default contrasts for the Fisher test. Note that each dose is compared with the control.

Output 48.4.3. *p*-Values

```
                                The Multtest Procedure

                                       p-Values

                    Variable    Contrast                Raw    Permutation

                    Liver       CTRL vs. 4PPM        0.2828         0.9690
                    Liver       CTRL vs. 8PPM        0.3069         0.9750
                    Liver       CTRL vs. 16PPM       0.7102         1.0000
                    Liver       CTRL vs. 50PPM       0.7718         1.0000
                    Lung        CTRL vs. 4PPM        0.7818         1.0000
                    Lung        CTRL vs. 8PPM        0.8858         1.0000
                    Lung        CTRL vs. 16PPM       0.5469         1.0000
                    Lung        CTRL vs. 50PPM       0.8498         1.0000
                    Lymph       CTRL vs. 4PPM        0.2423         0.9430
                    Lymph       CTRL vs. 8PPM        0.5898         1.0000
                    Lymph       CTRL vs. 16PPM       0.0350         0.2480
                    Lymph       CTRL vs. 50PPM       0.4161         0.9960
                    Cardio      CTRL vs. 4PPM        0.3163         0.9770
                    Cardio      CTRL vs. 8PPM        0.0525         0.3570
                    Cardio      CTRL vs. 16PPM       0.4506         1.0000
                    Cardio      CTRL vs. 50PPM       0.7576         1.0000
                    Pitui       CTRL vs. 4PPM        0.1250         0.7260
                    Pitui       CTRL vs. 8PPM        0.4948         1.0000
                    Pitui       CTRL vs. 16PPM       0.2157         0.9050
                    Pitui       CTRL vs. 50PPM       0.5051         1.0000
                    Ovary       CTRL vs. 4PPM        0.9437         1.0000
                    Ovary       CTRL vs. 8PPM        0.8126         1.0000
                    Ovary       CTRL vs. 16PPM       0.7760         1.0000
                    Ovary       CTRL vs. 50PPM       0.3689         0.9950
```

The preceding “p-Values” table lists p-values for the Fisher exact tests and their permutation-based adjustments. As noted by Brown and Fears, only one of the twenty-four tests is significant at the 5% level (Lymph, CTRL vs. 16PPM). Brown and Fears report a 12% chance of observing at least one significant raw p-value for 16PPM and a 9% chance of observing at least one significant raw p-value for Lymph (both at the 5% level). Adjusted p-values exhibit much lower chances of false significances. For this example, none of the adjusted p-values are close to significant.

Output 48.4.4. OUT= Data Set

```
Obs  _test_  _var_     _contrast_    _xval_  _mval_  _yval_  _nval_   raw_p   perm_p   sim_se

  1  FISHER  Liver   CTRL vs. 4PPM     14      49      18      50    0.28282  0.969  0.005481
  2  FISHER  Liver   CTRL vs. 8PPM     14      49      17      48    0.30688  0.975  0.004937
  3  FISHER  Liver   CTRL vs. 16PPM    14      49      11      43    0.71022  1.000  0.000000
  4  FISHER  Liver   CTRL vs. 50PPM    14      49      12      50    0.77175  1.000  0.000000
  5  FISHER  Lung    CTRL vs. 4PPM     12      49      10      50    0.78180  1.000  0.000000
  6  FISHER  Lung    CTRL vs. 8PPM     12      49       8      48    0.88581  1.000  0.000000
  7  FISHER  Lung    CTRL vs. 16PPM    12      49      11      43    0.54685  1.000  0.000000
  8  FISHER  Lung    CTRL vs. 50PPM    12      49       9      50    0.84978  1.000  0.000000
  9  FISHER  Lymph   CTRL vs. 4PPM      8      49      12      50    0.24228  0.943  0.007332
 10  FISHER  Lymph   CTRL vs. 8PPM      8      49       8      48    0.58977  1.000  0.000000
 11  FISHER  Lymph   CTRL vs. 16PPM     8      49      15      43    0.03498  0.248  0.013656
 12  FISHER  Lymph   CTRL vs. 50PPM     8      49      10      50    0.41607  0.996  0.001996
 13  FISHER  Cardio  CTRL vs. 4PPM      1      49       3      50    0.31631  0.977  0.004740
 14  FISHER  Cardio  CTRL vs. 8PPM      1      49       6      48    0.05254  0.357  0.015151
 15  FISHER  Cardio  CTRL vs. 16PPM     1      49       2      43    0.45061  1.000  0.000000
 16  FISHER  Cardio  CTRL vs. 50PPM     1      49       1      50    0.75758  1.000  0.000000
 17  FISHER  Pitui   CTRL vs. 4PPM      0      49       3      50    0.12496  0.726  0.014104
 18  FISHER  Pitui   CTRL vs. 8PPM      0      49       1      48    0.49485  1.000  0.000000
 19  FISHER  Pitui   CTRL vs. 16PPM     0      49       2      43    0.21572  0.905  0.009272
 20  FISHER  Pitui   CTRL vs. 50PPM     0      49       1      50    0.50505  1.000  0.000000
 21  FISHER  Ovary   CTRL vs. 4PPM      3      49       1      50    0.94372  1.000  0.000000
 22  FISHER  Ovary   CTRL vs. 8PPM      3      49       2      48    0.81260  1.000  0.000000
 23  FISHER  Ovary   CTRL vs. 16PPM     3      49       2      43    0.77596  1.000  0.000000
 24  FISHER  Ovary   CTRL vs. 50PPM     3      49       5      50    0.36889  0.995  0.002230
```

The preceding table lists the OUT= data set. The _test_, _var_, and _contrast_ variables provide the TEST name, TEST variable, and CONTRAST label, respectively. The _xval_, _mval_, _yval_, and _nval_ variables contain the components used to compute the Fisher exact tests from the hypergeometric distribution. The raw_p variable contains the p-values from the Fisher exact tests, and the perm_p variable contains their permutation-based adjustments. The variable sim_se is the simulation standard error from the permutation resampling.

Example 48.5. Inputting Raw *p*-Values

This example illustrates how to use PROC MULTTEST to multiplicity-adjust a collection of raw p-values obtained from some other source. This is a valuable option for those cases where PROC MULTTEST cannot compute the raw p-values directly.

```
data a;
   input Test$ Raw_P;
   datalines;
test1 .09108
test2 .69122
test3 .00177
test4 .57181
test5 .03121
test6 .01413
;
proc multtest pdata=a holm hoc fdr;
run;
```

Note that there are no statements other than the PROC MULTTEST statement using the p-value input mode. In this example, the raw p-values are adjusted using the

Holm, Hochberg, and Benjamini and Hocherg (FDR) methods.

The output from this analysis is listed in Output 48.5.1.

Output 48.5.1. Inputting Raw *p*-Values

```
                         The Multtest Procedure

                                p-Values

                                                                False
                                   Stepdown                 Discovery
        Test          Raw        Bonferroni      Hochberg        Rate

           1       0.0911            0.2732        0.2732      0.1366
           2       0.6912            1.0000        0.6912      0.6912
           3       0.0018            0.0106        0.0106      0.0106
           4       0.5718            1.0000        0.6912      0.6862
           5       0.0312            0.1248        0.1248      0.0624
           6       0.0141            0.0707        0.0707      0.0424
```

Note that the adjusted p-values for the Hochberg method (hoc_p) are less than or equal to those for the Holm method (stpbon_p). In turn, the adjusted p-values for the Benjamini and Hochberg method (fdr_p) are less than or equal to those for the Hochberg method. These comparisons hold generally for all p-value configurations. The FDR method controls the false discovery rate and not the familywise error rate. The Hochberg method controls the familywise error rate under independence. The Holm method controls the familywise error rate without assuming independence.

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Armitage, P. (1955), “Tests for Linear Trend in Proportions and Frequencies,” *Biometrics*, 11, 375–386.

Benjamini, Y. and Hochberg, Y. (1995), “Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing,” *Journal of the Royal Statistical Society. B*, 57, 289–300.

Bickis, M. and Krewski, D. (1986), “Statistical Issues in the Analysis of the Long Term Carcinogenicity Bioassay in Small Rodents: an Empirical Evaluation of Statistical Decision Rules,” *Environmental Health Directorate*, Health Protection Branch, Health and Welfare Canada, Ottawa, Ontario, Canada.

Brown, C.C. and Fears, T.R. (1981), “Exact Significance Levels for Multiple Binomial Testing with Application to Carcinogenicity Screens,” *Biometrics*, 37, 763–774.

Cochran, W. (1954), “Some Methods for Strengthening the Common χ^2 Tests,” *Biometrics*, 10, 417–451.

Dinse, G.E. (1985), “Testing for Trend in Tumor Prevalence Rates: I. Nonlethal Tumors,” *Biometrics*, 41, 751–770.

Freedman, D.A. (1981), “Bootstrapping Regression Models,” *Annals of Statistics,* 9, 1218–1228.

Freeman, M.F. and Tukey, J.W. (1950), “Transformations Related to the Angular and the Square Root,” *Annals of Mathematical Statistics,* 21, 607–611.

Good, I.J. (1987), “A Survey of the Use of the Fast Fourier Transform for Computing Distributions,” *Journal of Statistical Computation and Simulation,* 28, 87–93.

Heyse, J. and Rom, D. (1988), “Adjusting for Multiplicity of Statistical Tests in the Analysis of Carcinogenicity Studies,” *Biometrical Journal,* 30, 883–896.

Hochberg, Y. (1988), “A Sharper Bonferroni Procedure for Multiple Significance Testing,” *Biometrika,* 75, 800–803.

Hochberg, Y. and Tamhane, A.C. (1987), *Multiple Comparison Procedures,* New York: John Wiley & Sons, Inc.

Hoel, D.G. and Walburg, H.E. (1972), “Statistical Analysis of Survival Experiments,” *Journal of the National Cancer Institute,* 49, 361–372.

Holland, B.S. and Copenhaver, M.D. (1987), “An Improved Sequentially Rejective Bonferroni Test Procedure,” *Biometrics,* 43, 417–424.

Holm, S. (1979), “A Simple Sequentially Rejective Bonferroni Test Procedure,” *Scandinavian Journal of Statistics,* 6, 65–70.

Hommel, G. (1988), “A Comparison of Two Modified Bonferroni Procedures,” *Biometrika,* 75, 383–386.

Lagakos, S.W. and Louis, T.A. (1985), “The Statistical Analysis of Rodent Tumorigenicity Experiments,” in *Toxicological Risk Assessment,* eds. Clayson, D.B., Krewski, D., and Munro, I., Boca Raton, FL: CRC Press, 1, 144–163.

Mantel, N.(1980), “Assessing Laboratory Evidence for Neoplastic Activity,” *Biometrics,* 36, 381–399.

Mantel, N. and Haenszel, W. (1959), “Statistical Aspects of Analysis of Data from Retrospective Studies of Disease,” *Journal of the National Cancer Institute,* 22, 719–748.

Miller, J.J. (1978), “The Inverse of the Freeman-Tukey Double Arcsine Transformation,” *The American Statistician,* 32, 138.

Pagano, M. and Tritchler, D. (1983), “On Obtaining Permutation Distributions in Polynomial Time,” *Journal of the American Statistical Association,* 78, 435–440.

Peto, R., Pike, M.C., and Day, N.E. (1980), “Guidelines for Simple, Sensitive Significance Tests for Carcinogenic Effects in Long-Term Animal Experiments,” *Long-term and Short-term Screening Assays for Carcinogens: A Critical Appraisal,* IARC Monographs, Annex to Supplement 2, 311–426. Lyon: International Agency for Research on Cancer.

Sarkar, S., and Chang, C.K. (1997), “Simes Method for Multiple Hypothesis Testing with Positively Dependent Test Statistics,” *Journal of the American Statistical Association,* 92, 1601–1608.

Shaffer, J.P. (1986), “Modified Sequentially Rejective Multiple Test Procedures,” *Journal of the American Statistical Association,* 81, 826–831.

Sidak, Z. (1967), “Rectangular Confidence Regions for the Means of Multivariate Normal Distributions,” *Journal of the American Statistical Association,* 62, 626–633.

Simes, R.J. (1986), “An Improved Bonferroni Procedure for Multiple Tests of Significance,” *Biometrika*, 73, 751–754.

Soper, K.A. and Tonkonoh, N. (1993), “The Discrete Distribution Used for the Log-Rank Test Can Be Inaccurate,” *Biometrical Journal*, 35, 291–298.

Westfall, P.H. and Lin, Y. (1988), “Estimating Optimal Continuity Corrections in Run Time,” *Proceedings of the Statistical Computing Section*, New Orleans: American Statistical Association, 297–298.

Westfall, P.H. and Soper, K.A.(1994), “Nonstandard Uses of PROC MULTTEST: Permutational Peto Tests; Permutational and Unconditional t and Binomial Tests,” Proceedings of the Nineteenth Annual SAS User’s Group International Conference, 986–989.

Westfall, P.H., Tobias, R.D., Rom, D., Wolfinger, R.D., and Hochberg, Y. (1999). *Multiple Comparisons and Multiple Tests using the SAS System*, SAS Institute Inc., Cary, NC.

Westfall, P.H. and Wolfinger, R.D. (1997), “Multiple Tests with Discrete Distributions,” *The American Statistician,* 51, 3–8.

Westfall, P.H. and Wolfinger, R.D. (2000), “Closed Multiple Testing Procedures and PROC MULTTEST,” *Observations,* SAS Institute Inc., to appear.

Westfall, P.H. and Young, S.S. (1989), “P-value Adjustments for Multiple Tests in Multivariate Binomial Models,” *Journal of the American Statistical Association*, 84, 780–786.

Westfall, P.H. and Young, S.S. (1993), *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*, New York: John Wiley & Sons, Inc.

Yates, F. (1984), “Tests of Significance for 2×2 Contingency Tables,” *Journal of the Royal Statistical Society,* Series A, 147, 426–463.

Chapter 49
The NESTED Procedure

Chapter Contents

Chapter 49

The NESTED Procedure

Overview

The NESTED procedure performs random effects analysis of variance for data from an experiment with a nested (hierarchical) structure.* A random effects model for data from a completely nested design with two factors has the following form:

$$y_{ijr} = \mu + \alpha_i + \beta_{ij} + \epsilon_{ijr}$$

where

y_{ijr} is the value of the dependent variable observed at the rth replication with the first factor at its ith level and the second factor at its jth level.

μ is the overall (fixed) mean of the sampling population.

$\alpha_i, \beta_{ij}, \epsilon_{ijr}$ are mutually uncorrelated random effects with zero means and respective variances σ^2_α, σ^2_β, and σ^2_ϵ (the variance components).

This model is appropriate for an experiment with a multi-stage nested sampling design. An example of this is given in Example 49.1 on page 2994, where four turnip plants are randomly chosen (the first factor), then three leaves are randomly chosen from each plant (the second factor nested within the first), and then two samples are taken from each leaf (the different replications at fixed levels of the two factors).

Note that PROC NESTED is appropriate for models with only classification effects; it does not handle models that contain continuous covariates. For random effects models with covariates, use either the GLM or MIXED procedure.

Contrasted with Other SAS Procedures

The NESTED procedure performs a computationally efficient analysis of variance for data with a nested design, estimating the different components of variance and also testing for their significance if the design is balanced (see the "Unbalanced Data" section on page 2990). Although other procedures (such as GLM and MIXED) provide similar analyses, PROC NESTED is both easier to use and computationally more efficient for this special type of design. This is especially true when the design involves a large number of factors, levels, or observations.

For example, to specify a four-factor completely nested design in the GLM procedure, you use the form

*PROC NESTED is modeled after the General Purpose Nested Analysis of Variance program of the Dairy Cattle Research Branch of the United States Department of Agriculture. That program was originally written by M.R. Swanson, Statistical Reporting Service, United States Department of Agriculture.

```
class a b c d;
model y=a b(a) c(a b) d(a b c);
```

However, to specify the same design in PROC NESTED, you simply use the form

```
class a b c d;
var y;
```

In addition, other procedures require TEST statements to perform appropriate tests, whereas the NESTED procedure produces the appropriate tests automatically. However, PROC NESTED makes one assumption about the input data that the other procedures do not: **PROC NESTED assumes that the input data set is sorted by the classification (CLASS) variables defining the effects.** If you use PROC NESTED on data that is not sorted by the CLASS variables, then the results may not be valid.

Getting Started

Reliability of Automobile Models

A study is performed to compare the reliability of several models of automobiles. Three different automobile models (Model) from each of four domestic automobile manufacturers (Make) are tested. Three different cars of each make and model are subjected to a reliability test and given a score between 1 and 100 (Score), where higher scores indicate greater reliability.

The following statements create the SAS data set auto.

```
title 'Reliability of Automobile Models';
data auto;
   input Make $ Model Score @@;
   datalines;
a 1 62  a 2 77  a 3 59
a 1 67  a 2 73  a 3 64
a 1 60  a 2 79  a 3 60
b 1 72  b 2 58  b 3 80
b 1 75  b 2 63  b 3 84
b 1 69  b 2 57  b 3 89
c 1 94  c 2 76  c 3 81
c 1 90  c 2 75  c 3 85
c 1 88  c 2 78  c 3 85
d 1 69  d 2 73  d 3 90
d 1 72  d 2 88  d 3 87
d 1 76  d 2 87  d 3 92
;
```

The Make variable contains the make of the automobile, represented here by ‘a’, ‘b’, ‘c’, or ‘d’, while the Model variable represents the automobile model with a ‘1’, ‘2’, or ‘3’. The Score variable contains the reliability scores given to the three sampled cars from each Make-Model group. Since the automobile models are nested within their makes, the NESTED procedure is used to analyze this data. The NESTED procedure requires the data to be sorted by Make and, within Make, by Model, so the following statements execute a PROC SORT before completing the analysis.

```
proc sort;
   by Make Model;
proc nested;
   class Make Model;
   var Score;
run;
```

The Model variable appears after the Make variable in the CLASS statement because it is nested within Make. The VAR statement specifies the response variable. The output is displayed in Figure 49.1.

```
                          Reliability of Automobile Models

                                The NESTED Procedure

                        Coefficients of Expected Mean Squares

                        Source       Make       Model      Error

                        Make            9           3          1
                        Model           0           3          1
                        Error           0           0          1


            Nested Random Effects Analysis of Variance for Variable Score

Variance                Sum of                   Error                        Variance   Percent
Source         DF      Squares  F Value  Pr > F  Term    Mean Square         Component  of Total

Total          35  4177.888889                             119.368254       131.876543  100.0000
Make            3  1709.000000     2.15  0.1719  Model     569.666667        33.867284   25.6811
Model           8  2118.888889    18.16  <.0001  Error     264.861111        83.425926   63.2606
Error          24   350.000000                              14.583333        14.583333   11.0583


                        Score Mean                         75.94444444
                        Standard Error of Score Mean        3.97794848
```

Figure 49.1. Output from PROC NESTED

Figure 49.1 first displays the coefficients of the variance components that make up each of the expected mean squares, then the ANOVA table is displayed. The results do not indicate significant variation between the different automobile makes ($F = 2.15, p = 0.1719$). However, they do suggest that there is significant variation between the different models within the makes ($F = 18.16, p < 0.0001$). This is evident in the fact that the make of car accounts for only 25.7% of the total variation in the data, while the car model accounts for 63.3% (as shown in the Percent of Total column). The estimated variance components are shown in the Variance Component column.

Syntax

The following statements are available in PROC NESTED.

```
PROC NESTED < options > ;
    CLASS variables < / option > ;
    VAR variables ;
    BY variables ;
```

The PROC NESTED and CLASS statements are required. The BY, CLASS, and VAR statements are described after the PROC NESTED statement.

PROC NESTED Statement

PROC NESTED < *options* > **;**

The PROC NESTED statement has the following options:

AOV

displays only the analysis of variance statistics when there is more than one dependent variable. The “analysis of covariation” statistics are suppressed (see the “Analysis of Covariation” section on page 2990).

DATA=*SAS-data-set*

names the SAS data set to be used by PROC NESTED. By default, the procedure uses the most recently created SAS data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NESTED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables.

Note: When you use the NESTED procedure, your data must be sorted first by the BY variables and, within the BY variables, by the CLASS variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NESTED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < */ option* > **;**

You must include a CLASS statement with PROC NESTED specifying the classification variables for the analysis.

Values of a variable in the CLASS statement denote the levels of an effect. The name of that variable is also the name of the corresponding effect. The second effect is assumed to be nested within the first effect, the third effect is assumed to be nested within the second effect, and so on.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 8 characters of the formatted values, except for numeric variables with no explicit format, for which class levels were determined from the raw numeric values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide*, and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

Note: The data set must be sorted by the classification variables in the order that they are given in the CLASS statement. Use PROC SORT to sort the data if they are not already sorted.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

VAR Statement

VAR *variables* **;**

The VAR statement lists the dependent variables for the analysis. The dependent variables must be numeric variables. If you do not specify a VAR statement, PROC NESTED performs an analysis of variance for all numeric variables in the data set, except those already specified in the CLASS statement.

Details

Missing Values

An observation with missing values for any of the variables used by PROC NESTED is omitted from the analysis. Blank values of CLASS character variables are treated as missing values.

Unbalanced Data

A completely nested design is defined to be unbalanced if the groups corresponding to the levels of some classification variable are not all of the same size. The NESTED procedure can compute unbiased estimates for the variance components in an unbalanced design, but because the sums of squares on which these estimates are based no longer have χ^2 distributions under a Gaussian model for the data, F tests for the significance of the variance components cannot be computed. PROC NESTED checks to see that the design is balanced. If it is not, a warning to that effect is placed on the log, and the columns corresponding to the F tests in the analysis of variance are left blank.

General Random Effects Model

A random effects model for data from a completely nested design with n factors has the general form

$$y_{i_1 i_2 \cdots i_n r} = \mu + \alpha_{i_1} + \beta_{i_1 i_2} + \cdots + \epsilon_{i_1 i_2 \cdots i_n r}$$

where

$y_{i_1 i_2 \cdots i_n r}$ is the value of the dependent variable observed at the rth replication with factor j at level i_j, for $j = 1, \ldots, n$.

μ is the overall (fixed) mean of the sampled population.

$\alpha_{i_1}, \beta_{i_1 i_2}, \ldots, \epsilon_{i_1 i_2 \cdots i_n r}$ are mutually uncorrelated random effects with zero means and respective variances σ_α^2, σ_β^2, …, σ_ϵ^2.

Analysis of Covariation

When you specify more than one dependent variable, the NESTED procedure produces a descriptive analysis of the covariance between each pair of dependent variables in addition to a separate analysis of variance for each variable. The analysis of covariation is computed under the basic random effects model for each pair of dependent variables:

$$\begin{aligned} y_{i_1 i_2 \cdots i_n r} &= \mu + \alpha_{i_1} + \beta_{i_1 i_2} + \cdots + \epsilon_{i_1 i_2 \cdots i_n r} \\ y'_{i_1 i_2 \cdots i_n r} &= \mu' + \alpha'_{i_1} + \beta'_{i_1 i_2} + \cdots + \epsilon'_{i_1 i_2 \cdots i_n r} \end{aligned}$$

where the notation is the same as that used in the preceding general random effects model.

There is an additional assumption that all the random effects in the two models are mutually uncorrelated except for corresponding effects, for which

$$
\begin{aligned}
\mathrm{Corr}(\alpha_{i_1}, \alpha'_{i_1}) &= \rho_\alpha \\
\mathrm{Corr}(\beta_{i_1 i_2}, \beta'_{i_1 i_2}) &= \rho_\beta \\
&\vdots \\
\mathrm{Corr}(\epsilon_{i_1 i_2 \cdots i_n r}, \epsilon'_{i_1 i_2 \cdots i_n r}) &= \rho_\epsilon
\end{aligned}
$$

Error Terms in F Tests

Random effects ANOVAs are distinguished from fixed effects ANOVAs by which error mean squares are used as the denominator for F tests. Under a fixed effects model, there is only one true error term in the model, and the corresponding mean square is used as the denominator for all tests. This is how the usual analysis is computed in PROC ANOVA, for example. However, in a random effects model for a nested experiment, mean squares are compared sequentially. The correct denominator in the test for the first factor is the mean square due to the second factor; the correct denominator in the test for the second factor is the mean square due to the third factor; and so on. Only the mean square due to the last factor, the one at the bottom of the nesting order, should be compared to the error mean square.

Computational Method

The building blocks of the analysis are the sums of squares for the dependent variables for each classification variable within the factors that precede it in the model, corrected for the factors that follow it. For example, for a two-factor nested design, PROC NESTED computes the following sums of squares:

$$
\begin{array}{rl}
\text{Total SS} & \displaystyle\sum_{ijr} (y_{ijr} - y_{\cdots})^2 \\[2ex]
\text{SS for Factor 1} & \displaystyle\sum_{i} n_{i\cdot} \left(\frac{y_{i\cdot\cdot}}{n_{i\cdot}} - \frac{y_{\cdots}}{n_{\cdot\cdot}} \right)^2 \\[2ex]
\text{SS for Factor 2 within Factor 1} & \displaystyle\sum_{ij} n_{ij} \left(\frac{y_{ij\cdot}}{n_{ij}} - \frac{y_{i\cdot\cdot}}{n_{i\cdot}} \right)^2 \\[2ex]
\text{Error SS} & \displaystyle\sum_{ijr} \left(y_{ijr} - \frac{y_{ij\cdot}}{n_{ij}} \right)^2
\end{array}
$$

where y_{ijr} is the rth replication, n_{ij} is the number of replications at level i of the first factor and level j of the second, and a dot as a subscript indicates summation over the corresponding index. If there is more than one dependent variable, PROC NESTED also computes the corresponding sums of crossproducts for each pair. The expected value of the sum of squares for a given classification factor is a linear combination of the variance components corresponding to this factor and to the factors that are nested within it. For each factor, the coefficients of this linear combination are computed. (The efficiency of PROC NESTED is partly due to the fact that these various sums can be accumulated with just one pass through the data, assuming that the data have been sorted by the classification variables.) Finally, estimates of the variance components are derived as the solution to the set of linear equations that arise from equating the mean squares to their expected values.

Displayed Output

PROC NESTED displays the following items for each dependent variable:

- Coefficients of Expected Mean Squares, which are the coefficients of the $n+1$ variance components making up the expected mean square. Denoting the element in the ith row and jth column of this matrix by C_{ij}, the expected value of the mean square due to the ith classification factor is

  $$C_{i1}\sigma_1^2 + \cdots + C_{in}\sigma_n^2 + C_{i,n+1}\sigma_\epsilon^2 \ .$$

  C_{ij} is always zero for $i > j$, and if the design is balanced, C_{ij} is equal to the common size of all classification groups of the jth factor for $i \leq j$. Finally, the mean square for error is always an unbiased estimate of σ_ϵ^2. In other words, $C_{n+1,n+1} = 1$.

For every dependent variable, PROC NESTED displays an analysis of variance table. Each table contains the following:

- each Variance Source in the model (the different components of variance) and the total variance
- degrees of freedom (DF) for the corresponding sum of squares
- Sum of Squares for each classification factor. The sum of squares for a given classification factor is the sum of squares in the dependent variable within the factors that precede it in the model, corrected for the factors that follow it. (See the "Computational Method" section on page 2991.)
- F Value for a factor, which is the ratio of its mean square to the appropriate error mean square. The next column, labeled PR > F, gives the significance levels that result from testing the hypothesis that each variance component equals zero.
- the appropriate Error Term for an F test, which is the mean square due to the next classification factor in the nesting order. (See the "Error Terms in F Tests" section on page 2991.)

- Mean Square due to a factor, which is the corresponding sum of squares divided by the degrees of freedom
- estimates of the Variance Components. These are computed by equating the mean squares to their expected values and solving for the variance terms. (See the "Computational Method" section on page 2991.)
- Percent of Total, the proportion of variance due to each source. For the ith factor, the value is

$$100 \times \frac{\text{source variance component}}{\text{total variance component}}$$

- Mean, the overall average of the dependent variable. This gives an unbiased estimate of the mean of the population. Its variance is estimated by a certain linear combination of the estimated variance components, which is identical to the mean square due to the first factor in the model divided by the total number of observations when the design is balanced.

If there is more than one dependent variable, then the NESTED procedure displays an "analysis of covariation" table for each pair of dependent variables (unless the AOV option is specified in the PROC NESTED statement). See the "Analysis of Covariation" section on page 2990 for details. For each source of variation, this table includes the following:

- Degrees of Freedom
- Sum of Products
- Mean Products
- Covariance Component, the estimate of the covariance component

Items in the analysis of covariation table are computed analogously to their counterparts in the analysis of variance table. The analysis of covariation table also includes the following:

- Variance Component Correlation for a given factor. This is an estimate of the correlation between corresponding effects due to this factor. This correlation is the ratio of the covariance component for this factor to the square root of the product of the variance components for the factor for the two different dependent variables. (See the "Analysis of Covariation" section on page 2990.)
- Mean Square Correlation for a given classification factor. This is the ratio of the Mean Products for this factor to the square root of the product of the Mean Squares for the factor for the two different dependent variables.

ODS Table Names

PROC NESTED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 49.1. ODS Tables Produced in PROC NESTED

ODS Table Name	Description	Statement
ANCOVA	Analysis of covariance	default with more than one dependent variable
ANOVA	Analysis of variance	default
EMSCoef	Coefficients of expected mean squares	default
Statistics	Overall statistics for fit	default

Example

Example 49.1. Variability of Calcium Concentration in Turnip Greens

In the following example from Snedecor and Cochran (1976), an experiment is conducted to study the variability of calcium concentration in turnip greens. Four plants are selected at random; then three leaves are randomly selected from each plant. Two 100-mg samples are taken from each leaf. The amount of calcium is determined by microchemical methods.

Because the data are read in sorted order, it is not necessary to use PROC SORT on the CLASS variables. Leaf is nested in Plant; Sample is nested in Leaf and is left for the residual term. All the effects are random effects. The following statements read the data and invoke PROC NESTED. These statements produce Output 49.1.1:

```
title 'Calcium Concentration in Turnip Leaves'
      '--Nested Random Model';
title2 'Snedecor and Cochran, ''Statistical Methods'''
      ', 1976, p. 286';
data Turnip;
   do Plant=1 to 4;
      do Leaf=1 to 3;
         do Sample=1 to 2;
            input Calcium @@;
            output;
            end;
         end;
      end;
   datalines;
3.28 3.09 3.52 3.48 2.88 2.80 2.46 2.44
1.87 1.92 2.19 2.19 2.77 2.66 3.74 3.44
2.55 2.55 3.78 3.87 4.07 4.12 3.31 3.31
;
```

```
proc nested;
   class Plant Leaf;
   var Calcium;
run;
```

Output 49.1.1. Analysis of Calcium Concentration in Turnip Greens Using PROC NESTED

```
                 Calcium Concentration in Turnip Leaves--Nested Random Model
                  Snedecor and Cochran, ’Statistical Methods’, 1976, p. 286

                                    The NESTED Procedure

                           Coefficients of Expected Mean Squares

                           Source      Plant      Leaf      Error

                           Plant           6         2          1
                           Leaf            0         2          1
                           Error           0         0          1
```

```
                 Calcium Concentration in Turnip Leaves--Nested Random Model
                  Snedecor and Cochran, ’Statistical Methods’, 1976, p. 286

                                    The NESTED Procedure

              Nested Random Effects Analysis of Variance for Variable Calcium

Variance                Sum of                   Error                    Variance   Percent
Source        DF       Squares  F Value  Pr > F  Term    Mean Square     Component  of Total

Total         23     10.270396                             0.446539      0.532938  100.0000
Plant          3      7.560346     7.67  0.0097  Leaf      2.520115      0.365223   68.5302
Leaf           8      2.630200    49.41  <.0001  Error     0.328775      0.161060   30.2212
Error         12      0.079850                             0.006654      0.006654    1.2486


                      Calcium Mean                          3.01208333
                      Standard Error of Calcium Mean        0.32404445
```

The results indicate that there is significant (nonzero) variation from plant to plant (Pr > F is 0.0097) and from leaf to leaf within a plant (Pr > F is less than 0.0001). Notice that the variance component for Plant uses the Leaf mean square as an error term in the model rather than the error mean square.

References

Snedecor, G. W. and Cochran, W. G. (1976), *Statistical Methods*, Sixth Edition, Ames, IA: The Iowa State University Press.

Steel, R. G. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill Book Company.

Chapter 50
The NLIN Procedure

Chapter Contents

Chapter 50
The NLIN Procedure

Overview

The NLIN procedure produces least squares or weighted least squares estimates of the parameters of a nonlinear model. Nonlinear models are more difficult to specify and estimate than linear models. Instead of simply listing regressor variables, you must write the regression expression, declare parameter names, and supply initial parameter values. Some models are difficult to fit, and there is no guarantee that the procedure can fit the model successfully.

For each nonlinear model to be analyzed, you must specify the model (using a single dependent variable) and the names and starting values of the parameters to be estimated.

Using PROC NLIN, you can also

- confine the estimation procedure to a certain range of values of the parameters by imposing bounds on the estimates
- produce new SAS data sets containing predicted values, residuals, parameter estimates and SSE at each iteration, the covariance matrix of parameter estimates, and other statistics
- define your own objective function to be minimized

Estimation of a nonlinear model is an iterative process. To begin this process the NLIN procedure first examines the starting value specifications of the parameters. If a grid of values is specified, PROC NLIN evaluates the residual sum of squares at each combination of parameter values to determine the set of parameter values producing the lowest residual sum of squares. These parameter values are used for the initial step of the iteration.

Then PROC NLIN uses one of these five iterative methods:

- steepest-descent or gradient method
- Newton method
- modified Gauss-Newton method
- Marquardt method

These methods use derivatives or approximations to derivatives of the SSE with respect to the parameters to guide the search for the parameters producing the smallest SSE.

You can use the NLIN procedure for segmented models (see Example 50.1) or robust regression (see Example 50.2). You can also use it to compute maximum-likelihood estimates for certain models (refer to Jennrich and Moore 1975; Charnes, Frome, and Yu 1976).

Getting Started

The NLIN procedure performs univariate nonlinear regression using the least squares method. Nonlinear regression analysis is indicated when you have information specifying that the functional relationship between the predictor and response variables is nonlinear in the parameters. Such information might come from direct knowledge of the true model, theoretical developments, or previous studies. *Nonlinear*, in this sense, means that the mathematical relationship between the variables and parameters is not required to have a linear form. For example, consider the following two models:

$$Y = aX^2 + b$$

$$Y = \frac{1}{a}X + b$$

where a and b are parameters and X and Y are random variables. The first model is linear in the parameters; the second model is nonlinear.

Estimating the Nonlinear Model

As an example of a nonlinear regression analysis, consider the following theoretical model of enzyme kinetics. The model relates the initial velocity of an enzymatic reaction to the substrate concentration.

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \frac{\theta_1 x_i}{\theta_2 + x_i}, \text{ for } i = 1, 2, \ldots, n$$

where x_i represents the amount of substrate for n trials and $f(\boldsymbol{x}, \boldsymbol{\theta})$ is the velocity of the reaction. The vector $\boldsymbol{\theta}$ contains the rate parameters.

Suppose that you want to study the relationship between concentration and velocity for a particular enzyme/substrate pair. You record the reaction rate (velocity) observed at different substrate concentrations. Your data set is as follows:

```
data Enzyme;
   input Concentration Velocity @@;
   datalines;
0.26 124.7   0.30 126.9   0.48 135.9   0.50 137.6
0.54 139.6   0.68 141.1   0.82 142.8   1.14 147.6
1.28 149.8   1.38 149.4   1.80 153.9   2.30 152.5
2.44 154.5   2.48 154.7
;
```

The SAS data set Enzyme contains the two variables Concentration (substrate concentration) and Velocity (reaction rate). The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all of the values are read.

The following statements request a nonlinear regression analysis:

```
proc nlin data=Enzyme method=marquardt hougaard;
   parms theta1=155
         theta2=0 to 0.07 by 0.01;
   model Velocity = theta1*Concentration / (theta2 + Concentration);
run;
```

The DATA= option specifies that the SAS data set Enzyme be used in the analysis. The METHOD= option directs PROC NLIN to use the MARQUARDT iterative method. The HOUGAARD option requests that a skewness measure be calculated for the parameters.

The MODEL statement specifies the enzymatic reaction model

$$V = \frac{\theta_1 C}{\theta_2 + C}$$

where V represents the velocity or reaction rate and C represents the substrate concentration.

The PARMS statement declares the parameters and specifies their initial values. In this example, the initial estimates in the PARMS statement are obtained as follows. Since the model is a monotonic increasing function in C, and

$$\lim_{C \to \infty} \left(\frac{\theta_1 C}{\theta_2 + C} \right) = \theta_1$$

take the largest observed value of the variable Velocity (154.7) as the initial value for the parameter Theta1. Thus, the PARMS statement specifies 155 as the initial value for Theta1, which is approximately equal to the maximum observed velocity.

To obtain an initial value for the parameter $theta_2$, first rearrange the model equation to solve for θ_2:

$$\theta_2 = \frac{\theta_1 C}{V} - C$$

By substituting the initial value of Theta1 for θ_1 and taking each pair of observed values of Concentration and Velocity for C and V, respectively, you obtain a set of possible starting values for Theta2 ranging from about 0.01 to 0.07.

You can choose any value within this range as a starting value for Theta2, or you can direct PROC NLIN to perform a preliminary search for the best initial Theta2 value within that range of values. The PARMS statement specifies a range of values for Theta2, which results in a search over the grid points from 0 to 0.07 in increments of 0.01. The output from this PROC NLIN invocation are displayed in the following figures.

PROC NLIN evaluates the model at each point on the specified grid for the Theta2 parameter. Figure 50.1 displays the calculations resulting from the grid search.

```
                         The NLIN Procedure
                    Dependent Variable Velocity

                            Grid Search
                                             Sum of
                    theta1       theta2      Squares

                     155.0            0       3075.4
                     155.0       0.0100       2074.1
                     155.0       0.0200       1310.3
                     155.0       0.0300        752.0
                     155.0       0.0400        371.9
                     155.0       0.0500        147.2
                     155.0       0.0600      58.1130
                     155.0       0.0700      87.9662



                         The NLIN Procedure
                    Dependent Variable Velocity
                         Method: Marquardt

                          Iterative Phase
                                                         Sum of
              Iter       theta1       theta2            Squares

                 0        155.0       0.0600            58.1130
                 1        158.0       0.0736            19.7017
                 2        158.1       0.0741            19.6606
                 3        158.1       0.0741            19.6606


         NOTE: Convergence criterion met.
```

Figure 50.1. Nonlinear Least Squares Grid Search from the NLIN Procedure

The parameter Theta1 is held constant at its specified initial value of 155, the grid is traversed, and the residual sums of squares are computed at each point. The "best" starting value is the point that corresponds to the smallest value of the residual sum of squares. Figure 50.1 shows that the best starting value for Theta2 is 0.06. PROC NLIN uses this point as the initial value for Theta2 in the following iterative phase.

PROC NLIN determines convergence using the relative offset measure of Bates and Watts (1981). When this measure is less than 10^{-5}, convergence is declared. Figure 50.1 displays the iteration history.

```
                         The NLIN Procedure

                        Estimation Summary

               Method                    Marquardt
               Iterations                        3
               R                          5.861E-6
               PPC(theta2)                8.569E-7
               RPC(theta2)                0.000078
               Object                     2.902E-7
               Objective                  19.66059
               Observations Read                14
               Observations Used                14
               Observations Missing              0
```

Figure 50.2. Estimation Summary from the NLIN Procedure

Figure 50.2 displays a summary of the estimation including several convergence measures R, PPC, RPC, and Object.

The R measure is the relative offset convergence measure of Bates and Watts. A PPC value of 8.569E-7 indicates that the parameter Theta2 (which has the largest PPC value of all the parameters) would change by that relative amount were PROC NLIN to take an additional iteration step. The RPC value indicates that Theta2 changed by 0.000078, relative to its value in the last iteration. These changes are measured before steplength adjustments are made. The Object measure indicates that the objective function value changed 2.902E-7 in relative value from the last iteration.

```
                         The NLIN Procedure

           NOTE: An intercept was not specified for this model.

                                     Sum of        Mean               Approx
   Source                    DF     Squares      Square    F Value    Pr > F

   Model                      2      290116      145058    88537.2    <.0001
   Error                     12     19.6606      1.6384
   Uncorrected Total         14      290135
```

Figure 50.3. Nonlinear Least Squares Summary from the NLIN Procedure

Figure 50.3 displays the least squares summary statistics for the model. The degrees of freedom, sums of squares, and mean squares are listed.

```
                              The NLIN Procedure

                                     Approx        Approximate 95%
     Parameter       Estimate     Std Error      Confidence Limits        Skewness

     theta1             158.1        0.6737        156.6       159.6        0.0152
     theta2            0.0741       0.00313       0.0673      0.0809        0.0362
```

Figure 50.4. Parameter Estimates from the NLIN Procedure

Figure 50.4 displays the estimates for each parameter, the associated asymptotic standard error, and the upper and lower values for the asymptotic 95% confidence interval. PROC NLIN also displays the asymptotic correlations between the estimated parameters (not shown).

The skewness measures of 0.0152 and 0.0362 indicate that the parameters are nearly linear and that their standard errors and confidence intervals can be safely used for inferences.

Thus, the estimated nonlinear model relating reaction velocity and substrate concentration can be written as

$$\hat{\mathrm{V}} = \frac{158.105C}{0.0741 + C}$$

where V represents the velocity or rate of the reaction, and C represents the substrate concentration.

Syntax

PROC NLIN < *options* > **;**
MODEL *dependent=expression* **;**
PARAMETERS *parameter=values* <,. . ., *parameter=values*>**;**

other program statements

BOUNDS *inequality* < , . . . , *inequality* > **;**
BY *variables* **;**
DER.*parameter=expression* **;**
DER.*parameter***.***parameter=expression* **;**
ID *variables* **;**
OUTPUT OUT=*SAS-data-set keyword=names* <,. . .,*keyword=names*>**;**
CONTROL *variable* <*=values*> < . . . *variable* <*=values*>> **;**

A vertical bar (|) denotes a choice between two specifications. The *other program statements* are valid SAS expressions that can appear in the DATA step. PROC NLIN enables you to create new variables within the procedure and use them in the nonlinear analysis. The NLIN procedure automatically creates several variables that are also available for use in the analysis. See the section "Special Variables" beginning on page 3020 for more information. The PROC NLIN, PARMS, and MODEL statements are required.

The statements used in PROC NLIN, in addition to the PROC statement, are as follows:

BOUNDS	constrains the parameter estimates within specified bounds
BY	specifies variables to define subgroups for the analysis
DER	specifies the first or second partial derivatives
ID	specifies additional variables to add to the output data set
MODEL	defines the relationship between the dependent and independent variables
OUTPUT	creates an output data set containing statistics for each observation
PARMS	identifies parameters to be estimated and the starting values for each parameter
other program statements	includes assignment statements, ARRAY statements, DO loops, and program control statements

PROC NLIN Statement

PROC NLIN < *options* > **;**

The PROC NLIN statement invokes the procedure. The following table lists the options available with the PROC NLIN statement. Explanations follow in alphabetical order.

Task	Options
Specify data sets	DATA= OUTEST= SAVE
Grid search	BEST=
Choose an iteration method	METHOD=
Control step size	MAXSUBIT= NOHALVE RHO= SMETHOD= TAU=
Specify details of iteration	G4 UNCORRECTEDDF SIGSQ=
Minimization Tuning	CONVERGE= CONVERGEOBJ= CONVERGEPARM= SINGULAR= MAXITER=
Modify Amount of Output	HOUGAARD NOITPRINT NOPRINT
List Model Structure	LIST LISTALL LISTCODE LISTDEP LISTDER XREF
Trace Model Execution	FLOW PRINT TRACE

BEST=*n*

requests that PROC NLIN display the residual sums of squares only for the best n combinations of possible starting values from the grid. If you do not specify the BEST= option, PROC NLIN displays the residual sum of squares for every combination of possible parameter starting values.

CONVERGE=*c*

specifies the convergence criteria for PROC NLIN. For all iterative methods the relative offset convergence measure of Bates and Watts is used by default to determine convergence. This measure is labeled "R" in the Estimation Summary table. The iterations are said to have converged for CONVERGE=c if

$$\sqrt{\frac{r'X(X'X)^{-1}X'r}{\text{LOSS}^i}} < c$$

where r is the residual vector and X is the Jacobian matrix. The default LOSS function is the sum of squared errors (SSE). By default, CONVERGE=10^{-5}. The R convergence measure cannot be computed accurately in the special case of a perfect fit (residuals close to zero). When the SSE is less than the value of the SINGULAR= option, convergence is assumed.

CONVERGEOBJ=*c*

uses the change in the LOSS function as the convergence criterion. For more details on the LOSS function, see the section "Special Variable Used to Determine Convergence Criteria" on page 3021. The iterations are said to have converged for CONVERGEOBJ=c if

$$\frac{\text{LOSS}^{i-1} - \text{LOSS}^i}{\text{LOSS}^i + 10^{-6}} < c$$

where LOSS^i is the LOSS for the ith iteration. The default LOSS function is the sum of squared errors (SSE). The constant c should be a small positive number. See the "Computational Methods" section beginning on page 3024 for more details. If specified, the CONVERGEOBJ= option overrides the default CONVERGE= convergence criterion so that NLIN performs as it did in version 6 releases of the procedure.

CONVERGEPARM=*c*

uses the maximum change among parameter estimates as the convergence criterion. The iterations are said to have converged for CONVERGEPARM=c if

$$\max_j \left(\frac{|\beta_j^{i-1} - \beta_j^i|}{|\beta_j^{i-1}|} \right) < c$$

where β_j^i is the value of the jth parameter at the ith iteration.

The default convergence criterion is CONVERGE. If you specify CONVERGEPARM=c, the maximum change in parameters is used as the convergence criterion. If you specify both the CONVERGEOBJ= and CONVERGEPARM= options, PROC NLIN continues to iterate until the decrease in LOSS is sufficiently small (as determined by the CONVERGEOBJ= option) and the maximum change among the parameters is sufficiently small (as determined by the CONVERGEPARM= option).

DATA=*SAS-data-set*

specifies the SAS data set containing the data to be analyzed by PROC NLIN. If you omit the DATA= option, the most recently created SAS data set is used.

FLOW

displays a message for each statement in the model program as it is executed. This debugging option is rarely needed, and it produces large amounts of output.

G4

uses a Moore-Penrose (g_4) inverse in parameter estimation. Refer to Kennedy and Gentle (1980) for details.

HOUGAARD

adds Hougaard's measure of skewness to the parameter estimation table. Computation of the measure requires derivatives (see the section "Hougaard's Measure of Skewness" on page 3019).

LIST

displays the model program and variable lists, including the statements added by macros. Note that the expressions displayed by the LIST option do not necessarily represent the way the expression is actually calculated, since intermediate results for common subexpressions can be reused but are shown in expanded form by the LIST option. To see how the expression is actually evaluated, see the description for the LISTCODE option, which follows.

LISTALL

selects the LIST, LISTDEP, LISTDER, and LISTCODE options.

LISTCODE

displays the derivative tables and compiled model program code. The LISTCODE option is a debugging feature and is not normally needed.

LISTDEP

produces a report that lists, for each variable in the model program, the variables that depend on it and on which it depends.

LISTDER

displays a table of derivatives. The derivatives table lists each nonzero derivative computed for the problem. The derivative listed can be a constant, a variable in the model program, or a special derivative variable created to hold the result of the derivative expression.

MAXITER=*i*

limits the number of iterations PROC NLIN performs before it gives up trying to converge. The i value must be a positive integer. By default, MAXITER=100.

MAXSUBIT=*i*

places a limit on the number of step halvings. By default, MAXSUBIT=30. The value of MAXSUBIT must be a positive integer.

METHOD=GAUSS | MARQUARDT | NEWTON | GRADIENT

specifies the iterative method that PROC NLIN uses. The GAUSS, MARQUARDT and NEWTON methods are more robust than the GRADIENT method. If you omit the METHOD= option, METHOD=GAUSS is used. See the "Computational Methods" section beginning on page 3024 for more information.

NOITPRINT

suppresses the display of the results of each iteration.

NOHALVE

removes the restriction that the objective value must decrease at every iteration. Step halving is still used to satisfy BOUNDS and to ensure that the number of observations that can be evaluated does not decrease. NOHALVE is useful for iteratively reweighted least squares problems.

NOPRINT

suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUTEST=*SAS-data-set*

specifies an output data set to contain the parameter estimates produced at each iteration. See the "Output Data Sets" section on page 3028 for details. If you want to create a permanent SAS data set, you must specify a two-level name. See the chapter "SAS Files," in *SAS Language Reference: Concepts* for more information on permanent SAS data sets.

PRINT

displays the result of each statement in the program as it is executed. This option produces large amounts of output.

RHO=*value*

specifies a value to use in controlling the step-size search. By default, RHO=0.1 except when METHOD=MARQUARDT, in which case RHO=10. See the section "Computational Methods" beginning on page 3024 for more details.

SAVE

specifies that, when the iteration limit is exceeded, the parameter estimates from the final iteration are output to the OUTEST= data set. These parameter estimates are located in the observation with _TYPE_=FINAL. If you omit the SAVE option, the parameter estimates from the final iteration are not output to the data set unless convergence is attained.

SIGSQ=*value*

specifies a value to replace the mean square error for computing the standard errors of the estimates. The SIGSQ= option is used with maximum-likelihood estimation.

SINGULAR=*s*

specifies the singularity criterion, s, which is the absolute magnitude of the smallest pivot value allowed when inverting the Hessian or approximation to the Hessian. The default value is 1E-8.

SMETHOD=HALVE | GOLDEN | CUBIC

specifies the step-size search method that PROC NLIN uses. The default is SMETHOD=HALVE. See the section "Computational Methods" beginning on page 3024 for details.

TAU=*value*

specifies a value to use in controlling the step-size search. By default, TAU=1 except when METHOD=MARQUARDT, in which case TAU=0.01. See the section "Computational Methods" beginning on page 3024 for details.

TRACE

displays the result of each operation in each statement in the model program as it is executed, in addition to the information displayed by the FLOW and PRINT options. This debugging option is needed very rarely, and it produces even more output than the FLOW and PRINT options.

XREF

displays a cross-reference of the variables in the model program showing where each variable is referenced or given a value. The XREF listing does not include derivative variables.

UNCORRECTEDDF

specifies that no degrees of freedom are lost when a bound is active. When the UNCORRECTEDDF option is not specified, an active bound is treated as if a restriction was applied to the set of parameters so one parameter degree of freedom is deducted.

BOUNDS Statement

BOUNDS *inequality* $<, \ldots,$ *inequality* $>$ **;**

The BOUNDS statement restrains the parameter estimates within specified bounds. In each BOUNDS statement, you can specify a series of bounds separated by commas. The series of bounds is applied simultaneously. Each bound contains a list of parameters, an inequality comparison operator, and a value. In a single-bounded expression, these three elements follow one another in the order described. The following are examples of valid single-bounded expressions:

```
bounds a1-a10<=20;
bounds c>30;
bounds a b c > 0;
```

Multiple-bounded expressions are also permitted. For example,

```
bounds 0<=B<=10;
bounds 15<x1<=30;
bounds r <= s <= p < q;
```

If you need to restrict an expression involving several parameters (for example, $A + B < 1$), you can reparameterize the model so that the expression becomes a parameter.

For SAS versions 7.01 and later, lagrange multipliers are reported for all bounds that are enforced (active) when the estimation terminates. In the estimates table the Lagrange multiplier estimates are identified with names *Bound1*, *Bound2* ... . An active bound is treated as if a restriction was applied to the set of parameters so one parameter degree of freedom is deducted. The option UNCORRECTEDDF specifies that no degrees of freedom are lost when a bound is active.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NLIN to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NLIN procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CONTROL Statement

CONTROL *variable* <*=values*> < . . . *variable* <*=values*>> **;**

The CONTROL statement declares control variables and specifies their values. A control variable is like a retained variable (see the section "RETAIN Statement" on page 3016) except that it is retained *across* iterations and the derivative of the model with respect to a control variable is always zero.

DER Statements

DER. *parameter=expression* **;**

DER. *parameter.parameter=expression* **;**

The DER statement specifies first or second partial derivatives. By default, analytical derivatives are automatically computed. However, you can specify the derivatives yourself by using the DER.parm syntax. Use the first form shown to specify first partial derivatives, and use the second form to specify second partial derivatives. Note that the DER.parm syntax is retained for backward compatibility. The automatic analytical derivatives are, in general, a better choice. For additional information on automatic analytical derivatives, see the section "Automatic Derivatives" beginning on page 3017.

For most of the computational methods, you need only specify the first partial derivative for each parameter to be estimated. For the NEWTON method, specify both the first and the second derivatives. If any needed derivatives are not specified, they are automatically computed.

If you use the _LOSS_ variable, you can specify the derivative of _LOSS_ with respect to the parameters using the DER. syntax.For more information, see the "Special Variable Used to Determine Convergence Criteria" section on page 3021.

The expression can be an algebraic representation of the partial derivative of the expression in the MODEL statement with respect to the parameter or parameters that appear in the left-hand side of the DER statement. Numerical derivatives can also be used. The expression in the DER statement must conform to the rules for a valid SAS expression, and it can include any quantities that the MODEL statement expression contains.

ID Statement

ID *variables* **;**

The ID statement specifies additional variables to place in the output data set created by the OUTPUT statement. Any variable on the left-hand side of any assignment statement is eligible. Also, the special variables created by the procedure can be specified. Variables in the input data set do not need to be specified in the ID statement since they are automatically included in the output data set.

MODEL Statement

MODEL *dependent=expression* **;**

The MODEL statement defines the prediction equation by declaring the dependent variable and defining an expression that evaluates predicted values. The expression can be any valid SAS expression yielding a numeric result. The expression can include parameter names, variables in the data set, and variables created by program statements in the NLIN procedure. Any operators or functions that can be used in a DATA step can also be used in the MODEL statement.

A statement such as

```
model y=expression;
```

is translated into the form

```
model.y=expression;
```

using the compound variable name model.y to hold the predicted value. You can use this assignment as an alternative to the MODEL statement. Either a MODEL statement or an assignment to a compound variable such as model.y must appear.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set keyword=names* <, . . . ,*keyword=names*>**;**

The OUTPUT statement specifies an output data set to contain statistics calculated for each observation. For each statistic, specify the keyword, an equal sign, and a variable name for the statistic in the output data set. All of the names appearing in the OUTPUT statement must be valid SAS names, and none of the new variable names can match a variable already existing in the data set to which PROC NLIN is applied.

If an observation includes a missing value for one of the independent variables, both the predicted value and the residual value are missing for that observation. If the iterations fail to converge, all the values of all the variables named in the OUTPUT statement are missing values.

You can specify the following options in the OUTPUT statement. For a description of computational formulas, see Chapter 2, “Introduction to Regression Procedures.”

OUT=*SAS-data-set*

specifies the SAS data set to be created by PROC NLIN when an OUTPUT statement is included. The new data set includes all the variables in the data set to which PROC NLIN is applied. Also included are any ID variables specified in the ID statement, plus new variables with names that are specified in the OUTPUT statement.

The following values can be calculated and output to the new data set.

H=*name*

specifies a variable to contain the leverage, $x_i(\mathbf{X}'\mathbf{X})^{-1}x_i'$, where $\mathbf{X} = \partial\mathbf{F}/\partial\boldsymbol{\beta}$ and x_i is the ith row of $\mathbf{X}$. If you specify the special variable _WEIGHT_, the leverage is $w_i x_i(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}x_i'$.

L95M=*name*

specifies a variable to contain the lower bound of an approximate 95% confidence interval for the expected value (mean). See also the description for the U95M= option, which follows.

L95=*name*

specifies a variable to contain the lower bound of an approximate 95% confidence interval for an individual prediction. This includes the variance of the error as well as the variance of the parameter estimates. See also the description for the U95= option, which follows.

PARMS=*names*

specifies variables in the output data set to contain parameter estimates. These can be the same variable names as listed in the PARAMETERS statement; however, you can choose new names for the parameters identified in the sequence from the parameter estimates table. A note log indicates which variable in the output data set is associated with each parameter name. Note that, for each of these new variables, the values are the same for every observation in the new data set.

PREDICTED=*name*
P=*name*
specifies a variable in the output data set to contain the predicted values of the dependent variable.

RESIDUAL=*name*
R=*name*
specifies a variable in the output data set to contain the residuals (actual values minus predicted values).

SSE=*name*
ESS=*name*
specifies a variable to include in the new data set. The values for the variable are the residual sums of squares finally determined by the procedure. The values of the variable are the same for every observation in the new data set.

STDI=*name*
specifies a variable to contain the standard error of the individual predicted value.

STDP=*name*
specifies a variable to contain the standard error of the mean predicted value.

STDR=*name*
specifies a variable to contain the standard error of the residual.

STUDENT=*name*
specifies a variable to contain the studentized residuals, which are residuals divided by their standard errors.

U95M=*name*
specifies a variable to contain the upper bound of an approximate 95% confidence interval for the expected value (mean). See also the description for the L95M= option.

U95=*name*
specifies a variable to contain the upper bound of an approximate 95% confidence interval for an individual prediction. See also the description for the L95= option.

WEIGHT=*name*
specifies a variable in the output data set that contains values of the special variable _WEIGHT_.

PARAMETERS Statement

PARAMETERS *parameter=values* . . . **;**

PARMS *parameter=values* . . . **;**

A PARAMETERS (or PARMS) statement must come before the RUN statement. Several parameter names and values can appear. The parameter names must all be valid SAS names and must not duplicate the names of any variables in the data set to which the NLIN procedure is applied. Any parameters specified but not used in the MODEL statement are dropped from the estimation.

In each *parameter=values* specification, the parameter name identifies a parameter to be estimated, both in subsequent procedure statements and in the output. *Values* specify the possible starting values of the parameter.

Usually, only one value is specified for each parameter. If you specify several values for each parameter, PROC NLIN evaluates the model at each point on the grid. The value specifications can take any of several forms:

m	a single value
$m1, m2, \ldots, mn$	several values
m TO n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m TO n BY i	a sequence where m equals the starting value, n equals the ending value, and the increment is i
$m1$, $m2$ TO $m3$	mixed values and sequences

This PARMS statement specifies five parameters and sets their possible starting values as shown:

```
parms  b0=0
       b1=4 to 8
       b2=0 to .6 by .2
       b3=1, 10, 100
       b4=0, .5, 1 to 4;
```

Possible starting values

B0	B1	B2	B3	B4
0	4	0.0	1	0.0
	5	0.2	10	0.5
	6	0.4	100	1.0
	7	0.6		2.0
	8			3.0
				4.0

Residual sums of squares are calculated for each of the $1 \times 5 \times 4 \times 3 \times 6 = 360$ combinations of possible starting values. (This can take a long time.) See the "Special Variables" section beginning on page 3020 for information on programming parameter starting values.

RETAIN Statement

RETAIN *variable* <*=values*> < . . . *variable* <*=values*>> **;**

The RETAIN statement declares retained variables and specifies their values. A retained variable is like a control variable (see the section "CONTROL Statement" on page 3011) except that it is retained only *within* iterations. An iteration involves a single pass through the data set.

Other Program Statements with PROC NLIN

PROC NLIN supports many statements that are similar to SAS programming statements used in a DATA step. However, there are some differences in capabilities; for additional information, see the section "Incompatibilities with 6.11 and Earlier Versions of PROC NLIN" beginning on page 3031.

Several SAS program statements can be used after the PROC NLIN statement. These statements can appear anywhere in the PROC NLIN statement, but new variables must be created before they appear in other statements. For example, the following statements are valid since they create the variable temp before they use it in the MODEL statement:

```
proc nlin;
   parms b0=0 to 2 by 0.5 b1=0.01 to 0.09 by 0.01;
   temp=exp(-b1*x);
   model y=b0*(1-temp);
```

The following statements result in missing values for y because the variable temp is undefined before it is used:

```
proc nlin;
   parms b0=0 to 2 by 0.5 b1=0.01 to 0.09 by 0.01;
   model y=b0*(1-temp);
   temp=exp(-b1*x);
```

PROC NLIN can process assignment statements, explicitly or implicitly subscripted ARRAY statements, explicitly or implicitly subscripted array references, IF statements, SAS functions, and program control statements. You can use program statements to create new SAS variables for the duration of the procedure. These variables are not permanently included in the data set to which PROC NLIN is applied. Program statements can include variables in the DATA= data set, parameter names, variables created by preceding program statements within PROC NLIN, and special variables used by PROC NLIN. All of the following SAS program statements can be used in PROC NLIN:

- ARRAY
- assignment (`y = a*x + b;`)
- CALL

- DO
- iterative DO
- DO UNTIL
- DO WHILE
- END
- FILE
- GO TO
- IF-THEN/ELSE
- LINK-RETURN
- PUT (defaults to the list)
- RETURN
- SELECT
- sum (`y + 1;`)

These statements can use the special variables created by PROC NLIN. Consult the section "Special Variables" beginning on page 3020 for more information on special variables.

Details

Automatic Derivatives

Depending on the optimization method you select, analytical first- and second-order derivatives are computed automatically. Derivatives can still be supplied using the DER.parm syntax. These DER.parm derivatives are not verified by the differentiator. If any needed derivatives are not supplied, they are computed and added to the program statements. To view the computed derivatives, use the LISTDER or LIST option.

The following model is solved using Newton's method. Analytical first- and second-order derivatives are automatically computed.

```
proc nlin data=Enzyme method=newton list;
   parms x1=4 x2=2 ;
   model Velocity = x1 * exp (x2 * Concentration);
run;
```

```
                                The NLIN Procedure

                         Listing of Compiled Program Code
           Stmt    Line:Col      Statement as Parsed

              1    285:74        MODEL.Velocity = x1 * EXP(x2
                                 * Concentration);
              1    285:74        @MODEL.Velocity/@x1 = EXP(x2
                                 * Concentration);
              1    285:74        @MODEL.Velocity/@x2 = x1 * Concentration
                                 * EXP(x2 * Concentration);
              1    285:74        @@MODEL.Velocity/@x1/@x2 = Concentration
                                 * EXP(x2 * Concentration);
              1    285:74        @@MODEL.Velocity/@x2/@x1 = Concentration
                                 * EXP(x2 * Concentration);
              1    285:74        @@MODEL.Velocity/@x2/@x2 = x1
                                 * Concentration * Concentration
                                 * EXP(x2 * Concentration);
```

Figure 50.5. Model and Derivative Code Output

Note that all the derivatives require the evaluation of EXP(X2 * Concentration). If you specify the LISTCODE option in the PROC NLIN statement, the actual machine level code produced is as follows.

```
                                The NLIN Procedure

                                  Code Listing


     1 Stmt MODEL      line 296 column 78.
                       (1)
                       arg=MODEL.Velocity
                       argsave=MODEL.
                       Velocity
                       Source Text:          model Velocity = x1 * exp
                                             (x2 * Concentration);
         Oper *        at 296:108 (30,0,2).  * : _temp1 <- x2 Concentration
         Oper EXP      at 296:104            EXP : _temp2 <- _temp1
                       (103,0,1).
         Oper *        at 296:98 (30,0,2).   * : MODEL.Velocity <- x1 _temp2
         Oper eeocf    at 296:98 (18,0,1).   eeocf : _DER_ <- _DER_
         Oper =        at 296:98 (1,0,1).    = : @MODEL.Velocity/@x1 <- _temp2
         Oper *        at 296:104 (30,0,2).  * : @1dt1_1 <- Concentration _temp2
         Oper *        at 296:98 (30,0,2).   * : @MODEL.Velocity/@x2
                                             <- x1 @1dt1_1
         Oper =        at 296:98 (1,0,1).    = : @@MODEL.Velocity/@x1/@x2
                                             <- @1dt1_1
         Oper =        at 296:98 (1,0,1).    = : @@MODEL.Velocity/@x2/@x1
                                             <- @1dt1_1
         Oper *        at 296:104 (30,0,2).  * : @2dt1_1 <- Concentration
                                             @1dt1_1
         Oper *        at 296:98 (30,0,2).   * : @@MODEL.Velocity/@x2/@x2
                                             <- x1 @2dt1_1
```

Figure 50.6. LISTCODE Output

Note that, in the generated code, only one exponentiation is performed. The generated code reuses previous operations to be more efficient.

Hougaard's Measure of Skewness

A "close-to-linear" nonlinear regression model, first described by Ratkowsky (1990), is a model that produces parameters having properties similar to those produced by a linear regression model. That is, the least squares estimates of the parameters are close to being unbiased, normally distributed, minimum variance estimators.

A nonlinear regression model sometimes fails to be close to linear due to the properties of a single parameter. When this occurs, bias in the parameters can render inferences using the reported standard errors and confidence limits invalid. You can often fix the problem with reparameterization, replacing the offending parameter by one with better estimation properties.

You can use Hougaard's measure of skewness, g_{1i}, to assess whether a parameter is close to linear or whether it contains considerable nonlinearity. Specify the HOUGAARD option in the PROC NLIN statement to compute Hougaard's measure of skewness.

According to Ratkowsky (1990), if $|g_{1i}| < 0.1$, the estimator $\hat{\theta}_i$ of parameter θ_i is very close-to-linear in behavior and, if $0.1 < |g_{1i}| < .25$, the estimator is reasonably close-to-linear. If $|g_{1i}| > .25$, the skewness is very apparent. For $|g_{1i}| > 1$, the nonlinear behavior is considerable.

Hougaard's measure is computed as follows

$$E[\hat{\theta}_i - E(\hat{\theta}_i)]^3 = -(mse)^2 \sum_{jkl}^{np} L^{ij} L^{ik} L^{il} (W_{jkl} + W_{kjl} + W_{ljk})$$

where the sum is a triple sum over the number of parameters and

$$L = (X'X)^{-1}$$

$$W_{jkl} = \sum_{m=1}^{n} J_m^j H_m^{kl}$$

In the preceding equation, J_m is the Jacobian vector and H_m is the Hessian matrix evaluated at observation m. This third moment is normalized using the standard error as

$$g_{1i} = E[\hat{\theta}_i - E(\hat{\theta}_i)]^3 / (mse * L^{ii})^{3/2}$$

Missing Values

If the value of any one of the SAS variables involved in the model is missing from an observation, that observation is omitted from the analysis. If only the value of the dependent variable is missing, that observation has a predicted value calculated for it when you use an OUTPUT statement and specify the PREDICTED= option.

If an observation includes a missing value for one of the independent variables, both the predicted value and the residual value are missing for that observation. If the iterations fail to converge, all the values of all the variables named in the OUTPUT statement are missing values.

Special Variables

Several special variables are created automatically and can be used in PROC NLIN program statements.

Special Variables with Values that are Set by PROC NLIN

The values of the following six special variables are set by PROC NLIN and should not be reset to a different value by programming statements:

ERROR is set to 1 if a numerical error or invalid argument to a function occurs during the current execution of the program. It is reset to 0 before each new execution.

ITER represents the current iteration number. The variable _ITER_ is set to -1 during the grid search phase.

MODEL is set to 1 for passes through the data when only the predicted values are needed, not the derivatives. It is 0 when both predicted values and derivatives are needed. If your derivative calculations consume a lot of time, you can save resources by coding

```
   if _model_ then return;
```

after your MODEL statement but before your derivative calculations. The derivatives generated by PROC NLIN do this automatically.

_N_ indicates the number of times the PROC NLIN step has been executed. It is never reset for successive passes through the data set.

OBS indicates the observation number in the data set for the current program execution. It is reset to 1 to start each pass through the data set (unlike the _N_ variable).

SSE has the error sum of squares of the last iteration. During the grid search phase, the _SSE_ variable is set to 0. For iteration 0, the _SSE_ variable is set to the SSE associated with the point chosen from the grid search.

Special Variable Used to Determine Convergence Criteria

The special variable _LOSS_ can be used to determine convergence criteria:

LOSS is used to determine the criterion function for convergence and step shortening. PROC NLIN looks for the variable _LOSS_ in the program statements and, if it is defined, uses the (weighted) sum of this value instead of the residual sum of squares to determine the criterion function for convergence and step shortening. This feature is useful in certain types of maximum-likelihood estimation where the residual sum of squares is not the basic criterion.

Weighted Regression with the Special Variable _WEIGHT_

To get weighted least squares estimates of parameters, the _WEIGHT_ variable can be given a value in an assignment statement:

```
_weight_ =  expression;
```

When this statement is included, the expression on the right-hand side of the assignment statement is evaluated for each observation in the data set to be analyzed. The values obtained are taken as inverse elements of the diagonal variance-covariance matrix of the dependent variable.

When a variable name is given after the equal sign, the values of the variable are taken as the inverse elements of the variance-covariance matrix. The larger the _WEIGHT_ value, the more importance the observation is given.

The _WEIGHT_ variable can be a function of the estimated parameters. For estimation purposes the derivative of the _WEIGHT_ variable with-respect-to the parameters is not included in the gradient and the Hessian of the loss function. This is normally the desired approach for iteratively reweighted least squares estimation. With the _WEIGHT_ variable a function of parameters, the gradient and the Hessian used may lead to poor convergence or non-convergence of the requested estimation. To have the derivative of the _WEIGHT_ variable with-respect-to the parameters included in the gradient and the Hessian of the loss function, do not use the _WEIGHT_ variable. Instead, redefine the model as

$$(y - f(x,\beta))/\sqrt{wgt(\beta)}$$

where y is the original dependent variable, $f(x,\beta)$ is the nonlinear model, and $wgt(\beta)$ is the weight that is a function of the parameters.

If the _WEIGHT_= statement is not used, the default value of 1 is used, and regular least squares estimates are obtained.

Troubleshooting

This section describes a number of problems that can occur in your analysis with PROC NLIN.

Excessive Time

If you specify a grid of starting values that contains many points, the analysis may take excessive time since the procedure must go through the entire data set for each point on the grid.

The analysis may also take excessive time if your problem takes many iterations to converge since each iteration requires as much time as a linear regression with predicted values and residuals calculated.

Dependencies

The matrix of partial derivatives may be singular, possibly indicating an over-parameterized model. For example, if b0 starts at zero in the following model, the derivatives for b1 are all zero for the first iteration.

```
parms b0=0 b1=.022;
model pop=b0*exp(b1*(year-1790));
der.b0=exp(b1*(year-1790));
der.b1=(year-1790)*b0*exp(b1*(year-1790));
```

The first iteration changes a subset of the parameters; then the procedure can make progress in succeeding iterations. This singularity problem is local. The next example displays a global problem.

You may have a term b2 in the exponent that is nonidentifiable since it trades roles with b0.

```
parms b0=3.9 b1=.022 b2=0;
model pop=b0*exp(b1*(year-1790)+b2);
der.b0=exp(b1*(year-1790)+b2);
der.b1=(year-1790)*b0*exp(b1*(year-1790)+b2);
der.b2=b0*exp(b1*(year-1790)+b2);
```

Unable to Improve

The method may lead to steps that do not improve the estimates even after a series of step halvings. If this happens, the procedure issues a message stating that it is unable to make further progress, but it then displays the warning message

```
PROC NLIN failed to converge
```

and displays the results. This often means that the procedure has not converged at all. If you provided the derivatives, check them very closely and then check the sum-of-squares error surface before proceeding. If PROC NLIN has not converged, try a different set of starting values, a different METHOD= specification, the G4 option, or a different model.

Divergence

The iterative process may diverge, resulting in overflows in computations. It is also possible that parameters enter a space where arguments to such functions as LOG and SQRT become illegal. For example, consider the following model:

```
parms b=0;
model y=x / b;
```

Suppose that y happens to be all zero and x is nonzero. There is no least squares estimate for b since the SSE declines as b approaches infinity or minus infinity. The same model could be parameterized with no problem into y = a*x.

If you have divergence problems, try reparameterizing, selecting different starting values, increasing the maximum allowed number of iterations (the MAXITER= option), specifying an alternative METHOD= option, or including a BOUNDS statement.

Local Minimum

The program may converge to a local rather than a global minimum. For example, consider the following model.

```
parms a=1 b=-1;
model y=(1-a*x)*(1-b*x);
```

Once a solution is found, an equivalent solution with the same SSE can be obtained by swapping the values of a and b.

Discontinuities

The computational methods assume that the model is a continuous and smooth function of the parameters. If this is not true, the method does not work. For example, the following models do not work:

```
model y=a+int(b*x);

model y=a+b*x+4*(z>c);
```

Responding to Trouble

PROC NLIN does not necessarily produce a good solution the first time. Much depends on specifying good initial values for the parameters. You can specify a grid of values in the PARMS statement to search for good starting values. While most practical models should give you no trouble, other models may require switching to a different iteration method or an inverse computation method. Specifying the option METHOD=MARQUARDT sometimes works when the default method (Gauss-Newton) does not work.

Computational Methods

For the system of equations represented by the nonlinear model

$$\mathbf{Y} = \mathbf{F}(\beta_0, \beta_1, \ldots, \beta_r, \mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n) + \epsilon = \mathbf{F}(\boldsymbol{\beta}^*) + \epsilon$$

where $\mathbf{Z}$ is a matrix of the independent variables, $\boldsymbol{\beta}^*$ is a vector of the parameters, ϵ is the error vector, and $\mathbf{F}$ is a function of the independent variables and the parameters, there are two approaches to solving for the minimum. The first method is to minimize

$$L(\boldsymbol{\beta}) = 0.5(\mathbf{e}'\mathbf{e})$$

where $\mathbf{e} = \mathbf{Y} - \mathbf{F}(\boldsymbol{\beta})$ and $\boldsymbol{\beta}$ is an estimate of $\boldsymbol{\beta}^*$.

The second method is to solve the nonlinear “normal” equations

$$\mathbf{X}'\mathbf{F}(\boldsymbol{\beta}) = \mathbf{X}'\mathbf{Y}$$

where

$$\mathbf{X} = \frac{\partial \mathbf{F}}{\partial \boldsymbol{\beta}}$$

In the nonlinear situation, both $\mathbf{X}$ and $\mathbf{F}(\boldsymbol{\beta})$ are functions of $\boldsymbol{\beta}$ and a closed-form solution generally does not exist. Thus, PROC NLIN uses an iterative process: a starting value for $\boldsymbol{\beta}$ is chosen and continually improved until the error sum of squares $\epsilon'\epsilon$ is minimized.

The iterative techniques that PROC NLIN uses are similar to a series of linear regressions involving the matrix $\mathbf{X}$ evaluated for the current values of $\boldsymbol{\beta}$ and $\mathbf{e} = \mathbf{Y} - \mathbf{F}(\boldsymbol{\beta})$, the residuals evaluated for the current values of $\boldsymbol{\beta}$.

The iterative process begins at some point $\boldsymbol{\beta}_0$. Then $\mathbf{X}$ and $\mathbf{Y}$ are used to compute a $\boldsymbol{\Delta}$ such that

$$\text{SSE}(\boldsymbol{\beta}_0 + k\boldsymbol{\Delta}) < \text{SSE}(\boldsymbol{\beta}_0)$$

The four methods differ in how $\boldsymbol{\Delta}$ is computed to change the vector of parameters.

$$\begin{aligned}
\text{Steepest descent: } \boldsymbol{\Delta} &= \mathbf{X}'\mathbf{e} \\
\text{Gauss-Newton: } \boldsymbol{\Delta} &= (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{e} \\
\text{Newton: } \boldsymbol{\Delta} &= (\mathbf{G}^{-})\mathbf{X}'\mathbf{e} \\
\text{Marquardt: } \boldsymbol{\Delta} &= (\mathbf{X}'\mathbf{X} + \lambda \text{diag}(\mathbf{X}'\mathbf{X}))^{-}\mathbf{X}'\mathbf{e}
\end{aligned}$$

The default method used to compute $(\mathbf{X}'\mathbf{X})^{-}$ is the sweep operator producing a reflexive generalized (g_2) inverse. In some cases it would be preferable to use a Moore-Penrose (g_4) inverse. If the G4 option is specified in the PROC NLIN statement, a g_4 inverse is used to calculate $\mathbf{\Delta}$ on each iteration.

The Gauss-Newton and Marquardt iterative methods regress the residuals onto the partial derivatives of the model with respect to the parameters until the estimates converge. The Newton iterative method regresses the residuals onto a function of the first and second derivatives of the model with respect to the parameters until the estimates converge. Analytical first- and second-order derivatives are automatically computed.

Steepest Descent (Gradient)

The steepest descent method is based on the gradient of $\epsilon'\epsilon$:

$$\frac{1}{2}\frac{\partial L(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\mathbf{XY} + \mathbf{XF}(\boldsymbol{\beta}) = -\mathbf{X}'\mathbf{e}$$

The quantity $-\mathbf{X}'\mathbf{e}$ is the gradient along which $\epsilon'\epsilon$ increases. Thus $\mathbf{\Delta} = \mathbf{X}'\mathbf{e}$ is the direction of steepest descent.

If the automatic variables _WEIGHT_ and _RESID_ are used, then

$$\mathbf{\Delta} = \mathbf{X}'\mathbf{W}^{\mathbf{SSE}}\mathbf{r}$$

is the direction, where

$\mathbf{W}^{\mathbf{SSE}}$ is an $n \times n$ diagonal matrix with elements w_i^{SSE} of weights from the _WEIGHT_ variable. Each element w_i^{SSE} contains the value of _WEIGHT_ for the ith observation.

$\mathbf{r}$ is a vector with elements r_i from _RESID_. Each element r_i contains the value of _RESID_ evaluated for the ith observation.

Using the method of steepest descent, let

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \alpha\mathbf{\Delta}$$

where the scalar α is chosen such that

$$\text{SSE}(\boldsymbol{\beta}_i + \alpha\mathbf{\Delta}) < \text{SSE}(\boldsymbol{\beta}_i)$$

Note: The steepest descent method may converge very slowly and is therefore not generally recommended. It is sometimes useful when the initial values are poor.

Newton

The Newton method uses the second derivatives and solves the equation

$$\mathbf{\Delta} = \mathbf{G}^{-}\mathbf{X}'\mathbf{e}$$

where

$$\mathbf{G} = (\mathbf{X}'\mathbf{X}) + \sum_{i=1}^{n} H_i(\boldsymbol{\beta})\mathbf{e}_i$$

and $H_i(\boldsymbol{\beta})$ is the Hessian of $\mathbf{e}$:

$$[H_i]_{jk} = \left[\frac{\partial^2 \mathbf{e}_i}{\partial \boldsymbol{\beta}_j \partial \boldsymbol{\beta}_k} \right]_{jk}$$

If the automatic variables _WEIGHT_, _WGTJPJ_, and _RESID_ are used, then

$$\mathbf{\Delta} = \mathbf{G}^{-}\mathbf{X}'\mathbf{W}^{\mathbf{SSE}}\mathbf{r}$$

is the direction, where

$$\mathbf{G} = \mathbf{X}'\mathbf{W}^{\mathbf{XPX}}\mathbf{X} + \sum_{i=1}^{n} \mathbf{H}_i(\boldsymbol{\beta}) w_i^{XPX} r_i$$

and

$\mathbf{W}^{\mathbf{SSE}}$ is an $n \times n$ diagonal matrix with elements w_i^{SSE} of weights from the _WEIGHT_ variable. Each element w_i^{SSE} contains the value of _WEIGHT_ for the ith observation.

$\mathbf{W}^{\mathbf{XPX}}$ is an $n \times n$ diagonal matrix with elements w_i^{XPX} of weights from the _WGTJPJ_ variable.

Each element w_i^{XPX} contains the value of _WGTJPJ_ for the ith observation.

$\mathbf{r}$ is a vector with elements r_i from the _RESID_ variable. Each element r_i contains the value of _RESID_ evaluated for the ith observation.

Gauss-Newton

The Gauss-Newton method uses the Taylor series

$$\mathbf{F}(\boldsymbol{\beta}) = \mathbf{F}(\boldsymbol{\beta}_0) + \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_0) + \cdots$$

where $\mathbf{X} = \partial \mathbf{F}/\partial \boldsymbol{\beta}$ is evaluated at $\boldsymbol{\beta} = \boldsymbol{\beta}_0$.

Substituting the first two terms of this series into the normal equations

$$\begin{aligned} \mathbf{X}'\mathbf{F}(\boldsymbol{\beta}) &= \mathbf{X}'\mathbf{Y} \\ \mathbf{X}'(\mathbf{F}(\boldsymbol{\beta}_0) + \mathbf{X}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)) &= \mathbf{X}'\mathbf{Y} \\ (\mathbf{X}'\mathbf{X})(\boldsymbol{\beta} - \boldsymbol{\beta}_0) &= \mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{F}(\boldsymbol{\beta}_0) \\ (\mathbf{X}'\mathbf{X})\boldsymbol{\Delta} &= \mathbf{X}'\mathbf{e} \end{aligned}$$

and therefore

$$\boldsymbol{\Delta} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{e}$$

Caution: If $\mathbf{X}'\mathbf{X}$ is singular or becomes singular, PROC NLIN computes $\boldsymbol{\Delta}$ using a generalized inverse for the iterations after singularity occurs. If $\mathbf{X}'\mathbf{X}$ is still singular for the last iteration, the solution should be examined.

Marquardt

The Marquardt updating formula is as follows:

$$\boldsymbol{\Delta} = (\mathbf{X}'\mathbf{X} + \lambda \mathrm{diag}(\mathbf{X}'\mathbf{X}))^{-}\mathbf{X}'\mathbf{e}$$

The Marquardt method is a compromise between the Gauss-Newton and steepest descent methods (Marquardt 1963). As $\lambda \rightarrow 0$, the direction approaches Gauss-Newton. As $\lambda \rightarrow \infty$, the direction approaches steepest descent.

Marquardt's studies indicate that the average angle between Gauss-Newton and steepest descent directions is about 90°. A choice of λ between 0 and infinity produces a compromise direction.

By default, PROC NLIN chooses $\lambda = 10^{-7}$ to start and computes a $\boldsymbol{\Delta}$. If SSE$(\boldsymbol{\beta}_0 + \boldsymbol{\Delta}) <$ SSE$(\boldsymbol{\beta}_0)$, then $\lambda = \lambda/10$ for the next iteration. Each time SSE$(\boldsymbol{\beta}_0 + \boldsymbol{\Delta}) >$ SSE$(\boldsymbol{\beta}_0)$, then $\lambda = 10\lambda$.

Note: If the SSE decreases on each iteration, then $\lambda \rightarrow 0$, and you are essentially using the Gauss-Newton method. If SSE does not improve, then λ is increased until you are moving in the steepest descent direction.

Marquardt's method is equivalent to performing a series of ridge regressions and is useful when the parameter estimates are highly correlated or the objective function is not well approximated by a quadratic.

Step-Size Search

The default method of finding the step size k is step halving using SMETHOD=HALVE. If SSE$(\boldsymbol{\beta}_0 + \boldsymbol{\Delta}) > $ SSE$(\boldsymbol{\beta}_0)$, compute SSE$(\boldsymbol{\beta}_0 + 0.5\boldsymbol{\Delta})$, SSE$(\boldsymbol{\beta}_0 + 0.25\boldsymbol{\Delta}), \ldots,$ until a smaller SSE is found.

If you specify SMETHOD=GOLDEN, the step size k is determined by a golden section search. The parameter TAU determines the length of the initial interval to be searched, with the interval having length TAU or 2×TAU, depending on SSE$(\boldsymbol{\beta}_0 + \boldsymbol{\Delta})$. The RHO parameter specifies how fine the search is to be. The SSE at each endpoint of the interval is evaluated, and a new subinterval is chosen. The size of the interval is reduced until its length is less than RHO. One pass through the data is required each time the interval is reduced. Hence, if RHO is very small relative to TAU, a large amount of time can be spent determining a step size. For more information on the GOLDEN search, refer to Kennedy and Gentle (1980).

If you specify SMETHOD=CUBIC, the NLIN procedure performs a cubic interpolation to estimate the step size. If the estimated step size does not result in a decrease in SSE, step halving is used.

Output Data Sets

The data set produced by the OUTEST= option in the PROC NLIN statement contains the parameter estimates on each iteration including the grid search.

The variable _ITER_ contains the iteration number. The variable _TYPE_ denotes whether the observation contains iteration parameter estimates ('ITER'), final parameter estimates ('FINAL'), or covariance estimates ('COVB'). The variable _NAME_ contains the parameter name for covariances, and the variable _SSE_ contains the objective function value for the parameter estimates. The variable _STATUS_ indicates whether the estimates have converged.

The data set produced by the OUTPUT statement contains statistics calculated for each observation. In addition, the data set contains all the variables in the input data set and any ID variables that are specified in the ID statement.

Confidence Intervals

Parameter Confidence Intervals

The parameter confidence intervals are computed using the Wald based formula:

$$\hat{\beta}_i \pm stderr_i * t(N - P, 0.05/2)$$

where $stderr_i$ is the standard error of the i^{th} parameter $\hat{\beta}_i$ and $t(N - P, 0.05/2)$ is a t statistic with $N - P$ degrees of freedom, N is the number of observations, and P is the number of parameters. The confidence intervals are only asymptotically valid.

Model Confidence Intervals

Model confidence intervals are output when an OUT= data set is specified and one or more of the options L95M=, L95=, U95M=, or U95= is specified. The values of these terms are

$$\begin{aligned}
H &= w_i x_i (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1} x_i' \\
L95M &= f(\beta, \mathbf{z}_i) - \sqrt{MSE * H/w_i} * t(N-P, 0.975/2) \\
U95M &= f(\beta, \mathbf{z}_i) + \sqrt{MSE * H/w_i} * t(N-P, 0.975/2) \\
L95 &= f(\beta, \mathbf{z}_i) - \sqrt{MSE(H + 1/w_i)} * t(N-P, 0.975/2) \\
U95 &= f(\beta, \mathbf{z}_i) + \sqrt{MSE(H + 1/w_i)} * t(N-P, 0.975/2)
\end{aligned}$$

where $\mathbf{X} = \partial f / \partial \boldsymbol{\beta}$ and x_i is the ith row of $\mathbf{X}$. These results are derived for linear systems. The intervals are approximate for nonlinear models.

Parameter Covariance Matrix

For unconstrained estimates (no active bounds), the parameter covariance matrix is

$$(X'X)^{-1} * mse$$

for the gradient, Marquardt, and Gauss methods and

$$H^{-1} * mse$$

for Newton method. The mse is computed as

$$r'r/(nused - np)$$

where $nused$ is the number of non-missing observations and np is the number of estimable parameters. The standard error reported for the parameters is the sqrt of the corresponding diagonal element of this matrix.

Equality restrictions can be written as a vector function

$$h(\theta) = 0$$

Inequality restrictions are either active or inactive. When an inequality restriction is active, it is treated as an equality restriction.

For the following, assume the vector $h(\theta)$ contains all the current active restrictions. The constraint matrix A is

$$A(\hat{\theta}) = \frac{\partial h(\hat{\theta})}{\partial \hat{\theta}}$$

The covariance matrix for the restricted parameter estimates is computed as

$$Z(Z'HZ)^{-1}Z'$$

where H is Hessian or approximation to the Hessian, and Z is the last $(np - nc)$ columns of Q. Q is from an LQ factorization of the constraint matrix, *nc* is the number of active constraints, and *np* is the number of parameters. Refer to Gill, Murray, and Wright (1981) for more details on LQ factorization.

The covariance matrix for the Lagrange multipliers is computed as

$$(AH^{-1}A')^{-1}$$

Reported Convergence Measures

NLIN computes and reports four convergence measures labeled R, PPC, RPC, and OBJECT.

R is the primary convergence measure for the parameters. It measures the degree to which the residuals are orthogonal to the Jacobian columns, and it approaches 0 as the gradient of the objective function becomes small. R is defined as

$$\sqrt{\frac{r'X(X'X)^{-1}X'r}{\text{LOSS}^i}}$$

PPC is the prospective parameter change measure. PPC measures the maximum relative change in the parameters implied by the parameter-change vector computed for the next iteration. At the *k*th iteration, PPC is the maximum over the parameters

$$\frac{|\theta_i^{k+1} - \theta_i^k|}{|\theta|_i^k + 1E - 6}$$

where θ_i^k is the current value of the *i*th parameter and θ_i^{k+1} is the prospective value of this parameter after adding the change vector computed for the next iteration. These changes are measured before steplength adjustments are made. The parameter with the maximum prospective relative change is displayed with the value of PPC, unless the PPC is nearly 0.

RPC is the retrospective parameter change measure. RPC measures the maximum relative change in the parameters from the previous iteration. At the *k*th iteration, RPC is the maximum over *i* of

$$\frac{|\theta_i^k - \theta_i^{k-1}|}{|\theta_i^{k-1} + 1E - 6|}$$

where θ_i^k is the current value of the *i*th parameter and θ_i^{k-1} is the previous value of this parameter. These changes are measured before steplength adjustments are made. The name of the parameter with the maximum retrospective relative change is displayed with the value of RPC, unless the RPC is nearly 0.

OBJECT measures the relative change in the objective function value between iterations:

$$\frac{|O^k - O^{k-1}|}{|O^{k-1} + 1E-6|}$$

where O^{k-1} is the value of the objective function (O^k) from the previous iteration. This is the old CONVERGEOBJ= criterion.

Displayed Output

In addition to the output data sets, PROC NLIN also produces the following items:

- the estimates of the parameters and the residual Sums of Squares determined in each iteration
- a list of the residual Sums of Squares associated with all or some of the combinations of possible starting values of parameters
- an analysis-of-variance table including as sources of variation Regression, Residual, Uncorrected Total, Corrected Total, and F test

If the convergence criterion is met, PROC NLIN produces

- Estimation Summary Table
- Parameter Estimates
- an asymptotically valid standard error of the estimate, Asymptotic Standard Error.
- an Asymptotic 95% Confidence Interval for the estimate of the parameter
- an Asymptotic Correlation Matrix of the parameters

Incompatibilities with 6.11 and Earlier Versions of PROC NLIN

The NLIN procedure now uses a compiler that is different from the DATA step compiler. The compiler was changed so that analytical derivatives could be computed automatically. For the most part, the syntax accepted by the old NLIN procedure can be used in the new NLIN procedure. However, there are several differences that should be noted.

- You cannot specify a character index variable in the DO statement, and you cannot specify a character test in the IF statement. Thus DO I=1,2,3; is supported, but DO I='ONE','TWO','THREE'; is not supported. And IF 'THIS' < 'THAT' THEN . . .; is supported, but "IF 'THIS' THEN . . .;" is not supported.
- The PUT statement, which is used mostly for program debugging in PROC NLIN, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not.
  - The PUT statement does not support line pointers, factored lists, iteration factors, overprinting, the _INFILE_ option, the ':' format modifier, or the symbol '$'.
  - The PUT statement does support expressions inside of parentheses. For example, PUT (SQRT(X)); produces the square root of X.
  - The PUT statement also supports the option _PDV_ to display a formatted listing of all the variables in the program. The statement PUT _PDV_; prints a much more readable listing of the variables than PUT _ALL_; does.
- You cannot use the '*' subscript, but you can specify an array name in a PUT statement without subscripts. Thus, ARRAY A . . .; PUT A; is acceptable, but PUT A[*] ; is not. The statement PUT A; displays all the elements of the array A. The PUT A=; statement displays all the elements of A with each value labeled by the name of the element variable.
- You cannot specify any arguments in the ABORT statement.
- You can specify more than one target statement in the WHEN and OTHERWISE statements. That is, DO/END groups are not necessary for multiple WHEN statements, for example, SELECT; WHEN(exp1); stmt1; stmt2; WHEN(exp2); stmt3; stmt4; END;.
- You can specify only the options LOG, PRINT, and LIST in the FILE statement.
- The RETAIN statement retains only values across one pass through the data set. If you need to retain values across iterations, use the CONTROL statement to make a control variable.

The ARRAY statement in PROC NLIN is similar to, but not the same as, the ARRAY statement in the SAS DATA step. The ARRAY statement is used to associate a name (of no more than 8 characters) with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC NLIN does not support all the features of the DATA step ARRAY statement. You cannot specify implicit indexing variables; all array references must have explicit subscript expressions. You can specify simple array dimensions; lower bound specifications are not supported. A maximum of six dimensions are accepted.

On the other hand, the ARRAY statement supported by PROC NLIN does accept both variables and constants as array elements. (Constant array elements cannot be changed with assignment statements.)

```
proc nlin data=nld;
array b[4] 1 2 3 4;     /* Constant array */
array c[4] ( 1 2 3 4 ); /* Numeric array with initial values */

b[1] = 2;               /* This is an ERROR, b is a constant array*/
c[2] = 7.5;             /* This is allowed */
...
```

Both dimension specification and the list of elements are optional, but at least one must be specified. When the list of elements is not specified, or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

If the array is used as a pure array in the program rather than a list of symbols (the individual symbols of the array are not referenced in the code), the array is converted to a numerical array. A pure array is literally a vector of numbers that are accessed only by index. Using these types of arrays results in faster derivatives and compiled code.

```
proc nlin data=nld;
array c[4] ( 1 2 3 4 ); /* Numeric array with initial values */

c[2] = 7.5;             /* This is C used as a pure array  */
c1 = -92.5;             /* This forces C to be a list of symbols */
```

ODS Table Names

PROC NLIN assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 50.1. ODS Tables Produced in PROC NLIN

ODS Table Name	Description	Statement
ANOVA	Analysis of variance	default
CodeDependency	Variable cross reference	LISTDEP
CodeList	Listing of program statements	LISTCODE
ConvergenceStatus	Convergence status	default
CorrB	Correlation of the parameters	default
EstSummary	Summary of the estimation	default
FirstDerivatives	First derivative table	LISTDER
IterHistory	Iteration output	default
MissingValues	Missing values generated by the program	default
ParameterEstimates	Parameter estimates	default
ProgList	Listing of the compiled program	LIST

Convergence Status Table

The ConvergenceStatus table can be used to programmatically check on the status of an estimation. The ConvergenceStatus table contains the variable STATUS that takes on one of the values, 0, 1, or 3. If STATUS equals 0, then the convergence criterion were met. If STATUS equals 1, then the convergence criterion were met but notes were added to the log that may indicate a problem with the model. If STATUS equals 3, then the convergence criterion were not met.

The following sample program demonstrates how the ConvergenceStatus table can be used.

```
      /* Save the ConvergenceStatus    */
      /* table to the dataset "status" */
ods output ConvergenceStatus=status;
proc nlin data=a ;
   parameters a=1 b=1 c=1;
   model wgt=a+x/(b*y+c*z);
run;

data _NULL_; set status;
  if status > 0 then put "A problem Occurred";
run;
```

Examples

Example 50.1. Segmented Model

From theoretical considerations, you can hypothesize that

$$\begin{aligned} y &= a + b\,x + c\,x^2 && \text{if } x < x_0 \\ y &= p && \text{if } x >= x_0 \end{aligned}$$

That is, for values of x less than x_0, the equation relating y and x is quadratic (a parabola); and, for values of x greater than x_0, the equation is constant (a horizontal line). PROC NLIN can fit such a segmented model even when the joint point, x_0, is unknown.

The curve must be continuous (the two sections must meet at x_0), and the curve must be smooth (the first derivatives with respect to x are the same at x_0).

These conditions imply that

$$
\begin{aligned}
x_0 &= -b/2c \\
p &= a - b^2/4c
\end{aligned}
$$

The segmented equation includes only three parameters; however, the equation is nonlinear with respect to these parameters.

You can write program statements with PROC NLIN to conditionally execute different sections of code for the two parts of the model, depending on whether x is less than x_0 .

A PUT statement is used to print the constrained parameters every time the program is executed for the first observation (where x =1). The following statements perform the analysis.

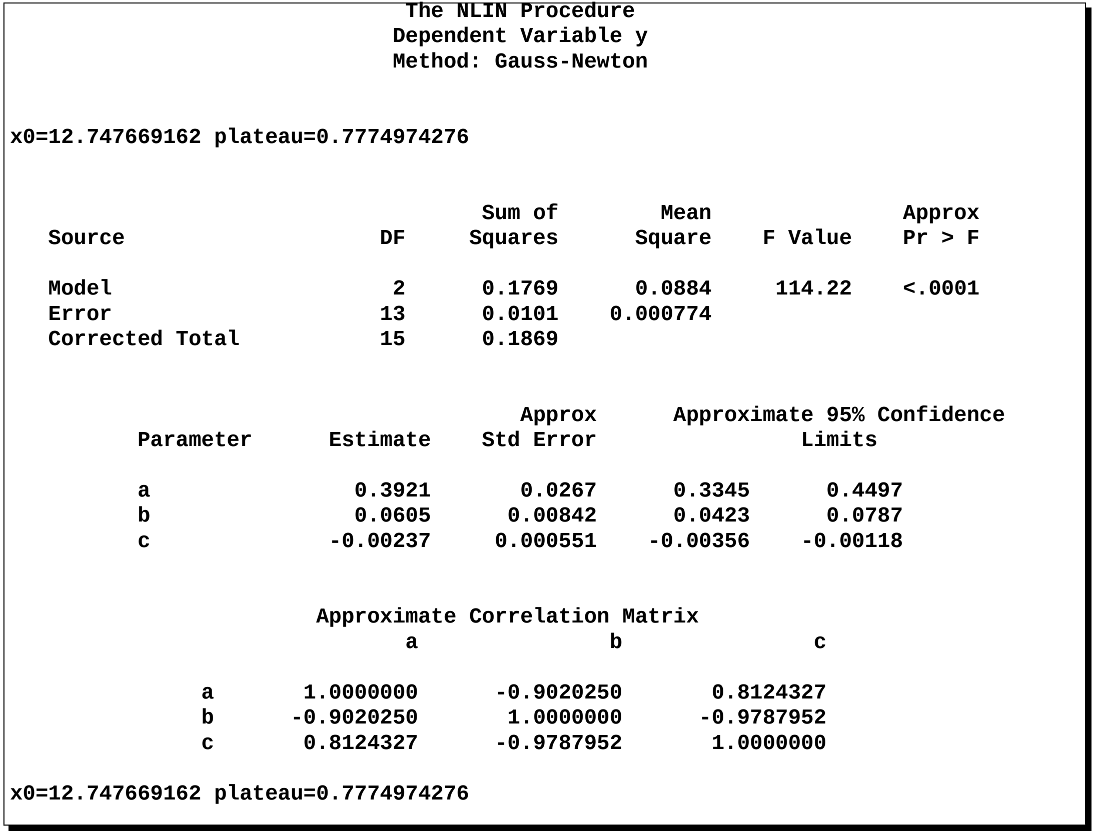
```
*---------FITTING A SEGMENTED MODEL USING NLIN-----*
|      |                                           |
|   Y  | QUADRATIC          PLATEAU                |
|      | Y=A+B*X+C*X*X      Y=P                    |
|      |                     ..................... |
|      |              .     :                      |
|      |          .         :                      |
|      |       .            :                      |
|      |     .              :                      |
|      |    .               :                      |
|      +-----------------------------------------X |
|                          X0                      |
|                                                  |
| CONTINUITY RESTRICTION: P=A+B*X0+C*X0**2         |
| SMOOTHNESS RESTRICTION: 0=B+2*C*X0 SO X0=-B/(2*C)|
*--------------------------------------------------*;

title 'Quadratic Model with Plateau';
data a;
   input y x @@;
   datalines;
.46 1  .47  2 .57  3 .61  4 .62  5 .68  6 .69  7
.78 8  .70  9 .74 10 .77 11 .78 12 .74 13 .80 13
.80 15 .78 16
;
proc nlin;
   parms a=.45 b=.05 c=-.0025;

   x0=-.5*b / c;                  * Estimate join point;
   if x<x0 then                   * Quadratic part of Model;
      model y=a+b*x+c*x*x;
   else                           * Plateau part of Model;
      model y=a+b*x0+c*x0*x0;

   if _obs_=1 and _iter_ =.  then do;
```

```
      plateau=a+b*x0+c*x0*x0;
      put /  x0= plateau=  ;
      end;
   output out=b predicted=yp;
run;

/* Setup for creating the graph */
legend1 frame cframe=ligr label=none cborder=black
        position=center value=(justify=center);
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

proc gplot;
   plot y*x yp*x/frame cframe=ligr legend=legend1
   vaxis=axis1 haxis=axis2 overlay ;
run;
```

Output 50.1.1. Nonlinear Least Squares Iterative Phase

```
                          Quadratic Model with Plateau

                               The NLIN Procedure
                              Dependent Variable y
                              Method: Gauss-Newton

                                Iterative Phase
                                                            Sum of
              Iter           a           b           c     Squares

                 0      0.4500      0.0500    -0.00250      0.0562
                 1      0.3881      0.0616    -0.00234      0.0118
                 2      0.3930      0.0601    -0.00234      0.0101
                 3      0.3922      0.0604    -0.00237      0.0101
                 4      0.3921      0.0605    -0.00237      0.0101
                 5      0.3921      0.0605    -0.00237      0.0101
                 6      0.3921      0.0605    -0.00237      0.0101

x0=12.747669162 plateau=0.7774974276


        NOTE: Convergence criterion met.

x0=12.747669162 plateau=0.7774974276
```

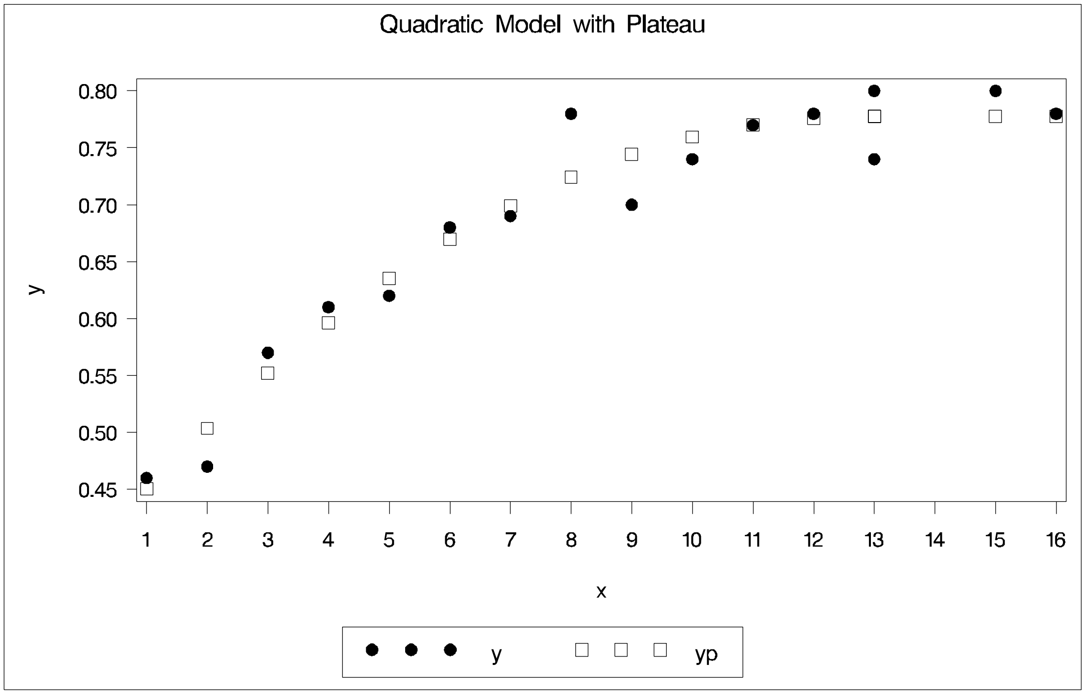
Output 50.1.2. Least Squares Analysis for the Quadratic Model



```
                              The NLIN Procedure
                             Dependent Variable y
                            Method: Gauss-Newton


x0=12.747669162 plateau=0.7774974276


                                    Sum of        Mean               Approx
   Source                    DF    Squares      Square    F Value    Pr > F

   Model                      2     0.1769      0.0884     114.22    <.0001
   Error                     13     0.0101    0.000774
   Corrected Total           15     0.1869


                                       Approx      Approximate 95% Confidence
          Parameter      Estimate    Std Error              Limits

          a                0.3921       0.0267      0.3345      0.4497
          b                0.0605      0.00842      0.0423      0.0787
          c              -0.00237     0.000551    -0.00356    -0.00118


                        Approximate Correlation Matrix
                              a               b               c

               a       1.0000000      -0.9020250       0.8124327
               b      -0.9020250       1.0000000      -0.9787952
               c       0.8124327      -0.9787952       1.0000000

x0=12.747669162 plateau=0.7774974276
```

Output 50.1.1 indicates that the join point is 12.75 and the plateau value is 0.78. As displayed in the following plot of the predicted values (YP) and the actual values, the selected join point and plateau value is reasonable. The predicted values for the estimation are written to the data set b with the OUTPUT statement.

Output 50.1.3. Observed and Predicted Values for the Quadratic Model



Example 50.2. Iteratively Reweighted Least Squares

The NLIN procedure is suited to methods that make the weight a function of the parameters in each iteration since the _WEIGHT_ variable can be computed with program statements.

The NOHALVE option is used because the SSE definition is modified at each iteration and the step-shortening criteria is thus circumvented.

Iteratively reweighted least squares (IRLS) can produce estimates for many of the robust regression criteria suggested in the literature. These methods act like automatic outlier rejectors since large residual values lead to very small weights. Holland and Welsch (1977) outline several of these robust methods. For example, the biweight criterion suggested by Beaton and Tukey (1974) tries to minimize

$$S_{biweight} = \Sigma\rho(r)$$

where

$$\rho(r) = (B^2/2)(1-(1-(r/B)^2)^2) \quad \text{if } |r| \leq B$$

or

$$\rho(r) = (B^2/2) \qquad \text{otherwise}$$

where r is $|residual|/\sigma$, σ is a measure of scale of the error, and B is a tuning constant.

The weighting function for the biweight is

$$w_i = (1 - (r_i/B)^2)^2 \quad \text{if } |r_i| \leq B$$

or

$$w_i = 0 \qquad \text{if } |r_i| > B$$

The biweight estimator depends on both a measure of scale (like the standard deviation) and a tuning constant; results vary if these values are changed.

The data are the population of the United States (in millions), recorded at ten-year intervals starting in 1790 and ending in 1990.

```
title 'U.S. Population Growth';
data uspop;
   input pop :6.3 @@;
   retain year 1780;
   year=year+10;
   yearsq=year*year;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
226542 248710
;

title 'Beaton/Tukey Biweight Robust Regression using IRLS';
proc nlin data=uspop nohalve;
   parms b0=20450.43 b1=-22.7806 b2=.0063456;
   model pop=b0+b1*year+b2*year*year;
   resid=pop-model.pop;
   sigma=2;
   b=4.685;
   r=abs(resid / sigma);
   if r<=b then _weight_=(1-(r / b)**2)**2;
   else _weight_=0;
   output out=c r=rbi;
run;

data c;
set c;
   sigma=2;
   b=4.685;
   r=abs(rbi / sigma);
   if r<=b then _weight_=(1-(r / b)**2)**2;
   else _weight_=0;
proc print;
run;
```

Output 50.2.1. Nonlinear Least Squares Analysis

```
                Beaton/Tukey Biweight Robust Regression using IRLS

                                The NLIN Procedure

                                      Sum of        Mean               Approx
 Source                    DF        Squares      Square    F Value    Pr > F

 Model                      2         113564     56782.0    49454.5    <.0001
 Error                     18        20.6670      1.1482
 Corrected Total           20         113585


                                         Approx      Approximate 95% Confidence
        Parameter      Estimate       Std Error                 Limits

        b0              20828.7           259.4      20283.8      21373.6
        b1             -23.2004          0.2746     -23.7773     -22.6235
        b2              0.00646        0.000073      0.00631      0.00661
```

Output 50.2.2. Listing of Computed Weights from PROC NLIN

```
                Beaton/Tukey Biweight Robust Regression using IRLS

Obs      pop     year    yearsq       rbi     sigma      b         r       _weight_

  1      3.929   1790   3204100   -0.93711     2      4.685   0.46855    0.98010
  2      5.308   1800   3240000    0.46091     2      4.685   0.23045    0.99517
  3      7.239   1810   3276100    1.11853     2      4.685   0.55926    0.97170
  4      9.638   1820   3312400    0.95176     2      4.685   0.47588    0.97947
  5     12.866   1830   3348900    0.32159     2      4.685   0.16080    0.99765
  6     17.069   1840   3385600   -0.62597     2      4.685   0.31298    0.99109
  7     23.191   1850   3422500   -0.94692     2      4.685   0.47346    0.97968
  8     31.443   1860   3459600   -0.43027     2      4.685   0.21514    0.99579
  9     39.818   1870   3496900   -1.08302     2      4.685   0.54151    0.97346
 10     50.155   1880   3534400   -1.06615     2      4.685   0.53308    0.97427
 11     62.947   1890   3572100    0.11332     2      4.685   0.05666    0.99971
 12     75.994   1900   3610000    0.25539     2      4.685   0.12770    0.99851
 13     91.972   1910   3648100    2.03607     2      4.685   1.01804    0.90779
 14    105.710   1920   3686400    0.28436     2      4.685   0.14218    0.99816
 15    122.775   1930   3724900    0.56725     2      4.685   0.28363    0.99268
 16    131.669   1940   3763600   -8.61325     2      4.685   4.30662    0.02403
 17    151.325   1950   3802500   -8.32415     2      4.685   4.16207    0.04443
 18    179.323   1960   3841600   -0.98543     2      4.685   0.49272    0.97800
 19    203.211   1970   3880900    0.95088     2      4.685   0.47544    0.97951
 20    226.542   1980   3920400    1.03780     2      4.685   0.51890    0.97562
 21    248.710   1990   3960100   -1.33067     2      4.685   0.66533    0.96007
```

Output 50.2.2 displays the computed weights. The observations for 1940 and 1950 are highly discounted because of their large residuals.

Example 50.3. Probit Model with Likelihood function

The data, taken from Lee (1974), consist of patient characteristics and a variable indicating whether cancer remission occurred. This example demonstrates how to use PROC NLIN with a likelihood function. In this case, the likelihood function to minimize is

$$-2 \log L \quad = \quad -2 \sum_{i=1}^{N} wght_i \log(\hat{p}_i(y_i, \mathbf{x}_i))$$

where

$$\hat{p}_i(y_i, \mathbf{x}_i) = \begin{cases} \Phi(\alpha + \beta' \mathbf{x}_i) & y_i = 0 \\ 1 - \Phi(\alpha + \beta' x_i) & y_i = 1 \end{cases}$$

and Φ is the normal probability function. This is the likelihood function for a binary probit model. This likelihood is strictly positive so that you can take a square root of $\log(\hat{p}_i(y_i, \mathbf{x}_i))$ and use this as your residual in PROC NLIN. The DATA step also creates a zero-valued dummy variable, like, that is used as the dependent variable.

```
Data remiss;
   input remiss cell smear infil li blast temp;
   label remiss = 'complete remission';
   like = 0;
   label like = 'dummy variable for nlin';
   datalines;
1 .8 .83 .66 1.9 1.1 .996
1 .9 .36 .32 1.4 .74 .992
0 .8 .88 .7 .8 .176 .982
0 1 .87 .87 .7 1.053 .986
1 .9 .75 .68 1.3 .519 .98
0 1 .65 .65 .6 .519 .982
1 .95 .97 .92 1 1.23 .992
0 .95 .87 .83 1.9 1.354 1.02
0 1 .45 .45 .8 .322 .999
0 .95 .36 .34 .5 0 1.038
0 .85 .39 .33 .7 .279 .988
0 .7 .76 .53 1.2 .146 .982
0 .8 .46 .37 .4 .38 1.006
0 .2 .39 .08 .8 .114 .99
0 1 .9 .9 1.1 1.037 .99
1 1 .84 .84 1.9 2.064 1.02
0 .65 .42 .27 .5 .114 1.014
0 1 .75 .75 1 1.322 1.004
0 .5 .44 .22 .6 .114 .99
1 1 .63 .63 1.1 1.072 .986
0 1 .33 .33 .4 .176 1.01 0
0 .9 .93 .84 .6 1.591 1.02
1 1 .58 .58 1 .531 1.002
0 .95 .32 .3 1.6 .886 .988
1 1 .6 .6 1.7 .964 .99
1 1 .69 .69 .9 .398 .986
0 1 .73 .73 .7 .398 .986
;
run;
```

```
proc nlin data=remiss method=newton sigsq=1;
   parms a -2 b  -1 c 6 int -10;

        /* Linear portion of model ------*/
   eq1 = a*cell + b*li + c*temp +int;

          /* probit */
   p = probnorm(eq1);

   if ( remiss = 1 ) then p = 1-p;

   model.like = sqrt(- 2 * log( p));
   output out=p p=predict;
run;
```

Note that the asymptotic standard errors of the parameters are computed under the least squares assumptions. The SIGSQ=1 option on the PROC NLIN statement forces PROC NLIN to replace the usual mean square error with 1. Also, METHOD=NEWTON is selected so the true Hessian of the likelihood function is used to calculate parameter standard errors rather than the crossproducts approximation to the Hessian.

Output 50.3.1. Nonlinear Least Squares Analysis from PROC NLIN

```
                Beaton/Tukey Biweight Robust Regression using IRLS

                              The NLIN Procedure

               NOTE: An intercept was not specified for this model.

                                   Sum of        Mean               Approx
Source                    DF      Squares      Square    F Value    Pr > F

Model                      4     -21.9002     -5.4750      -5.75     .
Error                     23      21.9002      0.9522
Uncorrected Total         27            0


                                      Approx      Approximate 95% Confidence
       Parameter      Estimate    Std Error                 Limits

       a               -5.6298       4.6376    -15.2234      3.9638
       b               -2.2513       0.9790     -4.2764     -0.2262
       c               45.1815      34.9095    -27.0337       117.4
       int            -36.7548      32.3607      -103.7     30.1879
```

The problem can be more simply solved using the following SAS statements.

```
proc probit data=remiss ;
   class remiss;
   model remiss=cell li temp ;
run;
```

References

Bard, J. (1970), “Comparison of Gradient Methods for the Solution of the Nonlinear Parameter Estimation Problem,” *SIAM Journal of Numerical Analysis*, 7, 157–186.

Bard, J. (1974), *Nonlinear Parameter Estimation*, New York: Academic Press, Inc.

Bates, D. M., and Watts, D.L. (1981), “A Relative Offset Orthogonality Convergence Criterion for Nonlinear Least Squares” *Technometrics*, 123, 179-183.

Beaton, A.E. and Tukey, J.W. (1974), “The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data,” *Technometrics*, 16, 147–185.

Charnes, A., Frome, E.L., and Yu, P.L. (1976), “The Equivalence of Generalized Least Squares and Maximum Likelihood Estimation in the Exponential Family,” *Journal of the American Statistical Association*, 71, 169–172.

Cox, D.R. (1970), *Analysis of Binary Data*, London: Chapman and Hall.

Finney, D.J. (1971), *Probit Analysis*, Third Edition, Cambridge: Cambridge University Press.

Gallant, A.R. (1975), “Nonlinear Regression,” *American Statistician*, 29, 73–81.

Gill, Philip E., Murray, Walter, and Wright, Margaret H. (1981), "Practical Optimization," New York: Academic Press Inc.

Hartley, H.O. (1961), “The Modified Gauss-Newton Method for the Fitting of Non-Linear Regression Functions by Least Squares,” *Technometrics*, 3, 269–280.

Holland, P.H. and Welsch, R.E. (1977), “Robust Regression Using Iteratively Reweighted Least-Squares,” *Communications Statistics: Theory and Methods*, 6, 813–827.

Jennrich, R.I. (1969), “Asymptotic Properties of Nonlinear Least Squares Estimators,” *Annals of Mathematical Statistics*, 40, 633–643.

Jennrich, R.I. and Moore, R.H. (1975), “Maximum Likelihood Estimation by Means of Nonlinear Least Squares,” *American Statistical Association*, *1975 Proceedings of the Statistical Computing Section*, 57–65.

Jennrich, R.I. and Sampson, P.F. (1968), “Application of Stepwise Regression to Non-Linear Estimation,” *Technometrics*, 10, 63–72.

Judge, G.G., Griffiths, W.E., Hill, R.C., and Lee, Tsoung-Chao (1980), *The Theory and Practice of Econometrics*, New York: John Wiley & Sons, Inc.

Kennedy, W.J. and Gentle, J.E. (1980), *Statistical Computing*, New York: Marcel Dekker, Inc.

Lee, E.T. (1974), “A Computer Program for Linear Logistic Regression Analysis,” *Computer Programs in Biomedicine*, 80–92.

Marquardt, D.W. (1963), “An Algorithm for Least-Squares Estimation of Nonlinear Parameters,” *Journal for the Society of Industrial and Applied Mathematics*, 11, 431–441.

Nelder, J.A. and Wedderburn, R.W.M. (1972), “Generalized Linear Models,” *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

Ratkowsky, D. (1990), "Handbook of Nonlinear Regression Models," Marcel Dekker: New York and Basel.

Chapter 51
The NLMIXED Procedure

Chapter Contents

Chapter 51
The NLMIXED Procedure

Overview

Introduction

The NLMIXED procedure fits nonlinear mixed models, that is, models in which both fixed and random effects enter nonlinearly. These models have a wide variety of applications, two of the most common being pharmacokinetics and overdispersed binomial data. PROC NLMIXED enables you to specify a conditional distribution for your data (given the random effects) having either a standard form (normal, binomial, Poisson) or a general distribution that you code using SAS programming statements.

PROC NLMIXED fits nonlinear mixed models by maximizing an approximation to the likelihood integrated over the random effects. Different integral approximations are available, the principal ones being adaptive Gaussian quadrature and a first-order Taylor series approximation. A variety of alternative optimization techniques are available to carry out the maximization; the default is a dual quasi-Newton algorithm.

Successful convergence of the optimization problem results in parameter estimates along with their approximate standard errors based on the second derivative matrix of the likelihood function. PROC NLMIXED enables you to use the estimated model to construct predictions of arbitrary functions using empirical Bayes estimates of the random effects. You can also estimate arbitrary functions of the nonrandom parameters, and PROC NLMIXED computes their approximate standard errors using the delta method.

Literature on Nonlinear Mixed Models

Davidian and Giltinan (1995) and Vonesh and Chinchilli (1997) provide good overviews as well as general theoretical developments and examples of nonlinear mixed models. Pinheiro and Bates (1995) is a primary reference for the theory and computational techniques of PROC NLMIXED. They describe and compare several different integrated likelihood approximations and provide evidence that adaptive Gaussian quadrature is one of the best methods. Davidian and Gallant (1993) also use Gaussian quadrature for nonlinear mixed models, although the smooth nonparametric density they advocate for the random effects is currently not available in PROC NLMIXED.

Traditional approaches to fitting nonlinear mixed models involve Taylor series expansions, expanding around either zero or the empirical best linear unbiased predictions of the random effects. The former is the basis for the well-known first-order method of Beal and Sheiner (1982, 1988) and Sheiner and Beal (1985), and it is optionally available in PROC NLMIXED. The latter is the basis for the estimation method of Lindstrom and Bates (1990), and it is not available in PROC NLMIXED. However, the closely related Laplacian approximation is an option; it is equivalent to adaptive

Gaussian quadrature with only one quadrature point. The Laplacian approximation and its relationship to the Lindstrom-Bates method are discussed by Beal and Sheiner (1992), Wolfinger (1993), Vonesh (1992, 1996), Vonesh and Chinchilli (1997), and Wolfinger and Lin (1997).

A parallel literature exists in the area of generalized linear mixed models, in which random effects appear as a part of the linear predictor inside of a link function. Taylor-series methods similar to those just described are discussed in articles such as Harville and Mee (1984), Stiratelli, Laird, and Ware (1984), Gilmour, Anderson, and Rae (1985), Goldstein (1991), Schall (1991), Engel and Keen (1992), Breslow and Clayton (1993), Wolfinger and O'Connell (1993), and McGilchrist (1994), but such methods have not been implemented in PROC NLMIXED because they can produce biased results in certain binary data situations (Rodriguez and Goldman 1995, Lin and Breslow 1996). Instead, a numerical quadrature approach is available in PROC NLMIXED, as discussed in Pierce and Sands (1975), Anderson and Aitkin (1985), Crouch and Spiegelman (1990), Hedeker and Gibbons (1994), Longford (1994), McCulloch (1994), Liu and Pierce (1994), and Diggle, Liang, and Zeger (1994).

Nonlinear mixed models have important applications in pharmacokinetics, and Roe (1997) provides a wide-ranging comparison of many popular techniques. Yuh et al. (1994) provide an extensive bibliography on nonlinear mixed models and their use in pharmacokinetics.

PROC NLMIXED Compared with Other SAS Procedures and Macros

The models fit by PROC NLMIXED can be viewed as generalizations of the random coefficient models fit by the MIXED procedure. This generalization allows the random coefficients to enter the model nonlinearly, whereas in PROC MIXED they enter linearly. With PROC MIXED you can perform both maximum likelihood and restricted maximum likelihood (REML) estimation, whereas PROC NLMIXED only implements maximum likelihood. This is because the analog to the REML method in PROC NLMIXED would involve a high dimensional integral over all of the fixed-effects parameters, and this integral is typically not available in closed form. Finally, PROC MIXED assumes the data to be normally distributed, whereas PROC NLMIXED enables you to analyze data that are normal, binomial, or Poisson or that have any likelihood programmable with SAS statements.

PROC NLMIXED does not implement the same estimation techniques available with the NLINMIX and GLIMMIX macros. These macros are based on the estimation methods of Lindstrom and Bates (1990), Breslow and Clayton (1993), and Wolfinger and O'Connell (1993), and they iteratively fit a set of generalized estimating equations (refer to Chapters 11 and 12 of Littell et al. 1996 and to Wolfinger 1997). In contrast, PROC NLMIXED directly maximizes an approximate integrated likelihood.

This remark also applies to the SAS/IML macros MIXNLIN (Vonesh and Chinchilli 1997) and NLMEM (Galecki 1998).

PROC NLMIXED has close ties with the NLP procedure in SAS/OR software. PROC NLMIXED uses a subset of the optimization code underlying PROC NLP and has many of the same optimization-based options. Also, the programming statement functionality used by PROC NLMIXED is the same as that used by PROC NLP and the MODEL procedure in SAS/ETS software.

Getting Started

Nonlinear Growth Curves with Gaussian Data

As an introductory example, consider the orange tree data of Draper and Smith (1981). These data consist of seven measurements of the trunk circumference (in millimeters) on each of five orange trees. You can input these data into a SAS data set as follows:

```
data tree;
   input tree day y;
   datalines;
1  118   30
1  484   58
1  664   87
1 1004  115
1 1231  120
1 1372  142
1 1582  145
2  118   33
2  484   69
2  664  111
2 1004  156
2 1231  172
2 1372  203
2 1582  203
3  118   30
3  484   51
3  664   75
3 1004  108
3 1231  115
3 1372  139
3 1582  140
4  118   32
4  484   62
4  664  112
4 1004  167
4 1231  179
4 1372  209
4 1582  214
```

```
5  118   30
5  484   49
5  664   81
5 1004  125
5 1231  142
5 1372  174
5 1582  177
;
```

Lindstrom and Bates (1990) and Pinheiro and Bates (1995) propose the following logistic nonlinear mixed model for these data:

$$y_{ij} = \frac{b_1 + u_{i1}}{1 + \exp[-(d_{ij} - b_2)/b_3]} + e_{ij}$$

Here, y_{ij} represents the jth measurement on the ith tree ($i = 1, \ldots, 5$; $j = 1, \ldots, 7$), d_{ij} is the corresponding day, b_1, b_2, b_3 are the fixed-effects parameters, u_{i1} are the random-effect parameters assumed to be iid $N(0, \sigma_u^2)$, and e_{ij} are the residual errors assumed to be iid $N(0, \sigma_e^2)$ and independent of the u_{i1}. This model has a logistic form, and the random-effect parameters u_{i1} enter the model linearly.

The statements to fit this nonlinear mixed model are as follows:

```
proc nlmixed data=tree;
   parms b1=190 b2=700 b3=350 s2u=1000 s2e=60;
   num = b1+u1;
   ex = exp(-(day-b2)/b3);
   den = 1 + ex;
   model y ~ normal(num/den,s2e);
   random u1 ~ normal(0,s2u) subject=tree;
run;
```

The PROC NLMIXED statement invokes the procedure and inputs the TREE data set. The PARMS statement identifies the unknown parameters and their starting values. Here there are three fixed-effects parameters (B1, B2, B3) and two variance components (S2U, S2E).

The next three statements are SAS programming statements specifying the logistic mixed model. A new variable U1 is included to identify the random effect. These statements are evaluated for every observation in the data set when PROC NLMIXED computes the log likelihood function and its derivatives.

The MODEL statement defines the dependent variable and its conditional distribution given the random effects. Here a normal (Gaussian) conditional distribution is specified with mean NUM/DEN and variance S2E.

The RANDOM statement defines the single random effect to be U1, and specifies that it follows a normal distribution with mean 0 and variance S2U. The SUBJECT= argument defines a variable indicating when the random effect obtains new realizations; in this case, it changes according to the values of the TREE variable. PROC NLMIXED assumes that the input data set is clustered according to the levels of the

TREE variable; that is, all observations from the same tree occur sequentially in the input data set.

The output from this analysis is as follows.

```
                         The NLMIXED Procedure

                            Specifications

     Data Set                                    WORK.TREE
     Dependent Variable                          y
     Distribution for Dependent Variable         Normal
     Random Effects                              u1
     Distribution for Random Effects             Normal
     Subject Variable                            tree
     Optimization Technique                      Dual Quasi-Newton
     Integration Method                          Adaptive Gaussian
                                                 Quadrature
```

The “Specifications” table lists some basic information about the nonlinear mixed model you have specified. Included are the input data set, dependent and subject variables, random effects, relevant distributions, and type of optimization.

```
                         The NLMIXED Procedure

                              Dimensions

                 Observations Used                  35
                 Observations Not Used               0
                 Total Observations                 35
                 Subjects                            5
                 Max Obs Per Subject                 7
                 Parameters                          5
                 Quadrature Points                   1
```

The “Dimensions” table lists various counts related to the model, including the number of observations, subjects, and parameters. These quantities are useful for checking that you have specified your data set and model correctly. Also listed is the number of quadrature points that PROC NLMIXED has selected based on the evaluation of the log likelihood at the starting values of the parameters. Here, only one quadrature point is necessary because the random-effect parameters u_{i1} enter the model linearly.

```
                         The NLMIXED Procedure

                              Parameters

       b1          b2          b3         s2u         s2e    NegLogLike

      190         700         350        1000          60    132.491787
```

The “Parameters” table lists the parameters to be estimated, their starting values, and the negative log likelihood evaluated at the starting values.

```
                              The NLMIXED Procedure

                                Iteration History

        Iter      Calls     NegLogLike        Diff     MaxGrad       Slope

           1          4     131.686742    0.805045    0.010269      -0.633
           2          6      131.64466    0.042082    0.014783     -0.0182
           3          8     131.614077    0.030583    0.009809    -0.02796
           4         10     131.572522    0.041555    0.001186    -0.01344
           5         11     131.571895    0.000627      0.0002    -0.00121
           6         13     131.571889    5.549E-6    0.000092    -7.68E-6
           7         15     131.571888    1.096E-6    6.097E-6    -1.29E-6


                   NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table records the history of the minimization of the negative log likelihood. For each iteration of the quasi-Newton optimization, values are listed for the number of function calls, the value of the negative log likelihood, the difference from the previous iteration, the absolute value of the largest gradient, and the slope of the search direction. The note at the bottom of the table indicates that the algorithm has converged successfully according to the GCONV convergence criterion, a standard criterion computed using a quadratic form in the gradient and inverse Hessian.

```
                              The NLMIXED Procedure

                                 Fit Statistics

                     -2 Log Likelihood                  263.1
                     AIC (smaller is better)            273.1
                     AICC (smaller is better)           275.2
                     BIC (smaller is better)            271.2
```

The“Fitting Information” table lists the final maximized value of the log likelihood as well as the information criteria of Akaike and Schwarz in two different forms. These statistics can be used to compare different nonlinear mixed models.

```
                             The NLMIXED Procedure

                              Parameter Estimates

                       Standard
Parameter    Estimate     Error     DF   t Value   Pr > |t|    Alpha      Lower

b1             192.05   15.6473      4     12.27     0.0003     0.05     148.61
b2             727.90   35.2472      4     20.65     <.0001     0.05     630.04
b3             348.07   27.0790      4     12.85     0.0002     0.05     272.88
s2u            999.88    647.44      4      1.54     0.1974     0.05    -797.70
s2e           61.5139   15.8831      4      3.87     0.0179     0.05    17.4153

                              Parameter Estimates

                        Parameter       Upper    Gradient

                        b1             235.50    1.154E-6
                        b2             825.76    5.289E-6
                        b3             423.25     -6.1E-6
                        s2u           2797.45    -3.84E-6
                        s2e            105.61    2.892E-6
```

The “Parameter Estimates” table lists the maximum likelihood estimates of the five parameters and their approximate standard errors computed using the final Hessian matrix. Approximate t-values and Wald-type confidence limits are also provided, with degrees of freedom equal to the number of subjects minus the number of random effects. You should interpret these statistics cautiously for variance parameters like S2U and S2E. The final column in the output is the gradient vector at the optimization solution. Each element appears to be sufficiently small to indicate a stationary point.

Since the random-effect parameters u_{i1} enter the model linearly, you can obtain equivalent results by using the first-order method (specify METHOD=FIRO in the PROC NLMIXED statement).

Logistic-Normal Model with Binomial Data

This example analyzes the data from Beitler and Landis (1985), which represent results from a multi-center clinical trial investigating the effectiveness of two topical cream treatments (active drug, control) in curing an infection. For each of eight clinics, the number of trials and favorable cures are recorded for each treatment. The SAS data set is as follows.

```
data infection;
   input clinic t x n;
   datalines;
1 1 11 36
1 0 10 37
2 1 16 20
2 0 22 32
3 1 14 19
3 0  7 19
4 1  2 16
```

```
4 0  1 17
5 1  6 17
5 0  0 12
6 1  1 11
6 0  0 10
7 1  1  5
7 0  1  9
8 1  4  6
8 0  6  7
run;
```

Suppose n_{ij} denotes the number of trials for the ith clinic and the jth treatment ($i = 1, \ldots, 8 \quad j = 0, 1$), and x_{ij} denotes the corresponding number of favorable cures. Then a reasonable model for the preceding data is the following logistic model with random effects:

$$x_{ij}|u_i \sim \text{Binomial}(n_{ij}, p_{ij})$$

and

$$\eta_{ij} = \log\left(\frac{p_{ij}}{(1 - p_{ij})}\right) = \beta_0 + \beta_1 t_j + u_i$$

The notation t_j indicates the jth treatment, and the u_i are assumed to be iid $N(0, \sigma_u^2)$.

The PROC NLMIXED statements to fit this model are as follows:

```
proc nlmixed data=infection;
   parms beta0=-1 beta1=1 s2u=2;
   eta = beta0 + beta1*t + u;
   expeta = exp(eta);
   p = expeta/(1+expeta);
   model x ~ binomial(n,p);
   random u ~ normal(0,s2u) subject=clinic;
   predict eta out=eta;
   estimate ’1/beta1’ 1/beta1;
run;
```

The PROC NLMIXED statement invokes the procedure, and the PARMS statement defines the parameters and their starting values. The next three statements define p_{ij}, and the MODEL statement defines the conditional distribution of x_{ij} to be binomial. The RANDOM statement defines U to be the random effect with subjects defined by the CLINIC variable.

The PREDICT statement constructs predictions for each observation in the input data set. For this example, predictions of η_{ij} and approximate standard errors of prediction are output to a SAS data set named ETA. These predictions include empirical Bayes estimates of the random effects u_i.

The ESTIMATE statement requests an estimate of the reciprocal of β_1.

The output for this model is as follows.

```
                          The NLMIXED Procedure

                              Specifications

    Data Set                                        WORK.INFECTION
    Dependent Variable                              x
    Distribution for Dependent Variable             Binomial
    Random Effects                                  u
    Distribution for Random Effects                 Normal
    Subject Variable                                clinic
    Optimization Technique                          Dual Quasi-Newton
    Integration Method                              Adaptive Gaussian
                                                    Quadrature
```

The “Specifications” table provides basic information about the nonlinear mixed model.

```
                          The NLMIXED Procedure

                                Dimensions

                  Observations Used                     16
                  Observations Not Used                  0
                  Total Observations                    16
                  Subjects                               8
                  Max Obs Per Subject                    2
                  Parameters                             3
                  Quadrature Points                      5
```

The “Dimensions” table provides counts of various variables. You should check this table to make sure the data set and model have been entered properly. PROC NLMIXED selects five quadrature points to achieve the default accuracy in the likelihood calculations.

```
                          The NLMIXED Procedure

                                Parameters

                  beta0       beta1         s2u    NegLogLike

                     -1           1           2    37.5945925
```

The “Parameters” table lists the starting point of the optimization.

```
                              The NLMIXED Procedure

                                Iteration History

         Iter     Calls    NegLogLike        Diff     MaxGrad       Slope

            1         2    37.3622692    0.232323    2.882077    -19.3762
            2         3    37.1460375    0.216232    0.921926    -0.82852
            3         5    37.0300936    0.115944    0.315897    -0.59175
            4         6    37.0223017    0.007792     0.01906    -0.01615
            5         7    37.0222472    0.000054    0.001743    -0.00011
            6         9    37.0222466     6.57E-7    0.000091    -1.28E-6
            7        11    37.0222466    5.38E-10    2.078E-6     -1.1E-9


                   NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table indicates successful convergence in seven iterations.

```
                              The NLMIXED Procedure

                                  Fit Statistics

                     -2 Log Likelihood                  74.0
                     AIC (smaller is better)            80.0
                     AICC (smaller is better)           82.0
                     BIC (smaller is better)            80.3
```

The “Fitting Information” table lists some useful statistics based on the maximized value of the log likelihood.

```
                              The NLMIXED Procedure

                               Parameter Estimates

                        Standard
Parameter   Estimate       Error     DF   t Value   Pr > |t|    Alpha       Lower

beta0        -1.1974      0.5561      7     -2.15     0.0683     0.05     -2.5123
beta1         0.7385      0.3004      7      2.46     0.0436     0.05     0.02806
s2u           1.9591      1.1903      7      1.65     0.1438     0.05     -0.8554

                               Parameter Estimates

                         Parameter      Upper    Gradient

                         beta0         0.1175     -3.1E-7
                         beta1         1.4488    -2.08E-6
                         s2u           4.7736    -2.48E-7
```

The “Parameter Estimates” table indicates marginal significance of the two fixed-effects parameters. The positive value of the estimate of β_1 indicates that the treatment significantly increases the chance of a favorable cure.

```
                                The NLMIXED Procedure

                                 Additional Estimates

                             Standard
Label      Estimate        Error     DF   t Value   Pr > |t|    Alpha      Lower      Upper

1/beta1      1.3542       0.5509      7      2.46     0.0436     0.05    0.05146     2.6569
```

The “Additional Estimates” table displays results from the ESTIMATE statement. The estimate of $1/\beta_1$ equals $1/0.7385 = 1.3541$ and its standard error equals $0.3004/0.7385^2 = 0.5509$ by the delta method (Billingsley 1986, Cox 1998). Note this particular approximation produces a t-statistic identical to that for the estimate of β_1.

Not shown is the ETA data set, which contains the original 16 observations and predictions of the η_{ij}.

Syntax

The following statements can be used with the NLMIXED procedure:

PROC NLMIXED *options* ;
ARRAY *array specification* ;
BOUNDS *boundary constraints* ;
BY *variables* ;
CONTRAST *’label’ expression <,expression>* ;
ESTIMATE *’label’ expression* ;
ID *names* ;
MODEL *model specification* ;
PARMS *parameters and starting values* ;
PREDICT *expression* ;
RANDOM *random effects specification* ;
REPLICATE *variable* ;
Program statements ;

The following sections provide a detailed description of each of these statements.

PROC NLMIXED Statement

PROC NLMIXED *options* **;**

This statement invokes the NLMIXED procedure. A large number of options are available in the PROC NLMIXED statement, and the following table categorizes them according to function.

Table 51.1. PROC NLMIXED statement options

Option	Description
Basic Options	
DATA=	input data set
METHOD=	integration method
Displayed Output Specifications	
START	gradient at starting values
HESS	Hessian matrix
ITDETAILS	iteration details
CORR	correlation matrix
COV	covariance matrix
ECORR	corr matrix of additional estimates
ECOV	cov matrix of additional estimates
EDER	derivatives of additional estimates
ALPHA=	alpha for confidence limits
DF=	degrees of freedom for p values and confidence limits
Debugging Output	
LIST	model program, variables
LISTCODE	compiled model program
LISTDEP	model dependency listing
LISTDER	model derivative
XREF	model cross reference
FLOW	model execution messages
TRACE	detailed model execution messages
Quadrature Options	
NOAD	no adaptive centering
NOADSCALE	no adaptive scaling
OUTQ=	output data set
QFAC=	search factor
QMAX=	maximum points
QPOINTS=	number of points
QSCALEFAC=	scale factor
QTOL=	tolerance
Empirical Bayes Options	
EBSTEPS=	number of Newton steps
EBSUBSTEPS=	number of substeps
EBSSFRAC=	step-shortening fraction
EBSSTOL=	step-shortening tolerance
EBTOL=	convergence tolerance
EBOPT	comprehensive optimization

Table 51.1. (continued)

Option	Description
EBZSTART	zero starting values
Optimization Specifications	
TECHNIQUE=	minimization technique
UPDATE=	update technique
LINESEARCH=	line-search method
LSPRECISION=	line-search precision
HESCAL=	type of Hessian scaling
INHESSIAN<=>	start for approximated Hessian
RESTART=	iteration number for update restart
OPTCHECK<=>	check optimality in neighborhood
Derivatives Specifications	
FD<=>	finite-difference derivatives
FDHESSIAN<=>	finite-difference second derivatives
DIAHES	use only diagonal of Hessian
Constraint Specifications	
LCEPSILON=	range for active constraints
LCDEACT=	LM tolerance for deactivating
LCSINGULAR=	tolerance for dependent constraints
Termination Criteria Specifications	
MAXFUNC=	maximum number of function calls
MAXITER=	maximum number of iterations
MINITER=	minimum number of iterations
MAXTIME=	upper limit seconds of CPU time
ABSCONV=	absolute function convergence criterion
ABSFCONV=	absolute function convergence criterion
ABSGCONV=	absolute gradient convergence criterion
ABSXCONV=	absolute parameter convergence criterion
FCONV=	relative function convergence criterion
FCONV2=	relative function convergence criterion
GCONV=	relative gradient convergence criterion
XCONV=	relative parameter convergence criterion
FDIGITS=	number accurate digits in objective function
FSIZE=	used in FCONV, GCONV criterion
XSIZE=	used in XCONV criterion
Step Length Specifications	
DAMPSTEP<=>	damped steps in line search
MAXSTEP=	maximum trust-region radius
INSTEP=	initial trust-region radius
Singularity Tolerances	
SINGCHOL=	tolerance for Cholesky roots
SINGHESS=	tolerance for Hessian
SINGSWEEP=	tolerance for sweep
SINGVAR=	tolerance for variances

Table 51.1. (continued)

Option	Description
Covariance Matrix Tolerances	
ASINGULAR=	absolute singularity for inertia
MSINGULAR=	relative M singularity for inertia
VSINGULAR=	relative V singularity for inertia
G4=	threshold for Moore-Penrose inverse
COVSING=	tolerance for singular COV matrix
CFACTOR=	multiplication factor for COV matrix

These options are described in alphabetical order. For a description of the mathematical notation used in the following sections, see the section "Modeling Assumptions and Notation."

ABSCONV=r
ABSTOL=r

specifies an absolute function convergence criterion. For minimization, termination requires $f(\theta^{(k)}) \leq r$. The default value of r is the negative square root of the largest double precision value, which serves only as a protection against overflows.

ABSFCONV=$r < [n] >$
ABSFTOL=$r < [n] >$

specifies an absolute function convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations:

$$|f(\theta^{(k-1)}) - f(\theta^{(k)})| \leq r$$

The same formula is used for the NMSIMP technique, but $\theta^{(k)}$ is defined as the vertex with the lowest function value, and $\theta^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r = 0$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV=$r < [n] >$
ABSGTOL=$r < [n] >$

specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small:

$$\max_j |g_j(\theta^{(k)})| \leq r$$

This criterion is not used by the NMSIMP technique. The default value is $r = 1\text{E}{-5}$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSXCONV=$r < [n] >$
ABSXTOL=$r < [n] >$

specifies an absolute parameter convergence criterion. For all techniques except NMSIMP, termination requires a small Euclidean distance between successive parameter vectors,

$$\parallel \theta^{(k)} - \theta^{(k-1)} \parallel_2 \leq r$$

For the NMSIMP technique, termination requires either a small length $\alpha^{(k)}$ of the vertices of a restart simplex,

$$\alpha^{(k)} \leq r$$

or a small simplex size,

$$\delta^{(k)} \leq r$$

where the simplex size $\delta^{(k)}$ is defined as the L1 distance from the simplex vertex $\xi^{(k)}$ with the smallest function value to the other n simplex points $\theta_l^{(k)} \neq \xi^{(k)}$:

$$\delta^{(k)} = \sum_{\theta_l \neq y} \parallel \theta_l^{(k)} - \xi^{(k)} \parallel_1$$

The default is $r = 1\mathrm{E} - 8$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

ALPHA=α

specifies the alpha level to be used in computing confidence limits. The default value is 0.05.

ASINGULAR=r

ASING=r

specifies an absolute singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is the square root of the smallest positive double precision value.

CFACTOR=f

specifies a multiplication factor f for the estimated covariance matrix of the parameter estimates.

COV

requests the approximate covariance matrix for the parameter estimates.

CORR

requests the approximate correlation matrix for the parameter estimates.

COVSING=$r > 0$

specifies a nonnegative threshold that determines whether the eigenvalues of a singular Hessian matrix are considered to be zero.

DAMPSTEP$<$**=**$r>$

DS$<$**=** $r>$

specifies that the initial step-size value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP technique) cannot be larger than r times the step-size value used in the former iteration. If you specify the DAMPSTEP option without factor r, the default value is $r = 2$. The DAMPSTEP=r option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they could lead to floating point overflows during the computation of objective functions and their derivatives. The DAMPSTEP=r option can save time-costly function calls that result in very small step sizes α. For more details on setting the start values of each line search, see the section "Restricting the Step Length" beginning on page 3096.

DATA=*SAS-data-set*

specifies the input data set. Observations in this data set are used to compute the log likelihood function that you specify with PROC NLMIXED statements.

NOTE: If you are using a RANDOM statement, the input data set must be clustered according to the SUBJECT= variable. One easy way to accomplish this is to sort your data by the SUBJECT= variable prior to calling PROC NLMIXED. PROC NLMIXED does not sort the input data set for you.

DF=d

specifies the degrees of freedom to be used in computing p values and confidence limits. The default value is the number of subjects minus the number of random effects for random effects models, and the number of observations otherwise.

DIAHES

specifies that only the diagonal of the Hessian is used.

EBOPT

requests that a more comprehensive optimization be carried out if the default empirical Bayes optimization fails to converge.

EBSSFRAC=$r > 0$

specifies the step-shortening fraction to be used while computing empirical Bayes estimates of the random effects. The default value is 0.8.

EBSSTOL=$r \geq 0$

specifies the objective function tolerance for determining the cessation of step-shortening while computing empirical Bayes estimates of the random effects. The default value is $r = 1\mathrm{E} - 8$.

EBSTEPS=$n \geq 0$

specifies the maximum number of Newton steps for computing empirical Bayes estimates of random effects. The default value is $n = 50$.

EBSUBSTEPS=$n \geq 0$

specifies the maximum number of step-shortenings for computing empirical Bayes estimates of random effects. The default value is $n = 20$.

EBTOL=$r \geq 0$

specifies the convergence tolerance for empirical Bayes estimation. The default value is $r = \epsilon\mathrm{E}4$, where ϵ is the machine precision. This default value equals approximately $1\mathrm{E} - 12$ on most machines.

EBZSTART

requests that a zero be used as starting values during empirical Bayes estimation. By default, the starting values are set equal to the estimates from the previous iteration (or zero for the first iteration).

ECOV

requests the approximate covariance matrix for all expressions specified in ESTIMATE statements.

ECORR

requests the approximate correlation matrix for all expressions specified in ESTIMATE statements.

EDER

requests the derivatives of all expressions specified in ESTIMATE statements with respect to each of the model parameters.

FCONV=$r < [n] >$
FTOL=$r < [n] >$

specifies a relative function convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(\theta^{(k)}) - f(\theta^{(k-1)})|}{\max(|f(\theta^{(k-1)})|, \mathrm{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. The same formula is used for the NMSIMP technique, but $\theta^{(k)}$ is defined as the vertex with the lowest function value, and $\theta^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default is r=$10^{-FDIGITS}$ where FDIGITS is the value of the FDIGITS= option. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FCONV2=$r < [n] >$
FTOL2=$r < [n] >$

specifies another function convergence criterion. For all techniques except NMSIMP, termination requires a small predicted reduction

$$df^{(k)} \approx f(\theta^{(k)}) - f(\theta^{(k)} + s^{(k)})$$

of the objective function. The predicted reduction

$$\begin{aligned} df^{(k)} &= -g^{(k)T}s^{(k)} - \frac{1}{2}s^{(k)T}H^{(k)}s^{(k)} \\ &= -\frac{1}{2}s^{(k)T}g^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step.

$$s^{(k)} = -[H^{(k)}]^{-1}g^{(k)}$$

For the NMSIMP technique, termination requires a small standard deviation of the function values of the $n+1$ simplex vertices $\theta_l^{(k)}$, $l = 0, \ldots, n$,

$$\sqrt{\frac{1}{n+1}\sum_l \left[f(\theta_l^{(k)}) - \overline{f}(\theta^{(k)})\right]^2} \leq r$$

where $\overline{f}(\theta^{(k)}) = \frac{1}{n+1} \sum_l f(\theta_l^{(k)})$. If there are n_{act} boundary constraints active at $\theta^{(k)}$, the mean and standard deviation are computed only for the $n + 1 - n_{act}$ unconstrained vertices. The default value is $r = 1\text{E} - 6$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FD$<$**= FORWARD | CENTRAL |**$r >$

specifies that all derivatives be computed using finite difference approximations. The following specifications are permitted:

FD is equivalent to FD=100.

FD=CENTRAL uses central differences.

FD=FORWARD uses forward differences.

FD=r uses central differences for the initial and final evaluations of the gradient, and Hessian. During iteration, start with forward differences and switch to a corresponding central-difference formula during the iteration process when one of the following two criteria is satisfied:

- The absolute maximum gradient element is less than or equal to r times the ABSGTOL threshold.
- The normalized predicted function reduction (see the GTOL option on page 3065) is less than or equal to $\max(1\text{E} - 6,\ r * \text{GTOL})$. The $1\text{E} - 6$ ensures that the switch is done, even if you set the GTOL threshold to zero.

Note that the FD and FDHESSIAN options cannot apply at the same time. The FDHESSIAN option is ignored when only first-order derivatives are used. See the section “Finite Difference Approximations of Derivatives” beginning on page 3091 for more information.

FDHESSIAN$<$**=FORWARD | CENTRAL**$>$
FDHES$<$**=FORWARD | CENTRAL**$>$
FDH$<$**=FORWARD | CENTRAL**$>$

specifies that second-order derivatives be computed using finite difference approximations based on evaluations of the gradients.

FDHESSIAN=FORWARD uses forward differences.

FDHESSIAN=CENTRAL uses central differences.

FDHESSIAN uses forward differences for the Hessian except for the initial and final output.

Note that the FD and FDHESSIAN options cannot apply at the same time. See the section “Finite Difference Approximations of Derivatives” beginning on page 3091 for more information.

FDIGITS=r

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -\log_{10} \epsilon$, where ϵ is the machine precision. The value of r is used to compute the interval size h for the computation of finite-difference approximations of the derivatives of the objective function and for the default value of the FCONV= option.

FLOW

displays a message for each statement in the model program as it is executed. This debugging option is very rarely needed and produces voluminous output.

FSIZE=r

specifies the FSIZE parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. For more details, see the FCONV= and GCONV= options.

G4=$n > 0$

specifies a dimension to determine the type of generalized inverse to use when the approximate covariance matrix of the parameter estimates is singular. The default value of n is 60. See the section "Covariance Matrix" beginning on page 3101 for more information.

GCONV=$r < [n] >$
GTOL=$r < [n] >$

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction is small,

$$\frac{g(\theta^{(k)})^T [H^{(k)}]^{-1} g(\theta^{(k)})}{\max(|f(\theta^{(k)})|, \mathrm{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate H is not available), the following criterion is used:

$$\frac{\parallel g(\theta^{(k)}) \parallel_2^2 \quad \parallel s(\theta^{(k)}) \parallel_2}{\parallel g(\theta^{(k)}) - g(\theta^{(k-1)}) \parallel_2 \max(|f(\theta^{(k)})|, \mathrm{FSIZE})} \leq r$$

This criterion is not used by the NMSIMP technique. The default value is $r = 1E-8$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

HESCAL=$0|1|2|3$
HS=$0|1|2|3$

specifies the scaling version of the Hessian matrix used in NRRIDG, TRUREG, NEWRAP, or DBLDOG optimization. If HS is not equal to 0, the first iteration and each restart iteration sets the diagonal scaling matrix $D^{(0)} = \mathrm{diag}(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

where $H_{i,i}^{(0)}$ are the diagonal elements of the Hessian. In every other iteration, the diagonal scaling matrix $D^{(0)} = \mathrm{diag}(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling is done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max\left[d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=2 specifies the Dennis, Gay, & Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max\left[0.6 * d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=3 specifies that d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In each scaling update, ϵ is the relative machine precision. The default value is HS=0. Scaling of the Hessian can be time consuming in the case where general linear constraints are active.

HESS

requests the display of the final Hessian matrix after optimization. If you also specify the START option, then the Hessian at the starting values is also printed.

INHESSIAN<=r >
INHESS<=r >

specifies how the initial estimate of the approximate Hessian is defined for the quasi-Newton techniques QUANEW and DBLDOG. There are two alternatives:

- If you do not use the r specification, the initial estimate of the approximate Hessian is set to the Hessian at $\theta^{(0)}$.
- If you do use the r specification, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix rI.

By default, if you do not specify the option INHESSIAN=r, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix rI, where the scalar r is computed from the magnitude of the initial gradient.

INSTEP=r

reduces the length of the first trial step during the line search of the first iterations. For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithm TRUREG or DBLDOG or the default step length of the line-search algorithms can result in arithmetic overflows. If this occurs, you should specify decreasing values of $0 < r < 1$ such as INSTEP=1E $-$ 1, INSTEP=1E $-$ 2, INSTEP=1E $-$ 4, and so on, until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG), the INSTEP= option specifies a factor $r > 0$ for the initial radius $\Delta^{(0)}$ of the trust region. The default initial trust-region radius is the length of the scaled gradient. This step corresponds to the default radius factor of $r = 1$.

- For line-search algorithms (NEWRAP, CONGRA, QUANEW), the INSTEP= option specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.
- For the Nelder-Mead simplex algorithm, using TECH=NMSIMP, the INSTEP=r option defines the size of the start simplex.

For more details, see the section "Computational Problems" beginning on page 3098.

ITDETAILS

requests a more complete iteration history, including the current values of the parameter estimates, their gradients, and additional optimization statistics. For further details, see the section "Iterations" beginning on page 3104.

LCDEACT=r
LCD=r

specifies a threshold r for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. During minimization, an active inequality constraint can be deactivated only if its Lagrange multiplier is less than the threshold value $r < 0$. The default value is

$$r = -\min(0.01, \max(0.1 * \text{ABSGCONV}, 0.001 * \text{gmax}^{(k)}))$$

where ABSGCONV is the value of the absolute gradient criterion, and $\text{gmax}^{(k)}$ is the maximum absolute element of the (projected) gradient $g^{(k)}$ or $Z^T g^{(k)}$. (See the section "Active Set Methods" beginning on page 3093 for a definition of Z.)

LCEPSILON=$r > 0$
LCEPS=$r > 0$
LCE=$r > 0$

specifies the range for active and violated boundary constraints. The default value is $r = 1\text{E}-8$. During the optimization process, the introduction of rounding errors can force PROC NLMIXED to increase the value of r by a factor of $10, 100, \ldots$. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR=$r > 0$
LCSING=$r > 0$
LCS=$r > 0$

specifies a criterion r, used in the update of the QR decomposition, that determines whether an active constraint is linearly dependent on a set of other active constraints. The default value is $r = 1\text{E}-8$. The larger r becomes, the more the active constraints are recognized as being linearly dependent. If the value of r is larger than 0.1, it is reset to 0.1.

LINESEARCH=i
LIS=i

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1987) for an introduction to line-search techniques. The value of i can be $1, \ldots, 8$. For CONGRA, QUANEW and NEWRAP, the default value is $i = 2$.

LIS=1	specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.
LIS=2	specifies a line-search method that needs more function than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.
LIS=3	specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.
LIS=4	specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.
LIS=5	specifies a line-search method that is a modified version of LIS=4.
LIS=6	specifies golden section line search (Polak 1971), which uses only function values for linear approximation.
LIS=7	specifies bisection line search (Polak 1971), which uses only function values for linear approximation.
LIS=8	specifies the Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

LIST

displays the model program and variable lists. The LIST option is a debugging feature and is not normally needed.

LISTCODE

displays the derivative tables and the compiled program code. The LISTCODE option is a debugging feature and is not normally needed.

LOGNOTE$<$=n $>$

writes periodic notes to the log describing the current status of computations. It is designed for use with analyses requiring extensive CPU resources. The optional integer value n specifies the desired level of reporting detail. The default is $n = 1$. Choosing $n = 2$ adds information about the objective function values at the end of each iteration. The most detail is obtained with $n = 3$, which also reports the results of function evaluations within iterations.

LSPRECISION=r
LSP=r

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and expensive line search may be necessary (Fletcher 1987). The second line-search method (which is the default for the NEWRAP, QUANEW, and CONGRA techniques) and the third line-search method approach exact line search for small LSPRECISION= values. If you have numerical problems, you should try to decrease the LSPRECISION= value to obtain a more precise line search. The default values are shown in the following table.

TECH=	**UPDATE=**	**LSP default**
QUANEW	DBFGS, BFGS	r = 0.4
QUANEW	DDFP, DFP	r = 0.06
CONGRA	all	r = 0.1
NEWRAP	no update	r = 0.9

For more details, refer to Fletcher (1987).

MAXFUNC=i
MAXFU=i

specifies the maximum number i of function calls in the optimization process. The default values are

- TRUREG, NRRIDG, NEWRAP: 125
- QUANEW, DBLDOG: 500
- CONGRA: 1000
- NMSIMP: 3000

Note that the optimization can terminate only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER=i
MAXIT=i

specifies the maximum number i of iterations in the optimization process. The default values are

- TRUREG, NRRIDG, NEWRAP: 50
- QUANEW, DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1000

These default values are also valid when i is specified as a missing value.

MAXSTEP=$r < [n] >$

specifies an upper bound for the step length of the line-search algorithms during the first n iterations. By default, r is the largest double precision value and n is the largest integer available. Setting this option can improve the speed of convergence for the CONGRA, QUANEW, and NEWRAP techniques.

MAXTIME=r

specifies an upper limit of r seconds of CPU time for the optimization process. The default value is the largest floating point double representation of your computer. Note that the time specified by the MAXTIME= option is checked only once at the end of each iteration. Therefore, the actual running time can be much longer than that specified by the MAXTIME= option. The actual running time includes the rest of the time needed to finish the iteration and the time needed to generate the output of the results.

METHOD=*value*

specifies the method for approximating the integral of the likelihood over the random effects. Valid values are as follows.

- FIRO
  specifies the first-order method of Beal and Sheiner (1982). When using METHOD=FIRO, you must specify the NORMAL distribution in the MODEL statement and you must also specify a RANDOM statement.
- GAUSS
  specifies adaptive Gauss-Hermite quadrature (Pinheiro and Bates 1995). You can prevent the adaptation with the NOAD option or prevent adaptive scaling with the NOADSCALE option. This is the default integration method.
- HARDY
  specifies Hardy quadrature based on an adaptive trapezoidal rule. This method is available only for one-dimensional integrals; that is, you must specify only one random effect.
- ISAMP
  specifies adaptive importance sampling (Pinheiro and Bates 1995) . You can prevent the adaptation with the NOAD option or prevent adaptive scaling with the NOADSCALE option. You can use the SEED= option to specify a starting seed for the random number generation used in the importance sampling. If you do not specify a seed, or specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock.

MINITER=i
MINIT=i

specifies the minimum number of iterations. The default value is 0. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms can behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations.

MSINGULAR=$r > 0$
MSING=$r > 0$

specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is $1\mathrm{E}-12$ if you do not specify the SINGHESS= option; otherwise, the default value is $\max(10\epsilon, (1\mathrm{E}-4) * \mathrm{SINGHESS})$. See the section "Covariance Matrix" beginning on page 3101 for more information.

NOAD

requests that the Gaussian quadrature be nonadaptive; that is, the quadrature points are centered at zero for each of the random effects and the current random-effects variance matrix is used as the scale matrix.

NOADSCALE

requests nonadaptive scaling for adaptive Gaussian quadrature; that is, the quadrature points are centered at the empirical Bayes estimates for the random effects, but the current random-effects variance matrix is used as the scale matrix. By default, the observed Hessian from the current empirical Bayes estimates is used as the scale matrix.

OPTCHECK$<$=$r > 0 >$

computes the function values $f(\theta_l)$ of a grid of points θ_l in a ball of radius of r about θ^*. If you specify the OPTCHECK option without factor r, the default value is $r = 0.1$ at the starting point and $r = 0.01$ at the terminating point. If a point θ_l^* is found with a better function value than $f(\theta^*)$, then optimization is restarted at θ_l^*.

OUTQ=*SAS-data-set*

specifies an output data set containing the quadrature points used for numerical integration.

QFAC=$r > 0$

specifies the additive factor used to adaptively search for the number of quadrature points. For METHOD=GAUSS, the search sequence is 1, 3, 5, 7, 9, 11, 11 + r, 11 + $2r$, . . ., where the default value of r is 10. For METHOD=ISAMP, the search sequence is 10, 10 + r, 10 + $2r$, . . ., where the default value of r is 50.

QMAX=$r > 0$

specifies the maximum number of quadrature points permitted before the adaptive search is aborted. The default values are 31 for adaptive Gaussian quadrature, 61 for non-adaptive Gaussian quadrature, 160 for adaptive importance sampling, and 310 for non-adaptive importance sampling.

QPOINTS=$n > 0$

specifies the number of quadrature points to be used during evaluation of integrals. For METHOD=GAUSS, n equals the number of points used in each dimension of the random effects, resulting in a total of n^r points, where r is the number of dimensions. For METHOD=ISAMP, n specifies the total number of quadrature points regardless of the dimension of the random effects. By default, the number of quadrature points is selected adaptively, and this option disables the adaptive search.

QSCALEFAC=$r > 0$

specifies a multiplier for the scale matrix used during quadrature calculations. The default value is 1.0.

QTOL=$r > 0$

specifies the tolerance used to adaptively select the number of quadrature points. When the relative difference between two successive likelihood calculations is less than r, then the search terminates and the lesser number of quadrature points is used during the subsequent optimization process. The default value is $1\mathrm{E} - 4$.

RESTART=$i > 0$
REST=$i > 0$

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after, at most, i iterations. Default values are

- CONGRA: UPDATE=PB: restart is performed automatically, i is not used.
- CONGRA: UPDATE≠PB: $i = \min(10n, 80)$, where n is the number of parameters.
- QUANEW: i is the largest integer available.

SEED=i

specifies the random number seed for METHOD=ISAMP. If you do not specify a seed, or specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. The value must be less than $2^{31} - 1$.

SINGCHOL=$r > 0$

specifies the singularity criterion r for Cholesky roots of the random-effects variance matrix and scale matrix for adaptive Gaussian quadrature. The default value is $1\mathrm{E}4$ times the machine epsilon; this product is approximately $1\mathrm{E}-12$ on most computers.

SINGHESS=$r > 0$

specifies the singularity criterion r for the inversion of the Hessian matrix. The default value is $1\mathrm{E} - 8$. See the ASINGULAR, MSINGULAR=, and VSINGULAR= options for more information.

SINGSWEEP=$r > 0$

specifies the singularity criterion r for inverting the variance matrix in the first-order method and the empirical Bayes Hessian matrix. The default value is $1\mathrm{E}4$ times the machine epsilon; this product is approximately $1\mathrm{E} - 12$ on most computers.

SINGVAR=$r > 0$

specifies the singularity criterion r below which statistical variances are considered to equal zero. The default value is $1\mathrm{E}4$ times the machine epsilon; this product is approximately $1\mathrm{E} - 12$ on most computers.

START

requests that the gradient of the log likelihood at the starting values be displayed. If you also specify the HESS option, then the starting Hessian is displayed as well.

TECHNIQUE=*value*
TECH=*value*

specifies the optimization technique. Valid values are

- CONGRA
  performs a conjugate-gradient optimization, which can be more precisely specified with the UPDATE= option and modified with the LINESEARCH= option. When you specify this option, UPDATE=PB by default.

- DBLDOG
  performs a version of double dogleg optimization, which can be more precisely specified with the UPDATE= option. When you specify this option, UPDATE=DBFGS by default.
- NMSIMP
  performs a Nelder-Mead simplex optimization.
- NONE
  does not perform any optimization. This option can be used
  - to perform a grid search without optimization
  - to compute estimates and predictions that cannot be obtained efficiently with any of the optimization techniques
- NEWRAP
  performs a Newton-Raphson optimization combining a line-search algorithm with ridging. The line-search algorithm LIS=2 is the default method.
- NRRIDG
  performs a Newton-Raphson optimization with ridging.
- QUANEW
  performs a quasi-Newton optimization, which can be defined more precisely with the UPDATE= option and modified with the LINESEARCH= option. This is the default estimation method.
- TRUREG
  performs a trust region optimization.

TRACE

displays the result of each operation in each statement in the model program as it is executed. This debugging option is very rarely needed, and it produces voluminous output.

UPDATE=*method*
UPD=*method*

specifies the update method for the quasi-Newton, double dogleg, or conjugate-gradient optimization technique. Not every update method can be used with each optimizer. See the section "Optimization Algorithms" beginning on page 3086 for more information.

Valid methods are

- BFGS
  performs the original Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the inverse Hessian matrix.
- DBFGS
  performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default update method.
- DDFP
  performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.

- DFP
  performs the original DFP update of the inverse Hessian matrix.
- PB
  performs the automatic restart update method of Powell (1977) and Beale (1972).
- FR
  performs the Fletcher-Reeves update (Fletcher 1987).
- PR
  performs the Polak-Ribiere update (Fletcher 1987).
- CD
  performs a conjugate-descent update of Fletcher (1987).

VSINGULAR=$r > 0$
VSING=$r > 0$

specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is $r = 1\text{E}-8$ if the SINGHESS= option is not specified, and it is the value of SINGHESS= option otherwise. See the section "Covariance Matrix" beginning on page 3101 for more information.

XCONV=$r < [n] > R$
XTOL=$r[n]$

specifies the relative parameter convergence criterion. For all techniques except NMSIMP, termination requires a small relative parameter change in subsequent iterations.

$$\frac{\max_j |\theta_j^{(k)} - \theta_j^{(k-1)}|}{\max(|\theta_j^{(k)}|, |\theta_j^{(k-1)}|, \text{XSIZE})} \leq r$$

For the NMSIMP technique, the same formula is used, but $\theta_j^{(k)}$ is defined as the vertex with the lowest function value and $\theta_j^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r = 1\text{E}-8$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE=$r > 0$

specifies the XSIZE parameter of the relative parameter termination criterion. The default value is $r = 0$. For more detail, see the XCONV= option.

ARRAY Statement

ARRAY *arrayname [{ dimensions }] [$] [variables and constants]* **;**

The ARRAY statement is similar to, but not the same as, the ARRAY statement in the SAS DATA step, and it is the same as the ARRAY statements in the NLIN, NLP, and MODEL procedures. The ARRAY statement is used to associate a name (of no more than eight characters) with a list of variables and constants. The array name is used with subscripts in the program to refer to the array elements. The following statements illustrate this.

```
array r[8] r1-r8;

do i = 1 to 8;
   r[i] = 0;
end;
```

The ARRAY statement does not support all the features of the ARRAY statement in the DATA step. It cannot be used to assign initial values to array elements. Implicit indexing of variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement does allow both variables and constants to be used as array elements. (Constant array elements cannot have values assigned to them.) Both dimension specification and the list of elements are optional, but at least one must be specified. When the list of elements is not specified or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

BOUNDS Statement

BOUNDS *b_con [, b_con...]* **;**

where *b_con* := number *operator* parameter_list *operator* number
or *b_con* := number *operator* parameter_list
or *b_con* := parameter_list *operator* number
and *operator* := $<=$, $<$, $>=$, or $>$

Boundary constraints are specified with a BOUNDS statement. One- or two-sided boundary constraints are allowed. The list of boundary constraints are separated by commas. For example,

```
bounds 0 <= a1-a9 X <= 1, -1 <= c2-c5;
bounds b1-b10 y >= 0;
```

You can specify more than one BOUNDS statement. If you specify more than one lower (upper) bound for the same parameter, the maximum (minimum) of these is taken.

If the maximum l_j of all lower bounds is larger than the minimum of all upper bounds u_j for the same variable θ_j, the boundary constraint is replaced by $\theta_j := l_j := \min(u_j)$ defined by the minimum of all upper bounds specified for θ_j.

BY Statement

BY *variables* **;**

You can use a BY statement with PROC NLMIXED to obtain separate analyses on DATA= data set observations in groups defined by the BY variables. This means that, unless TECH=NONE, an optimization problem is solved for each BY group separately. When a BY statement appears, the procedure expects the input DATA= data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Use the SORT procedure with a similar BY statement to sort the data.
- Use the BY statement option NOTSORTED or DESCENDING in the BY statement for the NLMIXED procedure. As a cautionary note, the NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Use the DATASETS procedure (in Base SAS software) to create an index on the BY variables.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CONTRAST Statement

CONTRAST *'label' expression <, expression> <options>* **;**

The CONTRAST statement enables you to conduct a statistical test that several expressions simultaneously equal zero. The expressions are typically contrasts, that is, differences whose expected values equal zero under the hypothesis of interest.

In the CONTRAST statement you must provide a quoted string to identify the contrast and then a list of valid SAS expressions separated by commas. Multiple CONTRAST statements are permitted, and results from all statements are listed in a common table. PROC NLMIXED constructs approximate F tests for each statement using the delta method (Cox 1998) to approximate the variance-covariance matrix of the constituent expressions.

The following option is available in the CONTRAST statement.

DF=d

specifies the denominator degrees of freedom to be used in computing p values for the F statistics. The default value corresponds to the DF= option in the PROC NLMIXED statement.

ESTIMATE Statement

ESTIMATE *'label' expression <options>* **;**

The ESTIMATE statement enables you to compute an additional estimate that is a function of the parameter values. You must provide a quoted string to identify the estimate and then a valid SAS expression. Multiple ESTIMATE statements are permitted, and results from all statements are listed in a common table. PROC NLMIXED computes approximate standard errors for the estimates using the delta method (Billingsley 1986). It uses these standard errors to compute corresponding t statistics, *p*-values, and confidence limits.

The ECOV option in the PROC NLMIXED statement produces a table containing the approximate covariance matrix of all of the additional estimates you specify. The ECORR option produces the corresponding correlation matrix. The EDER option produces a table of the derivatives of the additional estimates with respect to each of the model parameters.

The following options are available in the ESTIMATE statement:

ALPHA=α

specifies the alpha level to be used in computing confidence limits. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DF=d

specifies the degrees of freedom to be used in computing *p*-values and confidence limits. The default value corresponds to the DF= option in the PROC NLMIXED statement.

ID Statement

ID *names* **;**

The ID statement identifies additional quantities to be included in the OUT= data set of the PREDICT statement. These can be any symbols you have defined with SAS programming statements.

MODEL Statement

MODEL *dependent-variable* $\sim$ *distribution* **;**

The MODEL statement is the mechanism for specifying the conditional distribution of the data given the random effects. You must specify a single dependent variable from the input data set, a tilde ($\sim$), and then a distribution with its parameters. Valid distributions are as follows.

- *normal(m,v)* specifies a normal (Gaussian) distribution with mean m and variance v.
- *binary(p)* specifies a binary (Bernoulli) distribution with probability p.
- *binomial(n,p)* specifies a binomial distribution with count n and probability p.
- *gamma(a,b)* specifies a gamma distribution with shape a and scale b.

- *negbin(n,p)* specifies a negative binomial distribution with count n and probability p.
- *poisson(m)* specifies a Poisson distribution with mean m.
- *general(ll)* specifies a general log likelihood function that you construct using SAS programming statements.

The MODEL statement must follow any SAS programming statements you specify for computing parameters of the preceding distributions.

PARMS Statement

PARMS *<name_list* **[**=*numbers*] **[**, *name_list* **[**=*numbers*] ... **]**>
</ options> **;**

The PARMS statement lists names of parameters and specifies initial values, possibly over a grid. You can specify the parameters and values either directly in a list or provide the name of a SAS data set that contains them using the DATA= option.

While the PARMS statement is not required, you are encouraged to use it to provide PROC NLMIXED with accurate starting values. Parameters not listed in the PARMS statement are assigned an initial value of 1. PROC NLMIXED considers all symbols not assigned values to be parameters, so you should specify your modeling statements carefully and check the output from the “Parameters” table to make sure the proper parameters are identified.

A list of parameter names in the PARMS statement is not separated by commas and is followed by an equal sign and a list of numbers. If the number list consists of only one number, this number defines the initial value for all the parameters listed to the left of the equal sign.

If the number list consists of more than one number, these numbers specify the grid locations for each of the parameters listed to the left of the equal sign. You can use the TO and BY keywords to specify a number list for a grid search. If you specify a grid of points in a PARMS statement, PROC NLMIXED computes the objective function value at each grid point and chooses the best (feasible) grid point as an initial point for the optimization process. You can use the BEST= option to save memory for the storing and sorting of all grid point information.

The following options are available in the PARMS statement after a slash (/):

BEST=$i > 0$
specifies the maximum number of points displayed in the “Parameters” table, selected as the points with the maximum likelihood values. By default, all grid values are displayed.

DATA=*SAS-data-set*
specifies a SAS data set containing parameter names and starting values. The data set should be in one of two forms: narrow or wide. The narrow-form data set contains the variables PARAMETER and ESTIMATE, with parameters and values listed as

distinct observations. The wide-form data set has the parameters themselves as variables, and each observation provides a different set of starting values. BY groups are ignored in this data set, so the same starting grid is evaluated for each BY group.

PREDICT Statement

PREDICT *expression OUT=SAS-data-set <options>* **;**

The PREDICT statement enables you to construct predictions of an expression across all of the observations in the input data set. Any valid SAS programming expression involving the input data set variables, parameters, and random effects is valid. Predicted values are computed using the parameter estimates and empirical Bayes estimates of the random effects. Standard errors of prediction are computed using the delta method (Billingsley 1986, Cox 1998). Results are placed in an output data set that you specify with the OUT= option. Besides all variables from the input data set, the OUT= data set contains the following variables: Pred, StdErrPred, DF, tValue, Probt, Alpha, Lower, Upper. You can also add other computed quantities to this data set with the ID statement.

The following options are available in the PREDICT statement:

ALPHA=α

specifies the alpha level to be used in computing t statistics and intervals. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DER

requests that derivatives of the predicted expression with respect to all parameters be included in the OUT= data set. The variable names for the derivatives are the same as the parameter names with the prefix "Der_" appended. All of the derivatives are evaluated at the final estimates of the parameters and the empirical Bayes estimates of the random effects.

DF=d

specifies the degrees of freedom to be used in computing t statistics and intervals in the OUT= data set. The default value corresponds to the DF= option in the PROC NLMIXED statement.

RANDOM Statement

RANDOM *random-effects* $\sim$ *distribution SUBJECT=variable <options>* **;**

The RANDOM statement defines the random effects and their distribution. The random effects must be represented by symbols that appear in your SAS programming statements. They typically influence the mean value of the distribution specified in the MODEL statement. The RANDOM statement consists of a list of the random effects (usually just one or two symbols), a tilde ($\sim$), the distribution for the random effects, and then a SUBJECT= variable.

NOTE: The input data set must be clustered according to the SUBJECT= variable. One easy way to accomplish this is to sort your data by the SUBJECT= variable prior to calling PROC NLMIXED. PROC NLMIXED does not sort the input data set for you; rather, it processes the data sequentially and considers an observation to be from a new subject whenever the value of its SUBJECT= changes from the previous observation.

The only distribution currently available for the random effects is normal(m,v) with mean m and variance v. This syntax is illustrated as follows for one effect:

```
random u ~ normal(0,s2u) subject=clinic;
```

For multiple effects, you should specify bracketed vectors for m and v, the latter consisting of the lower triangle of the random-effects variance matrix listed in row order. This is illustrated for two and three random effects as follows.

```
random b1 b2 ~ normal([0,0],[g11,g21,g22]) subject=person;
random b1 b2 b3 ~ normal([0,0,0],[g11,g21,g22,g31,g32,g33])
   subject=person;
```

The SUBJECT= variable determines when new realizations of the random effects are assumed to occur. PROC NLMIXED assumes that a new realization occurs whenever the SUBJECT= variable changes from the previous observation, so your input data set should be clustered according to this variable. One easy way to accomplish this is to run PROC SORT prior to calling PROC NLMIXED using the SUBJECT= variable as the BY variable.

Only one RANDOM statement is permitted, so multilevel nonlinear mixed models are not currently accommodated.

The following options are available in the RANDOM statement:

ALPHA=α

specifies the alpha level to be used in computing t statistics and intervals. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DF=d

specifies the degrees of freedom to be used in computing t statistics and intervals in the OUT= data set. The default value corresponds to the DF= option in the PROC NLMIXED statement.

OUT=*SAS-data-set*

requests an output data set containing empirical Bayes estimates of the random effects and their approximate standard errors of prediction.

REPLICATE Statement

REPLICATE *variable* **;**

The REPLICATE statement provides a way to accommodate models in which different subjects have identical data. This occurs most commonly when the dependent variable is binary. When you specify a REPLICATE variable, PROC NLMIXED assumes that its value indicates the number of subjects having data identical to those for the current value of the SUBJECT= variable (specified in the RANDOM statement). Only the last observation of the REPLICATE variable for each subject is used, and the replicate variable must have only positive integer values.

Programming Statements

This section lists the programming statements used to code the log likelihood function in PROC NLMIXED. It also documents the differences between programming statements in PROC NLMIXED and programming statements in the DATA step. The syntax of programming statements used in PROC NLMIXED is identical to that used in the CALIS and GENMOD procedures (see Chapter 19 and Chapter 31, respectively), and the MODEL procedure (refer to the *SAS/ETS User's Guide*). Most of the programming statements that can be used in the SAS DATA step can also be used in the NLMIXED procedure. Refer to *SAS Language Reference: Dictionary* for a description of SAS programming statements. The following are valid statements:

ABORT;
CALL *name [(expression [, expression ...])];*
DELETE;
DO *[variable* **=** *expression*
  [**TO** *expression] [* **BY** *expression]*
  [, expression [**TO** *expression] [* **BY** *expression] ...]*
  ]
  [**WHILE** *expression] [* **UNTIL** *expression];*
END;
GOTO *statement_label;*
IF *expression;*
IF *expression* **THEN** *program_statement;*
  ELSE *program_statement;*
variable **=** *expression;*
variable **+** *expression;*
LINK *statement_label;*
PUT *[variable] [=] [...] ;*
RETURN*;*
SELECT [(*expression* **)]***;*
STOP*;*
SUBSTR(*variable, index, length* **) =** *expression;*
WHEN (*expression***)** *program_statement;*
  OTHERWISE *program_statement;*

For the most part, the SAS programming statements work the same as they do in the SAS DATA step, as documented in *SAS Language Reference: Concepts*; however, there are several differences.

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus

  ```
  do i = 1,2,3;
  ```

  is supported; however, the following statement is not supported.

  ```
  do i = 'A','B','C';
  ```

- The LAG function does work appropriately with PROC NLMIXED, but you can use the ZLAG function instead.
- The PUT statement, used mostly for program debugging in PROC NLMIXED, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not.
  - The PROC NLMIXED PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or "$".
  - The PROC NLMIXED PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

    ```
    put (sqrt(x));
    ```
  - The PROC NLMIXED PUT statement supports the item _PDV_ to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the _ALL_ print item:

    ```
    put _pdv_;
    ```
- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple statement WHENs. For example, the following syntax is valid.

```
select;
when ( exp1 ) stmt1;
              stmt2;
when ( exp2 ) stmt3;
              stmt4;
end;
```

When coding your programming statements, you should avoid defining variables that begin with an underscore (_), as they may conflict with internal variables created by PROC NLMIXED. The MODEL statement must come after any SAS programming statements you specify for computing parameters of the modeling distribution.

Details

This section contains details about the underlying theory and computations of PROC NLMIXED.

Modeling Assumptions and Notation

PROC NLMIXED operates under the following general framework for nonlinear mixed models. Assume that you have an observed data vector y_i for each of i subjects, $i = 1, \ldots, s$. The y_i are assumed to be independent across i, but within-subject covariance is likely to exist because each of the elements of y_i are measured on the same subject. As a statistical mechanism for modeling this within-subject covariance, assume that there exist latent random-effect vectors u_i of small dimension (typically one or two) that are also independent across i. Assume also that an appropriate model linking y_i and u_i exists, leading to the joint probability density function

$$p(y_i|X_i, \phi, u_i)q(u_i|\xi)$$

where X_i is a matrix of observed explanatory variables and ϕ and ξ are vectors of unknown parameters.

Let $\theta = (\phi, \xi)$ and assume that it is of dimension n. Then inferences about θ are based on the marginal likelihood function

$$m(\theta) = \prod_{i=1}^{s} \int p(y_i|X_i, \phi, u_i)q(u_i|\xi)du_i$$

In particular, the function

$$f(\theta) = -\log m(\theta)$$

is minimized over θ numerically in order to estimate θ, and the inverse Hessian (second derivative) matrix at the estimates provides an approximate variance-covariance matrix for the estimate of θ. The function $f(\theta)$ is referred to both as the negative log likelihood function and as the objective function for optimization.

As an example of the preceding general framework, consider the nonlinear growth curve example in the “Getting Started” section. Here, the conditional distribution $p(y_i|X_i, \phi, u_i)$ is normal with mean

$$\frac{b_1 + u_{i1}}{1 + \exp[-(d_{ij} - b_2)/b_3]}$$

and variance σ_e^2; thus $\phi = (b_1, b_2, b_3, \sigma_e^2)$. Also, u_i is a scalar and $q(u_i|\xi)$ is normal with mean 0 and variance σ_u^2; thus $\xi = \sigma_u^2$.

The following additional notation is also found in this chapter. The quantity $\theta^{(k)}$ refers to the parameter vector at the kth iteration, the function $g(\theta)$ refers to the gradient vector $\nabla f(\theta)$, and the matrix $H(\theta)$ refers to the Hessian $\nabla^2 f(\theta)$. Other symbols are used to denote various constants or option values.

Integral Approximations

An important part of the marginal maximum likelihood method described previously is the computation of the integral over the random effects. The default method in PROC NLMIXED for computing this integral is adaptive Gaussian quadrature as described in Pinheiro and Bates (1995). Another approximation method is the first-order method of Beal and Sheiner (1982, 1988). A description of these two methods follows.

Adaptive Gaussian Quadrature

A quadrature method approximates a given integral by a weighted sum over predefined abscissas for the random effects. A good approximation can usually be obtained with an adequate number of quadrature points as well as appropriate centering and scaling of the abscissas. Adaptive Gaussian quadrature for the integral over u_i centers the integral at the empirical Bayes estimate of u_i, defined as the vector $\hat{u}_i$ that minimizes

$$-\log\left[p(y_i|X_i,\phi,u_i)q(u_i|\xi)\right]$$

with ϕ and ξ set equal to their current estimates. The final Hessian matrix from this optimization can be used to scale the quadrature abscissas.

Suppose $(z_j, w_j; j = 1,\ldots,p)$ denote the standard Gauss-Hermite abscissas and weights (Golub and Welsch 1969, or Table 25.10 of Abramowitz and Stegun 1972). The adaptive Gaussian quadrature integral approximation is as follows.

$$\int p(y_i|X_i,\phi,u_i)q(u_i|\xi)du_i \approx$$
$$2^{r/2}\left|\Gamma(X_i,\theta)\right|^{-1/2}\sum_{j_1=1}^{p}\cdots\sum_{j_r=1}^{p}\left[p(y_i|X_i,\phi,a_{j_1,\ldots,j_r})q(a_{j_1,\ldots,j_r}|\xi)\prod_{k=1}^{r}w_{j_k}\exp z_{j_k}^2\right]$$

where r is the dimension of u_i, $\Gamma(X_i,\theta)$ is the Hessian matrix from the empirical Bayes minimization, $z_{j_1,\ldots,j_r}$ is a vector with elements $(z_{j_1},\ldots,z_{j_r})$, and

$$a_{j_1,\ldots,j_r} = \hat{u}_i + 2^{1/2}\Gamma(X_i,\theta)^{-1/2}z_{j_1,\ldots,j_r}$$

PROC NLMIXED selects the number of quadrature points adaptively by evaluating the log likelihood function at the starting values of the parameters until two successive evaluations have a relative difference less than the value of the QTOL= option. The specific search sequence is described under the QFAC= option. Using the QPOINTS= option, you can adjust the number of quadrature points p to obtain different levels of accuracy. Setting $p = 1$ results in the Laplacian approximation as described in Beal and Sheiner (1992), Wolfinger (1993), Vonesh (1992, 1996), Vonesh and Chinchilli (1997), and Wolfinger and Lin (1997).

The NOAD option in the PROC NLMIXED statement requests nonadaptive Gaussian quadrature. Here all $\hat{u}_i$ are set equal to zero, and the Cholesky root of the estimated variance matrix of the random effects is substituted for $\Gamma(X_i,\theta)^{-1/2}$ in the preceding expression for $a_{j_1,\ldots,j_r}$. In this case derivatives are computed using the algorithm of Smith (1995). The NOADSCALE option requests the same scaling substitution but with the empirical Bayes $\hat{u}_i$.

PROC NLMIXED computes the derivatives of the adaptive Gaussian quadrature approximation when carrying out the default dual quasi-Newton optimization.

First-Order Method

Another integral approximation available in PROC NLMIXED is the first-order method of Beal and Sheiner (1982, 1988) and Sheiner and Beal (1985). This approximation is used only in the case where $p(y_i|X_i,\phi,u_i)$ is normal, that is,

$$\begin{aligned} p(y_i|X_i,\phi,u_i) = {} & (2\pi)^{-n_i/2}\,|R_i(X_i,\phi)|^{-1/2} \\ & \exp\left\{-(1/2)\,[y_i - m_i(X_i,\phi,u_i)]^T\,R_i(X_i,\phi)^{-1}\,[y_i - m_i(X_i,\phi,u_i)]\right\} \end{aligned}$$

where n_i is the dimension of y_i, R_i is a diagonal variance matrix, and m_i is the conditional mean vector of y_i.

The first-order approximation is obtained by expanding $m(X_i,\phi,u_i)$ with a one-term Taylor series expansion about $u_i = 0$, resulting in the approximation

$$\begin{aligned} p(y_i|X_i,\phi,u_i) \quad \approx \quad & (2\pi)^{-n_i/2}\,|R_i(X_i,\phi)|^{-1/2} \\ & \exp\Big(-(1/2)\,[y_i - m_i(X_i,\phi,0) - Z_i(X_i,\phi)u_i]^T\,R_i(X_i,\phi)^{-1} \\ & \quad [y_i - m_i(X_i,\phi,0) - Z_i(X_i,\phi)u_i]) \end{aligned}$$

where $Z_i(X_i,\phi)$ is the Jacobian matrix $\partial m_i(X_i,\phi,u_i)/\partial u_i$ evaluated at $u_i = 0$.

Assuming that $q(u_i|\xi)$ is normal with mean 0 and variance matrix $G(\xi)$, the first-order integral approximation is computable in closed form after completing the square:

$$\begin{aligned} \int p(y_i|X_i,\phi,u_i)q(u_i|\xi)du_i \approx {} & (2\pi)^{-n_i/2}\,|V_i(X_i,\theta)|^{-1/2} \\ & \exp\Big(-(1/2)\,[y_i - m_i(X_i,\phi,0)]^T\,V_i(X_i,\theta)^{-1}\,[y_i - m_i(X_i,\phi,0)]\Big) \end{aligned}$$

where $V_i(X_i,\theta) = Z_i(X_i,\phi)G(\xi)Z_i(X_i,\phi)^T + R_i(X_i,\phi)$. The resulting approximation for $f(\theta)$ is then minimized over $\theta = (\phi,\xi)$ to obtain the first-order estimates. PROC NLMIXED uses finite-difference derivatives of the first-order integral approximation when carrying out the default dual quasi-Newton optimization.

Optimization Algorithms

There are several optimization techniques available in PROC NLMIXED. You can choose a particular optimizer with the TECH=$name$ option in the PROC NLMIXED statement.

Algorithm	TECH=
trust region Method	TRUREG
Newton-Raphson method with line search	NEWRAP
Newton-Raphson method with ridging	NRRIDG
quasi-Newton methods (DBFGS, DDFP, BFGS, DFP)	QUANEW
double-dogleg method (DBFGS, DDFP)	DBLDOG
conjugate gradient methods (PB, FR, PR, CD)	CONGRA
Nelder-Mead simplex method	NMSIMP

No algorithm for optimizing general nonlinear functions exists that always finds the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC NLMIXED provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC NLMIXED can find the correct solution. Moreover, nonlinear optimization can be computationally expensive in terms of time and memory, so you must be careful when matching an algorithm to a problem.

All optimization techniques in PROC NLMIXED use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters. Since the techniques are iterative, they require the repeated computation of

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)

However, since each of the optimizers requires different derivatives, some computational efficiencies can be gained. The following table shows, for each optimization technique, which derivatives are required (FOD: first-order derivatives; SOD: second-order derivatives).

Algorithm	FOD	SOD
TRUREG	x	x
NEWRAP	x	x
NRRIDG	x	x
QUANEW	x	-
DBLDOG	x	-
CONGRA	x	-
NMSIMP	-	-

Each optimization method employs one or more convergence criteria that determine when it has converged. The various termination criteria are listed and described in the "PROC NLMIXED Statement" section. An algorithm is considered to have converged when any one of the convergence criterion is satisfied. For example, under the default settings, the QUANEW algorithm will converge if $ABSGCONV < 1\mathrm{E}-5$, $FCONV < 10^{-FDIGITS}$, or $GCONV < 1\mathrm{E}-8$.

Choosing an Optimization Algorithm

The factors that go into choosing a particular optimization technique for a particular problem are complex and may involve trial and error.

For many optimization problems, computing the gradient takes more computer time than computing the function value, and computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that do use a Hessian matrix, and as a result the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can more easily terminate at stationary points rather than at global optima.

A few general remarks about the various optimization techniques are as follows.

- The second-derivative methods TRUREG, NEWRAP, and NRRIDG are best for small problems where the Hessian matrix is not expensive to compute. Sometimes the NRRIDG algorithm can be faster than the TRUREG algorithm, but TRUREG can be more stable. The NRRIDG algorithm requires only one matrix with $n(n+1)/2$ double words; TRUREG and NEWRAP require two such matrices.
- The first-derivative methods QUANEW and DBLDOG are best for medium-sized problems where the objective function and the gradient are much faster to evaluate than the Hessian. The QUANEW and DBLDOG algorithms, in general, require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG algorithms require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP (essentially one matrix with $n(n+1)/2$ double words). QUANEW is the default optimization method.
- The first-derivative method CONGRA is best for large problems where the objective function and the gradient can be computed much faster than the Hessian and where too much memory is required to store the (approximate) Hessian. The CONGRA algorithm, in general, requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Since CONGRA requires only a factor of n double-word memory, many large applications of PROC NLMIXED can be solved only by CONGRA.
- The no-derivative method NMSIMP is best for small problems where derivatives are not continuous or are very difficult to compute.

Algorithm Descriptions

Some details about the optimization techniques are as follows.

Trust Region Optimization (TRUREG)

The trust region method uses the gradient $g(\theta_{(k)})$ and the Hessian matrix $H(\theta_{(k)})$; thus, it requires that the objective function $f(\theta)$ have continuous first- and second-order derivatives inside the feasible region.

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyper-elliptic trust region with radius Δ that constrains the step size corresponding to the quality of the quadratic approximation. The trust region method is implemented using Dennis, Gay, and Welsch (1981), Gay (1983), and Moré and Sorensen (1983).

The trust region method performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms may be more efficient.

Newton-Raphson Optimization with Line Search (NEWRAP)

The NEWRAP technique uses the gradient $g(\theta_{(k)})$ and the Hessian matrix $H(\theta_{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region. If second-order derivatives are computed efficiently and precisely, the NEWRAP method may perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation (LIS=2).

Newton-Raphson Ridge Optimization (NRRIDG)

The NRRIDG technique uses the gradient $g(\theta_{(k)})$ and the Hessian matrix $H(\theta_{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

The NRRIDG method performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms may be more efficient.

Since the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than that of the NEWRAP technique, which works with Cholesky decomposition. Usually, however, NRRIDG requires fewer iterations than NEWRAP.

Quasi-Newton Optimization (QUANEW)

The (dual) quasi-Newton method uses the gradient $g(\theta_{(k)})$, and it does not need to compute second-order derivatives since they are approximated. It works well for medium to moderately large optimization problems where the objective function and the gradient are much faster to compute than the Hessian; but, in general, it requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. QUANEW is the default optimization algorithm because it provides an appropriate balance between the speed and stability required for most nonlinear mixed model applications.

The QUANEW technique is one of the following, depending upon the value of the UPDATE= option.

- the original quasi-Newton algorithm, which updates an approximation of the inverse Hessian
- the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian (default)

You can specify four update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- BFGS performs the original BFGS update of the inverse Hessian matrix.
- DFP performs the original DFP update of the inverse Hessian matrix.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted with an identity matrix, resulting in the steepest descent or ascent search direction. You can specify line-search algorithms other than the default with the LIS= option.

The QUANEW algorithm performs its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome. You can use the INSTEP= option to impose an upper bound for the step size α during the first five iterations. You can also use the INHESSIAN[=r] option to specify a different starting approximation for the Hessian. If you specify only the INHESSIAN option, the Cholesky factor of a (possibly ridged) finite difference approximation of the Hessian is used to initialize the quasi-Newton update process. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the QUANEW algorithm.

Double Dogleg Optimization (DBLDOG)

The double dogleg optimization method combines the ideas of the quasi-Newton and trust region methods. In each iteration, the double dogleg algorithm computes the step $s^{(k)}$ as the linear combination of the steepest descent or ascent search direction $s_1^{(k)}$ and a quasi-Newton search direction $s_2^{(k)}$.

$$s^{(k)} = \alpha_1 s_1^{(k)} + \alpha_2 s_2^{(k)}$$

The step is requested to remain within a prespecified trust region radius; refer to Fletcher (1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search. You can specify two update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell update of the Cholesky factor of the Hessian matrix.

The double dogleg optimization technique works well for medium to moderately large optimization problems where the objective function and the gradient are much faster to compute than the Hessian. The implementation is based on Dennis and Mei (1979) and Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, or NRRIDG technique, which requires second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for the update of the Cholesky factor of an approximate Hessian.

Conjugate Gradient Optimization (CONGRA)

Second-order derivatives are not required by the CONGRA algorithm and are not even approximated. The CONGRA algorithm can be expensive in function and gradient calls, but it requires only $O(n)$ memory for unconstrained optimization. In general, many iterations are required to obtain a precise solution, but each of the CONGRA iterations is computationally cheap. You can specify four different update formulas for generating the conjugate directions by using the UPDATE= option:

- PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.

- FR performs the Fletcher-Reeves update (Fletcher 1987).
- PR performs the Polak-Ribiere update (Fletcher 1987).
- CD performs a conjugate-descent update of Fletcher (1987).

The default, UPDATE=PB, behaved best in most test examples. You are advised to avoid the option UPDATE=CD, which behaved worst in most test examples.

The CONGRA subroutine should be used for optimization problems with large n. For the unconstrained or boundary constrained case, CONGRA requires only $O(n)$ bytes of working memory, whereas all other optimization methods require order $O(n^2)$ bytes of working memory. During n successive iterations, uninterrupted by restarts or changes in the working set, the conjugate gradient algorithm computes a cycle of n conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size. Other line-search algorithms can be specified with the LIS= option.

Nelder-Mead Simplex Optimization (NMSIMP)

The Nelder-Mead simplex method does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it may be unable to generate precise results for $n \gg 40$.

The original Nelder-Mead simplex algorithm is implemented and extended to boundary constraints. This algorithm does not compute the objective for infeasible points, but it changes the shape of the simplex adapting to the nonlinearities of the objective function, which contributes to an increased speed of convergence. It uses a special termination criteria.

Finite Difference Approximations of Derivatives

The FD= and FDHESSIAN= options specify the use of finite difference approximations of the derivatives. The FD= option specifies that all derivatives are approximated using function evaluations, and the FDHESSIAN= option specifies that second-order derivatives are approximated using gradient evaluations.

Computing derivatives by finite difference approximations can be very time consuming, especially for second-order derivatives based only on values of the objective function (FD= option). If analytical derivatives are difficult to obtain (for example, if a function is computed by an iterative process), you might consider one of the optimization techniques that uses first-order derivatives only (QUANEW, DBLDOG, or CONGRA).

Forward Difference Approximations

The forward difference derivative approximations consume less computer time, but they are usually not as precise as approximations that use central difference formulas.

- For first-order derivatives, n additional function calls are required:

$$g_i = \frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta + h_i e_i) - f(\theta)}{h_i}$$

- For second-order derivatives based on function calls only (Dennis and Schnabel 1983, p. 80), $n + n^2/2$ additional function calls are required for dense Hessian:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{f(\theta + h_i e_i + h_j e_j) - f(\theta + h_i e_i) - f(\theta + h_j e_j) + f(\theta)}{h_i h_j}$$

- For second-order derivatives based on gradient calls (Dennis and Schnabel 1983, p. 103), n additional gradient calls are required:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{g_i(\theta + h_j e_j) - g_i(\theta)}{2h_j} + \frac{g_j(\theta + h_i e_i) - g_j(\theta)}{2h_i}$$

Central Difference Approximations

Central difference approximations are usually more precise, but they consume more computer time than approximations that use forward difference derivative formulas.

- For first-order derivatives, $2n$ additional function calls are required:

$$g_i = \frac{\partial f}{\partial \theta_i} \approx \frac{f(\theta + h_i e_i) - f(\theta - h_i e_i)}{2h_i}$$

- For second-order derivatives based on function calls only (Abramowitz and Stegun 1972, p. 884), $2n + 4n^2/2$ additional function calls are required.

$$\frac{\partial^2 f}{\partial \theta_i^2} \approx \frac{-f(\theta + 2h_i e_i) + 16f(\theta + h_i e_i) - 30f(\theta) + 16f(\theta - h_i e_i) - f(\theta - 2h_i e_i)}{12h_i^2}$$

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{f(\theta + h_i e_i + h_j e_j) - f(\theta + h_i e_i - h_j e_j) - f(\theta - h_i e_i + h_j e_j) + f(\theta - h_i e_i - h_j e_j)}{4h_i h_j}$$

- For second-order derivatives based on gradient calls, $2n$ additional gradient calls are required:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{g_i(\theta + h_j e_j) - g_i(\theta - h_j e_j)}{4h_j} + \frac{g_j(\theta + h_i e_i) - g_j(\theta - h_i e_i)}{4h_i}$$

You can use the FDIGITS== option to specify the number of accurate digits in the evaluation of the objective function. This specification is helpful in determining an appropriate interval size h to be used in the finite difference formulas.

The step sizes h_j, $j = 1, \ldots, n$ are defined as follows.

- For the forward difference approximation of first-order derivatives using function calls and second-order derivatives using gradient calls, $h_j = \sqrt[2]{\eta}(1 + |\theta_j|)$.

- For the forward difference approximation of second-order derivatives using only function calls and all central difference formulas, $h_j = \sqrt[3]{\eta}(1 + |\theta_j|)$.

The value of η is defined by the FDIGITS= option:

- If you specify the number of accurate digits using FDIGITS=r, η is set to 10^{-r}.
- If you do not specify the FDIGITS= option, η is set to the machine precision ϵ.

Hessian Scaling

The rows and columns of the Hessian matrix can be scaled when you use the trust region, Newton-Raphson, and double dogleg optimization techniques. Each element $H_{i,j}$, $i, j = 1, \ldots, n$ is divided by the scaling factor $d_i d_j$, where the scaling vector $d = (d_1, \ldots, d_n)$ is iteratively updated in a way specified by the HESCAL=i option, as follows.

$i = 0:$ No scaling is done (equivalent to $d_i = 1$).

$i \neq 0:$ First iteration and each restart iteration sets:

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

$i = 1:$ Refer to Moré (1978):

$$d_i^{(k+1)} = \max\left[d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

$i = 2:$ Refer to Dennis, Gay, and Welsch (1981):

$$d_i^{(k+1)} = \max\left[.6 d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

$i = 3:$ d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In the preceding equations, ϵ is the relative machine precision or, equivalently, the largest double precision value that, when added to 1, results in 1.

Active Set Methods

The parameter vector $\theta \in \mathcal{R}^n$ can be subject to a set of m linear equality and inequality constraints:

$$\sum_{j=1}^{n} a_{ij}\theta_j = b_i \qquad i = 1, \ldots, m_e$$

$$\sum_{j=1}^{n} a_{ij}\theta_j \geq b_i \qquad i = m_e + 1, \ldots, m$$

The coefficients a_{ij} and right-hand sides b_i of the equality and inequality constraints are collected in the $m \times n$ matrix A and the m vector b.

The m linear constraints define a feasible region $\mathcal{G}$ in $\mathcal{R}^n$ that must contain the point θ_* that minimizes the problem. If the feasible region $\mathcal{G}$ is empty, no solution to the optimization problem exists.

In PROC NLMIXED, all optimization techniques use *active set methods*. The iteration starts with a feasible point $\theta_{(0)}$, which you can provide or which can be computed by the Schittkowski and Stoer (1979) algorithm implemented in PROC NLMIXED. The algorithm then moves from one feasible point $\theta_{(k-1)}$ to a better feasible point $\theta_{(k)}$ along a feasible search direction $s^{(k)}$,

$$\theta_{(k)} = \theta_{(k-1)} + \alpha^{(k)} s^{(k)} \quad , \quad \alpha^{(k)} > 0$$

Theoretically, the path of points $\theta_{(k)}$ never leaves the feasible region $\mathcal{G}$ of the optimization problem, but it can reach its boundaries. The active set $\mathcal{A}^{(k)}$ of point $\theta_{(k)}$ is defined as the index set of all linear equality constraints and those inequality constraints that are satisfied at $\theta_{(k)}$. If no constraint is active $\theta_{(k)}$, the point is located in the interior of $\mathcal{G}$, and the active set $\mathcal{A}^{(k)} = \emptyset$ is empty. If the point $\theta_{(k)}$ in iteration k hits the boundary of inequality constraint i, this constraint i becomes active and is added to $\mathcal{A}^{(k)}$. Each equality constraint and each active inequality constraint reduce the dimension (degrees of freedom) of the optimization problem.

In practice, the active constraints can be satisfied only with finite precision. The LCEPSILON=r option specifies the range for active and violated linear constraints. If the point $\theta_{(k)}$ satisfies the condition

$$\left| \sum_{j=1}^{n} a_{ij} \theta_j^{(k)} - b_i \right| \leq t$$

where $t = r(|b_i|+1)$, the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. Due to rounding errors in computing the projected search direction, error can be accumulated so that an iterate $\theta_{(k)}$ steps out of the feasible region.

In those cases, PROC NLMIXED may try to pull the iterate $\theta_{(k)}$ back into the feasible region. However, in some cases the algorithm needs to increase the feasible region by increasing the LCEPSILON=r value. If this happens, a message is displayed in the log output.

If the algorithm cannot improve the value of the objective function by moving from an active constraint back into the interior of the feasible region, it makes this inequality constraint an equality constraint in the next iteration. This means that the active set $\mathcal{A}^{(k+1)}$ still contains the constraint i. Otherwise, it releases the active inequality constraint and increases the dimension of the optimization problem in the next iteration.

A serious numerical problem can arise when some of the active constraints become (nearly) linearly dependent. PROC NLMIXED removes linearly dependent equality constraints before starting optimization. You can use the LCSINGULAR= option to specify a criterion r used in the update of the QR decomposition that determines whether an active constraint is linearly dependent relative to a set of other active constraints.

If the solution θ_* is subjected to n_{act} linear equality or active inequality constraints, the QR decomposition of the $n \times n_{act}$ matrix $\hat{A}^T$ of the linear constraints is computed by $\hat{A}^T = QR$, where Q is an $n \times n$ orthogonal matrix and R is an $n \times n_{act}$ upper triangular matrix. The n columns of matrix Q can be separated into two matrices, $Q = [Y, Z]$, where Y contains the first n_{act} orthogonal columns of Q and Z contains the last $n - n_{act}$ orthogonal columns of Q. The $n \times (n - n_{act})$ column-orthogonal matrix Z is also called the *nullspace matrix* of the active linear constraints $\hat{A}^T$. The $n - n_{act}$ columns of the $n \times (n - n_{act})$ matrix Z form a basis orthogonal to the rows of the $n_{act} \times n$ matrix $\hat{A}$.

At the end of the iterating, PROC NLMIXED computes the *projected gradient* g_Z,

$$g_Z = Z^T g$$

In the case of boundary-constrained optimization, the elements of the projected gradient correspond to the gradient elements of the free parameters. A necessary condition for θ_* to be a local minimum of the optimization problem is

$$g_Z(\theta_*) = Z^T g(\theta_*) = 0$$

The symmetric $n_{act} \times n_{act}$ matrix G_Z,

$$G_Z = Z^T G Z$$

is called a *projected Hessian matrix*. A second-order necessary condition for θ_* to be a local minimizer requires that the projected Hessian matrix is positive semidefinite.

Those elements of the n_{act} vector of first-order estimates of *Lagrange multipliers*,

$$\lambda = (\hat{A}\hat{A}^T)^{-1}\hat{A}ZZ^T g$$

that correspond to active inequality constraints indicate whether an improvement of the objective function can be obtained by releasing this active constraint. For minimization, a significant negative Lagrange multiplier indicates that a possible reduction of the objective function can be achieved by releasing this active linear constraint. The LCDEACT=r option specifies a threshold r for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. (In the case of boundary-constrained optimization, the Lagrange multipliers for active lower (upper) constraints are the negative (positive) gradient elements corresponding to the active parameters.)

Line-Search Methods

In each iteration k, the (dual) quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of f along a feasible descent search direction $s^{(k)}$

$$\theta_{(k+1)} = \theta_{(k)} + \alpha^{(k)} s^{(k)}, \quad \alpha^{(k)} > 0$$

by computing an approximately optimal scalar $\alpha^{(k)}$.

Therefore, a line-search algorithm is an iterative process that optimizes a nonlinear function $f(\alpha)$ of one parameter (α) within each iteration k of the optimization technique. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (refer to Fletcher 1987).

You can select various line-search algorithms by specifying the LIS= option. The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default method for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

You can modify the line-search methods LIS=2 and LIS=3 to be exact line searches by using the LSPRECISION= option and specifying the σ parameter described in Fletcher (1987). The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand side and right-hand side Goldstein conditions (refer to Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead them to (feasible) points where the objective function f is no longer defined or difficult to compute. Therefore, PROC NLMIXED provides options restricting the step length α or trust region radius Δ, especially during the first main iterations.

The inner product $g^T s$ of the gradient g and the search direction s is the slope of $f(\alpha) = f(\theta + \alpha s)$ along the search direction s. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(\theta + \alpha s)$) during the main iteration k is computed in three steps:

1. The first step uses either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final step-size value α– of the last iteration $k - 1$ to compute a first value of $\alpha_1^{(0)}$.

- If the DAMPSTEPoption is not used,

$$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \le step \le 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

  with

$$step = \begin{cases} df/|g^T s| & \text{if } |g^T s| \ge \epsilon \max(100 df, 1) \\ 1 & \text{otherwise} \end{cases}$$

  This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

- If the DAMPSTEP[=r] option is used,

$$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

  The initial value for the new step length can be no larger than r times the final step length $\alpha-$ of the former iteration. The default value is $r = 2$.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ using the INSTEP=r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

   After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

   where u is the maximum length of a step inside the feasible region.

The INSTEP=r option enables you to specify a smaller or larger radius Δ of the trust region used in the first iteration of the trust region and double dogleg algorithms. The default initial trust region radius $\Delta^{(0)}$ is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. In most practical applications of the TRUREG and DBLDOG algorithms, this choice is successful. However, for bad initial values and highly nonlinear objective functions (such as the EXP function), the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r, $0 < r < 1$, until the iteration starts successfully. A small factor r also affects the trust region radius $\Delta^{(k+1)}$ of the next steps because the radius is changed in each iteration by a factor $0 < c \le 4$, depending on the ratio ρ expressing the goodness of quadratic function approximation. Reducing the radius Δ corresponds to increasing the ridge parameter λ, producing smaller steps directed more closely toward the (negative) gradient direction.

Computational Problems

Floating Point Errors and Overflows

Numerical optimization of a numerically integrated function is a difficult task, and the computation of the objective function and its derivatives can lead to arithmetic exceptions and overflows. A typical cause of these problems is parameters with widely varying scales. If the scaling of your parameters varies by more than a few orders of magnitude, the numerical stability of the optimization problem can be seriously reduced and result in computational difficulties. A simple remedy is to rescale each parameter so that its final estimated value has a magnitude near 1.

If parameter rescaling does not help, consider the following actions:

- Specify the ITDETAILS option in the PROC NLMIXED statement to obtain more detailed information about when and where the problem is occurring.
- Provide different initial values or try a grid search of values.
- Use boundary constraints to avoid the region where overflows may happen.
- Delete outlying observations or subjects from the input data, if this is reasonable.
- Change the algorithm (specified in programming statements) that computes the objective function.

The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If an overflow occurs during a line search, you can use the INSTEP= option to reduce the length of the first trial step during the first five iterations, or you can use the DAMPSTEP or MAXSTEP option to restrict the step length of the initial α in subsequent iterations. If an arithmetic overflow occurs in the first iteration of the trust region or double dogleg algorithms, you can use the INSTEP= option to reduce the default trust region radius of the first iteration. You can also change the optimization technique or the line-search method.

Long Run Times

PROC NLMIXED can take a long time to run for problems with complex models, many parameters, or large input data sets. Although the optimization techniques used by PROC NLMIXED are some of the best ones available, they are not guaranteed to converge quickly for all problems. Ill-posed or misspecified models can cause the algorithms to use more extensive calculations designed to achieve convergence, and this can result in longer run times. So first make sure that your model is specified correctly, that your parameters are scaled to be of the same order of magnitude, and that your data reasonably match the model you are contemplating.

If you are using the default adaptive Gaussian quadrature algorithm and no iteration history is printing at all, then PROC NLMIXED may be bogged down trying to determine the number of quadrature points at the first set of starting values. Specifying the QPOINTS= option will bypass this stage and proceed directly to iterations; however, be aware that the likelihood approximation may not be accurate if there are too few quadrature points.

PROC NLMIXED may also have difficulty determining the number of quadrature points if the initial starting values are far from the optimum values. To obtain more accurate starting values for the model parameters, one easy method is to fit a model with no RANDOM statement. You can then use these estimates as starting values, although you will still need to specify values for the random-effects distribution. For normal-normal models, another strategy is to use METHOD=FIRO. If you can obtain estimates using this approximate method, then they can be used as starting values for more accurate likelihood approximations.

If you are running PROC NLMIXED multiple times, you will probably want to include a statement like the following in your program:

```
ods output ParameterEstimates=pe;
```

This statement creates a SAS data set named PE upon completion of the run. In your next invocation of PROC NLMIXED, you can then specify

```
parms / data=pe;
```

to read in the previous estimates as starting values.

To speed general computations, you should check over your programming statements to minimize the number of floating point operations. Using auxiliary variables and factoring amenable expressions can be useful changes in this regard.

Problems Evaluating Code for Objective Function

The starting point $\theta^{(0)}$ must be a point for which the programming statements can be evaluated. However, during optimization, the optimizer may iterate to a point $\theta^{(k)}$ where the objective function or its derivatives cannot be evaluated. In some cases, the specification of boundary for parameters can avoid such situations. In many other cases, you can indicate that the point $\theta^{(0)}$ is a bad point simply by returning an extremely large value for the objective function. In these cases, the optimization algorithm reduces the step length and stays closer to the point that has been evaluated successfully in the former iteration.

No Convergence

There are a number of things to try if the optimizer fails to converge.

- Change the initial values by using a grid search specification to obtain a set of good feasible starting values.
- Change or modify the update technique or the line-search algorithm.

  This method applies only to TECH=QUANEW and TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can

  - change the update formula with the UPDATE= option

  - change the line-search algorithm with the LIS= option
  - specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3

- Change the optimization technique.

  For example, if you use the default option, TECH=QUANEW, you can try one of the second-derivative methods if your problem is small or the conjugate gradient method if it is large.

- Adjust finite difference derivatives.

  The forward difference derivatives specified with the FD[=] or FDHESSIAN[=] option may not be precise enough to satisfy strong gradient termination criteria. You may need to specify the more expensive central difference formulas. The finite difference intervals may be too small or too big, and the finite difference derivatives may be erroneous.

- Double-check the data entry and program specification.

Convergence to Stationary Point

The gradient at a stationary point is the null vector, which always leads to a zero search direction. This point satisfies the first-order termination criterion. Search directions that are based on the gradient are zero, so the algorithm terminates. There are two ways to avoid this situation:

- Use the PARMS statement to specify a grid of feasible initial points.
- Use the OPTCHECK[=r] option to avoid terminating at the stationary point.

The signs of the eigenvalues of the (reduced) Hessian matrix contain information regarding a stationary point.

- If all of the eigenvalues are positive, the Hessian matrix is positive definite, and the point is a minimum point.
- If some of the eigenvalues are positive and all remaining eigenvalues are zero, the Hessian matrix is positive semidefinite, and the point is a minimum or saddle point.
- If all of the eigenvalues are negative, the Hessian matrix is negative definite, and the point is a maximum point.
- If some of the eigenvalues are negative and all of the remaining eigenvalues are zero, the Hessian matrix is negative semidefinite, and the point is a maximum or saddle point.
- If all of the eigenvalues are zero, the point can be a minimum, maximum, or saddle point.

Precision of Solution

In some applications, PROC NLMIXED may result in parameter values that are not precise enough. Usually, this means that the procedure terminated at a point too far from the optimal point. The termination criteria define the size of the termination region around the optimal point. Any point inside this region can be accepted for terminating the optimization process. The default values of the termination criteria are set to satisfy a reasonable compromise between the computational effort (computer time) and the precision of the computed estimates for the most common applications. However, there are a number of circumstances in which the default values of the termination criteria specify a region that is either too large or too small.

If the termination region is too large, then it can contain points with low precision. In such cases, you should determine which termination criterion stopped the optimization process. In many applications, you can obtain a solution with higher precision simply by using the old parameter estimates as starting values in a subsequent run in which you specify a smaller value for the termination criterion that was satisfied at the former run.

If the termination region is too small, the optimization process may take longer to find a point inside such a region, or it may not even find such a point due to rounding errors in function values and derivatives. This can easily happen in applications in which finite difference approximations of derivatives are used and the GCONV and ABSGCONV termination criteria are too small to respect rounding errors in the gradient values.

Covariance Matrix

The estimated covariance matrix of the parameter estimates is computed as the inverse Hessian matrix, and for unconstrained problems it should be positive definite. If the final parameter estimates are subjected to $n_{act} > 0$ active linear inequality constraints, the formulas of the covariance matrices are modified similar to Gallant (1987) and Cramer (1986, p. 38) and additionally generalized for applications with singular matrices.

There are several steps available that enable you to tune the rank calculations of the covariance matrix.

1. You can use the ASINGULAR=, MSINGULAR=, and VSINGULAR= options to set three singularity criteria for the inversion of the Hessian matrix H. The singularity criterion used for the inversion is

   $$|d_{j,j}| \leq \max(\text{ASING}, \text{VSING} * |H_{j,j}|, \text{MSING} * max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

   where $d_{j,j}$ is the diagonal pivot of the matrix H, and ASING, VSING, and MSING are the specified values of the ASINGULAR=, VSINGULAR=, and MSINGULAR= options. The default values are

   - ASING: the square root of the smallest positive double precision value
   - MSING: $1\text{E} - 12$ if you do not specify the SINGHESS= option and $\max(10\epsilon, \text{1E-4}*\text{SINGHESS})$ otherwise, where ϵ is the machine precision

- VSING: $1\text{E} - 8$ if you do not specify the SINGHESS= option and the value of SINGHESS otherwise

Note that, in many cases, a normalized matrix $D^{-1}AD^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

2. If the matrix H is found to be singular in the first step, a generalized inverse is computed. Depending on the G4= option, either a generalized inverse satisfying all four Moore-Penrose conditions is computed or a generalized inverse satisfying only two Moore-Penrose conditions is computed. If the number of parameters n of the application is less than or equal to G4=i, a G4 inverse is computed; otherwise, only a G2 inverse is computed. The G4 inverse is computed by the (computationally very expensive but numerically stable) eigenvalue decomposition, and the G2 inverse is computed by Gauss transformation. The G4 inverse is computed using the eigenvalue decomposition $A = Z\Lambda Z^T$, where Z is the orthogonal matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues, $\Lambda = diag(\lambda_1, ..., \lambda_n)$. The G4 inverse of H is set to

$$A^- = Z\Lambda^- Z^T$$

where the diagonal matrix $\Lambda^- = diag(\lambda_1^-, ..., \lambda_n^-)$ is defined using the COVSING= option.

$$\lambda_i^- = \begin{cases} 1/\lambda_i & \text{if } |\lambda_i| > \text{COVSING} \\ 0 & \text{if } |\lambda_i| \le \text{COVSING} \end{cases}$$

If you do not specify the COVSING= option, the nr smallest eigenvalues are set to zero, where nr is the number of rank deficiencies found in the first step.

For optimization techniques that do not use second-order derivatives, the covariance matrix is computed using finite difference approximations of the derivatives.

Prediction

The nonlinear mixed model is a useful tool for statistical prediction. Assuming a prediction is to be made regarding the ith subject, suppose that $f(\theta, u_i)$ is a differentiable function predicting some quantity of interest. Recall that θ denotes the vector of unknown parameters and u_i denotes the vector of random effects for the ith subject. A natural point prediction is $f(\hat{\theta}, \hat{u}_i)$, where $\hat{\theta}$ is the maximum likelihood estimate of θ and $\hat{u}_i$ is the empirical Bayes estimate of u_i described previously in “Integral Approximations.”

An approximate prediction variance matrix for $(\hat{\theta}, \hat{u}_i)$ is

$$P = \begin{bmatrix} \hat{H}^{-1} & \hat{H}^{-1}\left(\frac{\partial \hat{u}_i}{\partial \theta}\right)^T \\ \left(\frac{\partial \hat{u}_i}{\partial \theta}\right)\hat{H}^{-1} & \hat{\Gamma}^{-1} + \left(\frac{\partial \hat{u}_i}{\partial \theta}\right)\hat{H}^{-1}\left(\frac{\partial \hat{u}_i}{\partial \theta}\right)^T \end{bmatrix}$$

where $\hat{H}$ is the approximate Hessian matrix from the optimization for $\hat{\theta}$, $\hat{\Gamma}$ is the approximate Hessian matrix from the optimization for $\hat{u}_i$, and $(\partial \hat{u}_i / \partial \theta)$ is the derivative of $\hat{u}_i$ with respect to θ, evaluated at $(\hat{\theta}, \hat{u}_i)$. The approximate variance matrix for $\hat{\theta}$ is the standard one discussed in the previous section, and that for $\hat{u}_i$ is an approximation to the conditional mean squared error of prediction described by Booth and Hobert (1998).

The prediction variance for a general scalar function $f(\theta, u_i)$ is defined as the expected squared difference $E[f(\hat{\theta}, \hat{u}_i) - f(\theta, u_i)]^2$. PROC NLMIXED computes an approximation to it as follows. The derivative of $f(\theta, u_i)$ is computed with respect to each element of (θ, u_i) and evaluated at $(\hat{\theta}, \hat{u}_i)$. If a_i is the resulting vector, then the approximate prediction variance is $a_i^T P a_i$. This approximation is known as the delta method (Billingsley, 1986; Cox, 1998).

Computational Resources

Since nonlinear optimization is an iterative process that depends on many factors, it is difficult to estimate how much computer time is necessary to find an optimal solution satisfying one of the termination criteria. You can use the MAXTIME=, MAXITER=, and MAXFU= options to restrict the amount of CPU time, the number of iterations, and the number of function calls in a single run of PROC NLMIXED.

In each iteration k, the NRRIDG technique uses a symmetric Householder transformation to decompose the $n \times n$ Hessian matrix H

$$H = V'TV, \qquad V\text{: orthogonal}, \quad T\text{: tridiagonal}$$

to compute the (Newton) search direction s

$$s^{(k)} = -[H^{(k)}]^{-1} g^{(k)} \qquad k = 1, 2, 3, \ldots$$

The TRUREG and NEWRAP techniques use the Cholesky decomposition to solve the same linear system while computing the search direction. The QUANEW, DBLDOG, CONGRA, and NMSIMP techniques do not need to invert or decompose a Hessian matrix; thus, they require less computational resources than the other techniques.

The larger the problem, the more time is needed to compute function values and derivatives. Therefore, you may want to compare optimization techniques by counting and comparing the respective numbers of function, gradient, and Hessian evaluations.

Finite difference approximations of the derivatives are expensive because they require additional function or gradient calls:

- forward difference formulas
  - For first-order derivatives, n additional function calls are required.
  - For second-order derivatives based on function calls only, for a dense Hessian, $n + n^2/2$ additional function calls are required.
  - For second-order derivatives based on gradient calls, n additional gradient calls are required.

- central difference formulas
  - For first-order derivatives, $2n$ additional function calls are required.
  - For second-order derivatives based on function calls only, for a dense Hessian, $2n + 2n^2$ additional function calls are required.
  - For second-order derivatives based on gradient calls, $2n$ additional gradient calls are required.

Many applications need considerably more time for computing second-order derivatives (Hessian matrix) than for computing first-order derivatives (gradient). In such cases, a dual quasi-Newton technique is recommended, which does not require second-order derivatives.

Displayed Output

This section describes the displayed output from PROC NLMIXED. See the section "ODS Table Names" on page 3107 for details about how this output interfaces with the Output Delivery System.

Specifications

The NLMIXED procedure first displays the "Specifications" table, listing basic information about the nonlinear mixed model that you have specified. It includes the principal variables and estimation methods.

Dimensions

The "Dimensions" table lists counts of important quantities in your nonlinear mixed model, including the number of observations, subjects, parameters, and quadrature points.

Parameters

The "Parameters" table displays the information you provided with the PARMS statement and the value of the negative log likelihood function evaluated at the starting values.

Starting Gradient and Hessian

The START option in the PROC NLMIXED statement displays the gradient of the negative log likelihood function at the starting values of the parameters. If you also specify the HESS option, then the starting Hessian is displayed as well.

Iterations

The iteration history consists of one line of output for each iteration in the optimization process. The iteration history is displayed by default because it is important that you check for possible convergence problems. The default iteration history includes the following variables:

- Iter, the iteration number
- Calls, the number of function calls

- NegLogLike, the value of the objective function
- Diff, the difference between adjacent function values
- MaxGrad, the maximum of the absolute (projected) gradient components (except NMSIMP)
- Slope, the slope $g^T s$ of the search direction s at the current parameter iterate $\theta_{(k)}$ (QUANEW only)
- Rho, the ratio between the achieved and predicted value of Diff (NRRIDG only)
- Radius, the radius of the trust region (TRUREG only)
- StdDev, the standard deviation of the simplex values (NMSIMP only)
- Delta, the vertex length of the simplex (NMSIMP only)
- Size, the size of the simplex (NMSIMP only)

For the QUANEW method, the value of Slope should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently. If this difficulty is encountered, an asterisk (*) appears after the iteration number. If there is a tilde ($\sim$) after the iteration number, the BFGS update is skipped, and very high values of the Lagrange function are produced. A backslash ($\backslash$) after the iteration number indicates that Powell's correction for the BFGS update is used.

For methods using second derivatives, an asterisk (*) after the iteration number means that the computed Hessian approximation was singular and had to be ridged with a positive value.

For the NMSIMP method, only one line is displayed for several internal iterations. This technique skips the output for some iterations because some of the termination tests (StdDev and Size) are rather time consuming compared to the simplex operations, and they are performed only every five simplex operations.

The ITDETAILS option in the PROC NLMIXED statement provides a more detailed iteration history. Besides listing the current values of the parameters and their gradients, the following values are provided in addition to the default output:

- Restart, the number of iteration restarts
- Active, the number of active constraints
- Lambda, the value of the Lagrange multiplier (TRUREG and DBLDOG only)
- Ridge, the ridge value (NRRIDG only)
- Alpha, the line-search step size (QUANEW only)

An apostrophe (') trailing the number of active constraints indicates that at least one of the active constraints was released from the active set due to a significant Lagrange multiplier.

Fitting Information

The “Fitting Information” table lists the final minimized value of -2 times the log likelihood as well as the information criteria of Akaike (AIC) and Schwarz (BIC), as well as a finite-sample corrected version of AIC (AICC). The criteria are computed as follows:

$$\begin{aligned} AIC &= 2f(\hat{\theta}) + 2p \\ AICC &= 2f(\hat{\theta}) + 2pn/(n-p-1) \\ BIC &= 2f(\hat{\theta}) + p\log(s) \end{aligned}$$

where $f()$ is the negative of the marginal log likelihood function, $\hat{\theta}$ is the vector of parameter estimates, p is the number of parameters, n is the number of observations, and s is the number of subjects. Refer to Hurvich and Tsai (1989) and Burnham and Anderson (1998) for additional details.

Parameter Estimates

The “Parameter Estimates” table lists the estimates of the parameter values after successful convergence of the optimization problem or the final values of the parameters under nonconvergence. If the problem did converge, standard errors are computed from the final Hessian matrix. The ratio of the estimate with its standard error produces a t value, with approximate degrees of freedom computed as the number of subjects minus the number of random effects. A p-value and confidence limits based on this t distribution are also provided. Finally, the gradient of the negative log likelihood function is displayed for each parameter, and you should verify that they each are sufficiently small for non-constrained parameters.

Covariance and Correlation Matrices

Following standard maximum likelihood theory (for example, Serfling 1980), the asymptotic variance-covariance matrix of the parameter estimates equals the inverse of the Hessian matrix. You can display this matrix with the COV option in the PROC NLMIXED statement. The corresponding correlation form is available with the CORR option.

Additional Estimates

The “Additional Estimates” table displays the results of all ESTIMATE statements that you specify, with the same columns as the “Parameter Estimates” table. The ECOV and ECORR options in the PROC NLMIXED statement produce tables displaying the approximate covariance and correlation matrices of the additional estimates. They are computed using the delta method (Billingsley 1986; Cox 1998). The EDER option in the PROC NLMIXED statement produces a table displaying the derivatives of the additional estimates with respect to the model parameters evaluated at their final estimated values.

ODS Table Names

PROC NLMIXED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 51.2. ODS Tables Produced in PROC NLMIXED

ODS Table Name	Description	Statement or Option
AdditionalEstimates	Results from ESTIMATE statements	ESTIMATE
ConvergenceStatus	Convergence status	default
CorrMatAddEst	Correlation matrix of additional estimates	ECORR
CorrMatParmEst	Correlation matrix of parameter estimates	CORR
CovMatAddEst	Covariance matrix of additional estimates	ECOV
CovMatParmEst	Covariance matrix of parameter estimates	COV
DerAddEst	Derivatives of additional estimates	EDER
Dimensions	Dimensions of the problem	default
FitStatistics	Fit statistics	default
Hessian	Second derivative matrix	HESS
IterHistory	Iteration history	default
Parameters	Parameters	default
ParameterEstimates	Parameter estimates	default
Specifications	Model specifications	default
StartingHessian	Starting hessian matrix	START HESS
StartingValues	Starting values and gradient	START

Examples

Example 51.1. One-Compartment Model with Pharmacokinetic Data

A popular application of nonlinear mixed models is in the field of pharmacokinetics, which studies how a drug disperses through a living individual. This example considers the theophylline data from Pinheiro and Bates (1995). Serum concentrations of the drug theophylline are measured in 12 subjects over a 25-hour period after oral administration. The data are as follows.

```
data theoph;
   input subject time conc dose wt;
   datalines;
1  0.00  0.74 4.02 79.6
1  0.25  2.84 4.02 79.6
1  0.57  6.57 4.02 79.6
1  1.12 10.50 4.02 79.6
1  2.02  9.66 4.02 79.6
1  3.82  8.58 4.02 79.6
1  5.10  8.36 4.02 79.6
1  7.03  7.47 4.02 79.6
```

```
1  9.05  6.89 4.02 79.6
1 12.12  5.94 4.02 79.6
1 24.37  3.28 4.02 79.6
2  0.00  0.00 4.40 72.4
2  0.27  1.72 4.40 72.4
2  0.52  7.91 4.40 72.4
2  1.00  8.31 4.40 72.4
2  1.92  8.33 4.40 72.4
2  3.50  6.85 4.40 72.4
2  5.02  6.08 4.40 72.4
2  7.03  5.40 4.40 72.4
2  9.00  4.55 4.40 72.4
2 12.00  3.01 4.40 72.4
2 24.30  0.90 4.40 72.4
3  0.00  0.00 4.53 70.5
3  0.27  4.40 4.53 70.5
3  0.58  6.90 4.53 70.5
3  1.02  8.20 4.53 70.5
3  2.02  7.80 4.53 70.5
3  3.62  7.50 4.53 70.5
3  5.08  6.20 4.53 70.5
3  7.07  5.30 4.53 70.5
3  9.00  4.90 4.53 70.5
3 12.15  3.70 4.53 70.5
3 24.17  1.05 4.53 70.5
4  0.00  0.00 4.40 72.7
4  0.35  1.89 4.40 72.7
4  0.60  4.60 4.40 72.7
4  1.07  8.60 4.40 72.7
4  2.13  8.38 4.40 72.7
4  3.50  7.54 4.40 72.7
4  5.02  6.88 4.40 72.7
4  7.02  5.78 4.40 72.7
4  9.02  5.33 4.40 72.7
4 11.98  4.19 4.40 72.7
4 24.65  1.15 4.40 72.7
5  0.00  0.00 5.86 54.6
5  0.30  2.02 5.86 54.6
5  0.52  5.63 5.86 54.6
5  1.00 11.40 5.86 54.6
5  2.02  9.33 5.86 54.6
5  3.50  8.74 5.86 54.6
5  5.02  7.56 5.86 54.6
5  7.02  7.09 5.86 54.6
5  9.10  5.90 5.86 54.6
5 12.00  4.37 5.86 54.6
5 24.35  1.57 5.86 54.6
6  0.00  0.00 4.00 80.0
6  0.27  1.29 4.00 80.0
6  0.58  3.08 4.00 80.0
6  1.15  6.44 4.00 80.0
6  2.03  6.32 4.00 80.0
6  3.57  5.53 4.00 80.0
6  5.00  4.94 4.00 80.0
```

```
 6  7.00  4.02 4.00 80.0
 6  9.22  3.46 4.00 80.0
 6 12.10  2.78 4.00 80.0
 6 23.85  0.92 4.00 80.0
 7  0.00  0.15 4.95 64.6
 7  0.25  0.85 4.95 64.6
 7  0.50  2.35 4.95 64.6
 7  1.02  5.02 4.95 64.6
 7  2.02  6.58 4.95 64.6
 7  3.48  7.09 4.95 64.6
 7  5.00  6.66 4.95 64.6
 7  6.98  5.25 4.95 64.6
 7  9.00  4.39 4.95 64.6
 7 12.05  3.53 4.95 64.6
 7 24.22  1.15 4.95 64.6
 8  0.00  0.00 4.53 70.5
 8  0.25  3.05 4.53 70.5
 8  0.52  3.05 4.53 70.5
 8  0.98  7.31 4.53 70.5
 8  2.02  7.56 4.53 70.5
 8  3.53  6.59 4.53 70.5
 8  5.05  5.88 4.53 70.5
 8  7.15  4.73 4.53 70.5
 8  9.07  4.57 4.53 70.5
 8 12.10  3.00 4.53 70.5
 8 24.12  1.25 4.53 70.5
 9  0.00  0.00 3.10 86.4
 9  0.30  7.37 3.10 86.4
 9  0.63  9.03 3.10 86.4
 9  1.05  7.14 3.10 86.4
 9  2.02  6.33 3.10 86.4
 9  3.53  5.66 3.10 86.4
 9  5.02  5.67 3.10 86.4
 9  7.17  4.24 3.10 86.4
 9  8.80  4.11 3.10 86.4
 9 11.60  3.16 3.10 86.4
 9 24.43  1.12 3.10 86.4
10  0.00  0.24 5.50 58.2
10  0.37  2.89 5.50 58.2
10  0.77  5.22 5.50 58.2
10  1.02  6.41 5.50 58.2
10  2.05  7.83 5.50 58.2
10  3.55 10.21 5.50 58.2
10  5.05  9.18 5.50 58.2
10  7.08  8.02 5.50 58.2
10  9.38  7.14 5.50 58.2
10 12.10  5.68 5.50 58.2
10 23.70  2.42 5.50 58.2
11  0.00  0.00 4.92 65.0
11  0.25  4.86 4.92 65.0
11  0.50  7.24 4.92 65.0
11  0.98  8.00 4.92 65.0
11  1.98  6.81 4.92 65.0
11  3.60  5.87 4.92 65.0
```

```
11  5.02  5.22 4.92 65.0
11  7.03  4.45 4.92 65.0
11  9.03  3.62 4.92 65.0
11 12.12  2.69 4.92 65.0
11 24.08  0.86 4.92 65.0
12  0.00  0.00 5.30 60.5
12  0.25  1.25 5.30 60.5
12  0.50  3.96 5.30 60.5
12  1.00  7.82 5.30 60.5
12  2.00  9.72 5.30 60.5
12  3.52  9.75 5.30 60.5
12  5.07  8.57 5.30 60.5
12  7.07  6.59 5.30 60.5
12  9.03  6.11 5.30 60.5
12 12.05  4.57 5.30 60.5
12 24.15  1.17 5.30 60.5
run;
```

Pinheiro and Bates (1995) consider the following first-order compartment model for these data:

$$C_{it} = \frac{Dk_{e_i}k_{a_i}}{Cl_i(k_{a_i} - k_{e_i})}[\exp(-k_{e_i}t) - \exp(-k_{a_i}t)] + e_{it}$$

where C_{it} is the observed concentration of the ith subject at time t, D is the dose of theophylline, k_{e_i} is the elimination rate constant for subject i, k_{a_i} is the absorption rate constant for subject i, Cl_i is the clearance for subject i, and e_{it} are normal errors. To allow for random variability between subjects, they assume

$$\begin{aligned} Cl_i &= \exp(\beta_1 + b_{i1}) \\ k_{a_i} &= \exp(\beta_2 + b_{i2}) \\ k_{e_i} &= \exp(\beta_3) \end{aligned}$$

where the βs denote fixed-effects parameters and the b_is denote random-effects parameters with an unknown covariance matrix.

The PROC NLMIXED statements to fit this model are as follows.

```
proc nlmixed data=theoph;
   parms beta1=-3.22 beta2=0.47 beta3=-2.45
      s2b1=0.03 cb12=0 s2b2=0.4 s2=0.5;
   cl = exp(beta1 + b1);
   ka = exp(beta2 + b2);
   ke = exp(beta3);
   pred = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc ~ normal(pred,s2);
   random b1 b2 ~ normal([0,0],[s2b1,cb12,s2b2]) subject=subject;
run;
```

The PARMS statement specifies starting values for the three βs and four variance-covariance parameters. The clearance and rate constants are defined using SAS programming statements, and the conditional model for the data is defined to be normal with mean PRED and variance S2. The two random effects are B1 and B2, and their joint distribution is defined in the RANDOM statement. Brackets are used in defining their mean vector (two zeroes) and the lower triangle of their variance-covariance matrix (a general 2 × 2 matrix). The SUBJECT= variable is SUBJECT.

The results from this analysis are as follows.

```
                              The NLMIXED Procedure

                                  Specifications

        Data Set                                    WORK.THEOPH
        Dependent Variable                          conc
        Distribution for Dependent Variable         Normal
        Random Effects                              b1 b2
        Distribution for Random Effects             Normal
        Subject Variable                            subject
        Optimization Technique                      Dual Quasi-Newton
        Integration Method                          Adaptive Gaussian
                                                    Quadrature
```

The “Specifications” table lists the set up of the model.

```
                              The NLMIXED Procedure

                                    Dimensions

                    Observations Used                    132
                    Observations Not Used                  0
                    Total Observations                   132
                    Subjects                              12
                    Max Obs Per Subject                   11
                    Parameters                             7
                    Quadrature Points                      5
```

The “Dimensions” table indicates that there are 132 observations, 12 subjects, and 7 parameters. PROC NLMIXED selects 5 quadrature points for each random effect, producing a total grid of 25 points over which quadrature is performed.

```
                              The NLMIXED Procedure

                                    Parameters

   beta1      beta2      beta3       s2b1       cb12       s2b2         s2  NegLogLike

   -3.22       0.47      -2.45       0.03          0        0.4        0.5  177.789945
```

The “Parameters” table lists the 7 parameters, their starting values, and the initial evaluation of the negative log likelihood using adaptive Gaussian quadrature.

```
                              The NLMIXED Procedure

                                Iteration History

       Iter     Calls    NegLogLike         Diff     MaxGrad       Slope

          1         5    177.776248     0.013697    2.873367    -63.0744
          2         8      177.7643     0.011948    1.698144    -4.75239
          3        10    177.757264     0.007036    1.297439    -1.97311
          4        12    177.755688     0.001576    1.441408    -0.49772
          5        14      177.7467     0.008988    1.132279     -0.8223
          6        17    177.746401     0.000299    0.831293    -0.00244
          7        19    177.746318     0.000083    0.724198    -0.00789
          8        21     177.74574     0.000578    0.180018    -0.00583
          9        23    177.745736      3.88E-6    0.017958    -8.25E-6
         10        25    177.745736     3.222E-8    0.000143    -6.51E-8


                   NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table indicates that 10 steps are required for the dual quasi-Newton algorithm to achieve convergence.

```
                              The NLMIXED Procedure

                                  Fit Statistics

                     -2 Log Likelihood                  355.5
                     AIC (smaller is better)            369.5
                     AICC (smaller is better)           370.4
                     BIC (smaller is better)            372.9
```

The “Fitting Information” table lists the final optimized values of the log likelihood function and two information criteria in two different forms.

```
                              The NLMIXED Procedure

                               Parameter Estimates

                       Standard
Parameter   Estimate      Error     DF   t Value   Pr > |t|    Alpha      Lower

beta1        -3.2268    0.05950     10    -54.23     <.0001     0.05    -3.3594
beta2         0.4806     0.1989     10      2.42     0.0363     0.05    0.03745
beta3        -2.4592    0.05126     10    -47.97     <.0001     0.05    -2.5734
s2b1         0.02803    0.01221     10      2.30     0.0445     0.05   0.000833
cb12        -0.00127    0.03404     10     -0.04     0.9710     0.05   -0.07712
s2b2          0.4331     0.2005     10      2.16     0.0560     0.05   -0.01353
s2            0.5016    0.06837     10      7.34     <.0001     0.05     0.3493

                               Parameter Estimates

                        Parameter      Upper    Gradient

                        beta1        -3.0942    -0.00009
                        beta2         0.9238    3.645E-7
                        beta3        -2.3449    0.000039
                        s2b1         0.05523    -0.00014
                        cb12         0.07458    -0.00007
                        s2b2          0.8798    -6.98E-6
                        s2            0.6540    6.133E-6
```

The “Parameter Estimates” table contains the maximum likelihood estimates of the parameters. Both S2B1 and S2B2 are marginally significant, indicating between-subject variability in the clearances and absorption rate constants, respectively. There does not appear to be a significant covariance between them, as seen by the estimate of CB12.

The estimates of β_1, β_2, and β_3 are close to the adaptive quadrature estimates listed in Table 3 of Pinheiro and Bates (1995). However, Pinheiro and Bates use a Cholesky-root parameterization for the random-effects variance matrix and a logarithmic parameterization for the residual variance. The PROC NLMIXED statements using their parameterization are as follows, and results are similar.

```
proc nlmixed data=theoph;
   parms ll1=-1.5 l2=0 ll3=-0.1 beta1=-3 beta2=0.5 beta3=-2.5
      ls2=-0.7;
   s2 = exp(ls2);
   l1 = exp(ll1);
   l3 = exp(ll3);
   s2b1 = l1*l1*s2;
   cb12 = l2*l1*s2;
   s2b2 = (l2*l2 + l3*l3)*s2;
   cl = exp(beta1 + b1);
   ka = exp(beta2 + b2);
   ke = exp(beta3);
   pred = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc ~ normal(pred,s2);
   random b1 b2 ~ normal([0,0],[s2b1,cb12,s2b2]) subject=subject;
run;
```

Example 51.2. Probit-Normal Model with Binomial Data

For this example, consider the data from Weil (1970), also studied by Williams (1975), Ochi and Prentice (1984), and McCulloch (1994). In this experiment 16 pregnant rats receive a control diet and 16 receive a chemically treated diet, and the litter size for each rat is recorded after 4 and 21 days. The SAS data set is a follows.

```
data rats;
   input trt$ m x;
   if (trt=’c’) then do;
      x1 = 1;
      x2 = 0;
   end;
   else do;
      x1 = 0;
      x2 = 1;
   end;
   litter = _n_;
   datalines;
c 13 13
c 12 12
c  9  9
c  9  9
c  8  8
c  8  8
c 13 12
c 12 11
c 10  9
c 10  9
c  9  8
c 13 11
c  5  4
c  7  5
c 10  7
c 10  7
t 12 12
t 11 11
t 10 10
t  9  9
t 11 10
t 10  9
t 10  9
t  9  8
t  9  8
t  5  4
t  9  7
t  7  4
t 10  5
t  6  3
t 10  3
t  7  0
run;
```

Here, M represents the size of the litter after 4 days, and X represents the size of the litter after 21 days. Also, indicator variables X1 and X2 are constructed for the two treatment levels.

Following McCulloch (1994), assume a latent survival model of the form

$$y_{ijk} = t_i + \alpha_{ij} + e_{ijk}$$

where i indexes treatment, j indexes litter, and k indexes newborn rats within a litter. The t_i represent treatment means, the α_{ij} represent random litter effects assumed to be iid $N(0, s_i^2)$, and the e_{ijk} represent iid residual errors, all on the latent scale.

Instead of observing the survival times y_{ijk}, assume that only the binary variable indicating whether y_{ijk} exceeds 0 is observed. If x_{ij} denotes the sum of these binary variables for the ith treatment and the jth litter, then the preceding assumptions lead to the following generalized linear mixed model:

$$x_{ij}|\alpha_{ij} \sim \text{Binomial}(m_{ij}, p_{ij})$$

where m_{ij} is the size of each litter after 4 days and

$$p_{ij} = \Phi(t_i + \alpha_{ij})$$

The PROC NLMIXED statements to fit this model are as follows.

```
proc nlmixed data=rats;
   parms t1=1 t2=1 s1=.05 s2=1;
   eta = x1*t1 + x2*t2 + alpha;
   p = probnorm(eta);
   model x ~ binomial(m,p);
   random alpha ~ normal(0,x1*s1*s1+x2*s2*s2) subject=litter;
   estimate 'gamma2' t2/sqrt(1+s2*s2);
   predict p out=p;
run;
```

As in the previous example, the PROC NLMIXED statement invokes the procedure and the PARMS statement defines the parameters. The parameters for this example are the two treatment means, T1 and T2, and the two random-effect standard deviations, S1 and S2.

The indicator variables X1 and X2 are used in the program to assign the proper mean to each observation in the input data set as well as the proper variance to the random effects. Note that programming expressions are permitted inside the distributional specifications, as illustrated by the random-effects variance specified here.

The ESTIMATE statement requests an estimate of $\gamma_2 = t_2/\sqrt{1+s_2^2}$, which is a location-scale parameter from Ochi and Prentice (1984).

The PREDICT statement constructs predictions for each observation in the input data set. For this example, predictions of P and approximate standard errors of prediction are output to a SAS data set named P. These predictions are functions of the parameter estimates and the empirical Bayes estimates of the random effects α_i.

The output for this model is as follows.

```
                              The NLMIXED Procedure

                                  Specifications

         Data Set                                      WORK.RATS
         Dependent Variable                            x
         Distribution for Dependent Variable           Binomial
         Random Effects                                alpha
         Distribution for Random Effects               Normal
         Subject Variable                              litter
         Optimization Technique                        Dual Quasi-Newton
         Integration Method                            Adaptive Gaussian
                                                       Quadrature
```

The “Specifications” table provides basic information about this nonlinear mixed model.

```
                              The NLMIXED Procedure

                                    Dimensions

                     Observations Used                      32
                     Observations Not Used                   0
                     Total Observations                     32
                     Subjects                               32
                     Max Obs Per Subject                     1
                     Parameters                              4
                     Quadrature Points                       7
```

The “Dimensions” table provides counts of various variables.

```
                              The NLMIXED Procedure

                                    Parameters

                  t1          t2          s1          s2    NegLogLike

                   1           1        0.05           1    54.9362323
```

The “Parameters” table lists the starting point of the optimization.

```
                                 The NLMIXED Procedure

                                   Iteration History

         Iter     Calls    NegLogLike        Diff     MaxGrad       Slope

            1         2    53.9933934    0.942839    11.03261    -81.9428
            2         3     52.875353     1.11804    2.148952    -2.86277
            3         5    52.6350386    0.240314    0.329957    -1.05049
            4         6    52.6319939    0.003045    0.122926    -0.00672
            5         8    52.6313583    0.000636    0.028246    -0.00352
            6        11    52.6313174    0.000041    0.013551    -0.00023
            7        13    52.6313115    5.839E-6    0.000603    -0.00001
            8        15    52.6313115     9.45E-9    0.000022    -1.68E-8


                      NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table indicates successful convergence in 8 iterations.

```
                                 The NLMIXED Procedure

                                    Fit Statistics

                       -2 Log Likelihood                  105.3
                       AIC (smaller is better)            113.3
                       AICC (smaller is better)           114.7
                       BIC (smaller is better)            119.1
```

The “Fitting Information” table lists some useful statistics based on the maximized value of the log likelihood.

```
                                 The NLMIXED Procedure

                                  Parameter Estimates

                          Standard
Parameter   Estimate         Error     DF   t Value   Pr > |t|     Alpha       Lower

t1            1.3063        0.1685     31      7.75     <.0001      0.05      0.9626
t2            0.9475        0.3055     31      3.10     0.0041      0.05      0.3244
s1            0.2403        0.3015     31      0.80     0.4315      0.05     -0.3746
s2            1.0292        0.2988     31      3.44     0.0017      0.05      0.4198

                                  Parameter Estimates

                           Parameter      Upper    Gradient

                           t1            1.6499    -0.00002
                           t2            1.5705    9.283E-6
                           s1            0.8552    0.000014
                           s2            1.6385    -3.16E-6
```

The “Parameter Estimates” table indicates significance of all of the parameters except S1.

```
                              The NLMIXED Procedure

                              Additional Estimates

                    Standard
Label   Estimate     Error    DF  t Value  Pr > |t|   Alpha     Lower     Upper

gamma2    0.6603    0.2165    31     3.05    0.0047    0.05    0.2186    1.1019
```

The "Additional Estimates" table displays results from the ESTIMATE statement. The estimate of γ_2 equals 0.66, agreeing with that obtained by McCulloch (1994). The standard error 0.22 is computed using the delta method (Billingsley 1986; Cox, 1998).

Not shown is the P data set, which contains the original 32 observations and predictions of the p_{ij}.

Example 51.3. Probit-Normal Model with Ordinal Data

The data for this example are from Ezzet and Whitehead (1991), who describe a crossover experiment on two groups of patients using two different inhaler devices (A and B). Patients from group 1 used device A for one week and then device B for another week. Patients from group 2 used the devices in reverse order. The data entered as a SAS data set are as follows.

```
data inhaler;
   input clarity group time freq;
   gt = group*time;
   sub = floor((_n_+1)/2);
   datalines;
1 0 0 59
1 0 1 59
1 0 0 35
2 0 1 35
1 0 0  3
3 0 1  3
1 0 0  2
4 0 1  2
2 0 0 11
1 0 1 11
2 0 0 27
2 0 1 27
2 0 0  2
3 0 1  2
2 0 0  1
4 0 1  1
4 0 0  1
1 0 1  1
4 0 0  1
2 0 1  1
```

```
1 1 0 63
1 1 1 63
1 1 0 13
2 1 1 13
2 1 0 40
1 1 1 40
2 1 0 15
2 1 1 15
3 1 0  7
1 1 1  7
3 1 0  2
2 1 1  2
3 1 0  1
3 1 1  1
4 1 0  2
1 1 1  2
4 1 0  1
3 1 1  1
run;
```

The response measurement, CLARITY, is the patients' assessment on the clarity of the leaflet instructions for the devices. The CLARITY variable is on an ordinal scale, with 1=easy, 2=only clear after rereading, 3=not very clear, and 4=confusing. The GROUP variable indicates the treatment group and the TIME variable indicates the time of measurement. The FREQ variable indicates the number of patients with exactly the same responses. A variable GT is created to indicate a group by time interaction, and a variable SUB is created to indicate patients.

As in the previous example and in Hedeker and Gibbons (1994), assume an underlying latent continuous variable, here with the form

$$y_{ij} = \beta_0 + \beta_1 g_i + \beta_2 t_j + \beta_3 g_i t_j + u_i + e_{ij}$$

where i indexes patient and j indexes the time period, g_i indicates groups, t_j indicates time, u_i is a patient-level normal random effect, and e_{ij} are iid normal errors. The βs are unknown coefficients to be estimated.

Instead of observing y_{ij}, though, you observe only whether it falls in one of the four intervals: $(-\infty, 0)$, $(0, I1)$, $(I1, I1 + I2)$, or $(I1 + I2, \infty)$, where $I1$ and $I2$ are both positive. The resulting category is the value assigned to the CLARITY variable.

The following code sets up and fits this ordinal probit model:

```
proc nlmixed data=inhaler corr ecorr;
   parms b0=0 b1=0 b2=0 b3=0 sd=1 i1=1 i2=1;
   bounds i1 > 0, i2 > 0;
   eta = b0 + b1*group + b2*time + b3*gt + u;
   if (clarity=1) then p = probnorm(-eta);
   else if (clarity=2) then
      p = probnorm(i1-eta) - probnorm(-eta);
   else if (clarity=3) then
      p = probnorm(i1+i2-eta) - probnorm(i1-eta);
   else p = 1 - probnorm(i1+i2-eta);
   if (p > 1e-8) then ll = log(p);
   else ll = -1e100;
   model clarity ~ general(ll);
   random u ~ normal(0,sd*sd) subject=sub;
   replicate freq;
   estimate ’thresh2’ i1;
   estimate ’thresh3’ i1 + i2;
   estimate ’icc’ sd*sd/(1+sd*sd);
run;
```

The PROC statement specifies the input data set and requests correlations both for the parameter estimates (CORR option) and the additional estimates specified with ESTIMATE statements (ECORR option).

The parameters as defined in the PARMS statement are as follows. B0 (overall intercept), B1 (group main effect), B2 (time main effect), B3 (group by time interaction), SD (standard deviation of the random effect), I1 (increment between first and second thresholds), and I2 (increment between second and third thresholds). The BOUNDS statement restricts I1 and I2 to be positive.

The SAS programming statements begin by defining the linear predictor ETA, which is a linear combination of the B parameters and a single random effect U. The next statements define the ordinal likelihood according to the CLARITY variable, ETA, and the increment variables. An error trap is included in case the likelihood becomes too small.

A general log likelihood specification is used in the MODEL statement, and the RANDOM statement defines the random effect U to have standard deviation SD and subject variable SUB. The REPLICATE statement indicates that data for each subject should be replicated according to the FREQ variable.

The ESTIMATE statements specify the second and third thresholds in terms of the increment variables (the first threshold is assumed to equal zero for model identifiability). Also computed is the intraclass correlation.

The output is as follows.

```
                            The NLMIXED Procedure

                                Specifications

        Data Set                                    WORK.INHALER
        Dependent Variable                          clarity
        Distribution for Dependent Variable         General
        Random Effects                              u
        Distribution for Random Effects             Normal
        Subject Variable                            sub
        Replicate Variable                          freq
        Optimization Technique                      Dual Quasi-Newton
        Integration Method                          Adaptive Gaussian
                                                    Quadrature
```

The "Specifications" table echoes some primary information specified for this nonlinear mixed model.

```
                            The NLMIXED Procedure

                                  Dimensions

                    Observations Used                    38
                    Observations Not Used                 0
                    Total Observations                   38
                    Subjects                            286
                    Max Obs Per Subject                   2
                    Parameters                            7
                    Quadrature Points                     5
```

The "Dimensions" table reveals a total of 286 subjects, which is the sum of the values of the FREQ variable. Five quadrature points are selected for log likelihood evaluation.

```
                            The NLMIXED Procedure

                                  Parameters

      b0         b1         b2         b3         sd         i1         i2  NegLogLike

       0          0          0          0          1          1          1  538.484276
```

The "Parameters" table lists the simple starting values for this problem.

```
                              The NLMIXED Procedure

                                Iteration History

        Iter     Calls    NegLogLike        Diff     MaxGrad       Slope

           1         2    476.382511    62.10176    43.75062     -1431.4
           2         4    463.228197    13.15431    14.24648    -106.753
           3         5    458.528118     4.70008    48.31316    -33.0389
           4         6    450.975735    7.552383    22.60098    -40.9954
           5         8    448.012701    2.963033    14.86877    -16.7453
           6        10    447.245153    0.767549    7.774189    -2.26743
           7        11     446.72767    0.517483    3.793533    -1.59278
           8        13    446.518273    0.209396    0.868638    -0.37801
           9        16    446.514528    0.003745    0.328568    -0.02356
          10        18    446.513341    0.001187    0.056778    -0.00183
          11        20    446.513314    0.000027    0.010785    -0.00004
          12        22     446.51331    3.956E-6    0.004922    -5.41E-6
          13        24     446.51331    1.989E-7     0.00047       -4E-7


                   NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table indicates successful convergence in 13 iterations.

```
                              The NLMIXED Procedure

                                 Fit Statistics

                     -2 Log Likelihood                 893.0
                     AIC (smaller is better)           907.0
                     AICC (smaller is better)          910.8
                     BIC (smaller is better)           932.6
```

The “Fitting Information” table lists the log likelihood and information criteria.

```
                              The NLMIXED Procedure

                               Parameter Estimates

                       Standard
Parameter   Estimate      Error     DF   t Value   Pr > |t|    Alpha      Lower

b0           -0.6364     0.1342    285     -4.74     <.0001     0.05    -0.9006
b1            0.6007     0.1770    285      3.39     0.0008     0.05     0.2523
b2            0.6015     0.1582    285      3.80     0.0002     0.05     0.2900
b3           -1.4817     0.2385    285     -6.21     <.0001     0.05    -1.9512
sd            0.6599     0.1312    285      5.03     <.0001     0.05     0.4017
i1            1.7450     0.1474    285     11.84     <.0001     0.05     1.4548
i2            0.5985     0.1427    285      4.19     <.0001     0.05     0.3177

                               Parameter Estimates

                        Parameter      Upper    Gradient

                        b0           -0.3722     0.00047
                        b1            0.9491    0.000265
                        b2            0.9129     0.00008
                        b3           -1.0122    0.000102
                        sd            0.9181    -0.00009
                        i1            2.0352    0.000202
                        i2            0.8794    0.000087
```

The “Parameter Estimates” table indicates significance of all of the parameters.

```
                              The NLMIXED Procedure

                               Additional Estimates

                     Standard
Label     Estimate      Error    DF  t Value  Pr > |t|   Alpha     Lower     Upper

thresh2     1.7450     0.1474   285    11.84    <.0001    0.05    1.4548    2.0352
thresh3     2.3435     0.2073   285    11.31    <.0001    0.05    1.9355    2.7515
icc         0.3034    0.08402   285     3.61    0.0004    0.05    0.1380    0.4687
```

The “Additional Estimates” table displays results from the ESTIMATE statements.

Example 51.4. Poisson-Normal Model with Count Data

This example uses the pump failure data of Gaver and O'Muircheartaigh (1987). The number of failures and the time of operation are recorded for 10 pumps. Each of the pumps is classified into one of two groups corresponding to either continuous or intermittent operation. The data are as follows.

```
data pump;
   input y t group;
   pump = _n_;
   logtstd = log(t) - 2.4564900;
   datalines;
 5  94.320 1
 1  15.720 2
 5  62.880 1
14 125.760 1
 3   5.240 2
19  31.440 1
 1   1.048 2
 1   1.048 2
 4   2.096 2
22  10.480 2
run;
```

Each row denotes data for a single pump, and the variable LOGTSTD contains the centered operation times.

Letting y_{ij} denote the number of failures for the jth pump in the ith group, Draper (1996) considers the following hierarchical model for these data:

$$\begin{aligned} y_{ij}|\lambda_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\ \log\lambda_{ij} &= \alpha_i + \beta_i(\log t_{ij} - \overline{\log t}) + e_{ij} \\ e_{ij}|\sigma^2 &\sim \text{Normal}(0,\sigma^2) \end{aligned}$$

The model specifies different intercepts and slopes for each group, and the random effect is a mechanism for accounting for overdispersion.

The corresponding PROC NLMIXED statements are as follows.

```
proc nlmixed data=pump;
   parms logsig 0 beta1 1 beta2 1 alpha1 1 alpha2 1;
   if (group = 1) then eta = alpha1 + beta1*logtstd + e;
   else eta = alpha2 + beta2*logtstd + e;
   lambda = exp(eta);
   model y ~ poisson(lambda);
   random e ~ normal(0,exp(2*logsig)) subject=pump;
   estimate 'alpha1-alpha2' alpha1-alpha2;
   estimate 'beta1-beta2' beta1-beta2;
run;
```

The output is as follows.

```
                            The NLMIXED Procedure

                               Specifications

     Data Set                                      WORK.PUMP
     Dependent Variable                            y
     Distribution for Dependent Variable           Poisson
     Random Effects                                e
     Distribution for Random Effects               Normal
     Subject Variable                              pump
     Optimization Technique                        Dual Quasi-Newton
     Integration Method                            Adaptive Gaussian
                                                   Quadrature
```

The “Specifications” table displays some details for this Poisson-Normal model.

```
                            The NLMIXED Procedure

                                 Dimensions

                 Observations Used                       10
                 Observations Not Used                    0
                 Total Observations                      10
                 Subjects                                10
                 Max Obs Per Subject                      1
                 Parameters                               5
                 Quadrature Points                        5
```

The “Dimensions” table indicates that data for 10 pumps are used with one observation for each.

```
                            The NLMIXED Procedure

                                 Parameters

     logsig       beta1       beta2       alpha1      alpha2    NegLogLike

          0           1           1            1           1    32.8614614
```

The “Parameters” table lists the simple starting values for this problem and the initial evaluation of the negative log likelihood.

```
                              The NLMIXED Procedure

                                 Iteration History

        Iter      Calls     NegLogLike        Diff      MaxGrad       Slope

           1          2     30.6986932    2.162768     5.107253     -91.602
           2          5     30.0255468    0.673146     2.761738    -11.0489
           3          7      29.726325    0.299222     2.990401    -2.36048
           4          9     28.7390263    0.987299     2.074431    -3.93678
           5         10     28.3161933    0.422833     0.612531    -0.63084
           6         12       28.09564    0.220553     0.462162    -0.52684
           7         14     28.0438024    0.051838     0.405047    -0.10018
           8         16     28.0357134    0.008089     0.135059    -0.01875
           9         18      28.033925    0.001788     0.026279    -0.00514
          10         20     28.0338744    0.000051      0.00402    -0.00012
          11         22     28.0338727    1.681E-6     0.002864    -5.09E-6
          12         24     28.0338724    3.199E-7     0.000147    -6.87E-7
          13         26     28.0338724    2.532E-9     0.000017    -5.75E-9


                   NOTE: GCONV convergence criterion satisfied.
```

The “Iterations” table indicates successful convergence in 13 iterations.

```
                              The NLMIXED Procedure

                                  Fit Statistics

                     -2 Log Likelihood                  56.1
                     AIC (smaller is better)            66.1
                     AICC (smaller is better)           81.1
                     BIC (smaller is better)            67.6
```

The “Fitting Information” table lists the final log likelihood and associated information criteria.

```
                              The NLMIXED Procedure

                               Parameter Estimates

                       Standard
Parameter   Estimate      Error     DF   t Value   Pr > |t|    Alpha      Lower

logsig       -0.3161     0.3213      9     -0.98     0.3508     0.05    -1.0429
beta1        -0.4256     0.7473      9     -0.57     0.5829     0.05    -2.1162
beta2         0.6097     0.3814      9      1.60     0.1443     0.05    -0.2530
alpha1        2.9644     1.3826      9      2.14     0.0606     0.05    -0.1632
alpha2        1.7992     0.5492      9      3.28     0.0096     0.05     0.5568

                               Parameter Estimates

                        Parameter      Upper    Gradient

                        logsig        0.4107    -0.00002
                        beta1         1.2649    -0.00002
                        beta2         1.4724    -1.61E-6
                        alpha1        6.0921    -5.25E-6
                        alpha2        3.0415    -5.73E-6
```

```
                              The NLMIXED Procedure

                               Additional Estimates

                           Standard
  Label           Estimate      Error    DF  t Value  Pr > |t|   Alpha     Lower

  alpha1-alpha2     1.1653     1.4855     9     0.78    0.4529    0.05   -2.1952
  beta1-beta2      -1.0354     0.8389     9    -1.23    0.2484    0.05   -2.9331

                              Additional Estimates

                            Label              Upper

                            alpha1-alpha2     4.5257
                            beta1-beta2       0.8623
```

The “Parameter Estimates” and “Additional Estimates” tables list the maximum likelihood estimates for each of the parameters and two differences. The point estimates for the mean parameters agree fairly closely with the Bayesian posterior means reported by Draper (1996); however, the likelihood-based standard errors are roughly half the Bayesian posterior standard deviations. This is most likely due to the fact that the Bayesian standard deviations account for the uncertainty in estimating σ^2, whereas the likelihood values plug in its estimated value. This downward bias can be corrected somewhat by using the t_9 distribution shown here.

Example 51.5. Failure Time and Frailty Model

In this example an accelerated failure time model with proportional hazard is fitted with and without random effects. The data are from the "Getting Started" example of PROC LIFEREG, see Chapter 39, "The LIFEREG Procedure." Thirty-eight patients are divided into two groups of equal size, and different pain relievers are assigned to each group. The outcome reported is the time in minutes until headache relief. The variable censor indicates whether relief was observed during the course of the observation period (censor = 0) or whether the observation is censored (censor = 1).

```
data headache;
   input minutes group censor @@;
   patient = _n_;
   datalines;
11  1  0   12  1  0   19  1  0   19  1  0
19  1  0   19  1  0   21  1  0   20  1  0
21  1  0   21  1  0   20  1  0   21  1  0
20  1  0   21  1  0   25  1  0   27  1  0
30  1  0   21  1  1   24  1  1   14  2  0
16  2  0   16  2  0   21  2  0   21  2  0
23  2  0   23  2  0   23  2  0   23  2  0
25  2  1   23  2  0   24  2  0   24  2  0
26  2  1   32  2  1   30  2  1   30  2  0
32  2  1   20  2  1
;
```

In modeling survival data, censoring of observations must be taken into account carefully. In this example, only right censoring occurs. If $g(t,\boldsymbol{\beta})$, $h(t,\boldsymbol{\beta})$, and $G(t,\boldsymbol{\beta})$ denote the density of failure, hazard function, and survival distribution function at time t, respectively, the log-likelihood can be written as

$$\begin{aligned} l(\boldsymbol{\beta};\mathbf{t}) &= \sum_{i\in U_u} \log f(t_i,\boldsymbol{\beta}) + \sum_{i\in U_c} \log G(t_i,\boldsymbol{\beta}) \\ &= \sum_{i\in U_u} \log h(t_i,\boldsymbol{\beta}) + \sum_{i=1}^{n} \log G(t_i,\boldsymbol{\beta}) \end{aligned}$$

Refer to Cox and Oakes (1984, ch. 3). In these expressions U_u is the set of uncensored observations, U_c is the set of censored observations, and n denotes the total sample size.

The proportional hazards specification expresses the hazard in terms of a baseline hazard, multiplied by a constant. In this example the hazard is that of a Weibull model and is parameterized as $h(t,\boldsymbol{\beta}) = \gamma\alpha(\alpha t)^{\gamma-1}$ and $\alpha = \exp\{-\mathbf{x}'\boldsymbol{\beta}\}$.

The linear predictor is set equal to the intercept in the reference group (group = 2); this defines the baseline hazard. The corresponding distribution of survival past time t is $G(t,\boldsymbol{\beta}) = \exp\{-(\alpha t)^{\gamma}\}$. Refer to Cox and Oakes (1984, Table 2.1) and the section “Supported Distributions” in Chapter 39, “The LIFEREG Procedure,” for this and other survival distribution models and various parameterizations.

The following NLMIXED statements fit this accelerated failure time model and estimate the cumulative distribution function of time to headache relief.

```
proc nlmixed data=headache;
    bounds gamma > 0;
    linp  = b0 - b1*(group-2);
    alpha = exp(-linp);
    G_t   = exp(-(alpha*minutes)**gamma);
    g     = gamma*alpha*((alpha*minutes)**(gamma-1))*G_t;
    ll    = (censor=0)*log(g) + (censor=1)*log(G_t);
    model minutes ~ general(ll);
    predict 1-G_t out=cdf;
run;
proc print data=cdf;
  var group censor patient minutes pred;
run;
```

Output 51.5.1. Analysis Results for Failure Time Model

```
                                The NLMIXED Procedure

                                    Specifications

         Data Set                                        WORK.HEADACHE
         Dependent Variable                              minutes
         Distribution for Dependent Variable             General
         Optimization Technique                          Dual Quasi-Newton
         Integration Method                              None
```

The “Specifications” table shows that no integration is required, since the model does not contain random effects.

```
                                The NLMIXED Procedure

                                      Dimensions

                     Observations Used                    38
                     Observations Not Used                 0
                     Total Observations                   38
                     Parameters                            3
```

```
                              The NLMIXED Procedure

                                   Parameters

                     gamma          b0          b1    NegLogLike

                         1           1           1    263.990327
```

No starting values were given for the three parameters. The NLMIXED procedure assigns the default values in this case.

```
                              The NLMIXED Procedure

                                Iteration History

         Iter     Calls    NegLogLike        Diff     MaxGrad       Slope

            1         2    169.244311    94.74602     22.5599    -2230.83
            2         4    142.873508     26.3708    14.88631    -3.64643
            3         6    140.633695    2.239814    11.25234    -9.49454
            4         8    122.890659    17.74304    19.44959    -2.50807
            5         9    121.396959    1.493699    13.85584    -4.55427
            6        11    120.623843    0.773116    13.67062    -1.38064
            7        12    119.278196    1.345647    15.78014    -1.69072
            8        14    116.271325    3.006871    26.94029     -3.2529
            9        16    109.427401    6.843925    19.88382     -6.9289
           10        19    103.298102    6.129298    12.15647    -4.96054
           11        22    101.686239    1.611863    14.24868    -4.34059
           12        23    100.027875    1.658364    11.69853    -13.2049
           13        26    99.9189048    0.108971    3.602552    -0.55176
           14        28    99.8738836    0.045021    0.170712    -0.16645
           15        30    99.8736392    0.000244    0.050822    -0.00041
           16        32    99.8736351    4.071E-6    0.000705     -6.9E-6
           17        34    99.8736351     6.1E-10    4.768E-6    -1.23E-9


                   NOTE: GCONV convergence criterion satisfied.
```

```
                              The NLMIXED Procedure

                                 Fit Statistics

                     -2 Log Likelihood                 199.7
                     AIC (smaller is better)           205.7
                     AICC (smaller is better)          206.5
                     BIC (smaller is better)           210.7
```

After 17 iterations and 34 evaluations of the objective function, the procedure converges.

```
                                The NLMIXED Procedure

                                 Parameter Estimates

                           Standard
Parameter    Estimate        Error     DF    t Value    Pr > |t|     Alpha       Lower

gamma          4.7128       0.6742     38       6.99      <.0001      0.05      3.3479
b0             3.3091      0.05885     38      56.23      <.0001      0.05      3.1900
b1            -0.1933      0.07856     38      -2.46      0.0185      0.05     -0.3523

                                 Parameter Estimates

                            Parameter       Upper    Gradient

                            gamma          6.0777    5.327E-8
                            b0             3.4283    -4.77E-6
                            b1           -0.03426    -1.22E-6
```

The parameter estimates and their standard errors are identical to those obtained with the LIFEREG procedure and the statements

```
proc lifereg data=headache;
   class group;
   model minutes*censor(1) = group / dist=weibull;
run;
```

The *t* statistic and confidence limits are based on 38 degrees of freedom. The LIFEREG procedure computes *z*-intervals for the parameter estimates.

For the two groups you obtain

$$\begin{aligned}\widehat{\alpha}(\text{group} = 1) &= \exp\{-3.3091 + 0.1933\} = 0.04434 \\ \widehat{\alpha}(\text{group} = 2) &= \exp\{-3.3091\} = 0.03655\end{aligned}$$

The probabilities of headache relief by t minutes are estimated as

$$1 - G(t, \text{group} = 1) = 1 - \exp\{-(0.04434 * t)^{4.7128}\}$$
$$1 - G(t, \text{group} = 2) = 1 - \exp\{-(0.03655 * t)^{4.7128}\}$$

These probabilities, calculated at the observed times, are shown for the two groups in Output 51.5.2.

Output 51.5.2. Estimated Cumulative Distribution Function

```
          Obs    group    censor    patient    minutes      Pred

            1      1         0          1         11      0.03336
            2      1         0          2         12      0.04985
            3      1         0          3         19      0.35975
            4      1         0          4         19      0.35975
            5      1         0          5         19      0.35975
            6      1         0          6         19      0.35975
            7      1         0          7         21      0.51063
            8      1         0          8         20      0.43325
            9      1         0          9         21      0.51063
           10      1         0         10         21      0.51063
           11      1         0         11         20      0.43325
           12      1         0         12         21      0.51063
           13      1         0         13         20      0.43325
           14      1         0         14         21      0.51063
           15      1         0         15         25      0.80315
           16      1         0         16         27      0.90328
           17      1         0         17         30      0.97846
           18      1         1         18         21      0.51063
           19      1         1         19         24      0.73838
           20      2         0         20         14      0.04163
           21      2         0         21         16      0.07667
           22      2         0         22         16      0.07667
           23      2         0         23         21      0.24976
           24      2         0         24         21      0.24976
           25      2         0         25         23      0.35674
           26      2         0         26         23      0.35674
           27      2         0         27         23      0.35674
           28      2         0         28         23      0.35674
           29      2         1         29         25      0.47982
           30      2         0         30         23      0.35674
           31      2         0         31         24      0.41678
           32      2         0         32         24      0.41678
           33      2         1         33         26      0.54446
           34      2         1         34         32      0.87656
           35      2         1         35         30      0.78633
           36      2         0         36         30      0.78633
           37      2         1         37         32      0.87656
           38      2         1         38         20      0.20414
```

Since the slope estimate is negative with *p*-value of 0.0185, you can infer that pain reliever 1 leads to overall significantly faster relief, but the estimated probabilities give no information about patient-to-patient variation within and between groups. For example, while pain reliever 1 provides faster relief overall, some patients in group 2 may respond more quickly than other patients in group 1. A frailty model enables you to accommodate and estimate patient-to-patient variation in health status by introducing random effects into a subject's hazard function.

The following statements model the hazard for patient i in terms of $\alpha_i = \exp\{-\mathbf{x}_i'\boldsymbol{\beta} - z_i\}$, where z_i is a (normal) random patient effect. Notice that the only difference from the previous NLMIXED statements are the RANDOM statement and the addition of `z` in the linear predictor. The empirical Bayes estimates of the random effect (RANDOM statement), the parameter estimates (ODS OUTPUT statement), and the estimated cumulative distribution function (PREDICT statement) are saved to subsequently graph the patient-specific distribution functions.

```
ods output ParameterEstimates=est;
proc nlmixed data=headache;
    bounds gamma > 0;
    linp  = b0 - b1*(group-2) + z;
    alpha = exp(-linp);
    G_t   = exp(-(alpha*minutes)**gamma);
    g     = gamma*alpha*((alpha*minutes)**(gamma-1))*G_t;
    ll    = (censor=0)*log(g) + (censor=1)*log(G_t);
    model minutes ~ general(ll);
    random z ~ normal(0,exp(2*logsig)) subject=patient out=EB;
    predict 1-G_t out=cdf;
run;
proc print data=eb;
   var patient effect estimate stderrpred;
run;
```

Output 51.5.3. Analysis Results for Frailty Model

```
                                The NLMIXED Procedure

                                   Specifications

         Data Set                                       WORK.HEADACHE
         Dependent Variable                             minutes
         Distribution for Dependent Variable            General
         Random Effects                                 z
         Distribution for Random Effects                Normal
         Subject Variable                               patient
         Optimization Technique                         Dual Quasi-Newton
         Integration Method                             Adaptive Gaussian
                                                        Quadrature
```

The “Specifications” table shows that the objective function is computed by adaptive Gaussian quadrature. The “Dimensions” table reports that nine quadrature points are being used to integrate over the random effects.

```
                                The NLMIXED Procedure

                                     Dimensions

                     Observations Used                     38
                     Observations Not Used                  0
                     Total Observations                    38
                     Subjects                              38
                     Max Obs Per Subject                    1
                     Parameters                             4
                     Quadrature Points                      9
```

```
                              The NLMIXED Procedure

                                    Parameters

                gamma          b0          b1      logsig    NegLogLike

                    1           1           1           1     170.94366
```

```
                              The NLMIXED Procedure

                                 Iteration History

         Iter     Calls    NegLogLike        Diff     MaxGrad       Slope

            1         5    142.121411    28.82225    12.14484    -88.8664
            2         7    136.440369    5.681042    25.93096    -65.7217
            3         9    122.972041    13.46833    46.56546    -146.887
            4        11    120.904825    2.067216    23.77936    -94.2862
            5        13    109.224144    11.68068    57.65493    -92.4075
            6        15    105.064733    4.159411    4.824649    -19.5879
            7        16    101.902207    3.162526     14.1287    -6.33767
            8        18    99.6907395    2.211468    7.676822    -3.42364
            9        20    99.3654033    0.325336    5.689204    -0.93978
           10        22    99.2602178    0.105185    0.317643    -0.23408
           11        24     99.254434    0.005784     1.17351    -0.00556
           12        25    99.2456973    0.008737    0.247412    -0.00871
           13        27    99.2445445    0.001153    0.104942    -0.00218
           14        29    99.2444958    0.000049    0.005646     -0.0001
           15        31    99.2444957    9.147E-8    0.000271    -1.84E-7


                   NOTE: GCONV convergence criterion satisfied.
```

The procedure converges after 15 iterations. The achieved -2 log likelihood is only 1.2 less than that in the model without random effects. Compared to a chi-square distribution with one degree of freedom, the addition of the random effect appears not to improve the model significantly. Care must be exercised, however, in the interpretation of likelihood ratio tests when the value under the null hypothesis falls on the boundary of the parameter space (refer to, for example, Self and Liang 1987).

```
                              The NLMIXED Procedure

                                  Fit Statistics

                     -2 Log Likelihood                  198.5
                     AIC (smaller is better)            206.5
                     AICC (smaller is better)           207.7
                     BIC (smaller is better)            213.0
```

```
                              The NLMIXED Procedure

                               Parameter Estimates

                       Standard
Parameter   Estimate      Error     DF   t Value   Pr > |t|    Alpha       Lower

gamma         6.2867     2.1334     37      2.95     0.0055     0.05      1.9641
b0            3.2786    0.06576     37     49.86     <.0001     0.05      3.1453
b1           -0.1761    0.08264     37     -2.13     0.0398     0.05     -0.3436
logsig       -1.9027     0.5273     37     -3.61     0.0009     0.05     -2.9711

                               Parameter Estimates

                        Parameter      Upper    Gradient

                        gamma        10.6093    -1.89E-7
                        b0            3.4118    0.000271
                        b1          -0.00868    0.000111
                        logsig       -0.8343    0.000027
```

The estimate of the Weibull parameter has changed drastically from the model without random effects. The variance of the patient random effect is $\exp\{-2 * 1.9027\} = 0.02225$. The next listing shows the empirical Bayes estimates of the random effects. These are the adjustments made to the linear predictor in order to obtain a patient's survival distribution.

```
                                                StdErr
           Obs    patient    Effect    Estimate      Pred

             1        1        z       -0.13597    0.23249
             2        2        z       -0.13323    0.22793
             3        3        z       -0.06294    0.13813
             4        4        z       -0.06294    0.13813
             5        5        z       -0.06294    0.13813
             6        6        z       -0.06294    0.13813
             7        7        z       -0.02568    0.11759
             8        8        z       -0.04499    0.12618
             9        9        z       -0.02568    0.11759
            10       10        z       -0.02568    0.11759
            11       11        z       -0.04499    0.12618
            12       12        z       -0.02568    0.11759
            13       13        z       -0.04499    0.12618
            14       14        z       -0.02568    0.11759
            15       15        z        0.05980    0.11618
            16       16        z        0.10458    0.12684
            17       17        z        0.17147    0.14550
            18       18        z        0.06471    0.13807
            19       19        z        0.11157    0.14604
            20       20        z       -0.13406    0.22899
            21       21        z       -0.12698    0.21667
            22       22        z       -0.12698    0.21667
            23       23        z       -0.08506    0.15701
            24       24        z       -0.08506    0.15701
            25       25        z       -0.05797    0.13294
            26       26        z       -0.05797    0.13294
            27       27        z       -0.05797    0.13294
            28       28        z       -0.05797    0.13294
            29       29        z        0.06420    0.13956
            30       30        z       -0.05797    0.13294
            31       31        z       -0.04266    0.12390
            32       32        z       -0.04266    0.12390
            33       33        z        0.07618    0.14132
            34       34        z        0.16292    0.16460
            35       35        z        0.13193    0.15528
            36       36        z        0.06327    0.12124
            37       37        z        0.16292    0.16460
            38       38        z        0.02074    0.14160
```

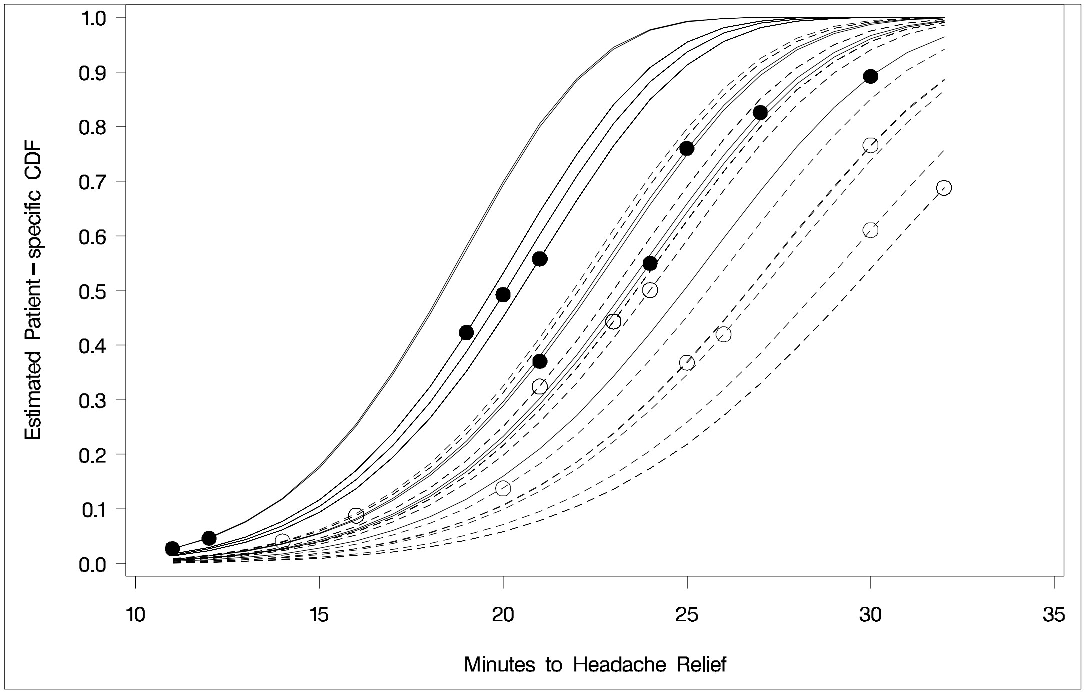
The predicted values and patient-specific survival distributions can be plotted with the SAS code that follows.

```
data cdf; set cdf; symbolid = int((patient-1)/19);
proc transpose data=est(keep=estimate) out=trest;
data trest; set trest; rename col1=gamma col2=b0 col3=b1;
data pred; merge eb(keep=estimate)
                 headache(keep=patient group);
    if _n_ = 1 then merge trest(keep=gamma b0 b1);
    do minutes=11 to 32;
      linp     = b0 - b1*(group-2) + estimate;
      pred     = 1-exp(- (exp(-linp)*minutes)**gamma);
      symbolid = patient+1;
      output;
    end;
    keep pred minutes symbolid;
run;
```

```
data pred; set cdf(keep=pred minutes symbolid) pred;
run;

axis1 label=(angle=90 rotate=0
             'Estimated Patient-specific CDF')
      minor=none;
axis2 label=('Minutes to Headache Relief'    )
      minor=none order=(10 to 35 by 5);

symbol1 value=dot    c=black h=0.15in i=none r=1;
symbol2 value=circle c=black h=0.15in i=none r=1;
symbol3 value=none   c=black l=1      i=join r=19;
symbol4 value=none   c=black l=2      i=join r=19;
proc gplot data=pred;
    plot pred*minutes=symbolid / frame cframe=ligr
             nolegend vaxis=axis1 haxis=axis2;
run;
```

The separation of the distribution functions by groups is evident in Output 51.5.4. Most of the distributions of patients in the first group are to the left of the distributions in the second group. The separation is not complete, however. Several patients assigned the second pain reliever experience headache relief more quickly than patients assigned to the first group.

Output 51.5.4. Patient-Specific CDFs and Predicted Values. Pain Reliever 1: Solid Line, Closed Circles. Pain Reliever 2: Dashed Lines, Open Circles.



References

Abramowitz, M. and Stegun, I.A. (1972), *Handbook of Mathematical Functions*, New York: Dover Publications, Inc.

Anderson, D.A. and Aitkin, M. (1985), "Variance Component Models with Binary Response: Interviewer Variability," *Journal of the Royal Statistical Society B*, 47, 203–210.

Beal, S.L. and Sheiner, L.B. (1982), "Estimating Population Kinetics," *CRC Crit. Rev. Biomed. Eng.*, 8, 195–222.

Beal, S.L. and Sheiner, L.B. (1988), "Heteroskedastic Nonlinear Regression," *Technometrics*, 30, 327–338.

Beal, S.L. and Sheiner, L.B., eds. (1992), *NONMEM User's Guide*, University of California, San Francisco, NONMEM Project Group.

Beale, E.M.L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F.A. Lootsma, London: Academic Press.

Beitler, P.J. and Landis, J.R. (1985), "A Mixed-effects Model for Categorical Data," *Biometrics*, 41, 991–1000.

Billingsley, P. (1986), *Probability and Measure*, Second Edition, New York: John Wiley & Sons, Inc.

Booth, J.G. and Hobert, J.P. (1998), "Standard Errors of Prediction in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 93, 262–272.

Breslow, N.E. and Clayton, D.G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association*, 88, 9–25.

Burnham, K.P. and Anderson, D.R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Cox, C. (1998), "Delta Method," *Encyclopedia of Biostatistics*, Eds. Peter Armitage and Theodore Colton, New York: John Wiley, 1125-1127.

Cox, D.R. and Oakes, D. (1984), *Analysis of Survival Data*, New York: Chapman & Hall.

Cramer, J. S. (1986), *Econometric Applications of Maximum Likelihood Methods*, Cambridge, England: Cambridge University Press.

Crouch, E.A.C. and Spiegelman, D. (1990), "The Evaluation of Integrals of the Form $\int_{-\infty}^{\infty} f(t)\exp(-t^2)dt$: Application to Logistic-normal Models," *Journal of the American Statistical Association*, 85, 464–469

Davidian, M. and Gallant, R.A. (1993), "The Nonlinear Mixed Effects Model with a Smooth Random Effects Density," *Biometrika*, 80, 475–488.

Davidian, M. and Giltinan, D.M. (1995), *Nonlinear Models for Repeated Measurement Data*, New York: Chapman & Hall.

Dennis, J.E., Gay, D.M., and Welsch, R.E. (1981), “An Adaptive Nonlinear Least-Squares Algorithm,” *ACM Transactions on Mathematical Software*, 7, 348–368.

Dennis, J.E. and Mei, H.H.W. (1979), “Two New Unconstrained Optimization Algorithms which Use Function and Gradient Values,” *J. Optim. Theory Appl.*, 28, 453–482.

Dennis, J.E. and Schnabel, R.B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Englewood, NJ: Prentice-Hall.

Diggle, P.J., Liang, K.Y., and Zeger, S.L. (1994), *Analysis of Longitudinal Data*, Oxford: Clarendon Press.

Draper, D. (1996), “Discussion of the Paper by Lee and Nelder,” *Journal of the Royal Statistical Society, Series B*, 58, 662–663.

Draper, N.R. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Engel, B. and Keen, A. (1992), “A Simple Approach for the Analysis of Generalized Linear Mixed Models,” LWA-92-6, Agricultural Mathematics Group (GLW-DLO). Wageningen, The Netherlands.

Eskow, E. and Schnabel, R.B. (1991), “Algorithm 695: Software for a New Modified Cholesky Factorization,” *Transactions on Mathematical Software*, 17(3), 306–312.

Ezzet, F. and Whitehead, J. (1991), “A Random Effects Model for Ordinal Responses from a Crossover Trial,” *Statistics in Medicine*, 10, 901–907.

Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons, Inc.

Galecki, A.T. (1998), “NLMEM: New SAS/IML Macro for Hierarchical Nonlinear Models,” *Computer Methods and Programs in Biomedicine*, 55, 207–216.

Gallant, A.R. (1987), *Nonlinear Statistical Models*, New York: John Wiley & Sons, Inc.

Gaver, D.P. and O’Muircheartaigh, I.G. (1987), “Robust Empirical Bayes Analysis of Event Rates,” *Technometrics*, 29, 1–15.

Gay, D.M. (1983), “Subroutines for Unconstrained Minimization,” *ACM Transactions on Mathematical Software*, 9, 503–524.

Gilmour, A.R., Anderson, R.D., and Rae, A.L. (1985), “The Analysis of Binomial Data by Generalized Linear Mixed Model,” *Biometrika*, 72, 593–599.

Goldstein, H. (1991), “Nonlinear Multilevel Models, with an Application to Discrete Response Data,” *Biometrika*, 78, 45–51.

Golub, G.H., and Welsch, J.H. (1969), “Calculation of Gaussian Quadrature Rules,” *Mathematical Computing,* 23, 221–230.

Harville, D.A. and Mee, R.W. (1984), “A Mixed-model Procedure for Analyzing Ordered Categorical Data,” *Biometrics*, 40, 393–408.

Hedeker, D. and Gibbons, R.D. (1994), “A Random Effects Ordinal Regression Model for Multilevel Analysis,” *Biometrics,* 50, 933–944.

Hurvich, C.M. and Tsai, C.-L. (1989), “Regression and Time Series Model Selection in Small Samples,” *Biometrika,* 76, 297–307.

Lin, X. and Breslow, N.E. (1996), “Bias Correction in Generalized Linear Mixed Models with Multiple Components of Dispersion,” *Journal of the American Statistical Association,* 91, 1007–1016.

Lindstrom, M.J. and Bates, D.M. (1990), “Nonlinear Mixed Effects Models for Repeated Measures Data,” *Biometrics,* 46, 673–687.

Littell, R.C., Milliken, G.A., Stroup, W.W., and Wolfinger, R.D. (1996), *SAS System for Mixed Models,* Cary, NC: SAS Institute Inc.

Liu, Q. and Pierce, D.A. (1994), “A Note on Gauss-Hermite Quadrature,” *Biometrika,* 81, 624–629.

Longford, N.T. (1994), “Logistic Regression with Random Coefficients,” *Computational Statistics and Data Analysis,* 17, 1–15.

McCulloch, C.E. (1994), “Maximum Likelihood Variance Components Estimation for Binary Data,” *Journal of the American Statistical Association,* 89, 330–335.

McGilchrist, C.E. (1994), “Estimation in Generalized Mixed Models,” *Journal of the Royal Statistical Society B,* 56, 61–69.

Moré, J.J. (1978), “The Levenberg-Marquardt Algorithm: Implementation and Theory,” in *Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin-Heidelberg-New York: Springer Verlag.

Moré, J.J. and Sorensen, D.C. (1983), “Computing a Trust-region Step,” *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.

Ochi, Y. and Prentice, R.L. (1984), “Likelihood Inference in a Correlated Probit Regression Model,” *Biometrika,* 71, 531–543.

Pierce, D.A. and Sands, B.R. (1975), *Extra-Bernoulli Variation in Binary Data,* Technical Report 46, Department of Statistics, Oregon State University.

Pinheiro, J.C. and Bates, D.M. (1995), “Approximations to the Log-likelihood Function in the Nonlinear Mixed-effects Model,” *Journal of Computational and Graphical Statistics,* 4, 12–35.

Polak, E. (1971), *Computational Methods in Optimization*, New York, Academic Press.

Powell, J.M.D. (1977), “Restart Procedures for the Conjugate Gradient Method,” *Math. Prog.*, 12, 241–254.

Roe, D.J. (1997) “Comparison of Population Pharmacokinetic Modeling Methods Using Simulated Data: Results from the Population Modeling Workgroup,” *Statistics in Medicine,* 16, 1241–1262.

Rodriguez, G. and Goldman, N. (1995), “An Assessment of Estimation Procedures for Multilevel Models with Binary Response,” *Journal of the Royal Statistical Society, Series A,* 158, 73–89.

Schall, R. (1991). “Estimation in Generalized Linear Models with Random Effects,” *Biometrika*, 78, 719–727.

Schittkowski, K. and Stoer, J. (1979), “A Factorization Method for the Solution of Constrained Linear Least Squares Problems Allowing Subsequent Data Changes,” *Numererische Mathematik*, 31, 431–463.

Self, S.G. and Liang, K.Y. (1987), “Asymptotic Properties of Maximum Likelihood estimators and Likelihood Ratio Tests Under Nonstandard Conditions,” *Journal of the American Statistical Association*, 82, 605–610

Serfling, R.J. (1980), *Approximation Theorems of Mathematical Statistics*, New York, John Wiley & Sons, Inc.

Sheiner L. B. and Beal S. L., “Evaluation of Methods for Estimating Population Pharmacokinetic Parameters. I. Michaelis-Menten Model: Routine Clinical Pharmacokinetic Data,” *Journal of Pharmacokinetics and Biopharmaceutics*, 8, (1980) 553–571.

Sheiner, L.B. and Beal, S.L. (1985), “Pharmacokinetic Parameter Estimates from Several Least Squares Procedures: Superiority of Extended Least Squares,” *Journal of Pharmacokinetics and Biopharmaceutics*, 13, 185–201.

Smith, S.P. (1995), “Differentiation of the Cholesky Algorithm,” *Journal of Computational and Graphical Statistics*, 4, 134–147.

Stiratelli, R., Laird, N.M., and Ware, J.H. (1984), “Random Effects Models for Serial Observations with Binary Response,” *Biometrics*, 40, 961-971.

Vonesh, E.F., (1992), “Nonlinear Models for the Analysis of Longitudinal Data,” *Statistics in Medicine*, 11, 1929–1954.

Vonesh, E.F., (1996), “A Note on Laplace’s Approximation in Nonlinear Mixed Effects Models,” *Biometrika*, 83, 447–452.

Vonesh, E.F. and Chinchilli, V.M. (1997), *Linear and Nonlinear Models for the Analysis of Repeated Measurements*, New York: Marcel Dekker.

Weil, C.S. (1970), “Selection of the Valid Number of Sampling Units and Consideration of their Combination in Toxicological Studies Involving Reproduction, Teratogenesis, or Carcinogenesis,” *Food and Cosmetic Toxicology*, 8, 177–182.

Williams, D.A. (1975), “The Analysis of Binary Responses from Toxicological Experiments Involving Reproduction and Teratogenicity,” *Biometrics*, 31, 949–952.

Wolfinger R.D. (1993), “Laplace’s Approximation for Nonlinear Mixed Models,” *Biometrika*, 80, 791–795.

Wolfinger, R.D. (1997), “Comment: Experiences with the SAS Macro NLINMIX,” *Statistics in Medicine*, 16, 1258–1259.

Wolfinger, R.D., and Lin, X. (1997), “Two Taylor-series Approximation Methods for Nonlinear Mixed Models,” *Computational Statistics and Data Analysis*, 25, 465–490.

Wolfinger, R.D. and O'Connell, M. (1993), "Generalized Linear Mixed Models: a Pseudo-likelihood Approach," *Journal of Statistical Computation and Simulation*, 48, 233–243.

Yuh, L., Beal, S., Davidian, M., Harrison, F., Hester, A., Kowalski, K., Vonesh, E., Wolfinger, R. (1994), "Population Pharmacokinetic/Pharmacodynamic Methodology and Applications: a Bibliography," *Biometrics*, 50, 566–575.

Chapter 52
The NPAR1WAY Procedure

Chapter Contents

Chapter 52
The NPAR1WAY Procedure

Overview

The NPAR1WAY procedure performs nonparametric tests for location and scale differences across a one-way classification. PROC NPAR1WAY also provides a standard analysis of variance on the raw data and tests based on the empirical distribution function.

PROC NPAR1WAY performs tests for location and scale differences based on the following scores of a response variable: Wilcoxon, median, Van der Waerden, Savage, Siegel-Tukey, Ansari-Bradley, Klotz, and Mood scores. Additionally, PROC NPAR1WAY provides tests using the raw input data as scores. When the data are classified into two samples, tests are based on simple linear rank statistics. When the data are classified into more than two samples, tests are based on one-way ANOVA statistics. Both asymptotic and exact p-values are available for these tests.

PROC NPAR1WAY also provides empirical distribution function (EDF) statistics, which test whether the distribution of a variable is the same across different groups. These include the Kolmogorov-Smirnov test, the Cramer-von Mises test, and, when the data are classified into only two samples, the Kuiper test. Exact p-values are available for the two-sample Kolmogorov-Smirnov test.

Getting Started

This example illustrates how you can use PROC NPAR1WAY to perform a one-way nonparametric analysis. The data from Halverson and Sherwood (1932) consist of weight gain measurements for five different levels of gossypol additive. Gossypol is a substance contained in cottonseed shells, and these data were collected to study the effect of gossypol on animal nutrition.

The following DATA step statements create the SAS data set Gossypol:

```
data Gossypol;
   input Dose n;
   do i=1 to n;
      input Gain @@;
      output;
   end;
   datalines;
 0 16
   228 229 218 216 224 208 235 229 233 219 224 220 232 200 208 232
.04 11
   186 229 220 208 228 198 222 273 216 198 213
.07 12
   179 193 183 180 143 204 114 188 178 134 208 196
.10 17
   130 87 135 116 118 165 151 59 126 64 78 94 150 160 122 110 178
```

```
.13 11
   154 130 130 118 118 104 112 134 98 100 104
;
```

The data set Gossypol contains the variable Dose, which represents the amount of gossypol additive, and the variable Gain, which represents the weight gain.

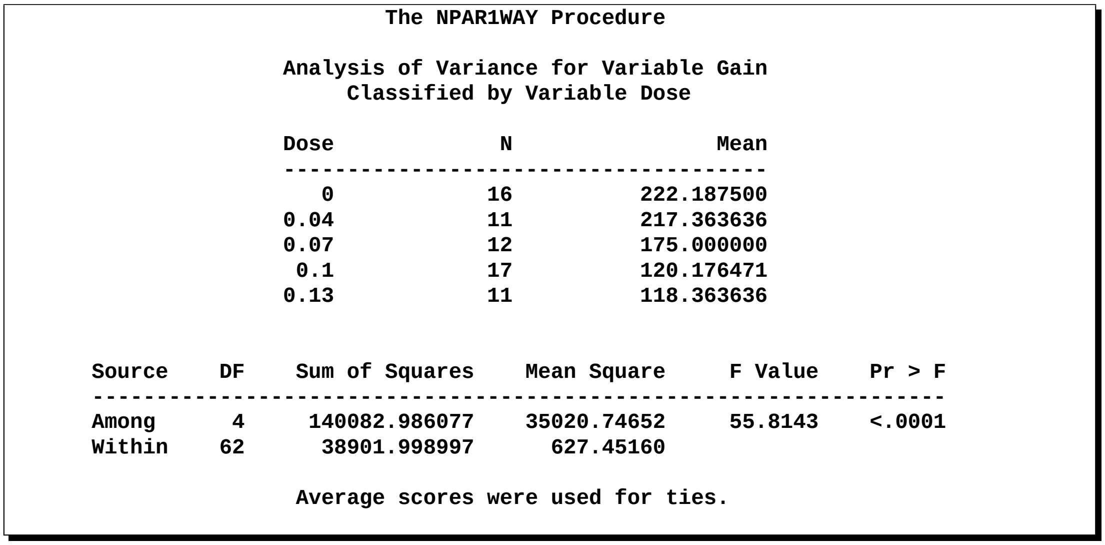
Researchers are interested in whether there is a difference in weight gain among the different dose levels of gossypol. The following statements invoke the NPAR1WAY procedure to perform a nonparametric analysis of this problem:

```
proc npar1way data=Gossypol;
   class Dose;
   var Gain;
run;
```

The variable Dose is the CLASS variable, and the VAR statement specifies the variable Gain is the response variable. The CLASS statement is required, and you must name only one CLASS variable. You may name one or more analysis variables in the VAR statement. If you omit the VAR statement, PROC NPAR1WAY analyzes all numeric variables in the data set except for the CLASS variable, the FREQ variable, and the BY variables.

Since no analysis options are specified in the PROC NPAR1WAY statement, the ANOVA, WILCOXON, MEDIAN, VW, SAVAGE, and EDF options are invoked by default. The following tables show the results of these analyses.

The tables in Figure 52.1 are produced with the ANOVA option. For each level of the CLASS variable Dose, PROC NPAR1WAY displays the number of observations and the mean of the analysis variable Gain. PROC NPAR1WAY displays a standard analysis of variance on the raw data. This gives the same results as the GLM and ANOVA procedures. The p-value for the F test is <.0001, which indicates that Dose accounts for a significant portion of the variability in the dependent variable Gain.

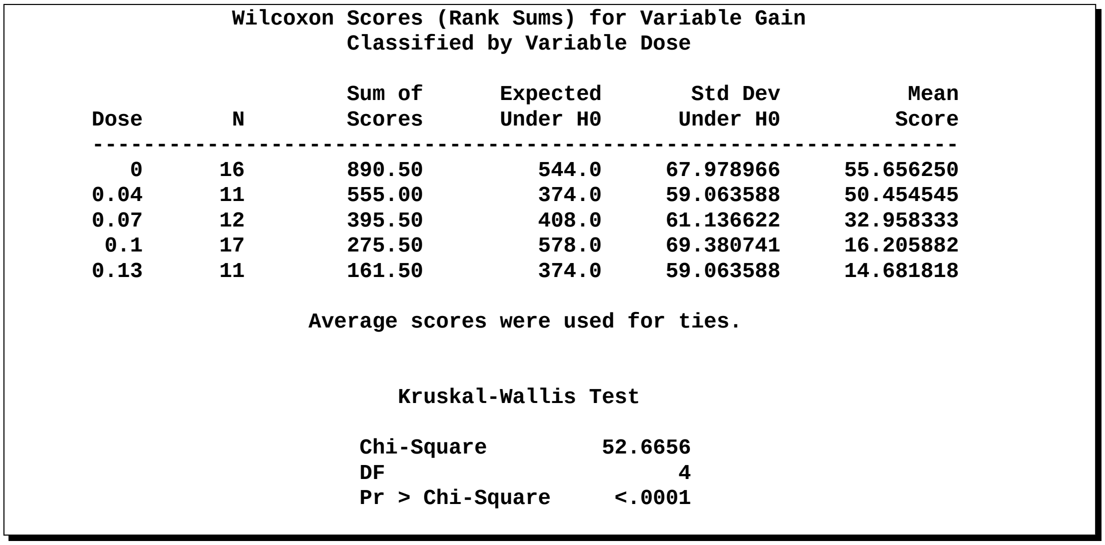


```
                       The NPAR1WAY Procedure

               Analysis of Variance for Variable Gain
                    Classified by Variable Dose

               Dose             N               Mean
               --------------------------------------
                  0            16         222.187500
               0.04            11         217.363636
               0.07            12         175.000000
                0.1            17         120.176471
               0.13            11         118.363636


Source    DF    Sum of Squares    Mean Square     F Value    Pr > F
-------------------------------------------------------------------
Among      4     140082.986077    35020.74652     55.8143    <.0001
Within    62      38901.998997      627.45160

                Average scores were used for ties.
```

Figure 52.1. Analysis of Variance

The WILCOXON option produces the output in Figure 52.2. PROC NPAR1WAY first provides a summary of the Wilcoxon scores for the analysis variable Gain by class level. For each level of the CLASS variable Dose, PROC NPAR1WAY displays the following information: number of observations, sum of the Wilcoxon scores, expected sum under the null hypothesis of no difference among class levels, standard deviation under the null hypothesis, and mean score.

Next PROC NPAR1WAY displays the one-way ANOVA statistic, which for Wilcoxon scores is known as the Kruskal-Wallis test. The statistic equals 52.6656, with four degrees of freedom, which is the number of class levels minus one. The p-value, or probability of a larger statistic under the null hypothesis, is <.0001. This leads to rejection of the null hypothesis that there is no difference in location for Gain among the levels of Dose. This p-value is asymptotic, computed from the asymptotic chi-square distribution of the test statistic. For certain data sets it may also be useful to compute the exact p-value; for example, for small data sets, or data sets that are sparse, skewed, or heavily tied. You can use the EXACT statement to request exact p-values for any of the location or scale tests available in PROC NPAR1WAY.



```
                 Wilcoxon Scores (Rank Sums) for Variable Gain
                          Classified by Variable Dose

                          Sum of      Expected       Std Dev          Mean
    Dose        N         Scores      Under H0      Under H0         Score
    --------------------------------------------------------------------
       0       16         890.50         544.0     67.978966     55.656250
    0.04       11         555.00         374.0     59.063588     50.454545
    0.07       12         395.50         408.0     61.136622     32.958333
     0.1       17         275.50         578.0     69.380741     16.205882
    0.13       11         161.50         374.0     59.063588     14.681818

                       Average scores were used for ties.


                              Kruskal-Wallis Test

                           Chi-Square         52.6656
                           DF                       4
                           Pr > Chi-Square     <.0001
```

Figure 52.2. Wilcoxon Score Analysis

Figure 52.3 through Figure 52.5 display the analyses produced by the MEDIAN, VW, and SAVAGE options. For each score type, PROC NPAR1WAY provides a summary of scores and the one-way ANOVA statistic, as previously described for Wilcoxon scores. Other score types available in PROC NPAR1WAY are Siegel-Tukey, Ansari-Bradley, Klotz, and Mood, which are used to test for scale differences. Additionally, you can request the SCORES=DATA option, which uses the input data as scores. This option gives you the flexibility to construct any scores for your data with the DATA step and then analyze these scores with PROC NPAR1WAY.

```
         Median Scores (Number of Points Above Median) for Variable Gain
                          Classified by Variable Dose

                          Sum of      Expected       Std Dev          Mean
      Dose        N       Scores      Under H0      Under H0         Score
      --------------------------------------------------------------------
         0       16         16.0      7.880597      1.757902          1.00
      0.04       11         11.0      5.417910      1.527355          1.00
      0.07       12          6.0      5.910448      1.580963          0.50
       0.1       17          0.0      8.373134      1.794152          0.00
      0.13       11          0.0      5.417910      1.527355          0.00

                       Average scores were used for ties.


                            Median One-Way Analysis

                           Chi-Square         54.1765
                           DF                       4
                           Pr > Chi-Square     <.0001
```

Figure 52.3. Median Score Analysis

```
                Van der Waerden Scores (Normal) for Variable Gain
                          Classified by Variable Dose

                          Sum of      Expected       Std Dev          Mean
      Dose        N       Scores      Under H0      Under H0         Score
      --------------------------------------------------------------------
         0       16    16.116474           0.0      3.325957      1.007280
      0.04       11     8.340899           0.0      2.889761      0.758264
      0.07       12    -0.576674           0.0      2.991186     -0.048056
       0.1       17   -14.688921           0.0      3.394540     -0.864054
      0.13       11    -9.191777           0.0      2.889761     -0.835616

                       Average scores were used for ties.


                        Van der Waerden One-Way Analysis

                           Chi-Square         47.2972
                           DF                       4
                           Pr > Chi-Square     <.0001
```

Figure 52.4. Van der Waerden Score Analysis

```
                  Savage Scores (Exponential) for Variable Gain
                           Classified by Variable Dose

                             Sum of      Expected       Std Dev          Mean
      Dose          N        Scores      Under H0      Under H0         Score
      -------------------------------------------------------------------------
         0         16     16.074391           0.0      3.385275      1.004649
      0.04         11      7.693099           0.0      2.941300      0.699373
      0.07         12     -3.584958           0.0      3.044534     -0.298746
       0.1         17    -11.979488           0.0      3.455082     -0.704676
      0.13         11     -8.203044           0.0      2.941300     -0.745731

                        Average scores were used for ties.


                             Savage One-Way Analysis

                            Chi-Square         39.4908
                            DF                       4
                            Pr > Chi-Square     <.0001
```

Figure 52.5. Savage Score Analysis

```
                    Kolmogorov-Smirnov Test for Variable Gain
                           Classified by Variable Dose

                                     EDF at    Deviation from Mean
               Dose         N       Maximum         at Maximum
               --------------------------------------------------
                  0        16      0.000000         -1.910448
               0.04        11      0.000000         -1.584060
               0.07        12      0.333333         -0.499796
                0.1        17      1.000000          2.153861
               0.13        11      1.000000          1.732565
               Total       67      0.477612

                   Maximum Deviation Occurred at Observation 36
                         Value of Gain at Maximum = 178.0

                    Kolmogorov-Smirnov Statistics (Asymptotic)
                          KS   0.457928    KSa   3.748300


                     Cramer-von Mises Test for Variable Gain
                           Classified by Variable Dose

                                               Summed Deviation
                    Dose            N              from Mean
                    ----------------------------------------
                       0           16              2.165210
                    0.04           11              0.918280
                    0.07           12              0.348227
                     0.1           17              1.497542
                    0.13           11              1.335745

                    Cramer-von Mises Statistics (Asymptotic)
                          CM   0.093508    CMa   6.265003
```

Figure 52.6. Empirical Distribution Function Analysis

The tables in Figure 52.6 display the empirical distribution function statistics, comparing the distribution of Gain for the different levels of Dose. These tables are produced by the EDF option, and they include Kolmogorov-Smirnov statistics and Cramer-von Mises statistics.

In the preceding example, the CLASS variable Dose has five levels, and the analyses examine possible differences among these five levels, or samples. The following statements invoke the NPAR1WAY procedure to perform a nonparametric analysis of the two lowest levels of Dose:

```
proc npar1way data=Gossypol;
   where Dose <= .04;
   class Dose;
   var Gain;
run;
```

The following tables show the results of this two-sample analysis. The tables in Figure 52.7 are produced by the ANOVA option.

```
                          The NPAR1WAY Procedure

                  Analysis of Variance for Variable Gain
                       Classified by Variable Dose

                  Dose             N               Mean
                  --------------------------------------
                     0            16         222.187500
                  0.04            11         217.363636


     Source    DF    Sum of Squares    Mean Square     F Value    Pr > F
     -------------------------------------------------------------------
     Among      1        151.683712     151.683712      0.5587    0.4617
     Within    25       6786.982955     271.479318

                   Average scores were used for ties.
```

Figure 52.7. Analysis of Variance for Two-Sample Data

```
                 Wilcoxon Scores (Rank Sums) for Variable Gain
                          Classified by Variable Dose

                          Sum of      Expected       Std Dev          Mean
Dose           N          Scores      Under H0      Under H0         Score
--------------------------------------------------------------------
   0          16          253.50         224.0     20.221565     15.843750
0.04          11          124.50         154.0     20.221565     11.318182

                       Average scores were used for ties.


                            Wilcoxon Two-Sample Test

                         Statistic             124.5000

                         Normal Approximation
                         Z                      -1.4341
                         One-Sided Pr <  Z       0.0758
                         Two-Sided Pr > |Z|      0.1515

                         t Approximation
                         One-Sided Pr <  Z       0.0817
                         Two-Sided Pr > |Z|      0.1635

                   Z includes a continuity correction of 0.5.


                              Kruskal-Wallis Test

                         Chi-Square              2.1282
                         DF                           1
                         Pr > Chi-Square         0.1446
```

Figure 52.8. Wilcoxon Two-Sample Analysis

Figure 52.8 displays the output produced by the WILCOXON option. PROC NPAR1WAY provides a summary of the Wilcoxon scores for the analysis variable Gain for each of the two class levels. Since there are only two levels, PROC NPAR1WAY displays the two-sample test, based on the simple linear rank statistic with Wilcoxon scores. The normal approximation includes a continuity correction. To remove this, you can specify the CORRECT=NO option. PROC NPAR1WAY also gives a *t* approximation for the Wilcoxon two-sample test. And as for the multisample analysis, PROC NPAR1WAY computes a one-way ANOVA statistic, which for Wilcoxon scores is known as the Kruskal-Wallis test. All these *p*-values show no difference in Gain for the two Dose levels at the .05 level of significance.

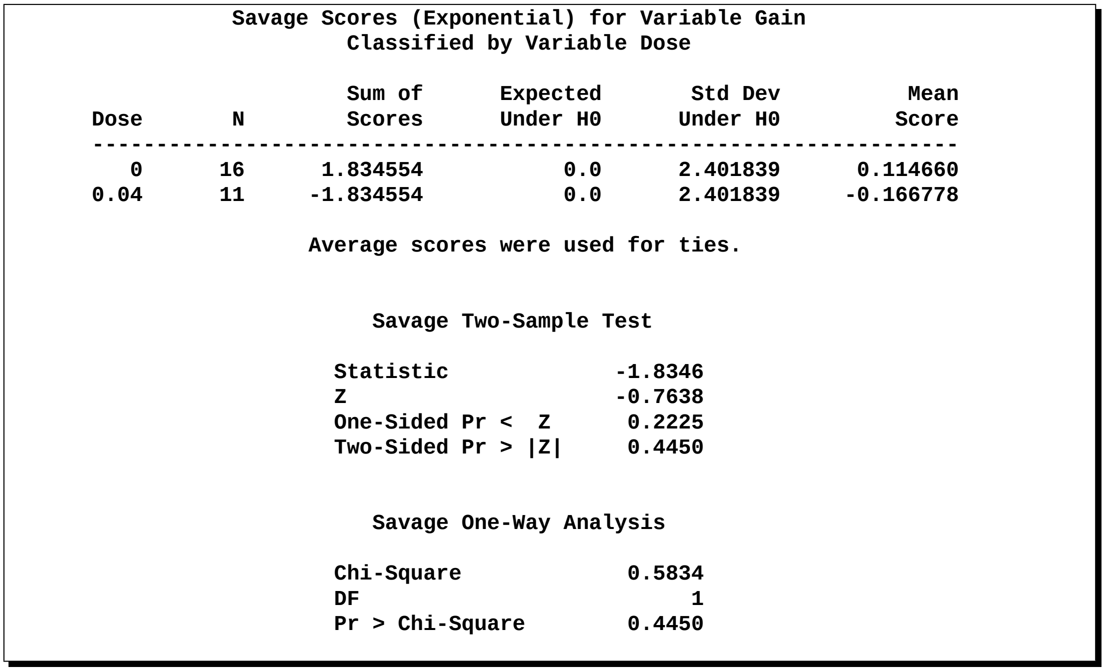
Figure 52.9 through Figure 52.11 display the two-sample analyses produced by the MEDIAN, VW, and SAVAGE options.

```
         Median Scores (Number of Points Above Median) for Variable Gain
                           Classified by Variable Dose

                             Sum of      Expected       Std Dev          Mean
      Dose        N          Scores      Under H0      Under H0         Score
      -------------------------------------------------------------------
         0       16             9.0      7.703704      1.299995      0.562500
      0.04       11             4.0      5.296296      1.299995      0.363636

                        Average scores were used for ties.


                             Median Two-Sample Test

                          Statistic              4.0000
                          Z                     -0.9972
                          One-Sided Pr <  Z      0.1593
                          Two-Sided Pr > |Z|     0.3187


                             Median One-Way Analysis

                          Chi-Square             0.9943
                          DF                          1
                          Pr > Chi-Square        0.3187
```

Figure 52.9. Median Two-Sample Analysis

```
                Van der Waerden Scores (Normal) for Variable Gain
                           Classified by Variable Dose

                             Sum of      Expected       Std Dev          Mean
      Dose        N          Scores      Under H0      Under H0         Score
      -------------------------------------------------------------------
         0       16        3.346520           0.0      2.320336      0.209157
      0.04       11       -3.346520           0.0      2.320336     -0.304229

                        Average scores were used for ties.


                         Van der Waerden Two-Sample Test

                          Statistic             -3.3465
                          Z                     -1.4423
                          One-Sided Pr <  Z      0.0746
                          Two-Sided Pr > |Z|     0.1492


                         Van der Waerden One-Way Analysis

                          Chi-Square             2.0801
                          DF                          1
                          Pr > Chi-Square        0.1492
```

Figure 52.10. Van der Waerden Two-Sample Analysis



```
                  Savage Scores (Exponential) for Variable Gain
                           Classified by Variable Dose

                                Sum of      Expected       Std Dev          Mean
      Dose          N           Scores      Under H0      Under H0         Score
      --------------------------------------------------------------------
         0         16         1.834554           0.0      2.401839      0.114660
      0.04         11        -1.834554           0.0      2.401839     -0.166778

                        Average scores were used for ties.


                             Savage Two-Sample Test

                           Statistic             -1.8346
                           Z                     -0.7638
                           One-Sided Pr <  Z      0.2225
                           Two-Sided Pr > |Z|     0.4450


                             Savage One-Way Analysis

                           Chi-Square             0.5834
                           DF                          1
                           Pr > Chi-Square        0.4450
```

Figure 52.11. Savage Two-Sample Analysis

```
    Kolmogorov-Smirnov Test for Variable Gain
           Classified by Variable Dose

                     EDF at    Deviation from Mean
Dose        N       Maximum        at Maximum
--------------------------------------------------
   0       16      0.250000         -0.481481
0.04       11      0.545455          0.580689
Total      27      0.370370

   Maximum Deviation Occurred at Observation 4
         Value of Gain at Maximum = 216.0

 Kolmogorov-Smirnov Two-Sample Test (Asymptotic)
       KS   0.145172    D         0.295455
       KSa  0.754337    Pr > KSa  0.6199


     Cramer-von Mises Test for Variable Gain
           Classified by Variable Dose

                             Summed Deviation
     Dose           N            from Mean
     ----------------------------------------
        0          16             0.098638
     0.04          11             0.143474

     Cramer-von Mises Statistics (Asymptotic)
          CM  0.008967    CMa  0.242112


          Kuiper Test for Variable Gain
           Classified by Variable Dose

                               Deviation
          Dose         N       from Mean
          ------------------------------
             0        16        0.090909
          0.04        11        0.295455

       Kuiper Two-Sample Test (Asymptotic)
  K  0.386364    Ka  0.986440    Pr > Ka  0.8383
```

Figure 52.12. Two-Sample EDF Tests

The tables in Figure 52.12 display the empirical distribution function statistics, comparing the distribution of Gain for the two levels of Dose. The *p*-value for the Kolmogorov-Smirnov two-sample test is 0.6199, which indicates no rejection of the null hypothesis that the Gain distributions are identical for the two levels of Dose.

Syntax

The following statements are available in PROC NPAR1WAY:

PROC NPAR1WAY < *options* > **;**
BY *variables* **;**
CLASS *variable* **;**
EXACT *statistic-options* < / *computation-options* > **;**
FREQ *variable* **;**
OUTPUT < **OUT=***SAS-data-set* > < *options* > **;**
VAR *variables* **;**

Both the PROC NPAR1WAY statement and the CLASS statement are required for the NPAR1WAY procedure. The rest of this section gives detailed syntax information for the BY, CLASS, EXACT, FREQ, OUTPUT, and VAR statements in alphabetical order after the description of the PROC NPAR1WAY statement. Table 52.1 summarizes the basic function of each PROC NPAR1WAY statement.

Table 52.1. Summary of PROC NPAR1WAY Statements

Statement	**Description**
BY	provides separate analyses for each BY group
CLASS	identifies the classification variable
EXACT	requests exact tests
FREQ	identifies a frequency variable
OUTPUT	requests an output data set
VAR	identifies analysis variables

PROC NPAR1WAY Statement

PROC NPAR1WAY < *options* > **;**

The PROC NPAR1WAY statement invokes the procedure and optionally identifies the input data set or requests particular analyses. By default, the procedure uses the most recently created SAS data set and omits missing values from the analysis. If you do not specify any analysis options, PROC NPAR1WAY performs an analysis of variance (option ANOVA), tests for location differences (options WILCOXON, MEDIAN, SAVAGE, and VW), and performs empirical distribution function tests (option EDF).

The following table lists the options available with the PROC NPAR1WAY statement. Descriptions follow in alphabetical order.

Table 52.2. PROC NPAR1WAY Statement Options

Task	Options
Specify the input data set	DATA=
Include missing CLASS values	MISSING
Suppress all displayed output	NOPRINT
Request analyses	AB
	ANOVA
	D
	EDF
	KLOTZ
	MEDIAN
	MOOD
	SAVAGE
	SCORES=DATA
	ST
	VW
	WILCOXON
Suppress continuity correction	CORRECT=NO

You can specify the following options in the PROC NPAR1WAY statement:

AB

requests an analysis using Ansari-Bradley scores. See the section "Ansari-Bradley Scores" on page 3168 for more information.

ANOVA

requests a standard analysis of variance on the raw data.

CORRECT=NO

suppresses the continuity correction for the Wilcoxon two-sample test and the Siegel-Tukey two-sample test. See the section "Simple Linear Rank Tests for Two-Sample Data" on page 3163 for more information.

D

requests the one-sided Kolmogorov-Smirnov $D+$ and $D-$ statistics and their asymptotic *p*-values, in addition to the two-sided D statistic produced by the EDF option for two-sample data. The D option invokes the EDF option. The statistics $D+$ and $D-$ are provided automatically if you request exact Kolmogorov-Smirnov statistics with the KS option in the EXACT statement for two-sample data. See the section "Tests Based on the Empirical Distribution Function" on page 3168 for details on Kolmogorov-Smirnov statistics.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC NPAR1WAY. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

EDF

requests statistics based on the empirical distribution function. These include the Kolmogorov-Smirnov and Cramer-von Mises tests and, if there are only two classification levels, the Kuiper test. See the section "Tests Based on the Empirical Distribution Function" on page 3168 for more information.

The EDF option produces the Kolmogorov-Smirnov D statistic for two-sample data. You can also request the one-sided $D+$ and $D-$ statistics for two-sample data with the D option.

KLOTZ

requests an analysis using Klotz scores. See the section "Klotz Scores" on page 3168 for more information.

MEDIAN

requests an analysis using median scores. When there are two classification levels, or two samples, this option produces the two-sample median test. When there are more than two samples, this option produces the multisample median test, which is also known as the Brown-Mood test. See the section "Median Scores" on page 3167 for more information.

MISSING

treats missing values of the CLASS variable as a valid class level.

MOOD

requests an analysis using Mood scores. See the section "Mood Scores" on page 3168 for more information.

NOPRINT

suppresses the display of all output. You can use the NOPRINT option when you only want to create an output data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

SAVAGE

requests an analysis using Savage scores. See the section "Savage Scores" on page 3167 for more information.

SCORES=DATA

requests an analysis using input data as scores. This option gives you the flexibility to construct any scores for your data with the DATA step and then analyze these scores with PROC NPAR1WAY. See the section "Scores for Linear Rank and One-Way ANOVA Tests" on page 3166 for more information. Using the SCORES=DATA option for raw (unscored) two-sample data produces a permutation test known as Pitman's test.

ST

requests an analysis using Siegel-Tukey scores. See the section "Siegel-Tukey Scores" on page 3167 for more information.

VW

requests an analysis using Van der Waerden scores. See the section "Van der Waerden Scores" on page 3167 for more information.

WILCOXON

requests an analysis using Wilcoxon scores. When there are two classification levels, or two samples, this option produces the Wilcoxon rank-sum test. For any number of classification levels, this option produces the Kruskal-Wallis test. See the section "Wilcoxon Scores" on page 3166 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NPAR1WAY to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NPAR1WAY procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The CLASS statement, which is required, names one and only one classification variable. The variable can be character or numeric. The CLASS variable identifies groups (or samples) in the data, and PROC NPAR1WAY provides analyses to examine differences among these groups. There may be two or more groups in the data.

EXACT Statement

EXACT *statistic-options* < / *computation-options* > **;**

The EXACT statement requests exact tests for the specified statistics. Optionally, PROC NPAR1WAY computes Monte Carlo estimates of the exact *p*-values. The *statistic-options* specify the statistics for which to provide exact tests, and the *computation-options* specify options for the computation of exact statistics.

CAUTION: PROC NPAR1WAY computes exact tests with fast and efficient algorithms that are superior to direct enumeration. Exact tests are appropriate when a data set is small, sparse, skewed, or heavily tied. For some large problems, computation of exact tests may require a large amount of time and memory. Consider using asymptotic tests for such problems. Alternatively, when asymptotic methods may not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values. See the section "Computational Resources" on page 3173 for more information.

Statistic-Options

The *statistic-options* specify the statistics for which to provide exact tests.

Exact *p*-values are available for all nonparametric tests of location and scale differences produced by PROC NPAR1WAY. These include tests based on the following scores: Wilcoxon, median, Van der Waerden, Savage, Siegel-Tukey, Ansari-Bradley, Klotz, and Mood scores. Additionally, exact *p*-values are available for tests using the raw input data as scores. The procedure computes exact *p*-values when the data are classified into two levels (two-sample tests) and when the data are classified into more than two levels (multisample tests). Two-sample tests are based on simple linear rank statistics. Multisample tests are based on one-way ANOVA statistics. Exact *p*-values are also available for the two-sample Kolmogorov-Smirnov test. See the section "Exact Tests" on page 3171 for details.

Table 52.3 lists the available *statistic-options* and the exact tests computed. The option names are identical to the corresponding options in the PROC NPAR1WAY statement and the OUTPUT statement.

If you list no *statistic-options* in the EXACT statement, then PROC NPAR1WAY computes all the available exact *p*-values for those tests requested in the PROC NPAR1WAY statement.

Table 52.3. EXACT Statement Statistic-Options

Option	Exact Test Computed
AB	Ansari-Bradley Test
KLOTZ	Klotz Test
KS	Two-Sample Kolmogorov-Smirnov Test
MEDIAN	Median Test
MOOD	Mood Test
SAVAGE	Savage Test
SCORES=DATA	Test Using Input Data as Scores
ST	Siegel-Tukey Test
WILCOXON	Wilcoxon Test for Two-Sample Data Kruskal-Wallis Test for Multisample Data
VW	Van der Waerden Test

Computation-Options

The *computation-options* specify options for computation of exact statistics. You can specify the following *computation-options* in the EXACT statement:

ALPHA=α

specifies the level of the confidence limits for Monte Carlo *p*-value estimates. The value of the ALPHA= option must be between 0 and 1, and the default is 0.01. A confidence level of α produces $100(1-\alpha)$% confidence limits. The default of ALPHA=0.01 produces 99% confidence limits for the Monte Carlo estimates. The ALPHA= option invokes the MC option.

MAXTIME=*value*

specifies the maximum clock time (in seconds) that PROC NPAR1WAY can use to compute an exact *p*-value. If the procedure does not complete the computation within the specified time, the computation terminates. The value of the MAXTIME= option must be a positive number. The MAXTIME= option is valid for Monte Carlo estimation of exact *p*-values, as well as for direct exact *p*-value computation.

See the section "Computational Resources" on page 3173 for more information.

MC

requests Monte Carlo estimation of exact *p*-values, instead of direct exact *p*-value computation. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations may not be sufficient. See the section "Monte Carlo Estimation" on page 3174 for more information.

The MC option is available for all the EXACT statement *statistic-options*. The ALPHA=, N=, and SEED= options also invoke the MC option.

N=*n*

specifies the number of samples for Monte Carlo estimation. The value of the N= option must be a positive integer, and the default is 10,000 samples. Larger values of *n* produce more precise estimates of exact *p*-values. Because larger values of *n* generate more samples, the computation time increases. The N= option invokes the MC option.

POINT

requests exact point probabilities for the test statistics.

The POINT option is available for all the EXACT statement *statistic-options*. The POINT option is not available with the MC option.

SEED=*number*

specifies the initial seed for random number generation for Monte Carlo estimation. The value of the SEED= option must be an integer. If you do not specify the SEED= option, or if the SEED= value is negative or zero, PROC NPAR1WAY uses the time of day from the computer's clock to obtain the initial seed. The SEED= option invokes the MC option.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a numeric variable that provides a frequency for each observation in the DATA= data set. If you use a FREQ statement, PROC NPAR1WAY assumes that an observation occurs n times, where n is the value of the FREQ variable for that observation. The sum of the FREQ variable values represents the total number of observations, and the analysis is based on this expanded number of observations.

If the value of the FREQ variable is missing or is less than one, PROC NPAR1WAY does not use that observation in the analysis. If the value of the FREQ variable is not an integer, PROC NPAR1WAY uses only the integer portion as the frequency of the observation.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *options* > **;**

The OUTPUT statement creates a SAS data set containing statistics computed by PROC NPAR1WAY. You specify which statistics to store in the output data set, using options identical to those used in the PROC NPAR1WAY statement. The output data set contains one observation for each analysis variable named in the VAR statement. For more information on the contents of the output data set, see the section "Output Data Set" on page 3175.

Note that you can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC NPAR1WAY output. For more information, see Table 52.6 on page 3184 and Chapter 14, "Using the Output Delivery System."

You can specify the following options in the OUTPUT statement:

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

options

specifies the statistics you want in the new data set. The options are identical to those used in the PROC NPAR1WAY statement to request analyses. Table 52.4 shows the available options. When you specify one of these options in the OUTPUT statement, the output data set contains statistics from that analysis. See the section "Output Data Set" on page 3175 for a list of the output data set variables corresponding to each option.

If you do not specify any statistics options in the OUTPUT statement, then the output data set includes statistics from all analyses specified in the PROC NPAR1WAY statement.

Table 52.4. OUTPUT Statement Options

Option	Output Data Set Statistics
AB	Ansari-Bradley Test
ANOVA	Standard analysis of variance
EDF	Kolmogorov-Smirnov Test Cramer-von Mises Test Kuiper Test for Two-Sample Data
KLOTZ	Klotz Test
MEDIAN	Median Test
MOOD	Mood Test
SAVAGE	Savage Test
SCORES=DATA	Test Using Input Data as Scores
ST	Siegel-Tukey Test
WILCOXON	Wilcoxon Test for Two-Sample Data Kruskal-Wallis Test
VW	Van der Waerden Test

VAR Statement

VAR *variables* **;**

The VAR statement names the response or dependent variables to be analyzed. These variables must be numeric. If the VAR statement is omitted, the procedure analyzes all numeric variables in the data set except for the CLASS variable, the FREQ variable, and the BY variables.

Details

Missing Values

If an observation has a missing value for a response variable, PROC NPAR1WAY excludes that observation from the analysis.

By default, PROC NPAR1WAY excludes observations with missing values of the CLASS variable. If you specify the MISSING option, PROC NPAR1WAY treats missing values of the CLASS variable as a valid class level and includes these observations in the analysis.

PROC NPAR1WAY treats missing BY variable values like any other BY variable value. The missing values form a separate BY group. When a value of the FREQ variable is missing, PROC NPAR1WAY excludes the observation from the analysis.

Tied Values

Tied values occur when two are more observations are equal, whether the observations occur in the same sample or in different samples. In theory, nonparametric tests were developed for continuous distributions where the probability of a tie is zero. In practice, however, ties often occur. PROC NPAR1WAY uses the same method to handle ties for all score types. The procedure computes the scores as if there were no ties, averages the scores for tied observations, and assigns this average score to each observation with the same value.

When there are tied values, PROC NPAR1WAY first sorts the observations in ascending order and assigns ranks as if there were no ties. Then the procedure computes the scores based on these ranks, using the formula for the specified score type. The procedure averages the scores for tied observations and assigns this average score to each of the tied observations. Thus, all equal data values have the same score value. PROC NPAR1WAY then computes the test statistic from these scores.

Note that the asymptotic tests may be less accurate when the distribution of the data is heavily tied. For such data, it may be appropriate to use the exact tests provided by PROC NPAR1WAY as described in the section "Exact Tests" on page 3171.

When computing empirical distribution function statistics for data with ties, PROC NPAR1WAY uses the formulas given in the section "Tests Based on the Empirical Distribution Function" on page 3168. No special handling of ties is necessary.

Note that PROC NPAR1WAY bases its computations on the internal numeric values of the analysis variables; the procedure does not format or round these values before analysis. When values differ in their internal representation, even slightly, PROC NPAR1WAY does not treat them as tied values. If this is a concern for your data, then round the analysis variables by an appropriate amount before invoking PROC NPAR1WAY. For information on the ROUND function, refer to the discussion in *SAS Language Reference: Dictionary*.

Statistical Computations

Simple Linear Rank Tests for Two-Sample Data

Statistics of the form

$$S \quad = \quad \sum_{j=1}^{n} c_j \; a(R_j)$$

are called *simple linear rank statistics*, where

R_j is the rank of the observation j

$a(R_j)$ is the score based on that rank

c_j is an indicator variable denoting the class to which the jth observation belongs

n is the total number of observations

For two-sample data (where the observations are classified into two levels), PROC NPAR1WAY calculates simple linear rank statistics for the scores that you specify. The section "Scores for Linear Rank and One-Way ANOVA Tests" on page 3166 describes the available scores, which you can use to test for differences in location and differences in scale.

To compute S, PROC NPAR1WAY sums the scores of the observations in the smaller of the two samples. If both samples have the same number of observations, PROC NPAR1WAY sums those scores for the sample that appears first in the input data set.

For each score that you specify, PROC NPAR1WAY computes an asymptotic test of the null hypothesis of no difference between the two classification levels. Exact tests are also available for these two-sample linear rank statistics. PROC NPAR1WAY computes exact tests for each score type that you specify in the EXACT statement. See the section "Exact Tests" on page 3171 for details.

To compute an asymptotic test for a linear rank sum statistic, PROC NPAR1WAY uses a standardized test statistic z, which has an asymptotic standard normal distribution under the null hypothesis. The standardized test statistic is computed as

$$z \quad = \quad \frac{S - E_0(S)}{\sqrt{Var_0(S)}}$$

where $E_0(S)$ is the expected value of S under the null hypothesis, and $Var_0(S)$ is the variance under the null hypothesis. As shown in Randles and Wolfe (1979),

$$E_0(S) \quad = \quad \frac{n_1}{n} \sum_{j=1}^{n} a(R_j)$$

where n_1 is the number of observations in the first (smaller) class level or sample, n_2 is the number of observations in the other class level, and

$$Var_0(S) \quad = \quad \frac{1}{(n-1)} \; \frac{n_1 \, n_2}{n} \; \left[\sum_{j=1}^{n} (a(R_j) - \bar{a})^2 \right]$$

where $\bar{a}$ is the average score,

$$\bar{a} \quad = \quad \sum_{j=1}^{n} a(R_j) \; / \; n$$

PROC NPAR1WAY computes one-sided and two-sided asymptotic p-values for each two-sample linear rank test. When the test statistic z is greater than its null hypothesis expected value of zero, PROC NPAR1WAY computes the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. When the test statistic is less than or equal to zero, PROC NPAR1WAY computes the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. The one-sided p-value P_1 can be expressed as

$$P_1 \quad = \quad \text{Prob} \; (\, Z > z \,) \qquad \text{if} \quad z > 0$$

$$P_1 \quad = \quad \text{Prob} \; (\, Z < z \,) \qquad \text{if} \quad z \leq 0$$

where Z has a standard normal distribution. The two-sided p-value P_2 is computed as

$$P_2 \quad = \quad \text{Prob} \; (\, |Z| > |z| \,)$$

For Wilcoxon scores and Siegel-Tukey scores, PROC NPAR1WAY incorporates a continuity correction when computing the standardized test statistic z, unless you specify the CORRECT=NO option. PROC NPAR1WAY applies the continuity correction by subtracting 0.5 from the numerator $S - E_0(S)$ if it is greater than zero. If the numerator is less than zero, PROC NPAR1WAY adds 0.5. Some sources recommend a continuity correction for nonparametric tests that use a continuous distribution to approximate a discrete distribution. Refer to Sheskin (1997). If you specify CORRECT=NO, PROC NPAR1WAY does not use a continuity correction for any test.

One-Way ANOVA Tests

PROC NPAR1WAY computes a one-way ANOVA test for each score type that you specify. Under the null hypothesis of no difference among class levels (or samples), this test statistic has an asymptotic chi-square distribution with $r - 1$ degrees of freedom, where r is the number of class levels. For Wilcoxon scores, this test is known as the Kruskal-Wallis test.

Exact one-way ANOVA tests are also available for multisample data (where the data are classified into more than two levels). For two-sample data, exact simple linear rank tests are available. PROC NPAR1WAY computes exact tests for each score type that you specify in the EXACT statement. See the section “Exact Tests” on page 3171 for details on exact tests.

PROC NPAR1WAY computes the one-way ANOVA test statistic as

$$C \quad = \quad \left(\sum_{i=1}^{r} (T_i - E_0(T_i))^2 \; / \; n_i \right) \; / \; S^2$$

where T_i is the total of scores for the class level i, $E_0(T_i)$ is the expected total for level i under the null hypothesis of no difference among levels, n_i is the number of observations in level i, and S^2 is the sample variance of the scores.

$$T_i \quad = \quad \sum_{j=1}^{n} c_{ij} \; a(R_j)$$

where $a(R_j)$ is the score for observation j, and c_{ij} indicates whether observation j is in level i.

$$E_0(T_i) \quad = \quad \frac{n_i}{n} \sum_{j=1}^{n} a(R_j)$$

$$S^2 \quad = \quad \frac{1}{(n-1)} \left[\sum_{j=1}^{n} \left(a(R_j) - \bar{a} \right)^2 \right]$$

where $\bar{a}$ is the average score,

$$\bar{a} \quad = \quad \sum_{j=1}^{n} a(R_j) \; / \; n$$

Scores for Linear Rank and One-Way ANOVA Tests

For each score type that you specify, PROC NPAR1WAY computes a one-way ANOVA statistic and also a linear rank statistic for two-sample data. The following score types are used primarily to test for differences in location: Wilcoxon, median, Van der Waerden, and Savage. The following scores types are used to test for scale differences: Siegel-Tukey, Ansari-Bradley, Klotz, and Mood. This section gives formulas for the score types. For further information on the formulas and the applicability of each score, refer to Randles and Wolfe (1979), Gibbons and Chakraborti (1992), Conover (1999), and Hollander and Wolfe (1999).

In addition to the score types described in this section, you can specify the SCORES=DATA option to use the input data observations as scores. This enables you to produce a very wide variety of tests. You can construct any scores using the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank and one-way ANOVA tests. You can also analyze the raw data with the SCORES=DATA option; for two-sample data, this permutation test is known as Pitman's test.

Wilcoxon Scores

Wilcoxon scores are the ranks of the observations.

$$a(R_j) = R_j$$

Using Wilcoxon scores in the linear rank statistic for two-sample data produces the rank sum statistic of the Mann-Whitney-Wilcoxon test. Using Wilcoxon scores in the one-way ANOVA statistic produces the Kruskal-Wallis test. Wilcoxon scores are locally most powerful for location shifts of a logistic distribution.

When computing the asymptotic Wilcoxon two-sample test, PROC NPAR1WAY uses a continuity correction by default, as described in the section "Simple Linear Rank Tests for Two-Sample Data" on page 3163. If you specify CORRECT=NO in the PROC NPAR1WAY statement, the procedure does not use a continuity correction.

Median Scores

Median scores equal 1 for observations greater than the median, and 0 otherwise.

$$a(R_j) = \begin{cases} 1 & \text{if } R_j > \frac{n+1}{2} \\ 0 & \text{if } R_j \le \frac{n+1}{2} \end{cases}$$

Using median scores in the linear rank statistic for two-sample data produces the two-sample median test. The one-way ANOVA statistic with median scores is equivalent to the Brown-Mood test. Median scores are particularly powerful for distributions that are symmetric and heavy-tailed.

Van der Waerden Scores

Van der Waerden scores are the quantiles of a standard normal distribution. These scores are also known as *quantile normal scores*.

$$a(R_j) = \Phi^{-1}\left(\frac{R_j}{n+1}\right)$$

where Φ is the cumulative distribution function of a standard normal distribution. These scores are powerful for normal distributions.

Savage Scores

Savage scores are expected values of order statistics from the exponential distribution, with 1 subtracted to center the scores around 0.

$$a(R_j) = \sum_{i=1}^{R_j} \frac{1}{n-i+1} - 1$$

Savage scores are powerful for comparing scale differences in exponential distributions or location shifts in extreme value distributions (Hajek 1969, p. 83).

Siegel-Tukey Scores

Siegel-Tukey scores are computed as

$$\begin{array}{llll} a(1) = 1, & a(n) = 2, & a(n-1) = 3, & a(2) = 4, \\ a(3) = 5, & a(n-2) = 6, & a(n-3) = 7, & a(4) = 8, \ldots \end{array}$$

where the score values continue to increase in this pattern towards the middle ranks until all observations have been assigned a score.

Ansari-Bradley Scores

Ansari-Bradley scores are similar to Siegel-Tukey scores, but Ansari-Bradley assigns the same scores to corresponding extreme ranks. (Siegel Tukey scores are just a permutation of the ranks $1, 2, \ldots, n$.)

$$\begin{aligned} &a(1) = 1, \quad a(n) = 1, \\ &a(2) = 2, \quad a(n-1) = 2, \ldots \end{aligned}$$

Equivalently, Ansari-Bradley scores are defined as

$$a(R_j) = \frac{n+1}{2} - \left| R_j - \frac{n+1}{2} \right|$$

Klotz Scores

Klotz scores are the squares of the Van der Waerden (or quantile normal) scores.

$$a(R_j) = \left[\Phi^{-1} \left(\frac{R_j}{n+1} \right) \right]^2$$

where Φ is the cumulative distribution function of a standard normal distribution.

Mood Scores

Mood scores are computed as the square of the difference between each rank and the average rank.

$$a(R_j) = \left[R_j - \frac{n+1}{2} \right]^2$$

Tests Based on the Empirical Distribution Function

If you specify the EDF option, PROC NPAR1WAY computes tests based on the empirical distribution function. These include the Kolmogorov-Smirnov and Cramer-von Mises tests, and also the Kuiper test for two-sample data. This section gives formulas for these test statistics. For further information on the formulas and the interpretation of EDF statistics, refer to Hollander and Wolfe (1999) and Gibbons and Chakraborti (1992). For details on the k-sample analogues of the Kolmogorov-Smirnov and Cramer-von Mises statistics used by NPAR1WAY, refer to Kiefer (1959).

The *empirical distribution function* (EDF) of a sample $\{x_j\}$, $j = 1, 2, \ldots, n$, is defined as the following function:

$$F(x) \quad = \quad \frac{1}{n} \, (\text{number of } x_j \leq x) \quad = \quad \frac{1}{n} \sum_{j=1}^{n} I(x_j \leq x)$$

where $I(\cdot)$ is an indicator function. PROC NPAR1WAY uses the subsample of values within the ith class level to generate an EDF for the class, F_i. The EDF for the overall sample, pooled over classes, can also be expressed as

$$F \;=\; \frac{1}{n} \sum_i (\, n_i \, F_i \,)$$

where n_i is the number of observations in the ith class level, and n is the total number of observations.

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov statistic measures the maximum deviation of the EDF within the classes from the pooled EDF. PROC NPAR1WAY computes the Kolmogorov-Smirnov statistic as

$$\mathrm{KS} = \max_j \sqrt{\frac{1}{n} \sum_i n_i \left(\, F_i(x_j) - F(x_j) \,\right)^2} \quad \text{where } j = 1, 2, \ldots, n$$

The asymptotic Kolmogorov-Smirnov statistic is computed as

$$\mathrm{KS}_a = \mathrm{KS} \;\times\; \sqrt{n}$$

For each class level i and overall, PROC NPAR1WAY displays the value of F_i at the maximum deviation from F and the value $(F_i - F)\,\sqrt{n_i}$ at the maximum deviation from F. PROC NPAR1WAY also gives the observation where the maximum deviation occurs.

If there are only two class levels, PROC NPAR1WAY computes the two-sample Kolmogorov-Smirnov test statistic D as

$$D = \max_j \left|\, F_1(x_j) - F_2(x_j) \,\right| \quad \text{where } j = 1, 2, \ldots, n$$

The *p*-value for this test is the probability that D is greater than the observed value d under the null hypothesis of no difference between class levels or samples. PROC NPAR1WAY computes the asymptotic *p*-value for D with the approximation

$$\mathrm{Prob}\;(\, D > d \,) \;=\; 1 - 2 \sum_{i=1}^{\infty} (-1)^{\,(i-1)} \, e^{\,(-2i^2 \; z^2)}$$

where

$$z \;=\; d \sqrt{\left(\frac{n_1 \, n_2}{n} \right)}$$

The quality of this approximation has been studied by Hodges (1957).

If you specify the D option, or if you request exact Kolmogorov-Smirnov *p*-values with the KS option in the EXACT statement, PROC NPAR1WAY also computes the one-sided Kolmogorov-Smirnov statistics $D+$ and $D-$ for two-sample data.

$$D+ \;=\; \max_j \,(\, F_1(x_j) - F_2(x_j)\,) \quad \text{where } j = 1, 2, \ldots, n$$

$$D- \;=\; \max_j \,(\, F_2(x_j) - F_1(x_j)\,) \quad \text{where } j = 1, 2, \ldots, n$$

The asymptotic probability that $D+$ is greater than the observed value d^+, under the null hypothesis of no difference between the two class levels, is computed as

$$\text{Prob}\;(\; D+ \;>\; d^+ \;) \;=\; e^{-2z^2} \;, \quad \text{where} \quad z \;=\; d^+ \sqrt{\left(\frac{n_1\, n_2}{n} \right)}$$

Similarly, the asymptotic probability that $D-$ is greater than the observed value d^- is computed as

$$\text{Prob}\;(\; D- \;>\; d^- \;) \;=\; e^{-2z^2} \;, \quad \text{where} \quad z \;=\; d^- \sqrt{\left(\frac{n_1\, n_2}{n} \right)}$$

To request exact *p*-values for the Kolmogorov-Smirnov statistics, you can specify the KS option in the EXACT statement. See the section "Exact Tests" on page 3171 for more information.

Cramer-von Mises Test

The Cramer-von Mises statistic is defined as

$$\text{CM} = \frac{1}{n^2} \sum_i \left(n_i \sum_{j=1}^{p} t_j \;(\, F_i(x_j) - F(x_j\,)\,)^2 \right)$$

where t_j is the number of ties at the jth distinct value and p is the number of distinct values. The asymptotic value is computed as

$$\text{CM}_a = \text{CM} \;\times\; n$$

PROC NPAR1WAY displays the contribution of each class level to the sum CM_a.

Kuiper Test

For data with two class levels, PROC NPAR1WAY computes the Kuiper statistic, its scaled value for the asymptotic distribution, and the asymptotic p-value. The Kuiper statistic is computed as

$$K = \max_j \left(F_1(x_j) - F_2(x_j)\right) - \min_j \left(F_1(x_j) - F_2(x_j)\right) \quad \text{where } j = 1, 2, \ldots, n$$

The asymptotic value is

$$K_a = K \sqrt{\frac{n_1\, n_2}{n}}$$

PROC NPAR1WAY displays $\max_j |F_1(x_j) - F_2(x_j)|$ for each class level.

The p-value for the Kuiper test is the probability of observing a larger value of K_a under the null hypothesis of no difference between the two classes. PROC NPAR1WAY computes this p-value according to Owen (1962), p. 441.

Exact Tests

PROC NPAR1WAY provides exact p-values for tests for location and scale differences based on the following scores: Wilcoxon, median, van der Waerden, Savage, Siegel-Tukey, Ansari-Bradley, Klotz, and Mood scores. Additionally, PROC NPAR1WAY provides exact p-values for tests using the raw data as scores. Exact tests are available for two-sample and multisample data. When the data are classified into two samples, tests are based on simple linear rank statistics. When the data are classified into more than two samples, tests are based on one-way ANOVA statistics.

Exact tests can be useful in situations where the asymptotic assumptions are not met and the asymptotic p-values are not close approximations for the true p-values. Standard asymptotic methods involve the assumption that the test statistic follows a particular distribution when the sample size is sufficiently large. When the sample size is not large, asymptotic results may not be valid, with the asymptotic p-values differing perhaps substantially from the exact p-values. Asymptotic results may also be unreliable when the distribution of the data is sparse, skewed, or heavily tied. Refer to Agresti (1996) and Bishop, Fienberg, and Holland (1975). Exact computations are based on the statistical theory of exact conditional inference for contingency tables, reviewed by Agresti (1992).

In addition to computation of exact p-values, PROC NPAR1WAY provides the option of estimating exact p-values by Monte Carlo simulation. This can be useful for problems that are so large that exact computations require a great amount of time and memory, but for which asymptotic approximations may not be sufficient.

The following sections summarize the exact computational algorithms, define the exact p-values that PROC NPAR1WAY computes, discuss the computational resource requirements, and describe the Monte Carlo estimation option.

Computational Algorithms

PROC NPAR1WAY computes exact p-values using the network algorithm developed by Mehta and Patel (1983). This algorithm provides a substantial advantage over direct enumeration, which can be very time consuming and feasible only for small problems. Refer to Agresti (1992) for a review of algorithms for computation of exact p-values, and refer to Mehta, Patel, and Tsiatis (1984) and Mehta, Patel, and Senchaudhuri (1991) for information on the performance of the network algorithm.

PROC NPAR1WAY constructs a contingency table from the input data, with rows formed by the levels of the classification variable and columns formed by the response variable values. The reference set for a given contingency table is the set of all contingency tables with the observed marginal row and column sums. Corresponding to this reference set, the network algorithm forms a directed acyclic network consisting of nodes in a number of stages. A path through the network corresponds to a distinct table in the reference set. The distances between nodes are defined so that the total distance of a path through the network is the corresponding value of the test statistic. At each node, the algorithm computes the shortest and longest path distances for all the paths that pass through that node. For the two-sample linear rank statistics, which can be expressed as a linear combination of cell frequencies multiplied by increasing row and column scores, PROC NPAR1WAY computes shortest and longest path distances using the algorithm given in Agresti, Mehta, and Patel (1990). For the multisample one-way test statistics, PROC NPAR1WAY computes an upper bound for the longest path and a lower bound for the shortest path, following the approach of Valz and Thompson (1994).

The longest and shortest path distances or bounds for a node are compared to the value of the test statistic to determine whether all paths through the node contribute to the p-value, none of the paths through the node contribute to the p-value, or neither of these situations occur. If all paths through the node contribute, the p-value is incremented accordingly, and these paths are eliminated from further analysis. If no paths contribute, these paths are eliminated from the analysis. Otherwise, the algorithm continues, still processing this node and the associated paths. The algorithm finishes when all nodes have been accounted for.

In applying the network algorithm, PROC NPAR1WAY uses full precision to represent all statistics, row and column scores, and other quantities involved in the computations. Although it is possible to use rounding to improve the speed and memory requirements of the algorithm, PROC NPAR1WAY does not do this since it can result in reduced accuracy of the p-values.

Definition of p-Values

For two-sample linear rank tests, PROC NPAR1WAY computes exact one-sided and two-sided p-values for each test specified in the EXACT statement. For the one-sided test, PROC NPAR1WAY displays the right-sided p-value when the observed value of the test statistic is greater than its expected value. The right-sided p-value is the sum of probabilities for those tables having a test statistic greater than or equal to the observed test statistic. Otherwise, when the test statistic is less than or equal to its expected value, PROC NPAR1WAY displays the left-sided p-value. The left-sided

p-value is the sum of probabilities for those tables having a test statistic less than or equal to the one observed. The one-sided p-value P_1 can be expressed as

$$\begin{aligned} P_1 &= \text{Prob}\,(\text{Test Statistic} \geq S) \qquad \text{if} \quad S > \text{Mean} \\ P_1 &= \text{Prob}\,(\text{Test Statistic} \leq S) \qquad \text{if} \quad S \leq \text{Mean} \end{aligned}$$

where S is the observed value of the test statistic and Mean is the expected value of the test statistic under the null hypothesis. PROC NPAR1WAY computes the two-sided p-value as the sum of the one-sided p-value and the corresponding area in the opposite tail of the distribution of the statistic, equidistant from the expected value. The two-sided p-value P_2 can be expressed as

$$P_2 = \text{Prob}\;(\,|\,\text{Test Statistic} - \text{Mean}\,| \geq |\,S - \text{Mean}\,|\,)$$

For multisample data, the tests are based on one-way ANOVA statistics. For a test of this form, large values of the test statistic indicate a departure from the null hypothesis; the test is inherently two-sided. The exact p-value is the sum of probabilities for those tables having a test statistic greater than or equal to the value of the observed test statistic.

If you specify the POINT option in the EXACT statement, PROC NPAR1WAY also displays exact point probabilities for the test statistics. The exact point probability is the exact probability that the test statistic equals the observed value.

Computational Resources

PROC NPAR1WAY uses relatively fast and efficient algorithms for exact computations. These recently developed algorithms, together with improvements in computer power, make it feasible now to perform exact computations for data sets where previously only asymptotic methods could be applied. Nevertheless, there are still large problems that may require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For large problems, consider whether exact methods are really needed or whether asymptotic methods might give results quite close to the exact results while requiring much less computer time and memory. When asymptotic methods may not be sufficient for such large problems, consider using Monte Carlo estimation of exact p-values, as described in the section "Monte Carlo Estimation" on page 3174.

A formula does not exist that can predict in advance how much time and memory are needed to compute an exact p-value for a certain problem. The time and memory required depend on several factors, including which test is being performed, the total sample size, the number of rows and columns, and the specific arrangement of the observations into table cells. Generally, larger problems (in terms of total sample size, number of rows, and number of columns) tend to require more time and memory. Additionally, for a fixed total sample size, time and memory requirements tend to increase as the number of rows and columns increase, since this corresponds to an increase in the number of tables in the reference set. Also for a fixed sample size, time and memory requirements increase as the marginal row and column totals become more homogeneous. Refer to Agresti, Mehta, and Patel (1990) and Gail and Mantel (1977).

At any time while PROC NPAR1WAY is computing exact p-values, you can terminate the computations by pressing the system interrupt key sequence (refer to the *SAS Companion* for your system) and choosing to stop computations. After you terminate exact computations, PROC NPAR1WAY completes all other remaining tasks. The procedure produces the requested output and reports missing values for any exact p-values not computed by the time of termination.

You can also use the MAXTIME= option in the EXACT statement to limit the amount of time PROC NPAR1WAY uses for exact computations. You specify a MAXTIME= value that is the maximum amount of time (in seconds) that PROC NPAR1WAY can use to compute an exact p-value. If PROC NPAR1WAY does not finish computing an exact p-value within that time, it terminates the computation and completes all other remaining tasks.

Monte Carlo Estimation

If you specify the MC option in the EXACT statement, PROC NPAR1WAY computes Monte Carlo estimates of the exact p-values instead of directly computing the exact p-values. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations may not be sufficient. To describe the precision of each Monte Carlo estimate, PROC NPAR1WAY provides the asymptotic standard error and $100(1-\alpha)\%$ confidence limits. The confidence level α is determined by the ALPHA= option in the EXACT statement, which, by default, equals 0.01, and produces 99% confidence limits. The N= option in the EXACT statement specifies the number of samples PROC NPAR1WAY uses for Monte Carlo estimation; the default is 10,000 samples. You can specify a larger value for n to improve the precision of the Monte Carlo estimates. Because larger values of n generate more samples, the computation time increases. Or you can specify a smaller value of n to reduce the computation time.

To compute a Monte Carlo estimate of an exact p-value, PROC NPAR1WAY generates a random sample of tables with the same total sample size, row totals, and column totals as the observed table. PROC NPAR1WAY uses the algorithm of Agresti, Wackerly, and Boyett (1979), which generates tables in proportion to their hypergeometric probabilities conditional on the marginal frequencies. For each sample table, PROC NPAR1WAY computes the value of the test statistic and compares it to the value for the observed table. When estimating a right-sided p-value, PROC NPAR1WAY counts all sample tables for which the test statistic is greater than or equal to the observed test statistic. Then the p-value estimate equals the number of these tables divided by the total number of tables sampled.

$$\begin{aligned}
\hat{P}_{\text{MC}} &= M \;/\; N \\
M &= \text{number of samples with } (\text{Test Statistic} \geq t) \\
N &= \text{total number of samples} \\
t &= \text{observed Test Statistic}
\end{aligned}$$

PROC NPAR1WAY computes left-sided and two-sided p-value estimates in a similar manner. For left-sided p-values, PROC NPAR1WAY evaluates whether the test statistic for each sampled table is less than or equal to the observed test statistic. For two-sided p-values, PROC NPAR1WAY examines the sample test statistics according to the expression for P_2 given in the section "Definition of p-Values" on page 3172.

The variable M is a binomial variable with N trials and success probability p. It follows that the asymptotic standard error of the Monte Carlo estimate is

$$se(\hat{P}_{\mathrm{MC}}) \quad = \quad \sqrt{\hat{P}_{\mathrm{MC}} \; (1 - \hat{P}_{\mathrm{MC}}) \; / \; (N-1)}$$

PROC NPAR1WAY constructs asymptotic confidence limits for the p-values according to

$$\hat{P}_{\mathrm{MC}} \quad \pm \quad z_{\alpha/2} \;\times\; se(\hat{P}_{\mathrm{MC}})$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the standard normal distribution, and the confidence level α is determined by the ALPHA= option in the EXACT statement.

When the Monte Carlo estimate $\hat{P}_{\mathrm{MC}}$ equals 0, then PROC NPAR1WAY computes the confidence limits for the p-value as

$$(\; 0, \;\; 1 - \alpha^{(1/N)} \;)$$

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 1, then PROC NPAR1WAY computes the confidence limits as

$$(\; \alpha^{(1/N)}, \;\; 1 \;)$$

Output Data Set

The OUTPUT statement creates a SAS data set that contains statistics computed by PROC NPAR1WAY. You specify which statistics to store in the output data set, using options identical to those used in the PROC NPAR1WAY statement. When you specify one of these options in the OUTPUT statement, PROC NPAR1WAY includes all available statistics from that analysis in the output data set.

The output data set contains one observation for each analysis variable within a BY-group. The OUTPUT data set can include the following variables:

- BY variables
- _VAR_, which identifies the analysis variable
- variables containing the specified statistics

The following table lists the variable names and descriptions for all available statistics. Note that some statistics are available only for the two-sample case (where the classification variable groups the data into two classes). Other statistics are available only for the multisample case.

When you request exact p-values for certain analyses using the EXACT statement, PROC NPAR1WAY also includes those p-values in the output data set if you specify the corresponding analysis options in the OUTPUT statement. If you do not request exact p-values, then they do not appear in the output data set.

Monte Carlo estimates of exact p-values are not available in this output data set, but you can use the Output Delivery System (ODS) to store Monte Carlo estimates in a SAS data set. You can use the Output Delivery System to create a SAS data set from any piece of PROC NPAR1WAY output. For more information, see Table 52.6 on page 3184 and Chapter 14, "Using the Output Delivery System."

Table 52.5. Output Data Set Variable Names and Descriptions

Option	**Output Variables**		**Variable Descriptions**
ANOVA	_MSA_		ANOVA Effect Mean Square, Among MS
	MSE		ANOVA Error Mean Square, Within MS
	_F_		F Statistic for ANOVA
	P_F		p-value, F Statistic for ANOVA
WILCOXON	_WIL_	*	Two-sample Wilcoxon Statistic
	Z_WIL	*	Wilcoxon Statistic, Standardized
	PL_WIL	*	p-value, Wilcoxon Test (Left-sided)
	PR_WIL	*	p-value, Wilcoxon Test (Right-sided)
	P2_WIL	*	p-value, Wilcoxon Test (Two-sided)
	PTL_WIL	*	p-value, Wilcoxon t Approximation (Left-sided)
	PTR_WIL	*	p-value, Wilcoxon t Approximation, (Right-sided)
	PT2_WIL	*	p-value, Wilcoxon t Approximation, (Two-sided)
	XPL_WIL	*	Exact p-value, Wilcoxon Test (Left-sided)
	XPR_WIL	*	Exact p-value, Wilcoxon Test (Right-sided)
	XPT_WIL	*	Exact Point Probability, Wilcoxon Test
	XP2_WIL	*	Exact p-value, Wilcoxon Test (Two-sided)
	_KW_		Kruskal-Wallis Statistic
	DF_KW		Degrees of Freedom, Kruskal-Wallis Test
	P_KW		p-value, Kruskal-Wallis Test
	XP_KW	**	Exact p-value, Kruskal-Wallis Test
	XPT_KW	**	Exact Point Probability, Kruskal-Wallis Test
MEDIAN	_MED_	*	Two-sample Median Statistic
	Z_MED	*	Median Statistic, Standardized
	PL_MED	*	p-value, Median Test (Left-sided)
	PR_MED	*	p-value, Median Test (Right-sided)
	P2_MED	*	p-value, Median Test (Two-sided)
	XPL_MED	*	Exact p-value, Median Test (Left-sided)
	XPR_MED	*	Exact p-value, Median Test (Right-sided)
	XPT_MED	*	Exact Point Probability, Median Test
	XP2_MED	*	Exact p-value, Median Test (Two-sided)

Table 52.5. (continued)

Option	Output Variables		Variable Descriptions
MEDIAN	_CHMED_		Median Chi-square (Brown-Mood Test)
	DF_CHMED		Degrees of Freedom, Median Chi-square
	P_CHMED		p-value, Median Chi-square Test
	XP_CHMED	**	Exact p-value, Median Chi-square
	XPT_CHME	**	Exact Point Probability, Median Chi-square
VW	_VW_	*	Two-sample Van der Waerden Statistic
	Z_VW	*	Van der Waerden Statistic, Standardized
	PL_VW	*	p-value, Van der Waerden Test (Left-sided)
	PR_VW	*	p-value, Van der Waerden Test (Right-sided)
	P2_VW	*	p-value, Van der Waerden Test (Two-sided)
	XPL_VW	*	Exact p-value, Van der Waerden Test (Left-sided)
	XPR_VW	*	Exact p-value, Van der Waerden Test (Right-sided)
	XPT_VW	*	Exact Point Probability, Van der Waerden Test
	XP2_VW	*	Exact p-value, Van der Waerden Test (Two-sided)
	CHVW		Van der Waerden Chi-square
	DF_CHVW		Degrees of Freedom, Van der Waerden Chi-square
	P_CHVW		p-value, Van der Waerden Chi-square Test
	XP_CHVW	**	Exact p-value, Van der Waerden Chi-square
	XPT_CHVW	**	Exact Point Prob, Van der Waerden Chi-square
SAVAGE	_SAV_	*	Two-sample Savage Statistic
	Z_SAV	*	Savage Statistic, Standardized
	PL_SAV	*	p-value, Savage Test (Left-sided)
	PR_SAV	*	p-value, Savage Test (Right-sided)
	P2_SAV	*	p-value, Savage Test (Two-sided)
	XPL_SAV	*	Exact p-value, Savage Test (Left-sided)
	XPR_SAV	*	Exact p-value, Savage Test (Right-sided)
	XPT_SAV	*	Exact Point Probability, Savage Test
	XP2_SAV	*	Exact p-value, Savage Test (Two-sided)
	CHSAV		Savage Chi-square
	DF_CHSAV		Degrees of Freedom, Savage Chi-square
	P_CHSAV		p-value, Savage Chi-square Test
	XP_CHSAV	**	Exact p-value, Savage Chi-square
	XPT_CHSA	**	Exact Point Probability, Savage Chi-square
ST	_ST_	*	Two-sample Siegel-Tukey Statistic
	Z_ST	*	Siegel-Tukey Statistic, Standardized
	PL_ST	*	p-value, Siegel-Tukey Test (Left-sided)
	PR_ST	*	p-value, Siegel-Tukey Test (Right-sided)
	P2_ST	*	p-value, Siegel-Tukey Test (Two-sided)
	XPL_ST	*	Exact p-value, Siegel-Tukey Test (Left-sided)
	XPR_ST	*	Exact p-value, Siegel-Tukey Test (Right-sided)
	XPT_ST	*	Exact Point Probability, Siegel-Tukey Test
	XP2_ST	*	Exact p-value, Siegel-Tukey Test (Two-sided)

Table 52.5. (continued)

Option	Output Variables		Variable Descriptions
ST	_CHST_		Siegel-Tukey Chi-square
	DF_CHST		Degrees of Freedom, Siegel-Tukey Chi-square
	P_CHST		p-value, Siegel-Tukey Chi-square Test
	XP_CHST	**	Exact p-value, Siegel-Tukey Chi-square
	XPT_CHST	**	Exact Point Probability, Siegel-Tukey Chi-square
AB	_AB_	*	Two-sample Ansari-Bradley Statistic
	Z_AB	*	Ansari-Bradley Statistic, Standardized
	PL_AB	*	p-value, Ansari-Bradley Test (Left-sided)
	PR_AB	*	p-value, Ansari-Bradley Test (Right-sided)
	P2_AB	*	p-value, Ansari-Bradley Test (Two-sided)
	XPL_AB	*	Exact p-value, Ansari-Bradley Test (Left-sided)
	XPR_AB	*	Exact p-value, Ansari-Bradley Test (Right-sided)
	XPT_AB	*	Exact Point Probability, Ansari-Bradley Test
	XP2_AB	*	Exact p-value, Ansari-Bradley Test (Two-sided)
	CHAB		Ansari Bradley Chi-square
	DF_CHAB		Degrees of Freedom, Ansari-Bradley Chi-square
	P_CHAB		p-value, Ansari-Bradley Chi-square Test
	XP_CHAB	**	Exact p-value, Ansari-Bradley Chi-square
	XPT_CHAB	**	Exact Point Probability, Ansari-Bradley Chi-square
KLOTZ	_KLOTZ_	*	Two-sample Klotz Statistic
	Z_K	*	Klotz Statistic, Standardized
	PL_K	*	p-value, Klotz Test (Left-sided)
	PR_K	*	p-value, Klotz Test (Right-sided)
	P2_K	*	p-value, Klotz Test (Two-sided)
	XPL_K	*	Exact p-value, Klotz Test (Left-sided)
	XPR_K	*	Exact p-value, Klotz Test (Right-sided)
	XPT_K	*	Exact Point Probability, Klotz Test
	XP2_K	*	Exact p-value, Klotz Test (Two-sided)
	CHK		Klotz Chi-square
	DF_CHK		Degrees of Freedom, Klotz Chi-square
	P_CHK		p-value, Klotz Chi-square Test
	XP_CHK	**	Exact p-value, Klotz Chi-square
	XPT_CHK	**	Exact Point Probability, Klotz Chi-square
MOOD	_MOOD_	*	Two-sample Mood Statistic
	Z_MOOD	*	Mood Statistic, Standardized
	PL_MOOD	*	p-value, Mood Test (Left-sided)
	PR_MOOD	*	p-value, Mood Test (Right-sided)
	P2_MOOD	*	p-value, Mood Test (Two-sided)
	XPL_MOOD	*	Exact p-value, Mood Test (Left-sided)
	XPR_MOOD	*	Exact p-value, Mood Test (Right-sided)
	XPT_MOOD	*	Exact Point Probability, Mood Test
	XP2_MOOD	*	Exact p-value, Mood Test (Two-sided)

Table 52.5. (continued)

Option	Output Variables		Variable Descriptions
MOOD	_CHMOOD_		Mood Chi-square
	DF_CHMOO		Degrees of Freedom, Mood Chi-square
	P_CHMOOD		p-value, Mood Chi-square Test
	XP_CHMOO	**	Exact p-value, Mood Chi-square
	XPT_CHMO	**	Exact Point Probability, Mood Chi-square
SCORES=DATA	_DATA_	*	Two-sample Data Scores Statistic
	Z_DATA	*	Data Scores Statistic, Standardized
	PL_DATA	*	p-value, Data Scores Test (Left-sided)
	PR_DATA	*	p-value, Data Scores Test (Right-sided)
	P2_DATA	*	p-value, Data Scores Test (Two-sided)
	XPL_DATA	*	Exact p-value, Data Scores Test (Left-sided)
	XPR_DATA	*	Exact p-value, Data Scores Test (Right-sided)
	XPT_DATA	*	Exact Point Probability, Data Scores Test
	XP2_DATA	*	Exact p-value, Data Scores Test (Two-sided)
	CHDATA		Data Scores Chi-square
	DF_CHDAT		Degrees of Freedom, Data Scores Chi-square
	P_CHDATA		p-value, Data Scores Chi-square Test
	XP_CHDAT	**	Exact p-value, Data Scores Chi-square
	XPT_CHDA	**	Exact Point Probability, Data Scores Chi-square
EDF	_KS_		Kolmogorov-Smirnov Statistic
	KSA		Kolmogorov-Smirnov Statistic (Asymptotic)
	_Dp_	*	Two-sample Kolmogorov-Smirnov D+
	P_Dp	*	p-value, Kolmogorov-Smirnov D+
	_Dm_	*	Two-sample Kolmogorov-Smirnov D-
	P_Dm	*	p-value, Kolmogorov-Smirnov D-
	_D_	*	Two-sample Kolmogorov-Smirnov Statistic
	P_KSA	*	p-value, Two-sample Kolmogorov-Smirnov
	XP_Dp	*	Exact p-value, Kolmogorov-Smirnov D+
	XPT_Dp	*	Exact Point Probability, Kolmogorov-Smirnov D+
	XP_Dm	*	Exact p-value, Kolmogorov-Smirnov D-
	XPT_Dm	*	Exact Point Probability, Kolmogorov-Smirnov D-
	XP_D	*	Exact p-value, Kolmogorov-Smirnov D
	XPT_D	*	Exact Point Probability, Kolmogorov-Smirnov D
	_CM_		Cramer-von Mises Statistic
	CMA		Cramer-von Mises Statistic (Asymptotic)
	_K_	*	Kuiper Two-sample Statistic
	_KA_	*	Kuiper Two-sample Statistic (Asymptotic)
	P_KA	*	p-value, Two-sample Kuiper (Asymptotic)

* Statistic included only for two-sample cases

** Statistic included only for multisample cases

Displayed Output

If you specify the ANOVA option, PROC NPAR1WAY displays a Class Means table and an Analysis of Variance table for each response variable. The Class Means table includes the following information for each CLASS variable value, or level:

- N, the number of observations
- the Mean of the response variable

The Analysis of Variance table includes the following information for each Source of variation (Among classes, and Within classes):

- DF, the degrees of freedom associated with the source
- the Sum of Squares
- the Mean Square, the sum of squares divided by the degrees of freedom

The Analysis of Variance table also includes the following:

- the F Value for testing the hypothesis that the group means are equal. This is computed by dividing the Mean Square (Among) by the Mean Square (Within).
- Pr > F, the significance probability corresponding to the F Value

For each score type that you specify, PROC NPAR1WAY displays a Class Scores table. The available score types include Wilcoxon, median, Van der Waerden, Savage, Siegel-Tukey, Ansari-Bradley, Klotz, Mood, and raw data scores. PROC NPAR1WAY assigns the specified scores to the response variable values, and classifies then according to the CLASS variable values. The Class Scores table includes the following information for each class:

- N, the number of observations
- Sum of Scores
- Expected Under H0, the expected sum of scores under the null hypothesis of no difference among classes
- Std Dev Under H0, the standard deviation under the null hypothesis
- Mean Score

When there are only two levels of the CLASS variable, PROC NPAR1WAY displays the following Two-Sample Test results for each analysis of scores:

- Statistic, which is the sum of scores for the class with the smaller sample size
- Z, the standardized test statistic, which has an asymptotic standard normal distribution under the null hypothesis
- One-Sided Pr < Z, or One-Sided Pr > Z, the asymptotic one-sided p-value, displayed as Pr < Z or Pr > Z, depending on whether Z is <= 0 or > 0
- Two-Sided Pr > |Z|, the asymptotic two-sided p-value

For Wilcoxon scores, PROC NPAR1WAY also displays a *t*-approximation for the two-sample test.

If you request an exact test by specifying the score type in the EXACT statement, PROC NPAR1WAY displays the following exact p-values for two-sample data:

- One-Sided Pr <= S, or One-Sided Pr >= S, the one-sided exact p-value, displayed as Pr <= S or Pr >= S, depending on whether S <= Mean or S > Mean, where S is the test statistic and Mean is its expected value under the null hypothesis
- Point Pr = S, the point probability, if you specify the POINT option in the EXACT statement
- Two-Sided Pr >= |S - Mean|, the two-sided exact p-value

If you request Monte Carlo estimates for the exact test by specifying the MC option in the EXACT statement, PROC NPAR1WAY displays the following information for two-sample data:

- Estimate of One-Sided Pr <= S or One-Sided Pr >= S, the one-sided exact p-value, together with its Lower and Upper Confidence Limits
- Estimate of Two-Sided Pr >= |S - Mean|, the two-sided exact p-value, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimates
- Initial Seed used to compute the Monte Carlo estimates

For both two-sample and multisample data, PROC NPAR1WAY displays the following One-Way Analysis for each score type:

- Chi-Square, the one-way ANOVA statistic for testing the null hypothesis of no difference among classes
- DF, the degrees of freedom
- Pr > Chi-Square, the asymptotic p-value

For multisample data, if you request an exact test by specifying the score type in the EXACT statement, PROC NPAR1WAY also displays the exact p-value as follows:

- Exact Pr >= Chi-Square
- Exact Pr = Chi-Square, the point probability, if you specify the POINT option in the EXACT statement

For multisample data, if you request a Monte Carlo estimate for the exact test by specifying the MC option in the EXACT statement, PROC NPAR1WAY displays the following information:

- Estimate of Pr >= Chi-Square, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimate
- Initial Seed used to compute the Monte Carlo estimate

If you specify the EDF option, PROC NPAR1WAY produces tables for the Kolmogorov-Smirnov Test, the Cramer-von Mises Test, and for two-sample data only, the Kuiper Test. The Kolmogorov-Smirnov Test table includes the following information for each CLASS variable value, or level:

- N, the number of observations
- EDF at Maximum, the value of the class EDF (empirical distribution function) at its maximum deviation from the pooled EDF
- Deviation from Mean at Maximum, the value of $\sqrt{n_i}\sqrt{F_i - F}$ at its maximum, where n_i it the class sample size, F_i is the class EDF, and F is the pooled EDF

PROC NPAR1WAY displays the following Kolmogorov-Smirnov statistics:

- KS, the Kolmogorov-Smirnov statistic
- KSa, the asymptotic Kolmogorov-Smirnov statistic, where KSa = $\sqrt{n}$ KS

For two-sample data, PROC NPAR1WAY displays the following Kolmogorov-Smirnov statistics:

- Pr > KSa, the asymptotic p-value for KSa, which equals Pr > D
- D = max | F1 - F2 |, the two-sample Kolmogorov-Smirnov statistic

For two-sample data, if you specify the D option, PROC NPAR1WAY also displays the following one-sided Kolmogorov-Smirnov statistics and their asymptotic *p*-values:

- D+ = max (F1 - F2)
- Pr > D+
- D- = max(F2 - F1)
- Pr > D-

For two-sample data, if you request an exact Kolmogorov-Smirnov test by specifying the KS option in the EXACT statement, PROC NPAR1WAY displays the following exact *p*-values:

- Exact Pr >= D
- Exact Pr >= D+
- Exact Pr >= D-
- Exact Point Pr = D, Exact Point Pr = D+, and Exact Point Pr = D-, if you specify the POINT option in the EXACT statement

If you request Monte Carlo estimates for the two-sample exact Kolmogorov-Smirnov test, PROC NPAR1WAY displays the following information for two-sample data:

- Estimate of Pr >= D, together with its Lower and Upper Confidence Limits
- Estimate of Pr >= D+, together with its Lower and Upper Confidence Limits
- Estimate of Pr >= D-, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimates
- Initial Seed used to compute the Monte Carlo estimates

The Cramer-von Mises Test table includes the following information for each CLASS variable value, or level:

- N, the number of observations
- Summed Deviation from Mean, which is $(n_i/n) \sum_{j=1}^{p} t_j \left(F_i(x_j) - F(x_j)\right)^2$

PROC NPAR1WAY also displays the following Cramer-von Mises statistics:

- CM, the Cramer-von Mises statistic
- CMa, the asymptotic Cramer-von Mises statistic, where CMa = n CM

For two-sample data, PROC NPAR1WAY displays the Kuiper Test table, which includes the following information for each class:

- N, the number of observations
- Deviation from Mean, which is $\max_j |F_1(x_j) - F_2(x_j)|$

PROC NPAR1WAY also displays the following Kuiper two-sample test statistics:

- K, the Kuiper two-sample test statistic
- Ka, the asymptotic Kuiper two-sample test statistic, where Ka = K $\sqrt{n_1 n_2 / n}$
- Pr > Ka

ODS Table Names

PROC NPAR1WAY assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

The WILCOXON, MEDIAN, VW, SAVAGE, and EDF options are the default if you do not specify any analysis options in the PROC NPAR1WAY statement.

Table 52.6. ODS Tables Produced in PROC NPAR1WAY

ODS Table Name	Description	Statement	Option
ANOVA	Analysis of variance	PROC	ANOVA
ABAnalysis	Ansari-Bradley one-way analysis	PROC	AB
ABMC	Monte Carlo estimates for the Ansari-Bradley exact test	EXACT	AB / MC
ABScores	Ansari-Bradley scores	PROC	AB
ABTest	Ansari-Bradley two-sample test	PROC	AB*
ClassMeans	Class Means	PROC	ANOVA
CVMStats	Cramer-von Mises statistics	PROC	EDF
CVMTest	Cramer-von Mises test	PROC	EDF
DataScores	Data scores	PROC	SCORES=DATA
DataScoresAnalysis	Data scores one-way analysis	PROC	SCORES=DATA
DataScoresMC	Monte Carlo estimates for the exact test based on data scores	EXACT	SCORES=DATA / MC
DataScoresTest	Data scores two-sample test	PROC	SCORES=DATA*
KlotzAnalysis	Klotz one-way analysis	PROC	KLOTZ
KlotzMC	Monte Carlo estimates for the Klotz exact test	EXACT	KLOTZ / MC
KlotzScores	Klotz scores	PROC	KLOTZ
KlotzTest	Klotz two-sample test	PROC	KLOTZ
KolSmirExactTest	Kolmogorov-Smirnov exact test	EXACT	KS*
KolSmir2Stats	Kolmogorov-Smirnov two-sample statistics	PROC	EDF*
KolSmirStats	Kolmogorov-Smirnov statistics	PROC	EDF**
KolSmirTest	Kolmogorov-Smirnov test	PROC	EDF
KruskalWallisMC	Monte Carlo estimates for the Kruskal-Wallis exact test	EXACT	WILCOXON / MC**
KruskalWallisTest	Kruskal-Wallis test	PROC	WILCOXON
KSMC	Monte Carlo estimates for the Kolmogorov-Smirnov exact test	EXACT	KS / MC*
KuiperStats	Kuiper two-sample statistics	PROC	EDF*
KuiperTest	Kuiper test	PROC	EDF*
MedianAnalysis	Median one-way analysis	PROC	MEDIAN
MedianMC	Monte Carlo estimates for the median exact test	EXACT	MEDIAN / MC
MedianScores	Median scores	PROC	MEDIAN
MedianTest	Median two-sample test	PROC	MEDIAN*

Table 52.6. (continued)

ODS Table Name	Description	Statement	Option
MoodAnalysis	Mood one-way analysis	PROC	MOOD
MoodMC	Monte Carlo estimates for the Mood exact test	EXACT	MOOD / MC
MoodScores	Mood scores	PROC	MOOD
MoodTest	Mood two-sample test	PROC	MOOD
SavageAnalysis	Savage one-way analysis	PROC	SAVAGE
SavageMC	Monte Carlo estimates for the Savage exact test	EXACT	SAVAGE / MC
SavageScores	Savage scores	PROC	SAVAGE
SavageTest	Savage two-sample test	PROC	SAVAGE*
STAnalysis	Siegel-Tukey one-way analysis	PROC	ST
STMC	Monte Carlo estimates for the Siegel-Tukey exact test	EXACT	ST / MC
STScores	Siegel-Tukey scores	PROC	ST
STTest	Siegel-Tukey two-sample test	PROC	ST*
VWAnalysis	Van der Waerden one-way analysis	PROC	VW
VWMC	Monte Carlo estimates for the Van der Waerden exact test	EXACT	VW / MC
VWScores	Van der Waerden scores	PROC	VW
VWTest	Van der Waerden two-sample test	PROC	VW*
WilcoxonMC	Monte Carlo estimates for the Wilcoxon two-sample exact test	EXACT	WILCOXON / MC*
WilcoxonScores	Wilcoxon scores	PROC	WILCOXON
WilcoxonTest	Wilcoxon two-sample test	PROC	WILCOXON*

* PROC NPAR1WAY produces this table only for two-sample data.

** PROC NPAR1WAY produces this table only for multisample data.

Examples

Example 52.1. Two-Sample Location Tests and EDF Statistics

Fifty-nine female patients with rheumatoid arthritis who participated in a clinical trial were assigned to two groups, active and placebo. The response status (excellent=5, good=4, moderate=3, fair=2, poor=1) of each patient was recorded.

The following SAS statements create the data set Arthritis, which contains the observed status values for all the patients. The variable Treatment denotes the treatment received by a patient, and the variable Response contains the response status of the patient. The variable Freq contains the frequency of the observation, which is the number of patients with the Treatment and Response combination.

```
data Arthritis;
   input Treatment $ Response Freq @@;
   datalines;
Active 5 5 Active 4 11 Active 3 5 Active 2 1 Active 1 5
Placebo 5 2 Placebo 4 4 Placebo 3 7 Placebo 2 7 Placebo 1 12
;
```

PROC NPAR1WAY tests the null hypothesis that there is no difference in the patient response status against an alternative hypothesis that the patient response status differs in the two treatment groups. The WILCOXON option requests the Wilcoxon test for difference in location, and the MEDIAN option requests the median test for difference in location. The EDF option requests empirical distribution function statistics. The variable Treatment is the CLASS variable, and the VAR statement specifies that the variable Response is the response variable.

```
proc npar1way wilcoxon median edf data=Arthritis;
   class Treatment;
   var Response;
   freq Freq;
run;
```

Output 52.1.1 shows the results of the Wilcoxon analysis. The Wilcoxon two-sample test statistic equals 999.0, which is the sum of the Wilcoxon scores for the smaller sample (Active). This sum is greater than 810.0, its expected value under the null hypothesis of no difference between the two samples Active and Placebo. The one-sided p-value is 0.0016, which shows that the patient response for the Active treatment is significantly more than for the Placebo group.

Output 52.1.1. Wilcoxon Two-Sample Test

```
                              The NPAR1WAY Procedure

                Wilcoxon Scores (Rank Sums) for Variable Response
                        Classified by Variable Treatment

                             Sum of      Expected       Std Dev          Mean
 Treatment         N         Scores      Under H0      Under H0         Score
 -------------------------------------------------------------------------
 Active           27          999.0         810.0     63.972744     37.000000
 Placebo          32          771.0         960.0     63.972744     24.093750

                       Average scores were used for ties.


                             Wilcoxon Two-Sample Test

                          Statistic             999.0000

                          Normal Approximation
                          Z                       2.9466
                          One-Sided Pr >  Z       0.0016
                          Two-Sided Pr > |Z|      0.0032

                          t Approximation
                          One-Sided Pr >  Z       0.0023
                          Two-Sided Pr > |Z|      0.0046

                    Z includes a continuity correction of 0.5.


                               Kruskal-Wallis Test

                          Chi-Square              8.7284
                          DF                           1
                          Pr > Chi-Square         0.0031
```

Output 52.1.2. Median Two-Sample Test

```
      Median Scores (Number of Points Above Median) for Variable Response
                       Classified by Variable Treatment

                            Sum of      Expected       Std Dev          Mean
Treatment        N          Scores      Under H0      Under H0         Score
------------------------------------------------------------------------
Active          27       18.916667     13.271186      1.728195      0.700617
Placebo         32       10.083333     15.728814      1.728195      0.315104

                      Average scores were used for ties.


                            Median Two-Sample Test

                         Statistic             18.9167
                         Z                      3.2667
                         One-Sided Pr >  Z      0.0005
                         Two-Sided Pr > |Z|     0.0011


                            Median One-Way Analysis

                         Chi-Square            10.6713
                         DF                          1
                         Pr > Chi-Square        0.0011
```

Output 52.1.2 shows the results of the median two-sample test. The statistic equals 18.9167, with a one-sided p-value of 0.0005. This shows that the response for the Active treatment is significantly more than for the Placebo group.

Output 52.1.3. Empirical Distribution Function Statistics

```
    Kolmogorov-Smirnov Test for Variable Response
           Classified by Variable Treatment

                         EDF at    Deviation from Mean
Treatment       N       Maximum        at Maximum
------------------------------------------------------
Active         27      0.407407         -1.141653
Placebo        32      0.812500          1.048675
Total          59      0.627119

     Maximum Deviation Occurred at Observation 3
          Value of Response at Maximum = 3.0

   Kolmogorov-Smirnov Two-Sample Test (Asymptotic)
         KS   0.201818    D         0.405093
         KSa  1.550191    Pr > KSa  0.0164


     Cramer-von Mises Test for Variable Response
          Classified by Variable Treatment

                                Summed Deviation
     Treatment          N           from Mean
     -------------------------------------------
     Active            27            0.526596
     Placebo           32            0.444316

       Cramer-von Mises Statistics (Asymptotic)
            CM  0.016456    CMa  0.970912


          Kuiper Test for Variable Response
          Classified by Variable Treatment

                                  Deviation
          Treatment        N      from Mean
          ---------------------------------
          Active          27       0.000000
          Placebo         32       0.405093

         Kuiper Two-Sample Test (Asymptotic)
    K  0.405093    Ka  1.550191    Pr > Ka  0.1409
```

Output 52.1.3 shows empirical distribution function statistics comparing these two samples. The asymptotic p-value for the Kolmogorov-Smirnov test is 0.0164. This indicates rejection of the null hypothesis that the distributions are identical for the two groups.

Example 52.2. The Exact Wilcoxon Two-Sample Test

Researchers conducted an experiment to compare the effects of two stimulants. Thirteen randomly selected subjects received the first stimulant and six randomly selected subjects received the second stimulant. The reaction times (in minutes) were measured while the subjects were under the influence of the stimulants.

The following SAS statements create the data set React, which contains the observed reaction times for each stimulant. The variable Stim represents Stimulant 1 or 2. The variable Time contains the reaction times observed for subjects under the stimulant.

```
data React;
   input Stim Time @@;
   datalines;
1 1.94   1 1.94   1 2.92   1 2.92   1 2.92   1 2.92   1 3.27
1 3.27   1 3.27   1 3.27   1 3.70   1 3.70   1 3.74
2 3.27   2 3.27   2 3.27   2 3.70   2 3.70   2 3.74
;
```

PROC NPAR1WAY tests the null hypothesis that there is no difference between the effects of the two stimulants against an alternative hypothesis that stimulant 1 has smaller reaction times than stimulant 2. The WILCOXON option specifies that Wilcoxon scores are to be used. The CLASS statement specifies that the variable Stim determines the classes. The VAR statement identifies Time as the response variable. The EXACT option requests the exact p-values. Since the sample size is small, the normal approximation may not be completely accurate, and it is appropriate to compute the exact test.

```
proc npar1way wilcoxon data=React;
   class Stim;
   var Time;
   exact;
run;
```

Output 52.2.1 displays the results of the Wilcoxon two-sample test. The Wilcoxon statistic equals 79.50. Since this value is greater than 60.0, the expected value under the null hypothesis, PROC NPAR1WAY displays the right-sided p-values. The one-sided exact p-value equals 0.0527, which is not significant at the 0.05 level. The normal approximation yields a one-sided p-value of 0.0421, which is significant at the 0.05 level. Thus, the normal approximation may result in rejection of the null hypothesis, while the exact test may result in acceptance.

Output 52.2.1. Wilcoxon Two-Sample Test

```
                       The NPAR1WAY Procedure

           Wilcoxon Scores (Rank Sums) for Variable Time
                    Classified by Variable Stim

                    Sum of      Expected       Std Dev          Mean
Stim       N        Scores      Under H0      Under H0         Score
--------------------------------------------------------------------
1         13        110.50         130.0     11.004784         8.500
2          6         79.50          60.0     11.004784        13.250

                 Average scores were used for ties.


                     Wilcoxon Two-Sample Test

                Statistic (S)               79.5000

                Normal Approximation
                Z                            1.7265
                One-Sided Pr >  Z            0.0421
                Two-Sided Pr > |Z|           0.0843

                t Approximation
                One-Sided Pr >  Z            0.0507
                Two-Sided Pr > |Z|           0.1014

                Exact Test
                One-Sided Pr >=  S           0.0527
                Two-Sided Pr >= |S - Mean|   0.1054

             Z includes a continuity correction of 0.5.


                        Kruskal-Wallis Test

                     Chi-Square         3.1398
                     DF                      1
                     Pr > Chi-Square    0.0764
```

Example 52.3. The Exact Savage Multisample Test

A researcher conducting a laboratory experiment randomly assigned 15 mice to receive one of three drugs. The survival time (in days) was then recorded.

The following SAS statements create the data set Mice, which contains the observed survival times for all the mice. The variable Trt denotes the treatment received by a mouse. The variable Days contains the number of days the mouse survived.

```
data Mice;
   input Trt $ Days @@;
   datalines;
1 1 1 1 1 3 1 3 1 4
2 3 2 4 2 4 2 4 2 15
3 4 3 4 3 10 3 10 3 26
;
```

PROC NPAR1WAY tests the null hypothesis that there is no difference in the survival times among the three drugs against an alternative hypothesis of difference among the drugs. The SAVAGE option specifies that Savage scores are to be used. The variable Trt is the CLASS variable, and the VAR statement specifies that the variable Days is the response variable. The EXACT statement requests the exact test.

```
proc npar1way savage data=Mice;
   class Trt;
   var Days;
   exact;
run;
```

Output 52.3.1. Savage Multisample Test

```
                       The NPAR1WAY Procedure

           Savage Scores (Exponential) for Variable Days
                    Classified by Variable Trt

                   Sum of      Expected       Std Dev          Mean
Trt       N        Scores      Under H0      Under H0         Score
--------------------------------------------------------------------
1         5     -3.367980           0.0      1.634555     -0.673596
2         5      0.095618           0.0      1.634555      0.019124
3         5      3.272362           0.0      1.634555      0.654472

                Average scores were used for ties.


                      Savage One-Way Analysis

               Chi-Square                     5.5047
               DF                                  2
               Asymptotic Pr >  Chi-Square    0.0638
               Exact      Pr >= Chi-Square    0.0445
```

Output 52.3.1 shows the results of the Savage test. The exact p-value is 0.0445, which is significant at the 0.05 level. However, the p-value based on the chi-square approximation is 0.0638, which results in nonrejection of the null hypothesis at the 0.05 level.

References

Agresti, A. (1992), “A Survey of Exact Inference for Contingency Tables,” *Statistical Science*, 7 (1), 131–177.

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A., Mehta, C.R., and Patel, N.R. (1990), “Exact Inference for Contingency Tables with Ordered Categories,” *Journal of American Statistical Association*, 85, 453–458.

Agresti, A., Wackerly, D., and Boyett, J.M. (1979), "Exact Conditional Tests for Cross-Classifications: Approximation of Attained Significance Levels," *Psychometrika*, 44, 75–83.

Bishop, Y., Fienberg, S.E., and Holland, P.W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Conover, W.J. (1999), *Practical Nonparametric Statistics*, Third Edition, New York: John Wiley & Sons, Inc.

Gail, M. and Mantel, N. (1977), "Counting the Number of $r \times c$ Contingency Tables with Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

Gibbons, J.D. and Chakraborti, S. (1992), *Nonparametric Statistical Inference*, Third Edition, New York: Marcel Dekker, Inc.

Hajek, J. (1969), *A Course in Nonparametric Statistics*, San Francisco: Holden-Day.

Halverson, J.O. and Sherwood, F.W. (1930), "Investigations in the Feeding of Cottonseed Meal to Cattle," *North Carolina Agr. Exp. Sta. Tech. Bulletin*, 39, 158pp.

Hodges, J.L. Jr. (1957), "The Significance Probability of the Smirnov Two-Sample Test," *Arkiv for Matematik*, 3, 469–486.

Hollander, M. and Wolfe, D.A. (1999), *Nonparametric Statistical Methods*, Second Edition, New York: John Wiley & Sons, Inc.

Kiefer, J. (1959), "K-Sample Analogues of the Kolmogorov-Smirnov and Cramer-von Mises Tests," *Annals of Mathematical Statistics*, 30, 420–447.

Mehta, C.R. and Patel, N.R. (1983), "A Network Algorithm for Performing Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of American Statistical Association*, 78, 427–434.

Mehta, C.R., Patel, N.R., and Senchaudhuri, P. (1991), "Exact Stratified Linear Rank Tests for Binary Data," *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (E.M. Keramidas, ed.), 200–207.

Mehta, C.R., Patel, N.R., and Tsiatis, A.A. (1984), "Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data," *Biometrics*, 40, 819–825.

Owen, D.B. (1962), *Handbook of Statistical Tables*, Reading, MA: Addison-Wesley Publishing Company, Inc.

Quade, D. (1966), "On Analysis of Variance for the k-Sample Problem," *Annals of Mathematical Statistics*, 37, 1747–1758.

Randles, R.H. and Wolfe, D.A. (1979), *Introduction to the Theory of Nonparametric Statistics*, New York: John Wiley & Sons, Inc.

Sheskin, D.J. (1997), *Handbook of Parametric and Nonparametric Statistical Procedures*, Boca Raton, Fla: CRC Press.

Valz, P.D. and Thompson, M.E. (1994), “Exact Inference for Kendall’s S and Spearman’s ρ with Extensions to Fisher’s Exact Test in $r \times c$ Contingency Tables,” *Journal of Computational and Graphical Statistics*, 3 (4), 459–472.

Chapter 53
The ORTHOREG Procedure

Chapter Contents

Chapter 53
The ORTHOREG Procedure

Overview

The ORTHOREG procedure fits general linear models by the method of least squares. Other SAS/STAT software procedures, such as GLM or REG, fit the same types of models, but PROC ORTHOREG can produce more accurate estimates than other regression procedures when your data are ill conditioned. Instead of collecting crossproducts, PROC ORTHOREG uses Gentleman-Givens transformations to update and compute the upper triangular matrix **R** of the QR decomposition of the data matrix, with special care for scaling (Gentleman 1972; 1973). This method has the advantage over other orthogonalization methods (for example, Householder transformations) of not requiring the data matrix to be stored in memory.

The standard SAS regression procedures (REG and GLM) are very accurate for most problems. However, if you have very ill-conditioned data, these procedures can produce estimates that yield an error sum of squares very close to the minimum but still different from the exact least-squares estimates. Normally, this coincides with estimates that have very high standard errors. In other words, the numerical error is much smaller than the statistical standard error.

Note that PROC ORTHOREG fits models by the method of linear least squares, minimizing the sum of the squared residuals for predicting the responses. It does *not* perform the modeling method known as "orthogonal regression," which minimizes a different criterion (the distance between the X/Y points taken together and the regression line.)

Getting Started

Longley Data

The labor statistics data set of Longley (1967) is noted for being ill conditioned. Both the ORTHOREG and GLM procedures are applied for comparison (only portions of the PROC GLM results are shown). **Note:** The results from this example vary from machine to machine, depending on floating-point configuration.

The following statements read the data into the SAS data set Longley.

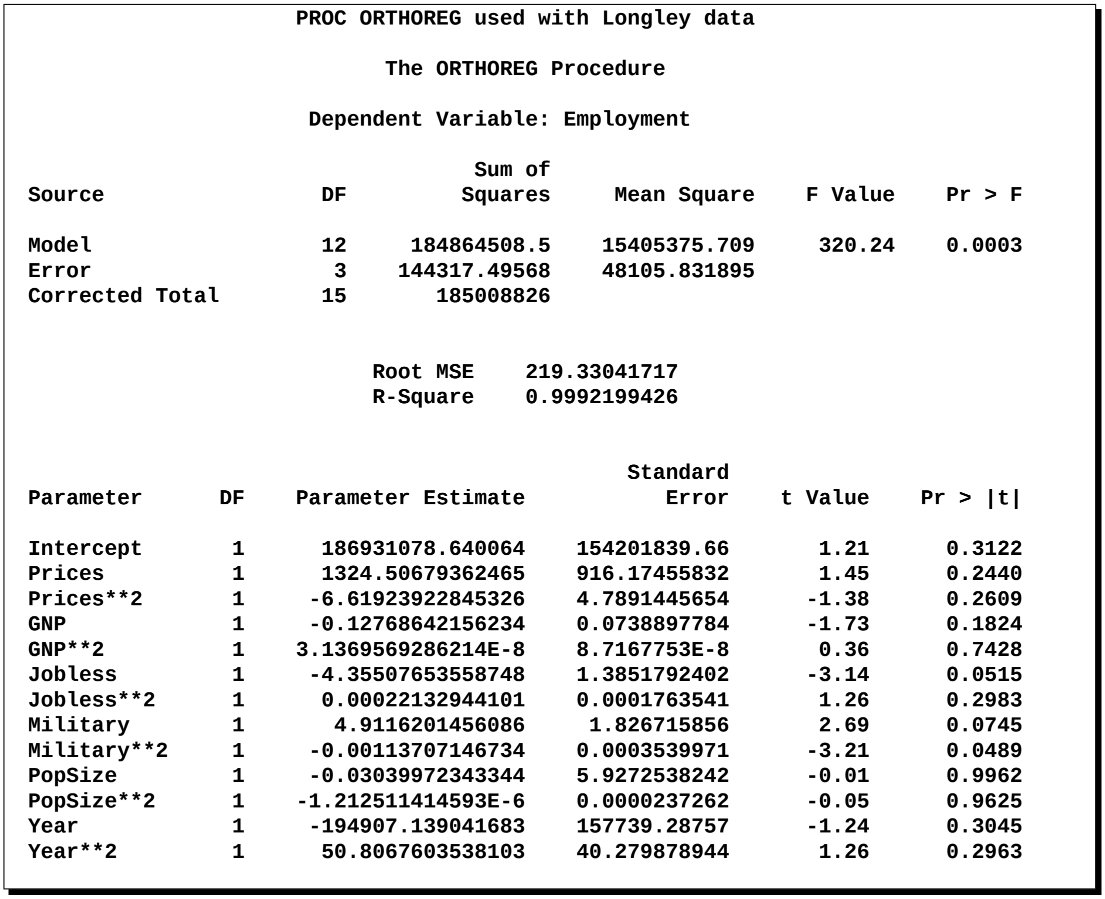
```
title 'PROC ORTHOREG used with Longley data';
data Longley;
   input Employment Prices GNP Jobless Military PopSize Year;
   datalines;
60323  83.0 234289 2356 1590 107608 1947
61122  88.5 259426 2325 1456 108632 1948
60171  88.2 258054 3682 1616 109773 1949
61187  89.5 284599 3351 1650 110929 1950
63221  96.2 328975 2099 3099 112075 1951
63639  98.1 346999 1932 3594 113270 1952
64989  99.0 365385 1870 3547 115094 1953
63761 100.0 363112 3578 3350 116219 1954
66019 101.2 397469 2904 3048 117388 1955
67857 104.6 419180 2822 2857 118734 1956
68169 108.4 442769 2936 2798 120445 1957
66513 110.8 444546 4681 2637 121950 1958
68655 112.6 482704 3813 2552 123366 1959
69564 114.2 502601 3931 2514 125368 1960
69331 115.7 518173 4806 2572 127852 1961
70551 116.9 554894 4007 2827 130081 1962
;
run;
```

The data set contains one dependent variable, Employment (total derived employment) and six independent variables: Prices (GNP implicit price deflator with year 1954 = 100), GNP (gross national product), Jobless (unemployment), Military (size of armed forces), PopSize (non-institutional population aged 14 and over), and Year (year).

The following statements use the ORTHOREG procedure to model the Longley data using a quadratic model in each independent variable, without interaction:

```
proc orthoreg data=Longley;
   model Employment = Prices   Prices*Prices
                      GNP      GNP*GNP
                      Jobless  Jobless*Jobless
                      Military Military*Military
                      PopSize  PopSize*PopSize
                      Year     Year*Year;
run;
```

Figure 53.1 shows the resulting analysis.



```
                       PROC ORTHOREG used with Longley data

                              The ORTHOREG Procedure

                        Dependent Variable: Employment

                                         Sum of
Source                      DF          Squares     Mean Square    F Value    Pr > F

Model                       12      184864508.5    15405375.709     320.24    0.0003
Error                        3     144317.49568    48105.831895
Corrected Total             15        185008826


                             Root MSE    219.33041717
                             R-Square    0.9992199426


                                                        Standard
Parameter      DF    Parameter Estimate                 Error    t Value    Pr > |t|

Intercept       1      186931078.640064         154201839.66       1.21      0.3122
Prices          1      1324.50679362465         916.17455832       1.45      0.2440
Prices**2       1     -6.61923922845326         4.7891445654      -1.38      0.2609
GNP             1     -0.12768642156234         0.0738897784      -1.73      0.1824
GNP**2          1    3.1369569286214E-8         8.7167753E-8       0.36      0.7428
Jobless         1     -4.35507653558748         1.3851792402      -3.14      0.0515
Jobless**2      1      0.00022132944101         0.0001763541       1.26      0.2983
Military        1       4.9116201456086          1.826715856       2.69      0.0745
Military**2     1     -0.00113707146734         0.0003539971      -3.21      0.0489
PopSize         1     -0.03039972343344         5.9272538242      -0.01      0.9962
PopSize**2      1    -1.212511414593E-6         0.0000237262      -0.05      0.9625
Year            1     -194907.139041683         157739.28757      -1.24      0.3045
Year**2         1      50.8067603538103         40.279878944       1.26      0.2963
```

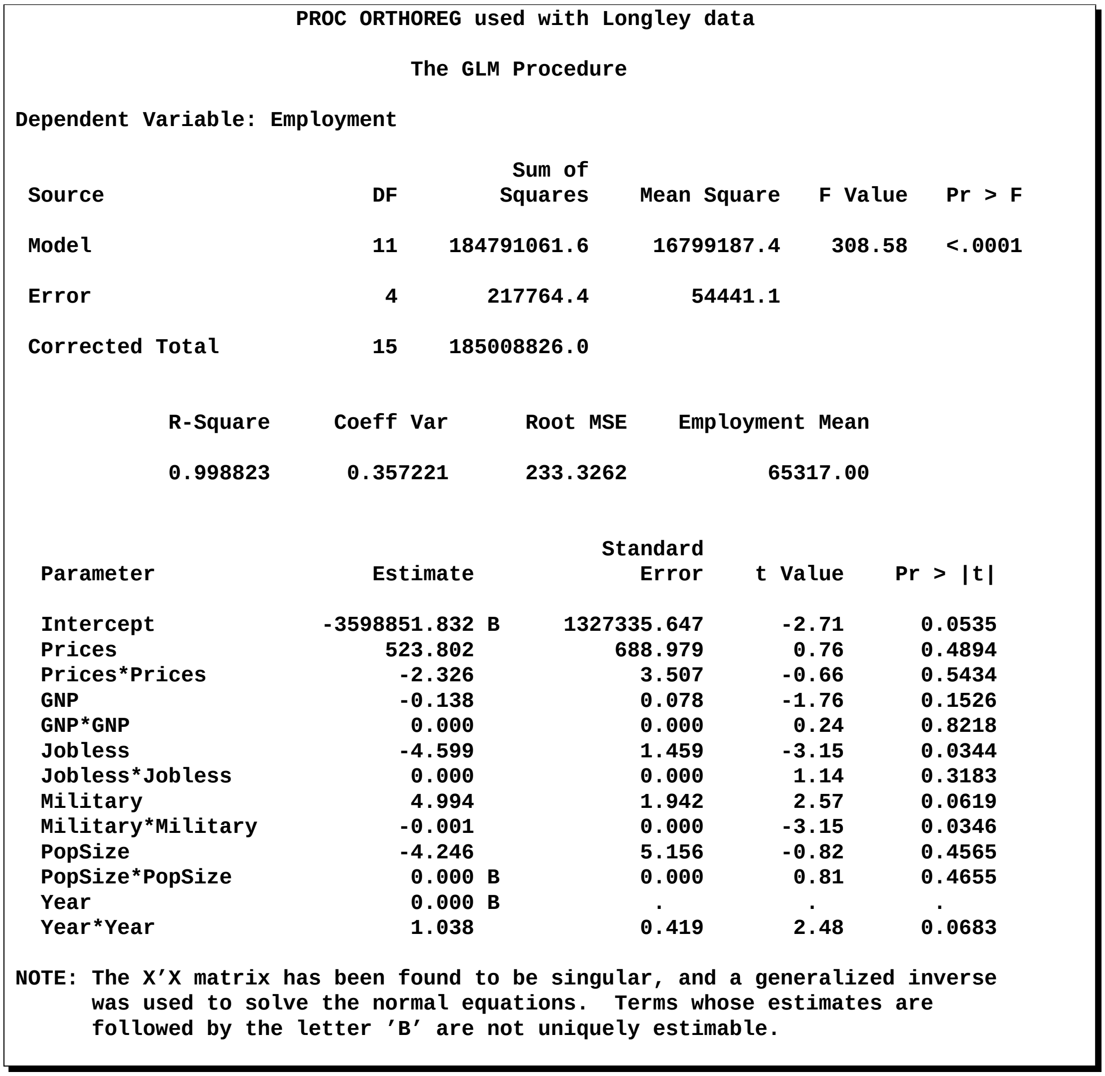
Figure 53.1. PROC ORTHOREG Results

The estimates in Figure 53.1 compare very well with the best estimates available; for additional information, refer to Longley (1967) and Beaton et al. (1976).

The following statements request the same analysis from the GLM procedure:

```
proc glm data=Longley;
   model Employment = Prices   Prices*Prices
                      GNP      GNP*GNP
                      Jobless  Jobless*Jobless
                      Military Military*Military
                      PopSize  PopSize*PopSize
                      Year     Year*Year;
   ods select OverallANOVA
              FitStatistics
              ParameterEstimates
              Notes;
run;
```

Figure 53.2 contains the over-all ANOVA table and the parameter estimates produced by PROC GLM. Notice that the ORTHOREG fit achieves a somewhat smaller root mean square error (RMSE) and also that the GLM procedure detects spurious singularities.



```
                      PROC ORTHOREG used with Longley data

                               The GLM Procedure

Dependent Variable: Employment

                                         Sum of
 Source                      DF         Squares     Mean Square    F Value    Pr > F

 Model                       11     184791061.6      16799187.4     308.58    <.0001

 Error                        4        217764.4         54441.1

 Corrected Total             15     185008826.0


            R-Square     Coeff Var      Root MSE    Employment Mean

            0.998823      0.357221      233.3262           65317.00


                                                Standard
  Parameter                 Estimate               Error    t Value    Pr > |t|

  Intercept             -3598851.832 B     1327335.647      -2.71      0.0535
  Prices                     523.802           688.979       0.76      0.4894
  Prices*Prices               -2.326             3.507      -0.66      0.5434
  GNP                         -0.138             0.078      -1.76      0.1526
  GNP*GNP                      0.000             0.000       0.24      0.8218
  Jobless                     -4.599             1.459      -3.15      0.0344
  Jobless*Jobless              0.000             0.000       1.14      0.3183
  Military                     4.994             1.942       2.57      0.0619
  Military*Military           -0.001             0.000      -3.15      0.0346
  PopSize                     -4.246             5.156      -0.82      0.4565
  PopSize*PopSize              0.000 B           0.000       0.81      0.4655
  Year                         0.000 B            .           .          .
  Year*Year                    1.038             0.419       2.48      0.0683

NOTE: The X'X matrix has been found to be singular, and a generalized inverse
      was used to solve the normal equations.  Terms whose estimates are
      followed by the letter 'B' are not uniquely estimable.
```

Figure 53.2. Partial PROC GLM Results

Syntax

The following statements are available in PROC ORTHOREG.

PROC ORTHOREG < *options* > **;**
  MODEL *dependent=independents* < / *option* > **;**
  BY *variables* **;**
  CLASS *variables* < / *option* > **;**
  WEIGHT *variable* **;**

The BY, CLASS, MODEL, and WEIGHT statements are described after the PROC ORTHOREG statement.

PROC ORTHOREG Statement

PROC ORTHOREG < *options* > **;**

The PROC ORTHOREG statement has the following options:

DATA=*SAS-data-set*

specifies the input SAS data set to use. By default, the procedure uses the most recently created SAS data set. The data set specified cannot be a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which you want the levels of the classification variables (specified in the CLASS statement) to be sorted. This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. In releases previous to Version 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The ORDER= option can take the following values.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

If you omit the ORDER= option, PROC ORTHOREG orders by the external formatted value.

OUTEST=*SAS-data-set*

produces an output data set containing the parameter estimates, the BY variables, and the special variables _TYPE_ (value PARMS), _NAME_ (blank), _RMSE_ (root mean squared error), and Intercept.

SINGULAR=*s*

specifies a singularity criterion $(s \geq 0)$ for the inversion of the triangular matrix $\mathbf{R}$. By default, SINGULAR=10E−12.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ORTHOREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ORTHOREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables < / option >* **;**

The CLASS statement names the classification variables to be used in the model. Typical class variables are Treatment, Sex, Race, Group, and Replication. If you use the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

MODEL Statement

MODEL *dependent=independents* < / *option* > **;**

The MODEL statement names the dependent variable and the independent effects. Only one MODEL statement is allowed. The specification of effects and the parameterization of the linear model is the same as in the GLM procedure; see Chapter 32, "The GLM Procedure," for further details.

The following option can be used in the MODEL statement:

NOINT

omits the intercept term from the model.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set whose values are relative weights for a weighted least-squares regression. If the weight value is proportional to the reciprocal of the variance for each observation, the weighted estimates are the best linear unbiased estimates (BLUE). For a more complete description of the WEIGHT statement, see the "WEIGHT Statement" section on page 1782 in Chapter 32, "The GLM Procedure." .

Details

Missing Values

If there is a missing value for any model variable in an observation, the entire observation is dropped from the analysis.

Output Data Set

The OUTEST= option produces a TYPE=EST output SAS data set containing the BY variables, parameter estimates, and four special variables. For each new value of the BY variables, PROC ORTHOREG outputs an observation to the OUTEST= data set. The variables in the data set are as follows:

- parameter estimates for all variables listed in the MODEL statement

- BY variables
- _TYPE_, which is a character variable with the value PARMS for every observation
- _NAME_, which is a character variable left blank for every observation
- _RMSE_, which is the root mean squared error (the estimate of the standard deviation of the true errors)
- Intercept, which is the estimated intercept. This variable does not exist in the OUTEST= data set if the NOINT option is specified.

Displayed Output

PROC ORTHOREG displays the parameter estimates and associated statistics. These include the following:

- overall model analysis of variance, including the error mean square, which is an estimate of σ^2 (the variance of the true errors), and the overall F test for a model effect
- root mean squared error, which is an estimate of the standard deviation of the true errors. It is calculated as the square root of the mean squared error.
- R-square, which is a measure between 0 and 1 that indicates the portion of the total variation that is attributed to the fit
- estimates for the parameters in the linear model

The table of parameter estimates consists of

- the terms used as regressors, including the Intercept, identifying the intercept parameter
- degrees of freedom (DF) for the variable. There is one degree of freedom for each parameter being estimated unless the model is not full rank.
- estimated linear coefficients
- estimates of the standard errors of the parameter estimates
- the critical t values for testing whether the parameters are zero. This is computed as the parameter estimate divided by its standard error.
- the two-sided p-value for the t-test, which is the probability that a t-statistic would obtain a greater absolute value than that observed given that the true parameter is zero

ODS Table Names

PROC ORTHOREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets.These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 53.1. ODS Tables Produced in PROC ORTHOREG

ODS Table Name	Description	Statement
ANOVA	Analysis of variance	default
FitStatistics	Overall statistics for fit	default
Levels	Table of class levels	CLASS statement
ParameterEstimates	Parameter estimates	default

Examples

Example 53.1. Precise Analysis of Variance

The data for the following example are from Powell et al. (1982). In order to calibrate an instrument for measuring atomic weight, 24 replicate measurements of the atomic weight of silver (chemical symbol **Ag**) are made with the new instrument and with a reference instrument.

Note: The results from this example vary from machine to machine depending on floating-point configuration.

The following statements read the measurements for the two instruments into the SAS data set AgWeight.

```
title 'Atomic Weight of Silver by Two Different Instruments';
data AgWeight;
   input Instrument AgWeight @@;
   datalines;
1 107.8681568   1 107.8681465   1 107.8681572   1 107.8681785
1 107.8681446   1 107.8681903   1 107.8681526   1 107.8681494
1 107.8681616   1 107.8681587   1 107.8681519   1 107.8681486
1 107.8681419   1 107.8681569   1 107.8681508   1 107.8681672
1 107.8681385   1 107.8681518   1 107.8681662   1 107.8681424
1 107.8681360   1 107.8681333   1 107.8681610   1 107.8681477
2 107.8681079   2 107.8681344   2 107.8681513   2 107.8681197
2 107.8681604   2 107.8681385   2 107.8681642   2 107.8681365
2 107.8681151   2 107.8681082   2 107.8681517   2 107.8681448
2 107.8681198   2 107.8681482   2 107.8681334   2 107.8681609
2 107.8681101   2 107.8681512   2 107.8681469   2 107.8681360
2 107.8681254   2 107.8681261   2 107.8681450   2 107.8681368
;
```

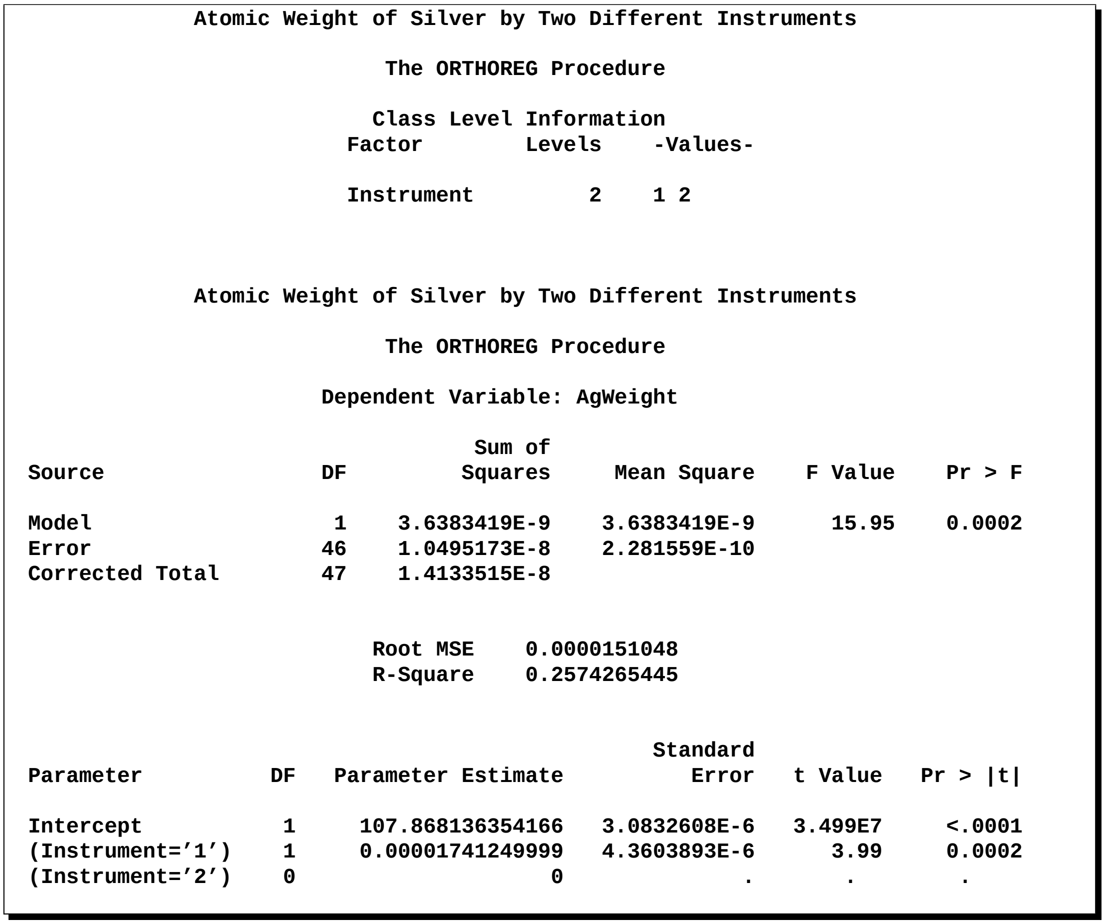
Notice that the variation in the atomic weight measurements is several orders of magnitude less than their mean. This is a situation that can be difficult for standard, regression-based analysis-of-variance procedures to handle correctly.

The following statements invoke the ORTHOREG procedure to perform a simple one-way analysis of variance, testing for differences between the two instruments.

```
proc orthoreg data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;
```

Output 53.1.1 shows the resulting analysis.

Output 53.1.1. PROC ORTHOREG Results for Atomic Weight Example



```
                Atomic Weight of Silver by Two Different Instruments

                               The ORTHOREG Procedure

                              Class Level Information
                            Factor        Levels    -Values-

                            Instrument         2    1 2


                Atomic Weight of Silver by Two Different Instruments

                               The ORTHOREG Procedure

                            Dependent Variable: AgWeight

                                           Sum of
Source                      DF           Squares     Mean Square    F Value    Pr > F

Model                        1    3.6383419E-9    3.6383419E-9      15.95    0.0002
Error                       46    1.0495173E-8    2.281559E-10
Corrected Total             47    1.4133515E-8


                              Root MSE    0.0000151048
                              R-Square    0.2574265445


                                                             Standard
Parameter           DF   Parameter Estimate         Error    t Value   Pr > |t|

Intercept            1     107.868136354166   3.0832608E-6   3.499E7     <.0001
(Instrument='1')     1     0.00001741249999   4.3603893E-6      3.99     0.0002
(Instrument='2')     0                    0              .         .          .
```

The mean difference between instruments is about 1.74×10^{-5} (the value of the `(Instrument='1')` parameter in the parameter estimates table), whereas the level of background variation in the measurements is about 1.51×10^{-5} (the value of the root mean squared error). The difference is significant, with a p-value of 0.0002.

The National Institute of Standards and Technology (1998) has provided certified ANOVA values for this data set. The following statements use ODS to examine the ANOVA values produced by both the ORTHOREG and GLM procedures more precisely for comparison with the NIST-certified values:

```
ods listing close;
ods output ANOVA         = OrthoregANOVA
           FitStatistics = OrthoregFitStat;


proc orthoreg data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;

ods output OverallANOVA  = GLMANOVA
           FitStatistics = GLMFitStat;
proc glm data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;
ods listing;

data _null_; set OrthoregANOVA  (in=inANOVA)
                 OrthoregFitStat(in=inFitStat);
   if (inANOVA) then do;
      if (Source = ’Model’) then put "Model SS: " ss e20.;
      if (Source = ’Error’) then put "Error SS: " ss e20.;
   end;
   if (inFitStat) then do;
      if (Statistic = ’Root MSE’) then
                            put "Root MSE: " nValue1 e20.;
      if (Statistic = ’R-Square’) then
                         put "R-Square: " nValue1 best20.;
   end;
data _null_; set GLMANOVA  (in=inANOVA)
                 GLMFitStat(in=inFitStat);
   if (inANOVA) then do;
      if (Source = ’Model’) then put "Model SS: " ss e20.;
      if (Source = ’Error’) then put "Error SS: " ss e20.;
   end;
   if (inFitStat) then      put "Root MSE: " RootMSE e20.;
   if (inFitStat) then   put "R-Square: " RSquare best20.;
run;
```

In releases of SAS/STAT software prior to Version 8, PROC GLM gave much less accurate results than PROC ORTHOREG, as shown in the following tables, which compare the ANOVA values certified by NIST with those produced by the two procedures.

	Model SS	**Error SS**
NIST-certified	3.6383418750000E-09	1.0495172916667E-08
ORTHOREG	3.6383418747907E-09	1.0495172916797E-08
GLM, Version 8	3.6383418747907E-09	1.0495172916797E-08
GLM, Previous releases	0	1.0331496763990E-08

	Root MSE	**R-Square**
NIST-certified	1.5104831444641E-05	0.25742654453832
ORTHOREG	1.5104831444735E-05	0.25742654452494
GLM, Version 8	1.5104831444735E-05	0.25742654452494
GLM, Previous releases	1.4986585859992E-05	0

While the ORTHOREG values and the GLM values for Version 8 are quite close to the certified ones, the GLM values for prior releases are not. In fact, since the model sum of squares is so small, in prior releases the GLM procedure set it (and consequently R^2) to zero.

Example 53.2. Wampler Data

This example applies the ORTHOREG procedure to a collection of data sets noted for being ill conditioned. The OUTEST= data set is used to collect the results for comparison with values certified to be correct by the National Institute of Standards and Technology (1998).

Note: The results from this example vary from machine to machine depending on floating-point configuration.

The data are from Wampler (1970). The independent variates for all five data sets are x^i, $i = 1, \ldots 5$, for $x = 0, 1, \ldots, 20$. Two of the five dependent variables are exact linear functions of the independent terms:

$$\begin{aligned} y_1 &= 1 + x + x^2 + x^3 + x^4 + x^5 \\ y_2 &= 1 + 0.1x + 0.01x^2 + 0.001x^3 + 0.0001x^4 + 0.00001x^5 \end{aligned}$$

The other three dependent variables have the same mean value as y_1, but with nonzero errors.

$$\begin{aligned} y_3 &= y_1 + \mathbf{e} \\ y_4 &= y_1 + 100\mathbf{e} \\ y_5 &= y_1 + 10000\mathbf{e} \end{aligned}$$

where $\mathbf{e}$ is a vector of values with standard deviation 2044, chosen to be orthogonal to the mean model for y_1.

The following statements create a SAS data set Wampler containing the Wampler data, run a SAS macro program using PROC ORTHOREG to fit a fifth-order polynomial in x to each of the Wampler dependent variables, and collect the results in a data set named ParmEst.

```
data Wampler;
   do x=0 to 20;
      input e @@;
      y1 = 1 +       x    +        x**2 +      x**3
             +       x**4 +        x**5;
      y2 = 1 + .1   *x    + .01   *x**2 + .001*x**3
             + .0001*x**4 + .00001*x**5;
      y3 = y1 +       e;
      y4 = y1 +   100*e;
      y5 = y1 + 10000*e;
      output;
   end;
   datalines;
759 -2048 2048 -2048 2523 -2048 2048 -2048 1838 -2048 2048
-2048 1838 -2048 2048 -2048 2523 -2048 2048 -2048 759
;

%macro WTest;
   data ParmEst; if (0); run;
   %do i = 1 %to 5;
      proc orthoreg data=Wampler outest=ParmEst&i noprint;
         model y&i = x x*x x*x*x x*x*x*x x*x*x*x*x;
      data ParmEst&i; set ParmEst&i; Dep = "y&i";
      data ParmEst; set ParmEst ParmEst&i;
         label Col1=’x’     Col2=’x**2’  Col3=’x**3’
               Col4=’x**4’  Col5=’x**5’;
      run;
   %end;
%mend;
%WTest;
```

Instead of displaying the raw values of the RMSE and parameter estimates, use a further DATA step to compute the deviations from the values certified to be correct by the National Institute of Standards and Technology (1998).

```
data ParmEst; set ParmEst;
   if      (Dep = ’y1’) then
      _RMSE_ = _RMSE_ - 0.00000000000000;
   else if (Dep = ’y2’) then
      _RMSE_ = _RMSE_ - 0.00000000000000;
   else if (Dep = ’y3’) then
      _RMSE_ = _RMSE_ - 2360.14502379268;
   else if (Dep = ’y4’) then
      _RMSE_ = _RMSE_ - 236014.502379268;
   else if (Dep = ’y5’) then
      _RMSE_ = _RMSE_ - 23601450.2379268;
   if (Dep ^= 'y2') then do;
      Intercept = Intercept - 1.00000000000000;
      Col1      = Col1      - 1.00000000000000;
      Col2      = Col2      - 1.00000000000000;
      Col3      = Col3      - 1.00000000000000;
      Col4      = Col4      - 1.00000000000000;
      Col5      = Col5      - 1.00000000000000;
   end;
   else do;
      Intercept = Intercept - 1.00000000000000;
      Col1      = Col1      - 0.100000000000000;
      Col2      = Col2      - 0.100000000000000e-1;
      Col3      = Col3      - 0.100000000000000e-2;
      Col4      = Col4      - 0.100000000000000e-3;
      Col5      = Col5      - 0.100000000000000e-4;
   end;
proc print data=ParmEst label noobs;
   title 'Wampler data: Deviations from Certified Values';
   format _RMSE_ Intercept Col1-Col5 e9.;
   var Dep _RMSE_ Intercept Col1-Col5;
run;
```

The results, shown in Output 53.2.1, indicate that the values computed by PROC ORTHOREG are quite close to the NIST-certified values.

Output 53.2.1. Wampler data: Deviations from Certified Values

```
                 Wampler data: Deviations from Certified Values

Dep     _RMSE_  Intercept          x       x**2       x**3       x**4       x**5

y1    0.00E+00   1.49E-10   9.08E-12  -5.99E-12   1.26E-12  -9.68E-14   2.00E-15
y2    0.00E+00  -6.33E-15   5.55E-16   1.37E-16  -1.13E-17   5.56E-19  -1.52E-20
y3    1.09E-11   3.02E-10  -1.70E-10   4.88E-11  -5.75E-12   3.18E-13  -6.88E-15
y4   -3.20E-10   2.74E-09  -5.60E-09   2.12E-09  -2.89E-10   1.63E-11  -3.24E-13
y5   -2.98E-08   2.46E-07  -5.54E-07   2.12E-07  -2.90E-08   1.64E-09  -3.27E-11
```

References

Beaton, A. E., Rubin, D. B., and Barone, J. L. (1976), "The Acceptability of Regression Solutions: Another Look at Computational Accuracy," *Journal of the American Statistical Association*, 71, 158–168.

Gentleman, W. M. (1972), "Basic Procedures for Large, Sparse orWeighted Least Squares Problems," Technical Report CSRR-2068, Univ. of Waterloo, Waterloo, Ontario, Canada.

Gentleman, W. M. (1973), "Least Squares Computations by Givens Transformations without Square Roots," *J. Inst. Math. Appl.*, 12, 329–336.

Lawson, C. L. and Hanson, R. J. (1974), *Solving Least Squares Problems*, Englewood Cliffs, NJ: Prentice-Hall, Inc.

Longley, J. W. (1967), “An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User,” *Journal of the American Statistical Association*, 62, 819–41.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, http://www.nist.gov/itl/div898/strd: last accessed Jan 20, 1998.

Powell, L. J., Murphy, T. J., and Gramlich, J. W. (1982), “The Absolute Isotopic Abundance and Atomic Weight of a Reference Sample of Silver,” *NBS Journal of Research*, 87, 9–19.

Wampler, R. H. (1970), “A Report of the Accuracy of Some Widely Used Least Squares Computer Programs,” *Journal of the American Statistical Association*, 65, 549–563.

Chapter 54
The PHREG Procedure

Chapter Contents

Chapter 54
The PHREG Procedure

Overview

The analysis of survival data requires special techniques because the data are almost always incomplete, and familiar parametric assumptions may be unjustifiable. Investigators follow subjects until they reach a prespecified endpoint (for example, death). However, subjects sometimes withdraw from a study, or the study is completed before the endpoint is reached. In these cases, the survival times (also known as failure times) are *censored*; subjects survived to a certain time beyond which their status is unknown. The uncensored survival times are sometimes referred to as *event* times. Methods for survival analysis must account for both censored and uncensored data.

There are many types of models that have been used for survival data. Two of the more popular types of models are the accelerated failure time model (Kalbfleisch and Prentice 1980) and the Cox proportional hazards model (Cox 1972). Each has its own assumptions on the underlying distribution of the survival times. Two closely related functions often used to describe the distribution of survival times are the survivor function and the hazard function (see the section "Failure Time Distribution" on page 3239 for definitions). The accelerated failure time model assumes a parametric form for the effects of the explanatory variables and usually assumes a parametric form for the underlying survivor function. Cox's proportional hazards model also assumes a parametric form for the effects of the explanatory variables, but it allows an unspecified form for the underlying survivor function.

The PHREG procedure performs regression analysis of survival data based on the Cox proportional hazards model. Cox's semiparametric model is widely used in the analysis of survival data to explain the effect of explanatory variables on hazard rates. The survival time of each member of a population is assumed to follow its own hazard function, $\lambda_i(t)$, expressed as

$$\lambda_i(t) = \lambda(t; \mathbf{Z}_i) = \lambda_0(t) \exp(\mathbf{Z}_i' \boldsymbol{\beta})$$

where $\lambda_0(t)$ is an arbitrary and unspecified baseline hazard function, $\mathbf{Z}_i$ is the vector of explanatory variables for the ith individual, and $\boldsymbol{\beta}$ is the vector of unknown regression parameters associated with the explanatory variables. The vector $\boldsymbol{\beta}$ is assumed to be the same for all individuals. The survivor function can be expressed as

$$S(t; \mathbf{Z}_i) = [S_0(t)]^{\exp(\mathbf{Z}_i' \boldsymbol{\beta})}$$

where $S_0(t) = \exp(-\int_0^t \lambda_0(u)du)$ is the baseline survivor function. To estimate $\boldsymbol{\beta}$, Cox (1972; 1975) introduced the partial likelihood function, which eliminates the unknown baseline hazard $\lambda_0(t)$ and accounts for censored survival times.

The partial likelihood of Cox also allows time-dependent explanatory variables. An explanatory variable is time-dependent if its value for any given individual can change over time. Time-dependent variables have many useful applications in survival analysis. You can use a time-dependent variable to model the effect of subjects changing treatment groups. Or you can include time-dependent variables such as blood pressure or blood chemistry measures that vary with time during the course of a study. You can also use time-dependent variables to test the validity of the proportional hazards model.

An alternative way to fit models with time-dependent explanatory variables is to use the counting process style of input. The counting process formulation enables PROC PHREG to fit a superset of the Cox model, known as the multiplicative hazards model. This extension also includes recurrent events data and left truncation of failure times. The theory of these models is based on the counting process pioneered by Andersen and Gill (1982), and the model is often referred to as the Andersen-Gill Model.

Multivariate failure time data arise when each study subject can potentially experience several events (for instance, multiple infections after surgery) or when there exists some natural or artificial clustering of subjects (for instance, a litter of mice) that induces dependence among the failure times of the same cluster. Data in the former situation are referred to as multiple events data, which include recurrent events data as a special case; data in the latter situation are referred to as clustered data. You can use PROC PHREG to carry out various methods for analyzing these data.

The population under study may consist of a number of subpopulations, each of which has its own baseline hazard function. PROC PHREG performs a stratified analysis to adjust for such subpopulation differences. Under the stratified model, the hazard function for the *j*th individual in the *i*th stratum is expressed as

$$\lambda_{ij}(t) = \lambda_{i0}(t) \exp(\mathbf{Z}'_{ij}\boldsymbol{\beta})$$

where $\lambda_{i0}(t)$ is the baseline hazard function for the *i*th stratum, and $\mathbf{Z}_{ij}$ is the vector of explanatory variables for the individual. The regression coefficients are assumed to be the same for all individuals across all strata.

Ties in the failure times may arise when the time scale is genuinely discrete or when survival times generated from the continuous-time model are grouped into coarser units. The PHREG procedure includes four methods of handling ties. The *discrete* logistic model is available for discrete time-scale data. The other three methods apply to continuous time-scale data. The *exact* method computes the exact conditional probability under the model that the set of observed tied event times occurs before all the censored times with the same value or before larger values. *Breslow* and *Efron* methods provide approximations to the exact method.

Variable selection is a typical exploratory exercise in multiple regression when the investigator is interested in identifying important prognostic factors from a large number of candidate variables. The PHREG procedure provides four model selection methods: forward selection, backward elimination, stepwise selection, and best subset selection. The best subset selection method is based on the likelihood score statistic. This method identifies a specified number of best models containing one, two, three variables and so on, up to the single model containing all of the explanatory variables.

The PHREG procedure also enables you to

- include an offset variable in the model
- weight the observations in the input data
- test linear hypotheses about the regression parameters
- perform conditional logistic regression analysis for matched case-control studies
- create a SAS data set containing survivor function estimates, residuals, and regression diagnostics
- create a SAS data set containing survival distribution estimates and confidence interval for the survivor function at each event time for a given realization of the explanatory variables

PROC PHREG can also be used to fit the multinomial logit choice model to discrete choice data. See [http://support.sas.com/techsup/tnote/tnote_stat.html#market] for more information on discrete choice modeling and the multinomial logit model. Look for the latest “Discrete Choice” report.

The remaining sections of this chapter contain information on how to use PROC PHREG, information on the underlying statistical methodology, and some sample applications of the procedure. The “Getting Started” section on page 3217 introduces PROC PHREG with two examples. The “Syntax” section on page 3221 describes the syntax of the procedure. The “Details” section on page 3239 summarizes the statistical techniques employed in PROC PHREG. The “Examples” section on page 3272 includes eight additional examples of useful applications. Experienced SAS/STAT software users may decide to proceed to the “Syntax” section, while other users may choose to read both the “Getting Started” and “Examples” sections before proceeding to “Syntax” and “Details.”

Experimental graphics are now available in PROC PHREG for model assessment. For more information, see the section “ODS Graphics” on page 3271.

Getting Started

PROC PHREG syntax is similar to that of the other regression procedures in the SAS System. For simple uses, only the PROC PHREG and MODEL statements are required.

Consider the following data from (Kalbfleisch and Prentice 1980). Two groups of rats received different pretreatment regimes and then were exposed to a carcinogen. Investigators recorded the survival times of the rats from exposure to mortality from vaginal cancer. Four rats died of other causes, so their survival times are censored. Interest lies in whether the survival curves differ between the two groups.

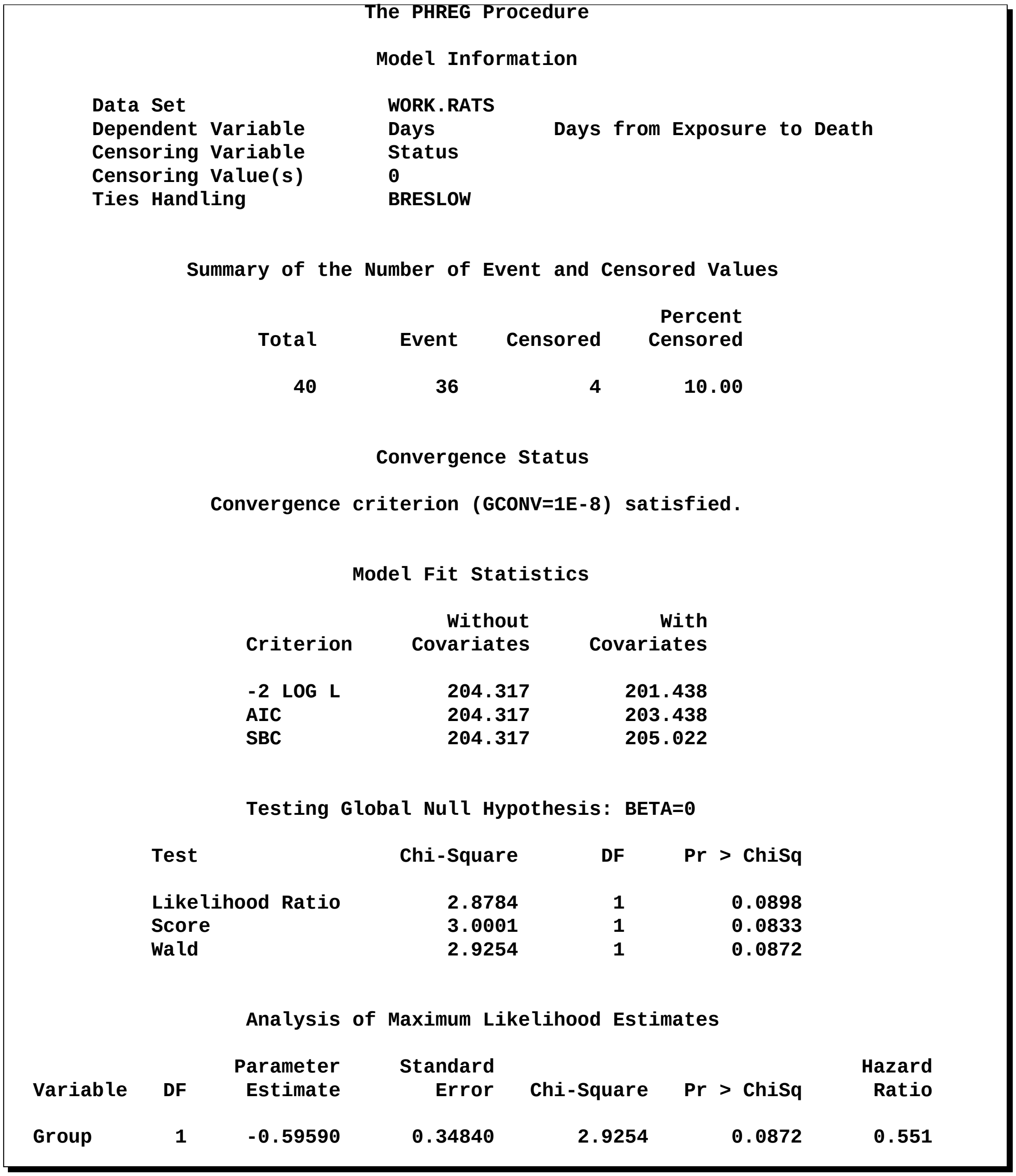
The data set Rats contains the variable Days (the survival time in days), the variable Status (the censoring indicator variable: 0 if censored and 1 if not censored), and the variable Group (the pretreatment group indicator).

```
data Rats;
   label Days  ='Days from Exposure to Death';
   input Days Status Group @@;
   datalines;
143 1 0   164 1 0   188 1 0   188 1 0
190 1 0   192 1 0   206 1 0   209 1 0
213 1 0   216 1 0   220 1 0   227 1 0
230 1 0   234 1 0   246 1 0   265 1 0
304 1 0   216 0 0   244 0 0   142 1 1
156 1 1   163 1 1   198 1 1   205 1 1
232 1 1   232 1 1   233 1 1   233 1 1
233 1 1   233 1 1   239 1 1   240 1 1
261 1 1   280 1 1   280 1 1   296 1 1
296 1 1   323 1 1   204 0 1   344 0 1
;
run;
```

In the MODEL statement, the response variable, Days, is crossed with the censoring variable, Status, with the value that indicates censoring enclosed in parentheses (0). The values of Days are considered censored if the value of Status is 0; otherwise, they are considered event times.

```
proc phreg data=Rats;
   model Days*Status(0)=Group;
run;
```

Results of the PROC PHREG analysis appear in Figure 54.1. Since Group takes only two values, the null hypothesis for no difference between the two groups is identical to the null hypothesis that the regression coefficient for Group is 0. All three tests in the "Testing Global Null Hypothesis: BETA=0" table (see the section "Testing the Global Null Hypothesis" on page 3246) suggest that the survival curves for the two pretreatment groups may not be the same. In this model, the hazards ratio (or risk ratio) for Group, defined as the exponentiation of the regression coefficient for Group, is the ratio of the hazard functions between the two groups. The estimate is 0.551, implying that the hazard function for Group=1 is smaller than that for Group=0. In other words, rats in Group=1 lived longer than those in Group=0.



```
                              The PHREG Procedure

                               Model Information

     Data Set                 WORK.RATS
     Dependent Variable       Days            Days from Exposure to Death
     Censoring Variable       Status
     Censoring Value(s)       0
     Ties Handling            BRESLOW


             Summary of the Number of Event and Censored Values

                                                     Percent
                   Total       Event    Censored    Censored

                      40          36           4       10.00


                              Convergence Status

               Convergence criterion (GCONV=1E-8) satisfied.


                            Model Fit Statistics

                                  Without           With
                  Criterion     Covariates     Covariates

                  -2 LOG L         204.317        201.438
                  AIC              204.317        203.438
                  SBC              204.317        205.022


                  Testing Global Null Hypothesis: BETA=0

          Test                 Chi-Square       DF     Pr > ChiSq

          Likelihood Ratio         2.8784        1         0.0898
          Score                    3.0001        1         0.0833
          Wald                     2.9254        1         0.0872


                  Analysis of Maximum Likelihood Estimates

                 Parameter     Standard                                Hazard
Variable   DF     Estimate        Error   Chi-Square   Pr > ChiSq       Ratio

Group       1     -0.59590      0.34840       2.9254       0.0872       0.551
```

Figure 54.1. Comparison of Two Survival Curves

In this example, the comparison of two survival curves is put in the form of a proportional hazards model. This approach is essentially the same as the log-rank (Mantel-Haenszel) test. In fact, if there are no ties in the survival times, the likelihood score test in the Cox regression analysis is identical to the log-rank test. The advantage of the Cox regression approach is the ability to adjust for the other variables by including them in the model. For example, the present model could be expanded by including a variable that contains the initial body weights of the rats.

Next, consider a simple test of the validity of the proportional hazards assumption. The proportional hazards model for comparing the two pretreatment groups is given by the following:

$$\lambda(t) = \begin{cases} \lambda_0(t) & \text{if GROUP} = 0 \\ \lambda_0(t)e^{\beta_1} & \text{if GROUP} = 1 \end{cases}$$

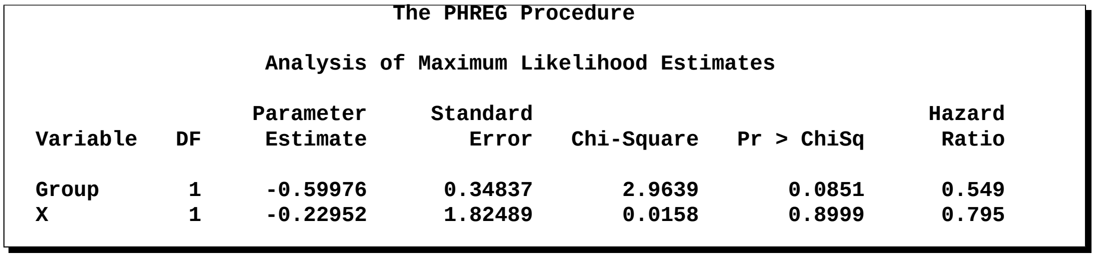
The ratio of hazards is e^{β_1}, which does not depend on time. If the hazard ratio changes with time, the proportional hazards model assumption is invalid. Simple forms of departure from the proportional hazards model can be investigated with the following time-dependent explanatory variable $x = x(t)$:

$$x(t) = \begin{cases} 0 & \text{if GROUP} = 0 \\ \log(t) - 5.4 & \text{if GROUP} = 1 \end{cases}$$

Here, $\log(t)$ is used instead of *t* to avoid numerical instability in the computation. The constant, 5.4, is the average of the logs of the survival times and is included to improve interpretability. The hazard ratio in the two groups then becomes $e^{\beta_1 - 5.4\beta_2}t^{\beta_2}$, where β_2 is the regression parameter for the time-dependent variable *x*. The term e^{β_1} represents the hazard ratio at the geometric mean of the survival times. A nonzero value of β_2 would imply an increasing $(\beta_2 > 0)$ or decreasing $(\beta_2 < 0)$ trend in the hazard ratio with time.

The MODEL statement in this analysis also includes the time-dependent explanatory variable X, which is defined within the procedure by the programming statement that follows the MODEL statement. At each event time, subjects in the risk set (those alive just before the event time) have their X values changed accordingly.

```
proc phreg data=Rats;
   model Days*Status(0)=Group X;
   X=Group*(log(Days) - 5.4);
run;
```



```
                                The PHREG Procedure

                      Analysis of Maximum Likelihood Estimates

                     Parameter      Standard                               Hazard
Variable     DF       Estimate         Error    Chi-Square    Pr > ChiSq    Ratio

Group         1       -0.59976       0.34837        2.9639        0.0851    0.549
X             1       -0.22952       1.82489        0.0158        0.8999    0.795
```

Figure 54.2. A Simple Test of Trend in the Hazard Ratio

The analysis of the parameter estimates is displayed in Figure 54.2. The Wald chi-square statistic for testing the null hypothesis that $\beta_2 = 0$ is 0.0158. The statistic is not statistically significant when compared to a chi-square distribution with one degree of freedom $(p = 0.8999)$. Thus, you can conclude that there is no evidence of an increasing or decreasing trend over time in the hazard ratio. See the "Examples" section beginning on page 3272 for additional illustrations of PROC PHREG usage.

Syntax

The following statements are available in PROC PHREG.

```
PROC PHREG < options > ;
    ASSESS keyword < / options > ;
    MODEL response < *censor(list) > = variables < /options > ;
    < programming statements >
    STRATA variable < (list) > < ...variable < (list) >>< /option > ;
    < label: > TEST equation1 < ,..., equationk >< /option > ;
    FREQ variable ;
    WEIGHT variable < /option >;
    ID variables ;
    OUTPUT < OUT=SAS-data-set >
          < keyword=name... keyword=name >< /options > ;
    BASELINE < OUT=SAS-data-set >
          < COVARIATES=SAS-data-set >
          < keyword=name... keyword=name >< /options > ;
    BY variables ;
```

The PROC PHREG statement invokes the procedure. All other statements except the MODEL statement are optional. Items within < > are optional, and there is no required order for the statements following the PROC PHREG statement. The MODEL statement specifies the variables that define the survival time, the censoring variable, and the explanatory variables. The STRATA statement specifies a variable or set of variables defining the strata for the analysis. The TEST statement contains equations that define linear hypotheses concerning the model parameters. The ID statement specifies the variables with values that are used to label the observations. The OUTPUT and BASELINE statements create data sets containing the survival estimates. DATA step programming statements can be included to create time-dependent explanatory variables.

PROC PHREG Statement

PROC PHREG < *options* > **;**

You can specify the following options in the PROC PHREG statement.

COVOUT

adds the estimated covariance matrix of the parameter estimates to the OUTEST= data set. The COVOUT option has no effect unless the OUTEST= option is specified.

COVM

requests the model-based covariance matrix (which is the inverse of the observed information matrix) be presented and used in the analysis if the COVS option is also specified. The COVM option has no effect if the COVS option is not specified.

COVSANDWICH < (AGGREGATE) >
COVS < (AGGREGATE) >

requests the robust sandwich estimate of Lin and Wei (1989) for the covariance matrix. When this option is specified, this robust sandwich estimate is used in the Wald tests for testing the global null hypothesis, null hypotheses of individual parameters, and the hypotheses in the TEST statements. In addition, a modified score test is computed in the testing of the global null hypothesis, and the parameter estimates table has an additional StdErrRatio column, which contains the ratios of the robust estimate of the standard error relative to the corresponding model-based estimate. Optionally, you can specify the keyword AGGREGATE enclosed in parentheses after the COVSANDWICH (or COVS) option, which requests a summing up of the score residuals for each distinct ID pattern in the computation of the robust sandwich covariance estimate. This AGGREGATE option has no effects if the ID statement is not specified.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MULTIPASS

requests that, for each Newton-Raphson iteration, PROC PHREG recompiles the risk sets corresponding to the event times for the (start,stop) style of response and recomputes the values of the time-dependent variables defined by the programming statements for each observation in the risk sets. If the MULTIPASS option is not specified, PROC PHREG computes all risk sets and all the variable values and saves them into a utility file. The MULTIPASS option decreases required disk space at the expense of increased execution time; however, for very large data, it may actually save time since it is time consuming to write and read large utility files. This option has an effect only when the (start,stop) style of response is used or when there are time-dependent explanatory variables.

NOPRINT

suppresses all displayed output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, “Using the Output Delivery System,” for more information.

NOSUMMARY

suppresses the display of the event and censored observation frequencies.

OUTEST=*SAS-data-set*

creates an output SAS data set that contains estimates of the regression coefficients. If you use the COVOUT option, the data set also contains the estimated covariance matrix of the parameter estimates. The data set includes

- any BY variables specified
- _TIES_, a character variable of length 8 with four possible values: BRESLOW, DISCRETE, EFRON, and EXACT. These are the four values of the TIES= option in the MODEL statement.

- _TYPE_, a character variable of length 8 with two possible values: PARMS for parameter estimates or COV for covariance estimates. If both the COVM and COVS options are specified in the PROC LIFETEST statement along with the COVOUT option, _TYPE_=COVM for the model-based covariance estimates and _TYPE_=COVS for the robust sandwich covariance estimates.
- _STATUS_, a character variable indicating whether the estimates have converged
- _NAME_, a character variable containing the name of the TIME variable for the row of parameter estimates and the name of each explanatory variable to label the rows of covariance estimates
- one variable for each explanatory variable in the MODEL statement. In a forward, backward, or stepwise regression analysis, if an explanatory variable is not included in the final model, the corresponding parameter estimate and covariances are set to missing.
- _LNLIKE_, a numeric variable containing the last computed value of the log likelihood

SIMPLE

displays simple descriptive statistics (mean, standard deviation, minimum, and maximum) for each explanatory variable in the MODEL statement.

ASSESS Statement (Experimental)

ASSESS < **VAR=***(list)* > < **PH** > < */options* > **;**

The ASSESS statement performs the graphical and numerical methods of Lin, Wei, and Ying (1993) for checking the adequacy of the Cox regression model. The methods are derived from cumulative sums of martingale residuals over follow-up times or covariate values. You can assess the functional form of a covariate or you can check the proportional hazards assumption for each covariate in the Cox model. PROC PHREG uses the experimental ODS graphics for the graphical displays. For specific information about the experimental graphics that is available in PROC PHREG, see the section "ODS Graphics" on page 3271. You must specify at least one of the following to create an analysis.

VAR=(list)

specifies the list of explanatory variables for which their functional forms are assessed. For each variable on the list, the observed cumulative martingale residuals are plotted against the values of the explanatory variable along with 20 (or *n* if NPATHS=*n* is specified) simulated residual patterns.

PROPORTIONALHAZARDS
PH

requests the checking of the proportional hazards assumption. For each explanatory variable in the model, the observed score process component is plotted against the follow-up time along with 20 (or *n* if NPATHS=*n* is specified) simulated patterns.

The following options can be specified after a slash (/).

NPATHS=*n*

specifies the number of simulated residual patterns to be displayed in a cumulative martingale residual plot or a score process plot. The default is *n*=20.

CRPANEL

requests that a plot with four panels, each containing the observed cumulative martingale residuals and two simulated residual patterns, be created.

RESAMPLE < =*n* **>**

requests that the Kolmogorov-type supremum test be computed on 1,000 simulated patterns or on *n* simulated patterns if *n* is specified.

SEED=*n*

specifies an integer seed for the random number generator used in creating simulated realizations for plots and for the Kolmogorov-type supremum tests. Specifying a seed enables you to reproduce identical graphs and p-values for the model assessments from the same PHREG specification. If the SEED= option is not specified, or if you specify a nonpositive seed, a random seed is derived from the time of day.

BASELINE Statement

BASELINE < **OUT=** *SAS-data-set* >< **COVARIATES=** *SAS-data-set* >
< *keyword=name ... keyword=name* >< */options* > **;**

The BASELINE statement creates a new SAS data set that contains the survivor function estimates at the event times of each stratum for every pattern of explanatory variable values ($\mathbf{x}$) given in the COVARIATES= data set. By default, the data set also contains the survivor function estimates corresponding to the means of the explanatory variables ($\mathbf{x} = \bar{\mathbf{z}}$) for each stratum. If you want only these estimates, you can omit the COVARIATES= option. No BASELINE data set is created if the model contains a time-dependent variable defined by means of programming statement.

The following list explains specifications in the BASELINE statement.

OUT=*SAS-data-set*

names the output BASELINE data set. If you omit the OUT= option, the data set is created and given a default name using the DATA*n* convention.

COVARIATES=*SAS-data-set*

names the SAS data set containing the set of explanatory variable values for which the survivor functions are estimated. There must be a corresponding variable in the COVARIATES= data set for each explanatory variable in the final model.

keyword=name

specifies the statistics included in the BASELINE data set and assigns names to the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic. The keywords and the corresponding statistics are

CMF	cumulative mean function estimate for recurrent events data. Specifying CMF=_ALL_ is equivalent to specifying CMF=CMF, STDCMF=StdErrCMF, LOWERCMF=LowerCMF,

and UPPERCMF=UpperCMF. Nelson (2002) refers to the mean function estimate as MCF (mean cumulative function).

CUMHAZ cumulative hazard function estimate for recurrent events data. Specifying CMFHAZ=_ALL_ is equivalent to specifying CUMHAZ=CumHaz, STDCUMHAZ=StdErrCumHaz, LOWERCUMHAZ=LowerCumHaz, and UPPERCUMHAZ=UpperCumHaz.

LOGLOGS log of the negative log of SURVIVAL

LOGSURV log of SURVIVAL

LOWER | L lower pointwise confidence limit for the survivor function. The confidence level is determined by the ALPHA= option.

LOWERCMF lower pointwise confidence limit for the cumulative mean function. The confidence level is determined by the ALPHA= option.

LOWERCUMHAZ lower pointwise confidence limit for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

STDERR standard error of the survivor function estimator

STDCMF standard error of the cumulative mean function estimator

STDCUMHAZ standard error of the cumulative hazard function estimator

STDXBETA standard error of the linear predictor estimator, $\sqrt{\mathbf{x}'\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{x}}$

SURVIVAL survivor function estimate $\widehat{S}(t) = [\widehat{S}_0(t)]^{\exp(\mathbf{x}'\widehat{\boldsymbol{\beta}})}$ Specifying SURVIVAL=_ALL_ is equivalent to specifying SURVIVAL=Survival, STDERR=StdErrSurvival, LOWER=LowerSurvival, and UPPER=UpperSurvival.

UPPER | U upper pointwise confidence limit for the survivor function. The confidence level is determined by the ALPHA= option.

UPPERCMF upper pointwise confidence limit for the cumulative mean function. The confidence level is determined by the ALPHA= option.

UPPERCUMHAZ upper pointwise confidence limit for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

XBETA estimate of the linear predictor, $\mathbf{x}'\widehat{\boldsymbol{\beta}}$

The following options can appear in the BASELINE statement after a slash (/).

ALPHA=*value*

specifies the significance level of the confidence interval for the survivor function. The value must be between 0 and 1. The default is 0.05, which results in a 95% confidence interval.

CLTYPE=*method*

specifies the method used to compute the confidence limits for $S(t, \mathbf{z})$, the survivor function for a subject with a fixed covariate vector $\mathbf{z}$ at event time t. The CLTYPE= option can take the following values:

LOG specifies that the confidence limits for $\log(S(t,\mathbf{z}))$ are to be computed using the normal theory approximation. The confidence limits for $S(t,\mathbf{z})$ are obtained by back-transforming the confidence limits for $\log(S(t,\mathbf{z}))$. The default is CLTYPE=LOG.

LOGLOG specifies that the confidence limits for the $\log(-\log(S(t,\mathbf{z})))$ are to be computed using normal theory approximation. The confidence limits for $S(t,\mathbf{z})$ are obtained by back-transforming the confidence limits for $\log(-\log(S(t,\mathbf{z})))$.

NORMAL specifies that the confidence limits for $S(t,\mathbf{z})$ are to be computed directly using normal theory approximation.

METHOD=*method*

specifies the method used to compute the survivor function estimates. The two available methods are

CH | EMP | NELSON specifies that the Nelson (empirical) cumulative hazard function estimate of the survivor function is to be computed; that is, the survivor function is estimated by exponentiating the negative empirical cumulative hazard function.

PL specifies that the product-limit estimate of the survivor function is to be computed. The default is METHOD=PL.

NOMEAN

excludes the survivor function estimates corresponding to the sample means of the explanatory variables.

The METHOD= and CLTYPE= options apply only to the survival estimates. For recurrent events data, both CMF= and CUMHAZ= statistics are the Nelson estimators, but their standard error are not the same. Confidence limits for the cumulative mean function and cumulative hazard function are based on the log transform.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PHREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PHREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Contents*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* < */option* > **;**

The *variable* in the FREQ statement identifies the variable (in the input data set) containing the frequency of occurrence of each observation. PROC PHREG treats each observation as if it appears *n* times, where *n* is the value of the FREQ variable for the observation. If not an integer, the frequency value is truncated to an integer. If the frequency value is missing, the observation is not used in the estimation of the regression parameters.

The following option can be specified in the FREQ statement after a slash (/):

NOTRUNCATE
NOTRUNC
specifies that frequency values are not truncated to integers.

ID Statement

ID *variables* **;**

The ID statement specifies additional variables for identifying observations in the input data. These variables are placed in the OUT= data set created by the OUTPUT statement. In the computation of the robust sandwich variance estimate, you can aggregate over distinct values of these ID variables.

Only variables in the input data set can be included in the ID statement.

MODEL Statement

MODEL *response* < **censor (list)* > = *variables* < */options* > **;**

MODEL *(t1, t2)* < **censor(list)* > = *variables* < */options* > **;**

The MODEL statement identifies the variables to be used as the failure time variables, the optional censoring variable, and the explanatory variables. Two forms of MODEL syntax can be specified; the first form allows one response variable, while the second form allows two variables for the counting process style of input (see the section "Counting Process Style of Input" on page 3241 for more information).

In the first MODEL statement, preceding the equal sign, is the name of the failure time variable. This can optionally be followed by an asterisk, the name of the censoring variable, and a list of censoring values (separated by blanks or commas if there is more than one) enclosed in parentheses. If the censoring variable takes on one of these values, the corresponding failure time is considered to be censored. The variables following the equal sign are the explanatory variables (sometimes called independent variables or covariates) for the model.

Instead of a single failure time variable, the second MODEL statement identifies a pair of failure time variables. Their names are enclosed in parentheses, and they signify the endpoints of a semi-closed interval $(t1, t2]$ during which the subject is at risk. If the censoring variable takes on one of the censoring values, the time $t2$ is considered to be censored.

The censoring variable and the explanatory variables must be numeric. The failure time variables must contain nonnegative values. Any observation with a negative failure time is excluded from the analysis, as is any observation with a missing value for any of the variables listed in the MODEL statement.

You can specify the following options in the MODEL statement.

Ties-Handling Option

TIES=*method*

specifies how to handle ties in the failure time. The TIES= option can take the following values:

BRESLOW uses the approximate likelihood of Breslow (1974). This is the default value.

DISCRETE replaces the proportional hazards model by the discrete logistic model

$$\frac{\lambda(t;\mathbf{z})}{1-\lambda(t;\mathbf{z})} = \frac{\lambda_0(t)}{1-\lambda_0(t)} \exp(\mathbf{z}'\boldsymbol{\beta})$$

where $\lambda_0(t)$ and $h(t;\mathbf{z})$ are discrete hazard functions.

EFRON uses the approximate likelihood of Efron (1977).

EXACT computes the exact conditional probability under the proportional hazards assumption that all tied event times occur before censored times of the same value or before larger values. This is equivalent to summing all terms of the marginal likelihood for $\boldsymbol{\beta}$ that are consistent with the observed data (Kalbfleisch and Prentice 1980; DeLong, Guirguis, and So 1994).

The EXACT method may take a considerable amount of computer resources. If ties are not extensive, the EFRON and BRESLOW methods provide satisfactory approximations to the EXACT method for the continuous time-scale model. In general, Efron's approximation gives results that are much closer to the EXACT method results than Breslow's approximation does. If the time scale is genuinely discrete, you should use the DISCRETE method. The DISCRETE method is also required in the analysis of case-control studies when there is more than one case in a matched set. If there are no ties, all four methods result in the same likelihood and yield identical estimates. The default, TIES=BRESLOW, is the most efficient method when there are no ties.

Model-Specification Options

ENTRYTIME=*variable*
ENTRY=*variable*

specifies the name of the variable that represents the left truncation time. This option has no effect when the counting process style of input is specified. See the section "Left Truncation of Failure Times" on page 3263 for more information.

NOFIT

performs the global score test, which tests the joint significance of all the explanatory variables in the MODEL statement. No parameters are estimated. If the NOFIT option is specified along with other MODEL statement options, NOFIT takes precedence, and all other options are ignored except the TIES= option.

OFFSET=*name*

specifies the name of an offset variable, which is an explanatory variable with a regression coefficient fixed as one. This option can be used to incorporate risk weights for the likelihood function.

SELECTION=*method*

specifies the method used to select the model. The *method*s available are

BACKWARD | B requests backward elimination.

FORWARD | F requests forward selection.

NONE | N fits the complete model specified in the MODEL statement. This is the default value.

SCORE requests best subset selection. It identifies a specified number of models with the highest score chi-square statistic for all possible model sizes ranging from one explanatory variable to the total number of explanatory variables listed in the MODEL statement.

STEPWISE | S requests stepwise selection.

For more information, see the section "Variable Selection Methods" on page 3264.

Model-Building Options

The following options enable you to provide additional specifications for the BACKWARD, FORWARD, SCORE, and STEPWISE model selection methods. They have no effect when SELECTION=NONE. Only the INCLUDE=, START=, STOP=, and BEST= options work with the SCORE method.

BEST=*n*

is used exclusively with the SCORE model selection method. The BEST=*n* option specifies that *n* models with the highest score chi-square statistics are to be displayed for each model size. If the option is omitted and there are no more than 10 explanatory variables, then all possible models are listed for each model size. If the option is omitted and there are more than 10 explanatory variables, then the number of models selected for each model size is, at most, equal to the number of explanatory variables listed in the MODEL statement. See Example 54.2 on page 3279 for an illustration of the SCORE selection method and the BEST= option.

DETAILS

produces a detailed display at each step of the model-building process. It produces an “Analysis of Variables Not in the Model” table before displaying the variable selected for entry for FORWARD or STEPWISE selection. For each model fitted, it produces the “Analysis of Maximum Likelihood Estimates” table. See Example 54.1 on page 3272 for a discussion of these tables.

INCLUDE=*n*

includes the first *n* explanatory variables listed in the MODEL statement in every model. The value for *n* ranges from 1 to *s*, where *s* is the number of explanatory variables in the MODEL statement. The default value of *n* is 0.

MAXSTEP=*n*

specifies the maximum number of times the explanatory variables can move in and out of the model before the STEPWISE model-building process ends. The default value for *n* is twice the number of explanatory variables in the MODEL statement. The option has no effect for other model selection methods.

SEQUENTIAL

forces variables to be added to the model in the order specified in the MODEL statement or to be eliminated from the model in the reverse order specified in the MODEL statement.

SLENTRY=*value*
SLE=*value*

specifies the significance level (a value between 0 and 1) for entering an explanatory variable into the model in the FORWARD or STEPWISE method. For all variables not in the model, the one with the smallest *p*-value is entered if the *p*-value is less than or equal to the specified significance level. The default value is 0.05.

SLSTAY=*value*
SLS=*value*

specifies the significance level (a value between 0 and 1) for removing an explanatory variable from the model in the BACKWARD or STEPWISE method. For all variables in the model, the one with the largest *p*-value is removed if the *p*-value exceeds the specified significance level. The default value is 0.05.

START=*n*

begins the FORWARD, BACKWARD, or STEPWISE model selection process with the first *n* explanatory variables listed in the MODEL statement. The value for *n* ranges from 0 to *s*, where *s* is the total number of explanatory variables in the MODEL statement. The default value of *n* is *s* for the BACKWARD method and 0 for the FORWARD and STEPWISE methods. Note that START=*n* specifies only that the first *n* explanatory variables appear in the first model, while INCLUDE=*n* specifies that the first *n* explanatory variables be included in every model. For the SCORE method, START=*n* specifies that the smallest models contain *n* explanatory variables, where *n* ranges from 1 to *s*. The default value of *n* is 1.

STOP=*n*

specifies the maximum (FORWARD method) or minimum (BACKWARD method) number of explanatory variables to be included in the final model. The value for *n* ranges from 0 to *s*, where *s* is the number of explanatory variables in the MODEL statement. The default value of *n* is 0 for the BACKWARD method and *s* for the FORWARD method. For the SCORE method, STOP=*n* specifies that the largest models contain *n* explanatory variables, where *n* ranges from 1 to *s*. The default value of *n* is *s*. The STOP= option has no effect for the STEPWISE method.

STOPRES
SR

specifies that the addition and deletion of variables are to be based on the result of the likelihood score test for testing the joint significance of variables not in the model. This score chi-square statistic is referred to as the residual chi-square. In the FORWARD method, the STOPRES option enters the explanatory variables into the model one at a time until the residual chi-square becomes insignificant (that is, until the *p*-value of the residual chi-square exceeds the SLENTRY= value). In the BACKWARD method, the STOPRES option removes variables from the model one at a time until the residual chi-square becomes significant (that is, until the *p*-value of the residual chi-square becomes less than the SLSTAY= value). The STOPRES option has no effect for the STEPWISE method.

Optimization Options

Four convergence criteria are allowed: ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E−8.

ABSFCONV=*value*

specifies the absolute function convergence criterion. Termination requires a small change in the objective function (log partial likelihood function) in subsequent iterations,

$$|l_k - l_{k-1}| < value$$

where l_k is the value of the objective function at iteration k.

CONVERGELIKE=*value*

is the same as specifying the ABSFCONV= option.

CONVERGEPARM=*value*

is the same as specifying the XCONV= option.

FCONV=*value*

specifies the relative function convergence criterion. Termination requires a small relative change in the objective function (log partial likelihood function) in subsequent iterations,

$$\frac{|l_k - l_{k-1}|}{|l_{k-1}| + 1\mathrm{E}-6} < value$$

where l_k is the value of the objective function at iteration k.

GCONV=*value*

specifies the relative gradient convergence criterion. Termination requires that the normalized prediction function reduction is small,

$$\frac{\mathbf{g}_k \mathbf{H}_k^{-1} \mathbf{g}_k}{|l_k| + 1\mathrm{E}-6} < value$$

where l_k is the log partial likelihood, $\mathbf{g}_k$ is the gradient vector (first partial derivatives of the log partial likelihood), and $\mathbf{H}_k$ is the negative Hessian matrix (second partial derivatives of the log partial likelihood), all at iteration k.

MAXITER=*n*

specifies the maximum number of iterations allowed. The default value for *n* is 25. If convergence is not attained in *n* iterations, the displayed output and all data sets created by PROC PHREG contain results that are based on the last maximum likelihood iteration.

RIDGING=ABSOLUTE | RELATIVE | NONE

specifies the technique to improve the log-likelihood when its value is worse than that of the previous step. For RIDGING=ABSOLUTE, the diagonal elements of the negative (expected) Hessian are inflated by adding the ridge value. For RIDGING=RELATIVE, the diagonal elements are inflated by the factor equal to 1 plus the ridge value. For RIDGING=NONE, the crude line-search method of taking half a step is used instead of ridging.

SINGULAR=*value*

specifies the singularity criterion for determining linear dependencies in the set of explanatory variables. The default value is 10^{-12}.

XCONV=*value*

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

$$\max_i |\delta_k^{(i)}| < value$$

where

$$\delta_k^{(i)} = \begin{cases} \theta_k^{(i)} - \theta_{k-1}^{(i)} & |\theta_{k-1}^{(i)}| < .01 \\ \frac{\theta_k^{(i)} - \theta_{k-1}^{(i)}}{\theta_{k-1}^{(i)}} & \text{otherwise} \end{cases}$$

where $\theta_k^{(i)}$ is the estimate of the ith parameter at iteration k.

Display Options

ALPHA=*value*

sets the significance level used for the confidence limits for the hazards ratios. The value must be between 0 and 1. The default value is 0.05, which results in the calculation of a 95% confidence interval. This option has no effect unless the RISKLIMITS option is specified.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COVB

displays the estimated covariance matrix of the parameter estimates.

ITPRINT

displays the iteration history, including the last evaluation of the gradient vector.

RISKLIMITS
RL

displays, for each explanatory variable, the $100(1-\alpha)\%$ confidence limits for the hazards ratio (e^{β_i}). The value for α is determined by the ALPHA= option.

OUTPUT Statement

OUTPUT <**OUT=** *SAS-data-set* >
< *keyword=name ... keyword=name* >< */options* > **;**

The OUTPUT statement creates a new SAS data set containing statistics calculated for each observation. These can include the estimated linear predictor ($\mathbf{z}_j'\widehat{\boldsymbol{\beta}}$) and its standard error, survival distribution estimates, residuals, and influence statistics. In addition, this data set includes the time variable, the explanatory variables listed in the MODEL statement, the censoring variable (if specified), and the BY, STRATA, FREQ, and ID variables (if specified).

For observations with missing values in the time variable or any explanatory variables, the output statistics are set to missing. However, for observations with missing values only in the censoring variable or the FREQ variable, survival estimates are still computed. Therefore, by adding observations with missing values in the FREQ variable or the censoring variable, you can compute the survivor function estimates for new observations or for settings of explanatory variables not present in the data without affecting the model fit.

No OUTPUT data set is created if the model contains a time-dependent variable defined by means of programming statements.

The following list explains specifications in the OUTPUT statement.

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the OUTPUT data set is created and given a default name using the DATA*n* convention.

keyword=name

specifies the statistics included in the OUTPUT data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and either a variable or a list of variables to contain the statistic. The keywords that accept a list of variables are DFBETA, RESSCH, RESSCO, and WTRESSCH. For these keywords, you can specify as many names in *name* as the number of explanatory variables specified in the MODEL statement. If you specify *k* names and *k* is less than the total number of explanatory variables, only the changes for the first *k* parameter estimates are output. The keywords and the corresponding statistics are as follows:

DFBETA approximate changes in the parameter estimates $(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_{(j)})$ when the jth observation is omitted. These variables are a weighted transform of the score residual variables and are useful in assessing local influence and in computing robust variance estimates.

LD approximate likelihood displacement when the observation is left out. This diagnostic can be used to assess the impact of each observation on the overall fit of the model.

LMAX relative influence of observations on the overall fit of the model. This diagnostic is useful in assessing the sensitivity of the fit of the model to each observation.

LOGLOGS log of the negative log of SURVIVAL

LOGSURV log of SURVIVAL

NUM_LEFT number of subjects at risk at the observation time τ_j (or at the right endpoint of the at risk interval when a counting process MODEL specification is used)

RESDEV deviance residual $\widehat{D}_j$. This is a transform of the martingale residual to achieve a more symmetric distribution.

RESMART martingale residual $\widehat{M}_j$. The residual at the observation time τ_j can be interpreted as the difference over $[0, \tau_j]$ in the observed number of events minus the expected number of events given by the model.

RESSCH Schoenfeld residuals. These residuals are useful in assessing the proportional hazards assumption.

RESSCO score residuals. These residuals are a decomposition of the first partial derivative of the log likelihood. They can be used to assess the leverage exerted by each subject in the parameter estimation. They are also useful in constructing robust sandwich variance estimators.

STDXBETA standard error of the estimated linear predictor, $\sqrt{\mathbf{z}_j'\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{z}_j}$

SURVIVAL survivor function estimate $\widehat{S}_j = [\widehat{S}_0(\tau_j)]^{\exp(\mathbf{z}_j'\widehat{\boldsymbol{\beta}})}$, where τ_j is the observation time

WTRESSCH weighted Schoenfeld residuals. These residuals are useful in investigating the nature of nonproportionality if the proportional hazard assumption does not hold.

XBETA estimate of the linear predictor, $\mathbf{z}_j'\widehat{\boldsymbol{\beta}}$

The following options can appear in the OUTPUT statement after a slash (/).

ORDER=*sort_order*

specifies the order of the observations in the OUTPUT data set. Available values for *sort_order* are

DATA requests that the output observations be sorted the same as the input data set.

SORTED requests that the output observations be sorted by strata and descending order of the time variable within each stratum.

The default is ORDER=DATA.

METHOD=*method*

specifies the method used to compute the survivor function estimates. The two available methods are

CH | EMP specifies that the empirical cumulative hazard function estimate of the survivor function is to be computed; that is, the survivor function is estimated by exponentiating the negative empirical cumulative hazard function.

PL specifies that the product-limit estimate of the survivor function is to be computed. The default is METHOD=PL.

Programming Statements

Programming statements are used to create or modify the values of the explanatory variables in the MODEL statement. They are especially useful in fitting models with time-dependent explanatory variables. Programming statements can also be used to create explanatory variables that are not time dependent. For example, you can create indicator variables from a categorical variable and incorporate them into the model. PROC PHREG programming statements cannot be used to create or modify the values of the response variable, the censoring variable, the frequency variable, or the strata variables.

The following DATA step statements are available in PROC PHREG:

```
ABORT
ARRAY
assignment statements
CALL
DO
iterative DO
DO UNTIL
DO WHILE
END
GOTO
IF-THEN/ELSE
LINK-RETURN
PUT
SELECT
SUM statement
```

By default, the PUT statement in PROC PHREG writes to the Output window instead of the Log window. If you want the results of the PUT statements to go to the Log window, add the following statement before the PUT statements:

```
FILE LOG;
```

DATA step functions are also available. Use these programming statements the same way you use them in the DATA step. For detailed information, refer to *SAS Language Reference: Dictionary*.

Consider the following example of using programming statements in PROC PHREG. Suppose blood pressure is measured at multiple times during the course of a study investigating the effect of blood pressure on some survival time. By treating the blood pressure as a time-dependent explanatory variable, you are able to use the value of the most recent blood pressure at each specific point of time in the modeling process rather than using the initial blood pressure or the final blood pressure. The values of the following variables are recorded for each patient, if they are available. Otherwise, the variables contain missing values.

Time	survival time
Censor	censoring indicator (with 0 as the censoring value)
BP0	blood pressure on entry to the study
T1	time 1
BP1	blood pressure at T1
T2	time 2
BP2	blood pressure at T2

The following programming statements create a variable BP. At each time T, the value of BP is the blood pressure reading for that time, if available. Otherwise, it is the last blood pressure reading.

```
proc phreg;
   model Time*Censor(0)=BP;
   BP = BP0;
   if Time>=T1 and T1^=. then BP=BP1;
   if Time>=T2 and T2^=. then BP=BP2;
run;
```

For other illustrations of using programming statements, see the "Getting Started" section on page 3217 and Example 54.4 on page 3285.

STRATA Statement

STRATA *variable* < *(list)* >< ... *variable* < *(list)* >>< */option* > **;**

The proportional hazards assumption may not be realistic for all data. If so, it may still be reasonable to perform a stratified analysis. The STRATA statement names the variables that determine the stratification. Strata are formed according to the nonmissing values of the STRATA variables unless the MISSING option is specified. In the STRATA statement, *variable* is a variable with values that are used to determine the strata levels, and *list* is an optional list of values for a numeric variable. Multiple variables can appear in the STRATA statement.

The values for *variable* can be formatted or unformatted. If the variable is a character variable, or if the variable is numeric and no list appears, then the strata are defined by the unique values of the variable. If the variable is numeric and is followed by a list, then the levels for that variable correspond to the intervals defined by the list. The corresponding strata are formed by the combination of levels and unique values. The list can include numeric values separated by commas or blanks, *value* to *value* by *value* range specifications, or combinations of these.

For example, the specification

```
   strata age (5, 10 to 40 by 10) sex ;
```

indicates that the levels for age are to be less than 5, 5 to 10, 10 to 20, 20 to 30, 30 to 40, and greater than 40. (Note that observations with exactly the cutpoint value fall into the interval preceding the cutpoint.) Thus, with the sex variable, this STRATA statement specifies 12 strata altogether.

The following option can be specified in the STRATA statement after a slash (/).

MISSING

allows missing values ('.' for numeric variables and blanks for character variables) as valid STRATA variable values. Otherwise, observations with missing STRATA variable values are deleted from the analysis.

TEST Statement

< *label:* > **TEST** *equation1* < , . . . , *equationk* >< */option* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. PROC PHREG performs a Wald test for the joint hypothesis specified in a single TEST statement. Each equation specifies a linear hypothesis; multiple equations (rows of the joint hypothesis) are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output, and should always be included. You can submit multiple TEST statements.

The form of an equation is as follows:

term < ±*term* **. . .** > < = < ±*term* < ±*term* **. . .** >>>

here *term* is a variable or a constant or a constant times a variable. The variable is any explanatory variable in the MODEL statement. When no equal sign appears, the expression is set to 0. The following code illustrates possible uses of the TEST statement:

```
proc phreg;
   model time= a1 a2 a3 a4;
   Test1: Test a1, a2;
   Test2: Test a1=0,a2=0;
   Test3: Test a1=a2=a3;
   Test4: Test a1=a2,a2=a3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

The following options can be specified in the TEST statement after a slash (/).

AVERAGE

enables you to assess the average effect of the variables in the given TEST statement. An overall estimate of the treatment effect is computed as a weighted average of the treatment coefficients as illustrated in the following code:

```
TREATMENT: test trt1, trt2, trt3, trt4 / average;
```

Let β_1, β_2, β_3 and β_4 be corresponding parameters for trt1, trt2, trt3, and trt4, respectively. Let $\widehat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4)'$ be estimated coefficient vector and let $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$ be the corresponding variance estimate. Assuming $\beta_1 = \beta_2 = \beta_3 = \beta_4$. The average treatment effect is estimated by $\mathbf{c}'\widehat{\boldsymbol{\beta}}$, where $\mathbf{c} = [1_4'\widehat{\mathbf{V}}^{-1}(\widehat{\boldsymbol{\beta}})1_4]^{-1}\widehat{\mathbf{V}}^{-1}(\widehat{\boldsymbol{\beta}})1_4$ and $1_4 = (1, 1, 1, 1)'$.

E

specifies that the linear coefficients and constants be printed. When the AVERAGE option is specified along with the E option, the optimal weights of the average effect are also printed in the same tables as the coefficients.

PRINT

displays intermediate calculations. This includes $\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}'$ bordered by $(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{c})$, and $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{c})$, where **L** is a matrix of linear coefficients and **c** is a vector of constants. See the section "Testing Linear Hypotheses about Regression Coefficients" on page 3247.

WEIGHT Statement

WEIGHT *variable* < */option* > **;**

The *variable* in the WEIGHT statement identifies the variable in the input data set that contains the case weights. When the WEIGHT statement appears, each observation in the input data set is weighted by the value of the WEIGHT variable. The WEIGHT values can be nonintegral and are not truncated. Observations with negative, zero or missing values for the WEIGHT variable are not used in the model fitting. When the WEIGHT statement is not specified, each observation is assigned a weight of 1. The WEIGHT statement is available for TIES=BRESLOW and TIES=EFRON only.

The following option can be specified in the WEIGHT statement after a slash (/):

NORMALIZE
NORM

causes the weights specified by the WEIGHT *variable* to be normalized so that they add up the actual sample size. With this option, the estimated covariance matrix of the parameter estimators is invariant to the scale of the WEIGHT variable.

Details

Failure Time Distribution

Let T be a nonnegative random variable representing the failure time of an individual from a homogeneous population. The survival distribution function (also known as the survivor function) of T is written as

$$S(t) = \Pr(T \geq t)$$

A mathematically equivalent way of specifying the distribution of T is through its hazard function. The hazard function $\lambda(t)$ specifies the instantaneous failure rate at t. If T is a continuous random variable, $\lambda(t)$ is expressed as

$$\lambda(t) = \lim_{\Delta t \to 0^+} \frac{\Pr(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} = \frac{f(t)}{S(t)}$$

where $f(t)$ is the probability density function of T. If T is discrete with masses at $x_1 < x_2 < \ldots$, then $\lambda(t)$ is given by

$$\lambda(t) = \sum_j \lambda_j \delta(t - x_j)$$

where

$$\delta(u) = \begin{cases} 0 & \text{if } u < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\lambda_j = \Pr(T = x_j \mid T \geq x_j) = \frac{\Pr(T = x_j)}{S(x_j)}$$

for $j = 1, 2, \ldots$

Partial Likelihood Function for the Cox Model

Let $\mathbf{Z}_l(t)$ denote the vector explanatory variables for the lth individual at time t. Let $t_1 < t_2 < \ldots < t_k$ denote the k distinct, ordered event times. Let d_i denote the multiplicity of failures at t_i; that is, d_i is the size of the set $\mathcal{D}_i$ of individuals that fail at t_i. Let w_l be the weight associated with the lth individual. Using this notation, the likelihood functions used in PROC PHREG to estimate $\boldsymbol{\beta}$ are described in the following sections.

Continuous Time Scale

Let $\mathcal{R}_i$ denote the risk set just before the ith ordered event time t_i . Let $\mathcal{R}_i^*$ denote the set of individuals whose event or censored times exceed t_i or whose censored times are equal to t_i.

Exact Likelihood

$$L_1(\boldsymbol{\beta}) = \prod_{i=1}^{k} \left\{ \int_0^\infty \prod_{j \in \mathcal{D}_i} \left[1 - \exp\left(-\frac{\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_j(t_i)}}{\sum_{l \in \mathcal{R}_i^*} \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)}} t \right) \right] \exp(-t)dt \right\}$$

Breslow Likelihood

$$L_2(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\mathrm{e}^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\left[\sum_{l \in \mathcal{R}_i} \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right]^{d_i}}$$

Incorporating weights, the Breslow likelihood becomes

$$L_2(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\mathrm{e}^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} w_j \mathbf{Z}_j(t_i)}}{\left[\sum_{l \in \mathcal{R}_i} w_l \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right]^{\sum_{j \in \mathcal{D}_i} w_i}}$$

Efron Likelihood

$$L_3(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\mathrm{e}^{\boldsymbol{\beta}' \sum_{j\in\mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\prod_{j=1}^{d_i} \left(\sum_{l\in\mathcal{R}_i} \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} - \frac{j-1}{d_i} \sum_{l\in\mathcal{D}_i} \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right)}$$

Incorporating weights, the Efron likelihood becomes

$$L_3(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\mathrm{e}^{\boldsymbol{\beta}' \sum_{j\in\mathcal{D}_i} w_j \mathbf{Z}_j(t_i)}}{\left[\prod_{j=1}^{d_i} \left(\sum_{l\in\mathcal{R}_i} w_l \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} - \frac{j-1}{d_i} \sum_{l\in\mathcal{D}_i} w_l \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right) \right]^{\frac{1}{d_i} \sum_{j\in\mathcal{D}_i} w_j}}$$

Discrete Time Scale

Let $\mathcal{Q}_i$ denote the set of all subsets of d_i individuals from the risk set $\mathcal{R}_i$. For each $\mathbf{q} \in \mathcal{Q}_i$, $\mathbf{q}$ is a d_i-tuple $(q_1, q_2, \ldots, q_{d_i})$ of individuals who might have failed at t_i.

Discrete Logistic Likelihood

$$L_4(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{\mathrm{e}^{\boldsymbol{\beta}' \sum_{j\in\mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\sum_{\mathbf{q}\in\mathcal{Q}_i} \mathrm{e}^{\boldsymbol{\beta}' \sum_{l=1}^{d_j} \mathbf{Z}_{q_l}(t_i)}}$$

The computation of $L_4(\boldsymbol{\beta})$ and its derivatives is based on an adaptation of the recurrence algorithm of Gail et al. (1981) to the logarithmic scale. When there are no ties on the event times (that is, $d_i \equiv 1$), all four likelihood functions $L_1(\boldsymbol{\beta})$, $L_2(\boldsymbol{\beta})$, $L_3(\boldsymbol{\beta})$, and $L_4(\boldsymbol{\beta})$ reduce to the same expression. In a stratified analysis, the partial likelihood is the product of the partial likelihood functions for the individual strata.

Counting Process Style of Input

In the counting process formulation, data for each subject are identified by a triple $\{N, Y, \mathbf{Z}\}$ of counting, at risk, and covariate processes. Here, $N(t)$ indicates the number of events that the subject experiences over the time interval $(0, t]$; $Y(t)$ indicates whether the subject is at risk at time t (one if at risk and zero otherwise); and $\mathbf{Z}(t)$ is a vector of explanatory variables for the subject at time t. The sample path of N is a step function with jumps of size +1 at the event times, and $N(0) = 0$. Unless $\mathbf{Z}(t)$ changes continuously with time, the data for each subject can be represented by multiple observations, each identifying a semiclosed time interval $(t1, t2]$, the values of the explanatory variables over that interval, and the event status at $t2$. The subject remains at risk during the interval $(t1, t2]$, and an event may occur at $t2$. Values of the explanatory variables for the subject remain unchanged in the interval. This style of data input was originated by Terry M. Therneau (1994).

For example, a patient has a tumor recurrence at weeks 3, 10, and 15 and is followed to week 23. The explanatory variables are Trt (treatment), Z1 (initial tumor number), and Z2 (initial tumor size), and, for this patient, the values of Trt, Z1, and Z2 are (1,1,3). The data for this patient are represented by the following four observations:

T1	T2	Event	Trt	Z1	Z2
0	3	1	1	1	3
3	10	1	1	1	3
10	15	1	1	1	3
15	23	0	1	1	3

Here (T1,T2] contains the at-risk intervals. The variable Event is a censoring variable indicating whether a recurrence has occurred at T2; a value of 1 indicates a tumor recurrence, and a value of 0 indicates nonrecurrence. The PHREG procedure fits the multiplicative hazards model, which is specified as follows:

```
proc phreg;
   model (T1,T2) * Event(0) = Trt Z1 Z2;
run;
```

Another useful application of the counting process formulation is delayed entry of subjects into the risk set. For example, in studying the mortality of workers exposed to a carcinogen, the survival time is chosen to be the worker's age at death by malignant neoplasm. Any worker joining the workplace at a later age than a given event failure time is not included in the corresponding risk set. The variables of a worker consist of Entry (age at which the worker entered the workplace), Age (age at death or age censored), Status (an indicator of whether the observation time is censored, with the value 0 identifying a censored time), and X1 and X2 (explanatory variables thought to be related to survival). The specification for such an application is as follows.

```
proc phreg;
   model (Entry, Age) * Status(0) = X1 X2;
run;
```

Alternatively, you can use a time-dependent variable to control the risk set, as illustrated in the following specification:

```
proc phreg;
   model Age * Status(0) = X1 X2;
   if Age < Entry then X1= .;
run;
```

Here, X1 becomes a time-dependent variable. At a given death time t, the value of X1 is reevaluated for each subject with Age $\geq t$; subjects with Entry $> t$ are given a missing value in X1 and are subsequently removed from the risk set. Computationally, this approach is not as efficient as the one that uses the counting process formulation.

The Multiplicative Hazards Model

Consider a set of n subjects such that the counting process $N_i \equiv \{N_i(t), t \geq 0\}$ for the ith subject represents the number of observed events experienced over time t. The sample paths of the process N_i are step functions with jumps of size $+1$, with $N_i(0) = 0$. Let $\boldsymbol{\beta}$ denote the vector of unknown regression coefficients. The multiplicative hazards function $\Lambda(t, \mathbf{Z}_i(t))$ for N_i is given by

$$Y_i(t) d\Lambda(t, \mathbf{Z}_i(t)) = Y_i(t) \exp(\boldsymbol{\beta}'\mathbf{Z}_i(t)) d\Lambda_0(t)$$

where

- $Y_i(t)$ indicates whether the ith subject is at risk at time t (specifically, $Y_i(t) = 1$ if at risk and $Y_i(t) = 0$ otherwise)
- $\mathbf{Z}_i(t)$ is the vector of explanatory variables for the ith subject at time t
- $\Lambda_0(t)$ is an unspecified baseline hazard function

Refer to Fleming and Harrington (1991) and Andersen et al. (1992). The Cox model is a special case of this multiplicative hazards model, where $Y_i(t) = 1$ until the first event or censoring, and $Y_i(t) = 0$ thereafter.

The partial likelihood for n independent triplets $(N_i, Y_i, \mathbf{Z}_i), i = 1, \ldots, n$, has the form

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^{n} \prod_{t \geq 0} \left\{ \frac{Y_i(t) \exp(\boldsymbol{\beta}'\mathbf{Z}_i(t))}{\sum_{j=1}^{n} Y_j(t) \exp(\boldsymbol{\beta}'\mathbf{Z}_j(t))} \right\}^{\Delta N_i(t)}$$

where $\Delta N_i(t) = 1$ if $N_i(t) - N_i(t-) = 1$, and $\Delta N_i(t) = 0$ otherwise.

Proportional Rates/Means Models for Recurrent Events

Let $N(t)$ be the number of events experienced by a subject over the time interval $(0, t]$. Let $dN(t)$ be the increment of the counting process N over $[t, t + dt)$. The rate function is given by

$$d\mu_{\mathbf{Z}}(t) = E[dN(t)|Z(t)] = \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}(t)} d\mu_0(t)$$

where $\mu_0(.)$ is an unknown continuous function. If the $\mathbf{Z}$ are time-independent, the rate model is reduced to the mean model

$$\mu_Z(t) = \mathrm{e}^{\boldsymbol{\beta}'Z} \mu_0(t)$$

The partial likelihood for n independent triplets $(N_i, Y_i, \mathbf{Z}_i), i = 1, \ldots, n$, of counting, at-risk, and covariate processes is the same as that of the multiplicative hazards model. However, a robust sandwich estimate is used for the covariance matrix of the parameter estimator instead of the model-based estimate.

Let T_{ki} be the kth event time of the ith subject. Let C_i be the censoring time of the ith subject. The at-risk indicator and the failure indicator are, respectively,

$$Y_i(t) = I(C_i \geq t) \quad \text{and} \quad \Delta_{ki} = I(T_{ki} \leq C_i)$$

Denote

$$S^{(i)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} Y_i(t) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_i(t)} \quad \text{and} \quad \bar{\mathbf{Z}}(\boldsymbol{\beta}, t) = \frac{\sum_{i=1}^{n} Y_i(t) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_i(t)} \mathbf{Z}_i(t)}{S^{(0)}(\boldsymbol{\beta}, t)}$$

Let $\hat{\boldsymbol{\beta}}$ be the maximum likelihood estimate of $\boldsymbol{\beta}$ and let $\mathcal{I}(\hat{\boldsymbol{\beta}})$ be the observed information matrix. The robust sandwich covariance matrix estimate is given by

$$\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{i=1}^{n} \left[W_i(\hat{\boldsymbol{\beta}}) W_i'(\hat{\boldsymbol{\beta}}) \right] \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})$$

where

$$\begin{aligned} W_i(\boldsymbol{\beta}) \quad &= \quad \sum_{k} \Delta_{ki} \left\{ Z_i(T_{ki}) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, T_{ki}) \right\} - \\ & \qquad \sum_{i=1}^{n} \sum_{l} \frac{\Delta_{lj} Y_i(T_{lj}) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_i(T_{lj})}}{S^{0}(\boldsymbol{\beta}, T_{lj})} \left\{ Z_i(T_{lj}) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, T_{lj}) \right\} \end{aligned}$$

For a given realization of the covariates $\boldsymbol{\xi}$, the Nelson estimator is used to predict the mean function

$$\hat{\mu}_{\boldsymbol{\xi}}(t) = \mathrm{e}^{\hat{\boldsymbol{\beta}}' \boldsymbol{\xi}} \sum_{i=1}^{n} \sum_{k} \frac{I(T_{ki} \leq t) \Delta_{ki}}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})}$$

with standard error estimate given by

$$\hat{\sigma}^2(\hat{\mu}_{\boldsymbol{\xi}}(t)) = \sum_{i=1}^{n} \left(\frac{1}{n} \widehat{\Psi}_i(t, \boldsymbol{\xi}) \right)^2$$

where

$$\begin{aligned} \frac{1}{n} \widehat{\Psi}_i(\boldsymbol{\xi}, t) \quad &= \quad \mathrm{e}^{\hat{\boldsymbol{\beta}}' \boldsymbol{\xi}} \Bigg\{ \sum_{k} \frac{I(T_{ki} \leq t) \Delta_{ik}}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})} - \sum_{j=1}^{n} \sum_{k} \frac{Y_i(T_{kj}) \mathrm{e}^{\hat{\boldsymbol{\beta}}' \mathbf{Z}_i(T_{kj})} I(T_{kj} \leq t) \Delta_{kj}}{[S^{(0)}(\hat{\boldsymbol{\beta}}, T_{kj})]^2} - \\ & \qquad \left[\sum_{i=1}^{n} \sum_{k} \frac{I(T_{ki} \leq t) \Delta_{ik} [\bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, T_{ki}) - \boldsymbol{\xi}]}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})} \right] \\ & \qquad \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \int_{0}^{\tau} [\mathbf{Z}_i(u) - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, u)] d\hat{M}_i(u) \Bigg\} \end{aligned}$$

Since the cumulative mean function is always nonnegative, the log transform is used to compute confidence intervals. The $100(1-\alpha)\%$ pointwise confidence limits for $\mu_{\boldsymbol{\xi}}(t)$ are

$$\hat{\mu}_{\boldsymbol{\xi}}(t)\mathrm{e}^{\pm z_{\alpha/2}\frac{\hat{\sigma}(\hat{\mu}_{\boldsymbol{\xi}}(t))}{\hat{\mu}_{\boldsymbol{\xi}}(t)}}$$

where $z_{\alpha/2}$ is the upper $100\alpha/2$ percentage point of the standard normal distribution.

Newton-Raphson Method

Let $L(\boldsymbol{\beta})$ be one of the likelihood functions described in the previous subsections. Let $l(\boldsymbol{\beta}) = \log L(\boldsymbol{\beta})$. Finding $\boldsymbol{\beta}$ such that $L(\boldsymbol{\beta})$ is maximized is equivalent to finding the solution $\widehat{\boldsymbol{\beta}}$ to the likelihood equations

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0$$

With $\widehat{\boldsymbol{\beta}}^0 = \mathbf{0}$ as the initial solution, the iterative scheme is expressed as

$$\widehat{\boldsymbol{\beta}}^{j+1} = \widehat{\boldsymbol{\beta}}^{j} - \left[\frac{\partial^2 l(\widehat{\boldsymbol{\beta}}^{j})}{\partial \boldsymbol{\beta}^2}\right]^{-1} \frac{\partial l(\widehat{\boldsymbol{\beta}}^{j})}{\partial \boldsymbol{\beta}}$$

The term after the minus sign is the Newton-Raphson step. If the likelihood function evaluated at $\widehat{\boldsymbol{\beta}}^{j+1}$ is less than that evaluated at $\widehat{\boldsymbol{\beta}}^{j}$, then $\widehat{\boldsymbol{\beta}}^{j+1}$ is recomputed using half the step size. The iterative scheme continues until convergence is obtained, that is, until $\widehat{\boldsymbol{\beta}}_{j+1}$ is sufficiently close to $\widehat{\boldsymbol{\beta}}_{j}$. Then the maximum likelihood estimate of $\boldsymbol{\beta}$ is $\widehat{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}}_{j+1}$.

The model-based variance estimate of $\widehat{\boldsymbol{\beta}}$ is obtained by inverting the information matrix $\mathcal{I}(\widehat{\boldsymbol{\beta}})$

$$\hat{\mathbf{V}}_m(\widehat{\boldsymbol{\beta}}) = \mathcal{I}^{-1}(\widehat{\boldsymbol{\beta}}) = -\left[\frac{\partial^2 l(\widehat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}^2}\right]^{-1}$$

Robust Sandwich Variance Estimate

For the ith subject, $i = 1, \ldots, n$, let X_i, w_i, and $\mathbf{Z}_i(t)$ be the observed time, weight, and the covariate vector at time t, respectively. Let Δ_i be the event indicator and let $Y_i(t) = I(X_i \geq t)$. Let

$$S^{(r)}(\boldsymbol{\beta}, t) = \sum_{j=1}^{n} w_j Y_j(t) \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_j(t)} \mathbf{Z}_j^{\bigotimes r}(t), \qquad r = 0, 1$$

Let $\bar{\mathbf{Z}}(\boldsymbol{\beta}, t) = \frac{S^{(1)}(\boldsymbol{\beta},t)}{S^{(0)}(\boldsymbol{\beta},t)}$. The score residual for the ith individual is

$$\mathbf{u}_i(\boldsymbol{\beta}) = \Delta_i \left\{ \mathbf{Z}_i(X_i) - \bar{\mathbf{Z}}(\beta, X_i) \right\} - \sum_{j=1}^{n} \Delta_j \frac{w_j Y_i(X_j) \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(X_j)}}{S^{(0)}(\boldsymbol{\beta}, X_j)} \left\{ \mathbf{Z}_i(X_j) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, X_j) \right\}$$

The robust sandwich variance estimate of $\widehat{\boldsymbol{\beta}}$ derived by Binder (1992) who incorporated weights into the analysis is

$$\widehat{\mathbf{V}}_s(\widehat{\boldsymbol{\beta}}) = \mathcal{I}^{-1}(\widehat{\boldsymbol{\beta}}) \left[\sum_{j=1}^{n} (w_j \mathbf{u}_j(\widehat{\boldsymbol{\beta}}))^{\bigotimes 2} \right] \mathcal{I}^{-1}(\widehat{\boldsymbol{\beta}})$$

where $\mathcal{I}(\widehat{\boldsymbol{\beta}})$ is the observed information matrix, and $\mathbf{a}^{\bigotimes 2} = \mathbf{a}\mathbf{a}'$. Note that when $w_i \equiv 1$,

$$\widehat{\mathbf{V}}_s(\widehat{\boldsymbol{\beta}}) = \mathbf{D}'\mathbf{D}$$

where $\mathbf{D}$ is the matrix of DFBETAS residuals. This robust variance estimate was proposed by Lin and Wei (1989) and Reid and Crèpeau (1985).

Testing the Global Null Hypothesis

The following three likelihood statistics can be used to test the global null hypothesis $H_0{:}\, \boldsymbol{\beta} = \mathbf{0}$. Under mild assumptions, each statistic has an asymptotic chi-square distribution with *p* degrees of freedom given the null hypothesis. The value *p* is the dimension of $\boldsymbol{\beta}$.

Likelihood Ratio Test

$$\chi^2_{LR} = 2 \left[l(\widehat{\boldsymbol{\beta}}) - l(\mathbf{0}) \right]$$

This formulation of the likelihood ratio test is not appropriate for the COVS option.

Wald's Test

$$\chi^2_W = \widehat{\boldsymbol{\beta}}' \left[\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) \right]^{-1} \widehat{\boldsymbol{\beta}}$$

where $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) = \widehat{\mathbf{V}}_m(\widehat{\boldsymbol{\beta}})$ for the model-based variance estimate and $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) = \widehat{\mathbf{V}}_s(\widehat{\boldsymbol{\beta}})$ for the robust sandwich variance estimate.

Score Test

$$\chi^2_S = \left[\frac{\partial l(\mathbf{0})}{\partial \boldsymbol{\beta}} \right]' \mathcal{I}^{-1}(\mathbf{0}) \left[\frac{\partial l(\mathbf{0})}{\partial \boldsymbol{\beta}} \right]$$

where $\mathcal{I}(\mathbf{0}) = -\frac{\partial^2 l(\mathbf{0})}{\partial \boldsymbol{\beta}^2}$. Replacing $\mathcal{I}^{-1}(\mathbf{0})$ by the robust sandwich variance estimate $\widehat{\mathbf{V}}_s(\mathbf{0})$, the modified score test is also printed when the COVS option is specified.

Hazards Ratio Estimates and Confidence Limits

Let β_i and $\widehat{\beta}_i$ denote the *i*th component of $\boldsymbol{\beta}$ and $\widehat{\boldsymbol{\beta}}$, respectively. The hazards ratio (also known as risk ratio) for the explanatory variable with regression coefficient β_i is defined as $\exp(\beta_i)$. The hazards ratio is estimated by $\exp(\widehat{\beta}_i)$. The $100(1-\alpha)\%$ confidence limits for the hazards ratio are calculated as

$$\exp\left(\widehat{\beta}_i \pm z_{\alpha/2}\sqrt{\widehat{\mathbf{V}}_{ii}(\widehat{\boldsymbol{\beta}})}\right)$$

where $\widehat{\mathbf{V}}_{ii}(\widehat{\boldsymbol{\beta}})$ is the *i*th diagonal element of the estimated covariance matrix $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$, and $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile point of the standard normal distribution.

The hazards ratio is the ratio of the hazards functions that correspond to a change of one unit of the given variable and conditional on fixed values of all other variables.

Testing Linear Hypotheses about Regression Coefficients

Linear hypotheses for $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\mathbf{c}$ is a vector of constants. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = \left(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{c}\right)' \left[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}'\right]^{-1} \left(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{c}\right)$$

where $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$ is the estimated covariance matrix. Under H_0, χ^2_W has an asymptotic chi-square distribution with *r* degrees of freedom, where *r* is the rank of $\mathbf{L}$.

Analysis of Multivariate Failure Time Data

Multivariate failure time data arise when each study subject can potentially experience several events (for instance, multiple infections after surgery) or when there exists some natural or artificial clustering of subjects (for instance, a litter of mice) that induces dependence among the failure times of the same cluster. Data in the former situation are referred to as multiple events data, and data in the latter situation are referred to as clustered data. The multiple events data can be further classified into ordered and unordered data. For ordered data, there is a natural ordering of the multiple failures within a subject, which includes recurrent events data as a special case. For unordered data, the multiple event times result from several concurrent failure processes.

Multiple events data can be analyzed by the Wei, Lin, and Weissfeld (1989), aka WLW, method based on the marginal Cox models. For the special case of recurrent events data, you can fit the intensity model (Andersen and Gill 1982), the proportional rates/means model (Pepe and Cai 1993; Lawless and Nadeau 1995; Lin, Wei, Yang, and Ying 2000), or the stratified models for total time and gap time proposed by Prentice, Williams, and Peterson (1981), aka PWP. For clustered data, you can carry out the analysis of Lee, Wei, and Amato (1992) based on the marginal Cox model. To use PROC PHREG to perform these analyses correctly and effectively, you have to array your data in a specific way to produce the correct risk sets.

All examples described in this section can be found in the program *phrmult.sas* in the SAS/STAT sample library. Furthermore, the "Examples" section in this chapter contains two examples to illustrate the methods for analyzing recurrent events data and clustered data.

Marginal Cox Models for Multiple Events Data

Suppose there are n subjects and each subject can experience up to K potential events. Let $\mathbf{Z}_{ki}(.)$ be the covariate process associated with the kth event for the ith subject. The marginal Cox models are given by

$$\lambda_k(t;\mathbf{Z}_{ki}) = \lambda_{k0}e^{\boldsymbol{\beta}_k'\mathbf{Z}_{ki}(t)}, \qquad k=1,\ldots,K; i=1,\ldots,n$$

where $\lambda_{k0}(t)$ is the (event-specific) baseline hazard function for the kth event and $\boldsymbol{\beta}_k$ is the (event-specific) column vector of regression coefficients for the kth event. WLW estimates $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_K$ by the maximum partial likelihood estimates $\hat{\boldsymbol{\beta}}_1,\ldots,\hat{\boldsymbol{\beta}}_K$, respectively, and uses a robust sandwich covariance matrix estimate for $(\hat{\boldsymbol{\beta}}_1',\ldots,\hat{\boldsymbol{\beta}}_K')'$ to account for the dependence of the multiple failure times.

By using a properly prepared input data set, you can estimate the regression parameters for all the marginal Cox models and compute the robust sandwich covariance estimates in one PROC PHREG invocation. For convenience of discussion, suppose each subject can potentially experience K=3 events and there are two explanatory variables Z1 and Z2. The event-specific parameters to be estimated are $\boldsymbol{\beta}_1 = (\beta_{11},\beta_{21})'$ for the first marginal model, $\boldsymbol{\beta}_2 = (\beta_{12},\beta_{22})'$ for the second marginal model, and $\boldsymbol{\beta}_3 = (\beta_{13},\beta_{23})'$ for the third marginal model. Inference of these parameters is based on the robust sandwich covariance matrix estimate of the parameter estimators. It is necessary that each row of the input data set represents the data for a potential event of a subject. The input data set should contain

- an ID variable for identifying the subject so that all observations of the same subject have the same ID value
- an Enum variable to index the multiple events. For example, Enum=1 for the first event, Enum=2 for the second event, and so on.
- a Time variable to represent the observed time from some time origin for the event. For recurrence events data, it is the time from the study entry to each recurrence.
- a Status variable to indicate whether the Time value is a censored or uncensored time. For example, Status=1 indicates an uncensored time and Status=0 indicates a censored time.
- independent variables (Z1 and Z2).

The WLW analysis can be carried out by specifying

```
proc phreg covs(aggregate);
   model Time*Status(0)=Z11 Z12 Z13 Z21 Z22 Z23;
   strata Enum;
   id ID;
   Z11= Z1 * (Enum=1);
   Z12= Z1 * (Enum=2);
   Z13= Z1 * (Enum=3);
   Z21= Z2 * (Enum=1);
   Z22= Z2 * (Enum=2);
   Z23= Z2 * (Enum=3);
   run;
```

Variable Enum is specified in the STRATA statement so that there is one marginal Cox model for each distinct value of Enum. Variables Z11, Z12, Z13, Z21, Z22, and Z23 in the MODEL statement are event-specific variables derived from the independent variables Z1 and Z2 by the given programming statements. In particular, variables Z11, Z12, and Z13 are the event-specific variables for the explanatory variable Z1; variables Z21, Z22, and Z23 are event-specific variables for the explanatory variable Z2. For $j = 1, 2$, and $k = 1, 2, 3$, variable Zjk contains the same values as the explanatory variable Zj for the rows that correspond to kth marginal model and the value 0 for all other rows; as such, β_{jk} is the regression coefficient for Zjk. You can avoid using the programming statements in PROC PHREG by creating these event-specific variables in the input data set using the same programming statements in a DATA step.

The option COVS(AGGREGATE) is specified in the PROC statement to obtain the robust sandwich estimate of the covariance matrix, and the score residuals used in computing the middle part of the sandwich estimate are aggregated over identical ID values. You can also include TEST statements in the PROC PHREG code to test various linear hypotheses of the regression parameters based on the robust sandwich covariance matrix estimate.

Consider the AIDS study data in Wei, Lin, and Weissfeld (1989) from a randomized clinical trial to assess the antiretrovial capacity of ribavirin over times in AIDS patients. Blood sample were collected at weeks 4, 8, and 12 from each patients in three treatment groups (placebo, low dose of ribavirin, and high dose). For each serum sample, the failure time is the number of days before virus positivity was detected. If the sample was contaminated or it took a longer period of time than was achievable in the laboratory, the sample was censored. For example,

- Patient #1 in the placebo group has uncensored times 9, 6, and 7 days (that is, it took 9 days to detect viral positivity in the first blood sample, 6 days for the second blood sample, and 7 days for the third blood sample).
- Patient #14 in the low dose group of rabavirin has uncensored times of 16 and 17 days for the first and second sample, respectively, and a censored time of 21 days for the third blood sample.

- Patient #28 in the High dose group has an uncensored time of 21 days for the first sample, no measurement for the second blood sample, and a censored time of 25 days for the third sample.

For a full-rank parameterization, two design variables are sufficient to represent three treatment groups. Based on the reference coding with placebo as the reference, the values of the two dummy explanatory variables Z1 and Z2 representing the treatments are

Treatment Group	Z1	Z2
Placebo	0	0
Low dose ribavirin	1	0
High dose ribavirin	0	1

The bulk of the task in using PROC PHREG to perform the WLW analysis lies in the preparation of the input data set. As discussed earlier, the input data set should contain the ID, Enum, Time, and Status variables, and event-specific independent variables Z11, Z12, Z13, Z21, Z22, and Z23. Data for the three patients described earlier are arrayed as follows:

ID	Time	Status	Enum	Z1	Z2
1	9	1	1	0	0
1	6	1	2	0	0
1	7	1	3	0	0
14	16	1	1	1	0
14	17	1	2	1	0
14	21	0	3	1	0
28	21	1	1	0	1
28	25	0	3	0	1

The first three rows are data for Patient #1 with event times at 9, 6, and 7 days, one row for each event. The next three rows are data for Patient #14, who has an uncensored time of 16 days for the first serum sample, an uncensored time of 17 days for the second sample, and a censored time of 21 days for the third sample. The last two rows are data for Patient #28 of the high dose group (Z1=0 and Z2=1). Since the patient did not have a second serum sample, there are only two rows of data.

To perform the WLW analysis, you specify

```
proc phreg covs(aggregate);
   model Time*Status(0)=Z11 Z12 Z13 Z21 Z22 Z23;
   strata Enum;
   id ID;
   Z11= Z1 * (Enum=1);
   Z12= Z1 * (Enum=2);
   Z13= Z1 * (Enum=3);
   Z21= Z2 * (Enum=1);
   Z22= Z2 * (Enum=2);
   Z23= Z2 * (Enum=3);
   EqualLowDose: test Z11=Z12, Z12=Z23;
   AverageLow: test Z11,Z12,Z13 / e average;
run;
```

Two linear hypotheses are tested using the TEST statements. The specification

```
      EqualLowDose: test Z11=Z12, Z12=Z13;
```

tests the null hypothesis $\beta_{11} = \beta_{12} = \beta_{13}$ of identical low-dose effects across three marginal models. The specification

```
    AverageLow: test Z11,Z12,Z13 / e average;
```

tests the null hypothesis of no low-dose effects (that is, $\beta_{11} = \beta_{12} = \beta_{13} = 0$). The AVERAGE option computes the optimal weights for estimating the average low-dose effect $\beta_1^* = \beta_{11} = \beta_{12} = \beta_{13}$ and performs a 1 DF test for testing the null hypothesis that $\beta_1^* = 0$. The E option displays the coefficients for the linear hypotheses, including the optimal weights.

Marginal Cox Models for Clustered Data

Suppose there are n clusters with K_i members in the ith cluster, $i = 1, \ldots, n$. Let $\mathbf{Z}_{ki}(.)$ be the covariate process associated with the kth member of the ith cluster. The marginal Cox model is given by

$$\lambda(t; \mathbf{Z}_{ki}) = \lambda_0(t) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_{ki}(t)} \qquad k = 1, \ldots, K_i; i = 1, \ldots, n$$

where $\lambda_0(t)$ is an arbitrary baseline hazard function and $\boldsymbol{\beta}$ is the vector of regression coefficients. Lee, Wei, and Amato (1992) estimate $\boldsymbol{\beta}$ by the maximum partial likelihood estimate $\hat{\boldsymbol{\beta}}$ under the independent working assumption, and use a robust sandwich covariance estimate to account for the intracluster dependence.

To use PROC PHREG to analyze the clustered data, each member of a cluster is represented by an observation in the input data set. The input data set to PROC PHREG should contain

- an ID variable to identify the cluster so that members of the same cluster have the same ID value

- a Time variable to represent the observed survival time of a member of a cluster
- a Status variable to indicate whether the Time value is an uncensored or censored time. For example, Status=1 indicates an uncensored time and Status=0 indicates a censored time.
- the explanatory variables thought to be related to the failure time

Consider a tumor study in which one of three female rats of the same litter was randomly subjected to a drug treatment. The failure time is the time from randomization to the detection of tumor. If a rat died before the tumor was detected, the failure time was censored. For example,

- In litter #1, the drug-treated rat has an uncensored time of 101 days, one untreated rat has a censored time of 49 days, and the other untreated rat has a failure time of 104 days.
- In litter #3, the drug-treated rat has a censored time of 104 days, one untreated rat has a censored time of 102 days, and the other untreated rat has a censored time of 104 days.

In this example, a litter is a cluster and the rats of the same litter are members of the cluster. Let Trt be a 0-1 variable representing the treatment a rat received, with value 1 for drug treatment and 0 otherwise. Data for the two litters of rats described earlier contribute six observations to the input data set:

Litter	Time	Status	Trt
1	101	0	1
1	49	1	0
1	104	0	0
3	104	0	1
3	102	0	0
3	104	0	0

The analysis of Lee, Wei, and Amato (1992) can be performed by PROC PHREG as follows:

```
proc phreg covs(aggregate);
   model Time*Status(0)=Treatment;
   id Litter;
run;
```

Intensity and Rate/Mean Models for Recurrent Events Data

Suppose each subject experiences recurrences of the same phenomenon. Let $N(t)$ be the number of events a subject experiences over the interval [0,t] and let $\mathbf{Z}(.)$ be the covariate process of the subject.

The intensity model (Andersen and Gill 1982) is given by

$$\lambda_{\mathbf{Z}}(t)dt = E\{dN(t)|\mathcal{F}_{t-}\} = \lambda_0(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}(t)}dt$$

where $\mathcal{F}_t$ represents all the information of the processes N and $\mathbf{Z}$ up to time t, $\lambda_0(t)$ is an arbitrary baseline intensity function, and $\boldsymbol{\beta}$ is the vector of regression coefficients. This model consists of two components: 1) all the influence of the prior events on future recurrences, if there is any, is mediated through the time-dependent covariates, and 2) the covariates have multiplicative effects on the instantaneous rate of the counting process. If the covariates are time invariant, the risk of recurrences is unaffected by the past events.

The proportional rates and means models (Pepe and Cai 1993; Lawless and Nadeau 1995; Lin, Wei, Yang, and Ying 2000) assume that the covariates have multiplicative effects on the mean and rate functions of the counting process. The rate function is given by

$$d\mu_{\mathbf{Z}}(t) = E\{dN(t)|\mathbf{Z}(t)\} = \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}(t)}d\mu_0(t)$$

where $\mu_0(t)$ is an unknown continuous function and $\boldsymbol{\beta}$ is the vector of regression parameters. If $\mathbf{Z}$ is time invariant, the mean function is given by

$$\mu_{\mathbf{Z}}(t) = E\{N(t)|\mathbf{Z}\} = \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}}\mu_0(t)$$

For both the intensity and the proportional rates/means models, estimates of the regression coefficients are obtained by solving the the partial likelihood score function. However, the covariance matrix estimate for the intensity model is computed as the inverse of the observed information matrix, while that for the proportional rates/means model is given by a sandwich estimate. For a given pattern of fixed covariates, the Nelson estimate for the cumulative intensity function is the same for the cumulative mean function, but their standard errors are not the same.

To fit the intensity or rate/mean model using PROC PHREG, the counting process style of input is needed. A subject with K events contributes K+1 observations to the input data set. The kth observation of the subject identifies the time interval from the $(k-1)$th event or time 0 (if $k=1$) to the kth event, $k=1,\ldots,K$. The $(K+1)$th observation represents the time interval from the Kth event to time of censorship. The input data set should contain

- a TStart variable to represent the $(k-1)$th recurrence time or the value 0 if $k=1$
- a TStop variable to represent the kth recurrence time or the follow-up time if $k=K+1$
- a Status variable indicating whether the TStop time is a recurrence time or a censored time; for example, Status=1 for a recurrence time and Status=0 for censored time
- explanatory variables thought to be related to the recurrence times

If the rate/mean model is used, the input data should also contain an ID variable for identifying the subjects.

Consider the Chronic Granulomatous Disease (CGD) data listed in Fleming and Harrington (1991). The disease is a rare disorder characterized by recurrent pyrogenic infections. The study is a placebo-controlled randomized clinical trial conducted by the International CGD Cooperative Study to assess the effect of gamma interferon to reduce the rate of infection. For each study patient the times of recurrent infections along with a number of prognostic factors were collected. For example,

- Patient #17404, age 38, in the gamma interferon group had a follow-up time of 293 without any infection.
- Patient #204001, age 12, in the placebo group had an infection at 219 days, a recurrent infection at 373 days, and was followed up to 414 days.

Let Trt be the variable representing the treatment status with value 1 for gamma interferon and value 2 for placebo. Let Age be a covariate representing the age of the CGD patient. Data for the two CGD patients described earlier are given in the following table.

ID	TStart	TStop	Status	Trt	Age
174054	0	293	0	1	38
204001	0	219	1	2	12
204001	219	373	1	2	12
204001	373	414	0	2	12

Since Patient #174054 had no infection through the end of the follow-up period (293 days), there is only one observation representing the period from time 0 to the end of the follow-up. Data for Patient #204001 are broken into three observations, since there are two infections. The first observation represents the period from time 0 to the first infection, the second observation represents the period from the first infection to the second infection, and the third time period represents the period from the second infection to the end of the follow-up.

The following specification fits the intensity model.

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt Age;
   run;
```

You can predict the cumulative intensity function for the a given pattern of fixed covariates by specifying the CUMHAZ= option in the BASELINE statement. Suppose you are interested in two fixed patterns, one for patients of age 30 in the gamma interferon group and the other for patients of age 1 in the placebo group. You first create the SAS data set as follows:

```
data Pattern;
   Trt=1; Age=30;
   output;
   Trt=2; Age=1;
   output;
   run;
```

You then include the following BASELINE statement in the PROC PHREG specification. The CUMHAZ=_all_ option produces the cumulative hazard function estimates, the standard error estimates, and the lower and upper pointwise confidence limits.

```
   baseline data=pattern out=out1 cumhaz=_all_ /nomean;
```

The following specification of PROC PHREG fits the mean model and predicts the cumulative mean function for the two patterns of covariates in the Pattern data set.

```
proc phreg covs(aggregate);
   model (Tstart,Tstop)*Status(0)=Trt Age;
   baseline data=Pattern out=out2 cmf=_all_ /nomean;
   id ID;
```

The COV(AGGREGATE) option computes the robust sandwich covariance matrix estimate. The CMF=_ALL_ option adds the cumulative mean funtion estimates, the standard error estimates, and the lower and upper pointwise confidence limits to the OUT=Out2 data set.

PWP Models for Recurrent Events Data

Let $N(t)$ be the number of events a subject experiences by time t. Let $\mathbf{Z}(t)$ be the covariate vectors of the subject at time t. For a subject who has K events before censorship takes place, let $t_0 = 0$, let t_k be the kth recurrence time, $k = 1, \ldots, K$, and let t_{K+1} be the censored time. Prentice, Williams, and Peterson (1981) consider two time scales, a total time from the beginning of the study and a gap time from immediately preceding failure. The PWP models are stratified Cox-type models that allow the shape of the hazard function to depend on the number of preceding events and possibly on other characteristics of $\{N(t)$ and $\mathbf{Z}(t)\}$. The total time and gap time models are given, respectively, as follows:

$$\begin{aligned}
\lambda(t|\mathcal{F}_{t-}) &= \lambda_{0k}(t)\mathrm{e}^{\boldsymbol{\beta}_k'\mathbf{Z}(t)}, \qquad t_{k-1} < t \le t_k \\
\lambda(t|\mathcal{F}_{t-}) &= \lambda_{0k}(t - t_{k-1})\mathrm{e}^{\boldsymbol{\beta}_k'\mathbf{Z}(t)}, \qquad t_{k-1} < t \le t_k
\end{aligned}$$

where λ_{0k} is an arbitrary baseline intensity functions, and $\boldsymbol{\beta}_k$ is a vector of stratum-specific regression coefficients. Here, a subject moves to the kth stratum immediately after his $(k-1)$th recurrence time and remains there until the kth recurrence occurs or until censorship takes place. For instance, a subject who experiences only one event moves from the first stratum to the second stratum after the event occurs and remains in the second stratum until the end of the follow-up.

You can use PROC PHREG to carry out the analyses of the PWP models, but you have to prepare the input data set to provide the correct risk sets. The input data set for analyzing the total time is the same as the AG model with an additional variable to represent the stratum that the subject is in. A subject with K events contributes K+1 observations to the input data set, one for each stratum that the subject moves to. The input data should contain

- a TStart variable to represent the $(k-1)$th recurrence time or the value 0 if $k = 1$
- a TStop variable to represent the kth recurrence time or the time of censorship if $k = K + 1$
- a Status variable with value 1 if the Time value is a recurrence time and value 0 if the Time value is a censored time
- an Enum variable representing the index of the stratum that the subject is in. For a subject who has only one event at t_1 and is followed to time t_c, Enum=1 for the first observation (where Time=t_1 and Status=1) and Enum=2 for the second observation (where Time=t_c and Status=0).
- explanatory variables thought to be related to the recurrence times

To analyze gap times, the input data set should also include a GapTime variable which is equal to (TStop $-$ TStart).

Consider the data of two subjects in CGD data described in the previous section.

- Patients #174054, age 38, in the gamma interferon group had a follow-up time of 293 without any infection.
- Patient #204001, age 12, in the placebo group had an infection at 219 days, a recurrent infection at 373 days, and a follow-up time of 414 days.

To illustrate, suppose all subjects have at most two observed events. The data for the two subjects in the input data set are as follows:

ID	TStart	TStop	Gaptime	Status	Enum	Trt	Age
174054	0	293	293	0	1	1	38
204001	0	219	219	1	1	2	12
204001	219	373	154	1	2	2	12
204001	373	414	41	0	3	2	12

Subject #174054 contributes only one observation to the input data, since there is no observed event. Subject #204001 contributes three observations, since there are two observed events.

To fit the total time model of PWP with stratum-specific slopes, you can either create the stratum-specific explanatory variables (Trt1, Trt2, and Trt3 for Trt, and Age1, Age2, and Age3 for Age) in a DATA step, or you can specify them in PROC PHREG using programming statements as follows:

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt1 Trt2 Trt3 Age1 Age2 Age3;
   strata Enum;
   Trt1= Trt * (Enum=1);
   Trt2= Trt * (Enum=2);
   Trt3= Trt * (Enum=3);
   Age1= Age * (Enum=1);
   Age2= Age * (Enum=2);
   Age3= Age * (Enum=3);
   run;
```

To fit the total time model of PWP with the common regression coefficients, you specify

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt Age;
   strata Enum;
   run;
```

To fit the gap time model of PWP with stratum-specific regression coefficients, you specify

```
proc phreg;
   model Gaptime*Status(0)=Trt1 Trt2 Trt3 Age1 Age2 Age3;
   strata Enum;
   Trt1= Trt * (Enum=1);
   Trt2= Trt * (Enum=2);
   Trt3= Trt * (Enum=3);
   Age1= Age * (Enum=1);
   Age2= Age * (Enum=2);
   Age3= Age * (Enum=3);
   run;
```

To fit the gap time model of PWP with common regression coefficients, you specify

```
proc phreg;
   model Gaptime*Status(0)=Trt Age;
   strata Enum;
   run;
```

Residuals

The cumulative baseline hazard function Λ_0 is estimated by

$$\hat{\Lambda}_0(t) = \sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{\sum_{j=1}^{n} Y_j(s) \exp(\widehat{\boldsymbol{\beta}}' \mathbf{Z}_j(s))}$$

Although this formula is for the BRESLOW=TIES option, the same formula is used for other TIES= options. The discrepancies between results obtained by using an appropriate formula for a nondefault TIES= option and those obtained by the given formula are minimal.

The martingale residual at t is defined as

$$\hat{M}_i(t) = N_i(t) - \int_0^t Y_i(s)\exp(\widehat{\boldsymbol{\beta}}'\mathbf{Z}_i(s))d\hat{\Lambda}_0(s)$$

Here $\hat{M}_i(t)$ estimates the difference over $(0,t]$ between the observed number of events for the ith subject and a conditional expected number of events. The quantity $\hat{M}_i \equiv \hat{M}_i(\infty)$ is referred to as the martingale residual for the ith subject. When the counting process MODEL specification is used, the RESMART= variable contains the component $(\hat{M}_i(t_2) - \hat{M}_i(t_1))$ instead of the martingale residual at t_2. The martingale residual for a subject can be obtained by summing up these component residuals within the subject. For the Cox model with no time-dependent explanatory variables, the martingale residual for the ith subject with observation time t_i and event status δ_i, where

$$\delta_i = \begin{cases} 0 & \text{if } t_i \text{ is a censored time} \\ 1 & \text{if } t_i \text{ is an event time} \end{cases}$$

is

$$\hat{M}_i = \delta_i - \hat{\Lambda}_0(t_i)\exp(\widehat{\boldsymbol{\beta}}'\mathbf{z}_i)$$

The deviance residuals d_i are a transform of the martingale residuals:

$$d_i = sign(\hat{M}_i)\sqrt{2\left[-\hat{M}_i - N_i(\infty)\log\left(\frac{N_i(\infty)-\hat{M}_i}{N_i(\infty)}\right)\right]}$$

The square root shrinks large negative martingale residuals, while the logarithmic transformation expands martingale residuals that are close to unity. As such, the deviance residuals are more symmetrically distributed about zero than the martingale residuals. For the Cox model, the deviance residual reduces to the form

$$d_i = sign(\hat{M}_i)\sqrt{2[-\hat{M}_i - \delta_i\log(\delta_i - \hat{M}_i)]}$$

When the counting process MODEL specification is used, values of the RESDEV= variable are set to missing because the deviance residuals can be calculated on a per subject basis only.

The Schoenfeld (1982) residual vector is calculated on a per event time basis as

$$\mathbf{U}_i(t) = \mathbf{Z}_i(t) - \bar{\mathbf{Z}}(t)$$

where t is an event time, and $\bar{\mathbf{Z}}(t)$ is a weighted average of the covariates over the risk set at time t and is given by

$$\bar{\mathbf{Z}}(t) = \frac{\sum_{l=1}^{n} Y_l(t)\mathbf{Z}_l(t)\exp(\widehat{\boldsymbol{\beta}}'\mathbf{Z}_l(t))}{\sum_{l=1}^{n} Y_l(t)\exp(\widehat{\boldsymbol{\beta}}'\mathbf{Z}_l(t))}$$

Under the proportional hazards assumption, the Schoenfeld residuals have the sample path of a random walk; therefore, they are useful in assessing time trend or lack of proportionality. Harrell (1986) proposed a z-transform of the Pearson correlation between these residuals and the rank order of the failure time as a test statistic for nonproportional hazards. Therneau, Grambsch, and Fleming (1990) considered a Kolmogorov-type test using the cumulative sum of the residuals.

The score process for the ith subject at time t is

$$\mathbf{L}_i(t) = \int_0^t [\mathbf{Z}_i(s) - \bar{\mathbf{Z}}(s)]d\hat{M}_i(s)$$

The vector $\mathbf{L}_i \equiv \mathbf{L}_i(\infty)$ is the score residual for the ith subject. When the counting process MODEL specification is used, the RESSCO= variables contain the components of $(\mathbf{L}_i(t2) - \mathbf{L}_i(t1))$ instead of the score process at $t2$. The score residual for a subject can be obtained by summing up these component residuals within the subject.

The score residuals are a decomposition of the first partial derivative of the log likelihood. They are useful in assessing the influence of each subject on individual parameter estimates. They also play an important role in the computation of the robust sandwich variance estimators of Lin and Wei (1989) and Wei, Lin, and Weissfeld (1989).

Diagnostics Based on Weighted Residuals

The vector of weighted Schoenfeld residuals, $\mathbf{r}_i$, is computed as

$$\mathbf{r}_i = n_e\widehat{\mathbf{V}}\mathbf{U}_i(t_i)$$

where n_e is the total number of events, $\widehat{\mathbf{V}} = \widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$, and $\mathbf{U}_i(t_i)$ is the vector of Schoenfeld residuals at the event time t_i. The components of $\mathbf{r}_i$ are output to the WTRESSCH= variables.

The weighted Schoenfeld residuals are useful in assessing the proportional hazards assumption. The idea is that most of the common alternatives to the proportional hazards can be cast in terms of a time-varying coefficient model

$$\lambda(t, \mathbf{Z}) = \lambda_0(t)\exp(\beta_1(t)Z_1 + \beta_2(t)Z_2 + \ldots)$$

where $\lambda(t, \mathbf{Z})$ and $\lambda_0(t)$ are hazards rates. Let $\hat{\beta}_j$ and r_{ij} be the jth component of $\hat{\boldsymbol{\beta}}$ and $\mathbf{r}_i$, respectively. Grambsch and Therneau (1994) suggest using a smoothed plot of $(\hat{\beta}_j + r_{ij})$ versus t_i to discover the functional form of the time-varying coefficient $\beta_j(t)$. A zero slope indicates that the coefficient is not varying with time.

The weighted score residuals are used more often than their unscaled counterparts in assessing local influence. Let $\hat{\boldsymbol{\beta}}_{(i)}$ be the estimate of $\boldsymbol{\beta}$ when the ith subject is left out, and let $\boldsymbol{\Delta}_i = \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)}$. The jth component of $\boldsymbol{\Delta}_i$ can be used to assess any untoward effect of the ith subject on $\hat{\beta}_j$. The exact computation of $\boldsymbol{\Delta}_i$ involves refitting the model each time a subject is omitted. Cain and Lange (1984) derived the following approximation of $\boldsymbol{\Delta}_i$ as weighted score residuals:

$$\widehat{\boldsymbol{\Delta}}_i = \widehat{\mathbf{V}}\mathbf{L}_i$$

Here, $\widehat{\mathbf{V}} = \widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$, and $\mathbf{L}_i$ is the vector of the score residuals for the ith subject. Values of $\widehat{\boldsymbol{\Delta}}_i$ are output to the DFBETA= variables. Again, when the counting process MODEL specification is used, the DFBETA= variables contain the component $(\widehat{\mathbf{V}}\mathbf{L}_i(t2) - \widehat{\mathbf{V}}\mathbf{L}_i(t1))$. The vector $\widehat{\boldsymbol{\Delta}}_i$ for a subject can be obtained by summing these components within the subject.

Note that these DFBETA statistics are a transform of the score residuals. In computing the robust sandwich variance estimators of Lin and Wei (1989) and Wei, Lin, and Weissfeld (1989), it is more convenient to use the DFBETA statistics than the score residuals (see Example 54.8 on page 3304).

Influence of Observations on Overall Fit of the Model

The LD statistic approximates the likelihood displacement, which is the amount by which minus twice the log likelihood $(-2\log\mathcal{L}(\hat{\boldsymbol{\beta}}))$, under a fitted model, changes when each subject in turn is left out. When the ith subject is omitted, the likelihood displacement is

$$2\log\mathcal{L}(\hat{\boldsymbol{\beta}}) - 2\log\mathcal{L}(\hat{\boldsymbol{\beta}}_{(i)})$$

where $\hat{\boldsymbol{\beta}}_{(i)}$ is the vector of parameter estimates obtained by fitting the model without the ith subject. Instead of refitting the model without the ith subject, Pettitt and Bin Daud (1989) propose that the likelihood displacement for the ith subject be approximated by

$$LD_i = \mathbf{L}_i'\widehat{\mathbf{V}}\mathbf{L}_i$$

This approximation is output to the LD= variable.

The LMAX statistic is another global influence statistic. This statistic is based on the symmetric matrix

$$\mathbf{B} = \mathbf{L}\widehat{\mathbf{V}}\mathbf{L}'$$

where $\mathbf{L}$ is the matrix with rows that are the score residual vectors $\mathbf{L}_i$. The elements of the eigenvector associated with the largest eigenvalue of the matrix $\mathbf{B}$, standardized to unit length, give a measure of the sensitivity of the fit of the model to each observation in the data. The influence of the ith subject on the global fit of the model is proportional to the magnitude of ζ_i, where ζ_i is the ith element of the vector $\boldsymbol{\zeta}$ that satisfies

$$\mathbf{B}\boldsymbol{\zeta} = \lambda_{\max}\boldsymbol{\zeta} \quad \text{and} \quad \boldsymbol{\zeta}'\boldsymbol{\zeta} = 1$$

with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{B}$. The sign of ζ_i is irrelevant, and its absolute value is output to the LMAX= variable.

When the counting process MODEL specification is used, the LD= and LMAX= variables are set to missing, because these two global influence statistics can be calculated on a per subject basis only.

Survival Distribution Estimates for the Cox Model

Two estimators of the survivor function are available: one is the product-limit estimate and the other is based on the empirical cumulative hazard function.

Product-Limit Estimates

Let $\mathcal{C}_i$ denote the set of individuals censored in the half-open interval $[t_i, t_{i+1})$, where $t_0 = 0$ and $t_{k+1} = \infty$. Let γ_l denote the censoring times in $[t_i, t_{i+1})$; l ranges over $\mathcal{C}_i$. The likelihood function for all individuals is given by

$$\mathcal{L} = \prod_{i=0}^{k} \left\{ \prod_{l \in \mathcal{D}_i} \left([S_0(t_i)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} - [S_0(t_i+0)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right) \prod_{l \in \mathcal{C}_i} [S_0(\gamma_l+0)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right\}$$

where $\mathcal{D}_0$ is empty. The likelihood $\mathcal{L}$ is maximized by taking $S_0(t) = S_0(t_i+0)$ for $t_i < t \leq t_{i+1}$ and allowing the probability mass to fall only on the observed event times $t_1, \ldots, t_k$. By considering a discrete model with hazard contribution $1-\alpha_i$ at t_i, you take $S_0(t_i) = S_0(t_{i-1}+0) = \prod_{j=0}^{i-1} \alpha_j$, where $\alpha_0 = 1$. Substitution into the likelihood function produces

$$\mathcal{L} = \prod_{i=0}^{k} \left\{ \prod_{j \in \mathcal{D}_i} \left(1 - \alpha_i^{\exp(\mathbf{z}_j'\boldsymbol{\beta})} \right) \prod_{l \in \mathcal{R}_i - \mathcal{D}_i} \alpha_i^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right\}$$

If you replace $\boldsymbol{\beta}$ with $\widehat{\boldsymbol{\beta}}$ estimated from the partial likelihood function and then maximize with respect to α_1 , . . . ,α_k , the maximum likelihood estimate $\widehat{\alpha}_i$ of α_i becomes a solution of

$$\sum_{j \in \mathcal{D}_i} \frac{\exp(\mathbf{z}_j'\widehat{\boldsymbol{\beta}})}{1 - \widehat{\alpha}_i^{\exp(\mathbf{z}_j'\widehat{\boldsymbol{\beta}})}} = \sum_{l \in \mathcal{R}_i} \exp(\mathbf{z}_l'\widehat{\boldsymbol{\beta}})$$

When only a single failure occurs at t_i, $\widehat{\alpha}_i$ can be found explicitly. Otherwise, an iterative solution is obtained by the Newton method.

The estimated baseline cumulative hazard function is

$$\widehat{H}_0(t) = -\log(\widehat{S}_0(t))$$

where $\widehat{S}_0(t)$ is the estimated baseline survivor function given by

$$\widehat{S}_0(t) = \widehat{S}_0(t_{i-1}+0) = \prod_{j=0}^{i-1} \widehat{\alpha}_j, \qquad t_{i-1} < t \leq t_i$$

For details, refer to Kalbfleisch and Prentice (1980). For a given realization of the explanatory variables $\boldsymbol{\xi}$, the product-limit estimate of the survival function at $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\hat{S}(t,\boldsymbol{\xi}) = [\widehat{S}_0(t)]^{\exp(\boldsymbol{\beta}'\boldsymbol{\xi})}$$

Empirical Cumulative Hazards Function Estimates

Let $\boldsymbol{\xi}$ be a given realization of the explanatory variables. The empirical cumulative hazard function estimate at $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\widehat{\Lambda}(t,\boldsymbol{\xi}) = \sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{\sum_{j=1}^n Y_j(s)\exp(\widehat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))}$$

The variance estimator of $\widehat{\Lambda}(t,\boldsymbol{\xi})$ is given by the following (Tsiatis 1981):

$$\begin{aligned} & \widehat{var}\{n^{\frac{1}{2}}(\widehat{\Lambda}(t,\boldsymbol{\xi}) - \Lambda(t,\boldsymbol{\xi}))\} \\ = \quad & n\Bigg\{\sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{[\sum_{j=1}^n Y_j(s)\exp(\widehat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))]^2} + \mathbf{H}'(t,\boldsymbol{\xi})\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{H}(t,\boldsymbol{\xi})\Bigg\} \end{aligned}$$

where $\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$ and

$$\mathbf{H}(t,\boldsymbol{\xi}) = \sum_{i=1}^{n} \int_0^t \frac{\sum_{l=1}^n Y_l(s)(\mathbf{Z}_l - \boldsymbol{\xi})\exp(\widehat{\boldsymbol{\beta}}'(\mathbf{z}_l - \boldsymbol{\xi}))}{[\sum_{j=1}^n Y_j(s)\exp(\widehat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))]^2} dN_i(s)$$

The empirical cumulative hazard function (CH) estimate of the survivor function for $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\tilde{S}(t,\boldsymbol{\xi}) = \exp(-\widehat{\Lambda}(t,\boldsymbol{\xi}))$$

Confidence Intervals for the Survivor Function

Let $\hat{S}(t,\boldsymbol{\xi})$ and $\tilde{S}(t,\boldsymbol{\xi})$ correspond to the product-limit (PL) and empirical cumulative hazard function (CH) estimates of the survivor function for $\mathbf{Z}=\boldsymbol{\xi}$, respectively. Both the standard error of log($\hat{S}(t,\boldsymbol{\xi})$) and the standard error of log($\tilde{S}(t,\boldsymbol{\xi})$) are approximated by $\tilde{\sigma}_0(t,\boldsymbol{\xi})$, which is the square root of the variance estimate of $\widehat{\Lambda}(t,\boldsymbol{\xi})$; refer to Kalbfleisch and Prentice (1980, p.116). By the delta method, the standard errors of $\hat{S}(t,\boldsymbol{\xi})$ and $\tilde{S}(t,\boldsymbol{\xi})$ are given by

$$\hat{\sigma}_1(t,\boldsymbol{\xi}) \doteq \hat{S}(t,\boldsymbol{\xi})\tilde{\sigma}_0(t,\boldsymbol{\xi}) \qquad \text{and} \qquad \tilde{\sigma}_1(t,\boldsymbol{\xi}) \doteq \tilde{S}(t,\boldsymbol{\xi})\tilde{\sigma}_0(t,\boldsymbol{\xi})$$

respectively. The standard errors of log[-log($\hat{S}(t,\boldsymbol{\xi})$)] and log[-log($\tilde{S}(t,\boldsymbol{\xi})$)] are given by

$$\hat{\sigma}_2(t,\boldsymbol{\xi}) \doteq \frac{-\tilde{\sigma}_0(t,\boldsymbol{\xi})}{\log(\hat{S}(t,\boldsymbol{\xi}))} \qquad \text{and} \qquad \tilde{\sigma}_2(t,\boldsymbol{\xi}) \doteq \frac{\tilde{\sigma}_0(t,\boldsymbol{\xi})}{\widehat{\Lambda}(t,\boldsymbol{\xi})}$$

respectively.

Let $z_{\alpha/2}$ be the upper $100(1-\frac{\alpha}{2})$ percentile point of the standard normal distribution. A $100(1-\alpha)\%$ confidence interval for the survivor function $S(t,\boldsymbol{\xi})$ is given in the following table.

Method	CLTYPE	Confidence Limits
LOG	PL	$\exp[\log(\hat{S}(t,\boldsymbol{\xi})) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_0(t,\boldsymbol{\xi})]$
LOG	CH	$\exp[\log(\tilde{S}(t,\boldsymbol{\xi})) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_0(t,\boldsymbol{\xi})]$
LOGLOG	PL	$\exp\{-\exp[\log(-\log(\hat{S}(t,\boldsymbol{\xi}))) \pm z_{\frac{\alpha}{2}}\hat{\sigma}_2(t,\boldsymbol{\xi})]\}$
LOGLOG	CH	$\exp\{-\exp[\log(-\log(\tilde{S}(t,\boldsymbol{\xi}))) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_2(t,\boldsymbol{\xi})]\}$
NORMAL	PL	$\hat{S}(t,\boldsymbol{\xi}) \pm z_{\frac{\alpha}{2}}\hat{\sigma}_1(t,\boldsymbol{\xi})$
NORMAL	CH	$\tilde{S}(t,\boldsymbol{\xi}) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_1(t,\boldsymbol{\xi})$

Left Truncation of Failure Times

Left truncation arises when individuals only come under observation some known time after the natural time origin of the phenomenon under study. The risk set just prior to an event time does not include individuals whose left truncation times exceed the given event time. Thus, any contribution to the likelihood must be conditional on the truncation limit having been exceeded.

Although left truncation can be accommodated in PROC PHREG through the counting process style of input, such specification does not allow survival estimates to be output. Using the ENTRY= option in PROC PHREG for left truncation does not suppress computing the survival estimates.

Consider the following specifications of PROC PHREG:

```
proc phreg data=one;
   model t2*dead(0)=x1-x10/entry=t1;
   baseline out=out1 survival=s;
   title 'The ENTRY= option is Specified';
run;

proc phreg data=one;
   model (t1,t2)*dead(0)=x1-x10;
   baseline out=out2 survival=s;
   title 'Counting Process Style of Input';
run;
```

Both specifications yield the same model estimates; however, the baseline data set out2 is empty, since survivor function estimates are not computed when you use the counting process style of input.

Variable Selection Methods

Five variable selection methods are available. The simplest method (and the default) is SELECTION=NONE, for which PROC PHREG fits the complete model as specified in the MODEL statement. The other four methods are FORWARD for forward selection, BACKWARD for backward elimination, STEPWISE for stepwise selection, and SCORE for best subsets selection. These methods are specified with the SELECTION= option in the MODEL statement. Intercept parameters are forced to stay in the model unless the NOINT option is specified.

When SELECTION=FORWARD, PROC PHREG first estimates parameters for variables forced into the model. These variables are the intercepts and the first n explanatory variables in the MODEL statement, where n is the number specified by the START= or INCLUDE= option in the MODEL statement (n is zero by default). Next, the procedure computes the adjusted chi-square statistics for each variable not in the model and examines the largest of these statistics. If it is significant at the SLSENTRY= level, the corresponding variable is added to the model. Once a variable is entered in the model, it is never removed from the model. The process is repeated until none of the remaining variables meet the specified level for entry or until the STOP= value is reached.

When SELECTION=BACKWARD, parameters for the complete model as specified in the MODEL statement are estimated unless the START= option is specified. In that case, only the parameters for the intercepts and the first n explanatory variables in the MODEL statement are estimated, where n is the number specified by the START= option. Results of the Wald test for individual parameters are examined. The least significant variable that does not meet the SLSSTAY= level for staying in the model is removed. Once a variable is removed from the model, it remains excluded. The process is repeated until no other variable in the model meets the specified level for removal or until the STOP= value is reached.

The SELECTION=STEPWISE option is similar to the SELECTION=FORWARD option except that variables already in the model do not necessarily remain. Variables are entered into and removed from the model in such a way that each forward selection step can be followed by one or more backward elimination steps. The stepwise selection process terminates if no further variable can be added to the model or if the variable just entered into the model is the only variable removed in the subsequent backward elimination.

For SELECTION=SCORE, PROC PHREG uses the branch and bound algorithm of Furnival and Wilson (1974) to find a specified number of models with the highest likelihood score (chi-square) statistic for all possible model sizes, from 1, 2, 3 variables, and so on, up to the single model containing all of the explanatory variables. The number of models displayed for each model size is controlled by the BEST= option. You can use the START= option to impose a minimum model size, and you can use the STOP= option to impose a maximum model size. For instance, with BEST=3, START=2, and STOP=5, the SCORE selection method displays the best three models (that is, the three models with the highest score chi-squares) containing 2, 3, 4, and 5 variables.

The SEQUENTIAL and STOPRES options can alter the default criteria for adding variables to or removing variables from the model when they are used with the FORWARD, BACKWARD, or STEPWISE selection methods.

Assessment of the Proportional Hazards Model (Experimental)

The proportional hazards model specifies that the hazard function for the failure time T associated with a $p \times 1$ column covariate vector $\mathbf{Z}$ takes the form

$$\lambda(t;\mathbf{Z}) = \lambda_0(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}}$$

where $\lambda_0(.)$ is an unspecifed baseline hazard function and $\boldsymbol{\beta}$ is a $p \times 1$ column vector of regression parameters. Lin et al. (1993) present graphical and numerical methods for model assessment based on the cumulative sums of martingale residuals and their transforms over certain coordinates (e.g., covariate values or follow-up times). The distributions of these stochastic processes under the assumed model can be approximated by the distributions of certain zero-mean Gaussian processes whose realizations can be generated by simulation. Each observed residual pattern can then be compared, both graphically and numerically, with a number of realizations from the null distribution. Such comparisons enable you to assess objectively whether the observed residual pattern reflects anything beyond random fluctuation. These procedures are useful in determining appropriate functional forms of covariates and assessing the proportional hazards assumption. You use the ASSESS statement to carry out these model-checking procedures.

For a sample of n subjects, let $(X_i, \Delta_i, \mathbf{Z}_i)$ be the data of the ith subject; that is, X_i represents the observed failure time, Δ_i has a value of 1 if X_i is an uncensored time and 0 otherwise, and $\mathbf{Z}_i = (Z_{1i}, \ldots, Z_{pi})'$ is a p-vector of covariates. Let $N_i(t) = \Delta_i I(X_i \le t)$ and $Y_i(t) = I(X_i \ge t)$. Let

$$S^{(0)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} Y_i(t) \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i} \quad \text{and} \quad \mathbf{Z}(\boldsymbol{\beta}, t) = \frac{\sum_{i=1}^{n} Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i}\mathbf{Z}_i}{S^{(0)}(\boldsymbol{\beta}, t)}$$

Let $\hat{\boldsymbol{\beta}}$ be the maximum partial likelihood estimate of $\boldsymbol{\beta}$, and let $\mathcal{I}(\hat{\boldsymbol{\beta}})$ be the observed information matrix.

The martingale residuals are defined as

$$\hat{M}_i(t) = N_i(t) - \int_0^t Y_i(u)\mathrm{e}^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_i} d\hat{\Lambda}_0(u), \qquad i = 1, \ldots, n$$

where $\hat{\Lambda}_0(t) = \int_0^t \frac{\sum_{i=1}^{n} dN_i(u)}{S^{(0)}(\hat{\boldsymbol{\beta}}, u)}$.

The empirical score process $\mathbf{U}(\hat{\boldsymbol{\beta}}, t) = (U_1(\hat{\boldsymbol{\beta}}, t), \ldots, U_p(\hat{\boldsymbol{\beta}}, t))'$ is a transform of the martingale residuals:

$$\mathbf{U}(\hat{\boldsymbol{\beta}}, t) = \sum_{i=1}^{n} \mathbf{Z}_i \hat{M}_i(t)$$

Checking the Functional Form of a Covariate

To check the functional form of the jth covariate, consider the partial-sum process of $\hat{M}_i = \hat{M}_i(\infty)$:

$$W_j(z) = \sum_{i=1}^{n} I(Z_{ji} \le z)\hat{M}_i$$

Under that null hypothesis that the model holds, $W_j(z)$ can be approximated by the zero-mean gaussian process

$$\begin{aligned} \hat{W}_j(z) \;\; = \;\; & \sum_{l=1}^{n} \Delta_l \Bigg\{ I(Z_{jl} \le z) - \frac{\sum_{i=1}^{n} Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i} I(Z_{ij} \le z)}{S^{(0)}(\hat{\boldsymbol{\beta}}, X_l)} \Bigg\} G_l - \\ & \sum_{k=1}^{n} \int_0^\infty Y_k(s)\mathrm{e}^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_k} I(Z_{jk} \le z)[\mathbf{Z}_k - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, s)]' d\hat{\Lambda}_0(s) \\ & \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{l=1}^{n} \Delta_l [\mathbf{Z}_k - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, X_l)] G_l \end{aligned}$$

where $(G_1, \ldots, G_n)$ are independent standard normal variables that are independent of $(X_i, \Delta_i, \mathbf{Z}_i)$, $i = 1, \ldots, n$.

You can assess the functional form of the jth covariate by plotting a small number of realizations (the default is 20) of $\hat{W}_j(z)$ on the same graph as the observed $W_j(z)$ and visually comparing them to see how typical the observed pattern of $W_j(z)$ is of the null distribution samples. You can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j be the observed value of $S_j = \sup_z |W_j(z)|$ and let $\hat{S}_j = \sup_z |\hat{W}_j(z)|$. The p-value $\Pr(S_j \geq s_j)$ is approximated by $\Pr(\hat{S}_j \geq s_j)$, which in turn is approximated by generating a large number of realizations (1000 is the default) of $\hat{W}_j(.)$.

Checking the Proportional Hazards Assumption

Consider the standardized empirical score process for the jth component of **Z**

$$U_j^*(t) = [\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})_{jj}]^{\frac{1}{2}} U_j(\hat{\boldsymbol{\beta}}, t),$$

Under the null hypothesis that the model holds, $U_j^*(t)$ can be approximated by

$$\begin{aligned}\hat{U}_j^*(t) \quad = \quad & [\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})_{jj}]^{\frac{1}{2}} \Bigg\{ \sum_{l=1}^{n} I(X_l \leq t)\Delta_l [Z_{jl} - \bar{Z}_j(\hat{\boldsymbol{\beta}}, t)] G_l - \\ & \sum_{k=1}^{n} \int_0^t Y_k(s) \mathrm{e}^{\hat{\boldsymbol{\beta}}' \mathbf{Z}_k} Z_{jk} [\mathbf{Z}_k - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, s)]' d\hat{\Lambda}_0(s) \\ & \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{l=1}^{n} \Delta_l [\mathbf{Z}_l - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, X_l)] G_l \Bigg\}\end{aligned}$$

where $\bar{Z}_j(\hat{\boldsymbol{\beta}}, t)$ is the jth component of $\bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, t)$, and $(G_1, \ldots, G_n)$ are independent standard normal variables that are independent of $(X_i, \Delta_i, \mathbf{Z}_i, (i = 1, \ldots, n)$.

You can assess the proportional hazards assumption for the jth covariate by plotting a few realizations of $\hat{U}_j^*(t)$ on the same graph as the observed $U_j^*(t)$ and visually comparing them to see how typical the observed pattern of $U_j^*(t)$ is of the null distribution samples. Again you can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j^* be the observed value of $S_j^* = \sup_t |U_j^*(t)|$ and let $\hat{S}_j^* = \sup_t |\hat{U}_j^*(t)|$. The p-value $\Pr[S_j^* \geq s_j^*]$ is approximated by $\Pr[\hat{S}_j^* \geq s_j^*]$, which in turn is approximated by generating a large number of realizations (1000 is the default) of $\hat{U}_j^*(.)$.

Computational Resources

Let n be the number of observations in a BY group. Let p be the number of explanatory variables. The minimum working space (in bytes) needed to process the BY group is

$$\max\{12n, 24p^2 + 160p\}$$

Extra memory is needed for certain TIES= options. Let k be the maximum multiplicity of tied times. The TIES=DISCRETE option requires extra memory (in bytes) of

$$4k(p^2 + 4p)$$

The TIES=EXACT option requires extra memory (in bytes) of

$$24(k^2 + 5k)$$

If sufficient space is available, the input data are also kept in memory. Otherwise, the input data are reread from the utility file for each evaluation of the likelihood function and its derivatives, with the resulting execution time substantially increased.

Displayed Output

If you use the NOPRINT option in the PROC PHREG statement, the procedure does not display any output. Otherwise, the displayed output of the PHREG procedure includes the following:

- the “Model Information” table, which contains:
  - the two-level name of the input data set
  - the name and label of the failure time variable
  - if you specify the censoring variable,
    - the name and label of the censoring variable
    - the values that the censoring variable assumes to indicate censored times
  - if you use the OFFSET= option in the MODEL statement, the name and label of the offset variable
  - if you specify the FREQ statement, the name and label of the frequency variable
  - if you specify the WEIGHT statement, the name and label of the weight variable
  - the method of handling ties in the failure time
- the “Summary of the Number of Event and Censored Values” table, which displays, for each stratum, the breakdown of the number of events and censored values. This table is not produced if the NOSUMMARY option is specified.
- if you specify the SIMPLE option in the PROC PHREG statement, the “Simple Statistics for Explanatory Variables” table, which displays, for each stratum, the mean, standard deviation, and minimum and maximum for each explanatory variable in the MODEL statement

- if you specify the ITPRINT option in the MODEL statement, the “Iteration History” table, which displays the iteration number, step size, log likelihood, and parameter estimates at each iteration The last evaluation of the gradient vector is also displayed.
- the “Model Fit Statistics” table, which gives the values of -2 log likelihood for fitting a model with no explanatory variable and for fitting a model with all the explanatory variables. The AIC and SBC are also given in this table.
- the “Testing Global Null Hypothesis: BETA=0” table, which displays results of the likelihood ratio test, the score test, and the Wald test
- the “Analysis of Maximum Likelihood Estimates” table, which contains:
  - the maximum likelihood estimate of the parameter
  - the estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
  - if you specify the COVS option in the PROC statement, the ratio of the robust standard error estimate to the model-based standard error estimate
  - the Wald Chi-Square statistic, computed as the square of the parameter estimate divided by its standard error estimate
  - the degrees of freedom of the Wald chi-square statistic. It has a value of 1 unless the corresponding parameter is redundant or infinite, in which case the value is 0.
  - the *p*-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom
  - the hazards ratio estimate computed by exponentiating the parameter estimate
  - if you specified the RISKLIMITS option in the MODEL statement, the confidence limits for the hazards ratio
- if you specify SELECTION=SCORE in the MODEL statement, the “Regression Models Selected by Score Criterion” table, which gives the number of explanatory variables in each model, the score chi-square statistic, and the names of the variables included in the model
- if you use the FORWARD or STEPWISE selection method and you specify the DETAILS option in the MODEL statement, the “Analysis of Variables Not in the Model” table, which gives the Score chi-square statistic for testing the significance of each variable not in the model (after adjusting for the variables already in the model), and the *p*-value of the chi-square statistic with respect to a chi-square distribution with one degree of freedom. This table is produced before a variable is selected for entry in a forward selection step.
- if you specify the FORWARD, BACKWARD, or STEPWISE selection method, a table summarizing the model-building process, which gives the step number, the explanatory variable entered or removed at each step, the chi-square statistic, and the corresponding *p*-value on which the entry or removal is based

- if you use the COVB option in the MODEL statement, the estimated covariance matrix of the parameter estimates
- if you use the CORRB option in the MODEL statement, the estimated correlation matrix of the parameter estimates
- if you specify a TEST statement,
  - the “Linear Coefficients” table, which gives the coefficients and constants of the linear hypothesis (if the E option is specified)
  - the printing of the intermediate calculations of the Wald test (if the option PRINT is specified)
  - the “Test Results” table, which gives the Wald chi-square statistic, the degrees of freedom, and the p-value
  - the “Average Effect” table, which gives the weighted average of the parameter estimates for the variables in the TEST statement, the estimated standard error, the z-score, and the *p*-value (if the AVERAGE option is specified)

ODS Table Names

PROC PHREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 54.1. ODS Tables Produced in PROC PHREG

ODS Table Name	Description	Statement	Option
BestSubsets	Best subset selection	MODEL	SELECTION=SCORE
CensoredSummary	Summary of event and censored observations	MODEL	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB
FitStatistics	Model fit statistics	MODEL	default
FunctionalFormSupTest (experimental)	Supremum test for functional form	ASSESS	VAR=
GlobalScore	Global chi-square test	MODEL	NOFIT
GlobalTests	Tests of the global null hypothesis	MODEL	default
IterHistory	Iteration history	MODEL	ITPRINT
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
ModelBuildingSummary	Summary of model building	MODEL	SELECTION=B/F/S
ModelInfo	Model information	PROC	default
NObs	Number of observations		default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default

Table 54.1. (continued)

ODS Table Name	Description	Statement	Option
ProportionalHazardsSupTest (experimental)	Supremum test for proportional hazards assumption	ASSESS	PH
ResidualChiSq	Residual chi-square	MODEL	SELECTION=F/B
SimpleStatistics	Summary statistics for explanatory variables	PROC	SIMPLE
TestAverage	Average effect for test	TEST	AVERAGE
TestCoeff	Coefficients for linear hypotheses	TEST	E
TestPrint1	**L**[cov(**b**)]**L**' and **Lb-c**	TEST	PRINT
TestPrint2	Ginv(**L**[cov(**b**)]**L**') and Ginv(**L**[cov(**b**)]**L**')(**Lb-c**)	TEST	PRINT
TestStmts	Linear hypotheses testing results	TEST	
VariablesNotInModel	Analysis of variables not in the model	MODEL	SELECTION=F/S

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs for model assessment with the PHREG procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the ASSESS statement in PROC PHREG. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

ODS Graph Names

PROC PHREG assigns a name to each graph it creates using the ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 54.2.

Table 54.2. ODS Graphics Produced by PROC PHREG

ODS Graph Name	Description	Statement	Option
CumulativeResiduals	Cumulative martingale residual plot	ASSESS	VAR=
CumResidPanel	Panel plot of cumulative martingale residuals	ASSESS	VAR= and CRPANEL
ScoreProcess	Standardized score process plot	ASSESS	PH

Examples

Example 54.1. Stepwise Regression

Krall, Uthoff, and Harley (1975) analyzed data from a study on multiple myeloma in which researchers treated 65 patients with alkylating agents. Of those patients, 48 died during the study and 17 survived. In the data set Myeloma, the variable Time represents the survival time in months from diagnosis. The variable VStatus consists of two values, 0 and 1, indicating whether the patient was alive or dead, respectively, at the end of the study. If the value of VStatus is 0, the corresponding value of Time is censored. The variables thought to be related to survival are LogBUN (log(BUN) at diagnosis), HGB (hemoglobin at diagnosis), Platelet (platelets at diagnosis: 0=abnormal, 1=normal), Age (age at diagnosis in years), LogWBC (log(WBC) at diagnosis), Frac (fractures at diagnosis: 0=none, 1=present), LogPBM (log percentage of plasma cells in bone marrow), Protein (proteinuria at diagnosis), and SCalc (serum calcium at diagnosis). Interest lies in identifying important prognostic factors from these nine explanatory variables.

```
data Myeloma;
   input Time VStatus LogBUN HGB Platelet Age LogWBC Frac
         LogPBM Protein SCalc;
   label Time='Survival Time'
         VStatus='0=Alive 1=Dead';
   datalines;
 1.25  1  2.2175   9.4  1  67  3.6628  1  1.9542  12  10
 1.25  1  1.9395  12.0  1  38  3.9868  1  1.9542  20  18
 2.00  1  1.5185   9.8  1  81  3.8751  1  2.0000   2  15
 2.00  1  1.7482  11.3  0  75  3.8062  1  1.2553   0  12
 2.00  1  1.3010   5.1  0  57  3.7243  1  2.0000   3   9
 3.00  1  1.5441   6.7  1  46  4.4757  0  1.9345  12  10
 5.00  1  2.2355  10.1  1  50  4.9542  1  1.6628   4   9
 5.00  1  1.6812   6.5  1  74  3.7324  0  1.7324   5   9
 6.00  1  1.3617   9.0  1  77  3.5441  0  1.4624   1   8
 6.00  1  2.1139  10.2  0  70  3.5441  1  1.3617   1   8
 6.00  1  1.1139   9.7  1  60  3.5185  1  1.3979   0  10
 6.00  1  1.4150  10.4  1  67  3.9294  1  1.6902   0   8
 7.00  1  1.9777   9.5  1  48  3.3617  1  1.5682   5  10
 7.00  1  1.0414   5.1  0  61  3.7324  1  2.0000   1  10
 7.00  1  1.1761  11.4  1  53  3.7243  1  1.5185   1  13
 9.00  1  1.7243   8.2  1  55  3.7993  1  1.7404   0  12
11.00  1  1.1139  14.0  1  61  3.8808  1  1.2788   0  10
11.00  1  1.2304  12.0  1  43  3.7709  1  1.1761   1   9
11.00  1  1.3010  13.2  1  65  3.7993  1  1.8195   1  10
11.00  1  1.5682   7.5  1  70  3.8865  0  1.6721   0  12
11.00  1  1.0792   9.6  1  51  3.5051  1  1.9031   0   9
13.00  1  0.7782   5.5  0  60  3.5798  1  1.3979   2  10
14.00  1  1.3979  14.6  1  66  3.7243  1  1.2553   2  10
15.00  1  1.6021  10.6  1  70  3.6902  1  1.4314   0  11
16.00  1  1.3424   9.0  1  48  3.9345  1  2.0000   0  10
16.00  1  1.3222   8.8  1  62  3.6990  1  0.6990  17  10
17.00  1  1.2304  10.0  1  53  3.8808  1  1.4472   4   9
17.00  1  1.5911  11.2  1  68  3.4314  0  1.6128   1  10
18.00  1  1.4472   7.5  1  65  3.5682  0  0.9031   7   8
19.00  1  1.0792  14.4  1  51  3.9191  1  2.0000   6  15
```

```
   19.00  1  1.2553   7.5  0  60  3.7924  1  1.9294   5   9
   24.00  1  1.3010  14.6  1  56  4.0899  1  0.4771   0   9
   25.00  1  1.0000  12.4  1  67  3.8195  1  1.6435   0  10
   26.00  1  1.2304  11.2  1  49  3.6021  1  2.0000  27  11
   32.00  1  1.3222  10.6  1  46  3.6990  1  1.6335   1   9
   35.00  1  1.1139   7.0  0  48  3.6532  1  1.1761   4  10
   37.00  1  1.6021  11.0  1  63  3.9542  0  1.2041   7   9
   41.00  1  1.0000  10.2  1  69  3.4771  1  1.4771   6  10
   41.00  1  1.1461   5.0  1  70  3.5185  1  1.3424   0   9
   51.00  1  1.5682   7.7  0  74  3.4150  1  1.0414   4  13
   52.00  1  1.0000  10.1  1  60  3.8573  1  1.6532   4  10
   54.00  1  1.2553   9.0  1  49  3.7243  1  1.6990   2  10
   58.00  1  1.2041  12.1  1  42  3.6990  1  1.5798  22  10
   66.00  1  1.4472   6.6  1  59  3.7853  1  1.8195   0   9
   67.00  1  1.3222  12.8  1  52  3.6435  1  1.0414   1  10
   88.00  1  1.1761  10.6  1  47  3.5563  0  1.7559  21   9
   89.00  1  1.3222  14.0  1  63  3.6532  1  1.6232   1   9
   92.00  1  1.4314  11.0  1  58  4.0755  1  1.4150   4  11
    4.00  0  1.9542  10.2  1  59  4.0453  0  0.7782  12  10
    4.00  0  1.9243  10.0  1  49  3.9590  0  1.6232   0  13
    7.00  0  1.1139  12.4  1  48  3.7993  1  1.8573   0  10
    7.00  0  1.5315  10.2  1  81  3.5911  0  1.8808   0  11
    8.00  0  1.0792   9.9  1  57  3.8325  1  1.6532   0   8
   12.00  0  1.1461  11.6  1  46  3.6435  0  1.1461   0   7
   11.00  0  1.6128  14.0  1  60  3.7324  1  1.8451   3   9
   12.00  0  1.3979   8.8  1  66  3.8388  1  1.3617   0   9
   13.00  0  1.6628   4.9  0  71  3.6435  0  1.7924   0   9
   16.00  0  1.1461  13.0  1  55  3.8573  0  0.9031   0   9
   19.00  0  1.3222  13.0  1  59  3.7709  1  2.0000   1  10
   19.00  0  1.3222  10.8  1  69  3.8808  1  1.5185   0  10
   28.00  0  1.2304   7.3  1  82  3.7482  1  1.6721   0   9
   41.00  0  1.7559  12.8  1  72  3.7243  1  1.4472   1   9
   53.00  0  1.1139  12.0  1  66  3.6128  1  2.0000   1  11
   57.00  0  1.2553  12.5  1  66  3.9685  0  1.9542   0  11
   77.00  0  1.0792  14.0  1  60  3.6812  0  0.9542   0  12
   ;
```

The stepwise selection process consists of a series of alternating step-up and step-down phases. The former adds variables to the model, while the latter removes variables from the model.

Stepwise regression analysis is requested by specifying the SELECTION=STEPWISE option in the MODEL statement. The option SLENTRY=0.25 specifies that a variable has to be significant at the 0.25 level before it can be entered into the model, while the option SLSTAY=0.15 specifies that a variable in the model has to be significant at the 0.15 level for it to remain in the model. The DETAILS option requests detailed results for the variable selection process.

```
proc phreg data=Myeloma;
   model Time*VStatus(0)=LogBUN HGB Platelet Age LogWBC
                         Frac LogPBM Protein SCalc
                         / selection=stepwise slentry=0.25
                           slstay=0.15 details;
run;
```

Results of the stepwise regression analysis are displayed in Output 54.1.1 through Output 54.1.7.

Output 54.1.1. Individual Score Test Results for All Variables

```
                              The PHREG Procedure

                               Model Information

             Data Set                 WORK.MYELOMA
             Dependent Variable       Time              Survival Time
             Censoring Variable       VStatus           0=Alive 1=Dead
             Censoring Value(s)       0
             Ties Handling            BRESLOW


                Summary of the Number of Event and Censored Values

                                                     Percent
                      Total        Event    Censored    Censored

                         65           48          17       26.15


                      Analysis of Variables Not in the Model

                                       Score
                       Variable    Chi-Square    Pr > ChiSq

                       LogBUN          8.5164        0.0035
                       HGB             5.0664        0.0244
                       Platelet        3.1816        0.0745
                       Age             0.0183        0.8924
                       LogWBC          0.5658        0.4519
                       Frac            0.9151        0.3388
                       LogPBM          0.5846        0.4445
                       Protein         0.1466        0.7018
                       SCalc           1.1109        0.2919


                             Residual Chi-Square Test

                        Chi-Square       DF     Pr > ChiSq

                           18.4550        9         0.0302
```

Individual score tests are used to determine which of the nine explanatory variables is first selected into the model. In this case, the score test for each variable is the global score test for the model containing that variable as the only explanatory variable. The chi-square statistic is compared to a chi-square distribution with one degree of freedom. Output 54.1.1 displays the chi-square statistics and the corresponding p-values. The variable LogBUN has the largest chi-square value (8.5164), and it is significant ($p = 0.0035$) at the SLENTRY=0.25 level. The variable LogBUN is thus entered into the model. Output 54.1.2 displays the model results. Since the Wald chi-square statistic is significant ($p = 0.0039$) at the SLSTAY=0.15 level, LogBUN stays in the model.

Output 54.1.2. First Model in the Stepwise Selection Process

```
Step  1. Variable LogBUN is entered.  The model contains the following
         explanatory variables:

         LogBUN


                               Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                     Without        With
                    Criterion     Covariates     Covariates

                    -2 LOG L         309.716        301.959
                    AIC              309.716        303.959
                    SBC              309.716        305.830


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         7.7572        1         0.0053
            Score                    8.5164        1         0.0035
            Wald                     8.3392        1         0.0039


                    Analysis of Maximum Likelihood Estimates

                   Parameter     Standard                                Hazard
  Variable   DF     Estimate        Error   Chi-Square   Pr > ChiSq       Ratio

  LogBUN      1      1.74595      0.60460       8.3392       0.0039       5.731
```

Output 54.1.3. Score Tests Adjusted for the Variable LogBUN

```
                     Analysis of Variables Not in the Model

                                       Score
                      Variable    Chi-Square    Pr > ChiSq

                      HGB             4.3468        0.0371
                      Platelet        2.0183        0.1554
                      Age             0.7159        0.3975
                      LogWBC          0.0704        0.7908
                      Frac            1.0354        0.3089
                      LogPBM          1.0334        0.3094
                      Protein         0.5214        0.4703
                      SCalc           1.4150        0.2342


                            Residual Chi-Square Test

                       Chi-Square       DF      Pr > ChiSq

                           9.3164        8          0.3163
```

Output 54.1.4. Second Model in the Stepwise Selection Process

```
Step  2. Variable HGB is entered.  The model contains the following explanatory
         variables:

         LogBUN  HGB


                               Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                     Without        With
                    Criterion     Covariates     Covariates

                    -2 LOG L        309.716       297.767
                    AIC             309.716       301.767
                    SBC             309.716       305.509


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        11.9493        2         0.0025
            Score                   12.7252        2         0.0017
            Wald                    12.1900        2         0.0023


                    Analysis of Maximum Likelihood Estimates

                   Parameter     Standard                                Hazard
  Variable   DF     Estimate        Error   Chi-Square    Pr > ChiSq      Ratio

  LogBUN      1      1.67440      0.61209       7.4833       0.0062       5.336
  HGB         1     -0.11899      0.05751       4.2811       0.0385       0.888
```

The next step consists of selecting another variable to add to the model. Output 54.1.3 displays the chi-square statistics and *p*-values of individual score tests (adjusted for LogBUN) for the remaining eight variables. The score chi-square for a given variable is the value of the likelihood score test for testing the significance of the variable in the presence of LogBUN. The variable HGB is selected because it has the highest chi-square value (4.3468), and it is significant ($p = 0.0371$) at the SLENTRY=0.25 level. Output 54.1.4 displays the fitted model containing both LogBUN and HGB. Based on the Wald statistics, neither LogBUN nor HGB is removed from the model.

Output 54.1.5. Third Model in the Stepwise Regression

```
Step  3. Variable SCalc is entered.  The model contains the following
         explanatory variables:

         LogBUN  HGB  SCalc


                               Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                      Without         With
                    Criterion      Covariates   Covariates

                    -2 LOG L          309.716      296.078
                    AIC               309.716      302.078
                    SBC               309.716      307.692


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        13.6377        3         0.0034
            Score                   15.3053        3         0.0016
            Wald                    14.4542        3         0.0023


                    Analysis of Maximum Likelihood Estimates

                       Parameter     Standard                                Hazard
  Variable   DF         Estimate        Error   Chi-Square   Pr > ChiSq       Ratio

  LogBUN      1          1.63593      0.62359       6.8822       0.0087       5.134
  HGB         1         -0.12643      0.05868       4.6419       0.0312       0.881
  SCalc       1          0.13286      0.09868       1.8127       0.1782       1.142
```

Output 54.1.5 shows Step 3 of the selection process, in which the variable SCalc is added, resulting in the model with LogBUN, HGB, and SCalc as the explanatory variables. Note that SCalc has the smallest Wald chi-square statistic, and it is not significant ($p = 0.1782$) at the SLSTAY=0.15 level. The variable SCalc is then removed from the model in a step-down phase in Step 4 (Output 54.1.6). The removal of SCalc brings the stepwise selection process to a stop in order to avoid repeatedly entering and removing the same variable.

Output 54.1.6. Final Model in the Stepwise Regression

```
Step  4. Variable SCalc is removed.  The model contains the following
         explanatory variables:

         LogBUN  HGB


                                   Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                                 Model Fit Statistics

                                         Without           With
                    Criterion         Covariates     Covariates

                    -2 LOG L             309.716        297.767
                    AIC                  309.716        301.767
                    SBC                  309.716        305.509


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio        11.9493        2         0.0025
            Score                   12.7252        2         0.0017
            Wald                    12.1900        2         0.0023


                    Analysis of Maximum Likelihood Estimates

                     Parameter     Standard                                Hazard
  Variable   DF       Estimate        Error   Chi-Square   Pr > ChiSq       Ratio

  LogBUN      1        1.67440      0.61209       7.4833       0.0062       5.336
  HGB         1       -0.11899      0.05751       4.2811       0.0385       0.888


NOTE: Model building terminates because the variable to be entered is the
      variable that was removed in the last step.
```

The procedure also displays a summary table of the steps in the stepwise selection process, as shown in Output 54.1.7.

The stepwise selection process results in a model with two explanatory variables, LogBUN and HGB.

Output 54.1.7. Model Selection Summary

```
                            Summary of Stepwise Selection

              Variable            Number         Score          Wald
Step     Entered      Removed       In        Chi-Square     Chi-Square     Pr > ChiSq

   1     LogBUN                      1           8.5164          .             0.0035
   2     HGB                         2           4.3468          .             0.0371
   3     SCalc                       3           1.8225          .             0.1770
   4                  SCalc          2            .            1.8127          0.1782
```

Example 54.2. Best Subset Selection

An alternative to stepwise selection of variables is best subset selection. The procedure uses the branch and bound algorithm of Furnival and Wilson (1974) to find a specified number of best models containing one, two, three variables, and so on, up to the single model containing all of the explanatory variables. The criterion used to determine "best" is based on the global score chi-square statistic. For two models A and B, each having the same number of explanatory variables, model A is considered to be better than model B if the global score chi-square statistic for A exceeds that for B.

Best subset selection analysis is requested by specifying the SELECTION=SCORE option in the MODEL statement. The BEST=3 option requests the procedure to identify only the three best models for each size. In other words, PROC PHREG will list the three models having the highest score statistics of all the models possible for a given number of covariates.

```
proc phreg data=Myeloma;
   model Time*VStatus(0)=LogBUN HGB Platelet Age LogWBC
                         Frac LogPBM Protein SCalc
                         / selection=score best=3;
run;
```

Output 54.2.1 displays the results of this analysis. The number of explanatory variables in the model is given in the first column, and the names of the variables are listed on the right. The models are listed in descending order of their score chi-square values within each model size. For example, among all models containing two explanatory variables, the model that contains the variables LogBUN and HGB has the largest score value (12.7252), the model that contains the variables LogBUN and Platelet has the second largest score value (11.1842), and the model that contains the variables LogBUN and SCalc has the third largest score value (9.9962).

Output 54.2.1. Best Variable Combinations

```
                                The PHREG Procedure

                  Regression Models Selected by Score Criterion

Number of       Score
Variables  Chi-Square  Variables Included in Model

        1      8.5164  LogBUN
        1      5.0664  HGB
        1      3.1816  Platelet

        2     12.7252  LogBUN HGB
        2     11.1842  LogBUN Platelet
        2      9.9962  LogBUN SCalc

        3     15.3053  LogBUN HGB SCalc
        3     13.9911  LogBUN HGB Age
        3     13.5788  LogBUN HGB Frac

        4     16.9873  LogBUN HGB Age SCalc
        4     16.0457  LogBUN HGB Frac SCalc
        4     15.7619  LogBUN HGB LogPBM SCalc

        5     17.6291  LogBUN HGB Age Frac SCalc
        5     17.3519  LogBUN HGB Age LogPBM SCalc
        5     17.1922  LogBUN HGB Age LogWBC SCalc

        6     17.9120  LogBUN HGB Age Frac LogPBM SCalc
        6     17.7947  LogBUN HGB Age LogWBC Frac SCalc
        6     17.7744  LogBUN HGB Platelet Age Frac SCalc

        7     18.1517  LogBUN HGB Platelet Age Frac LogPBM SCalc
        7     18.0568  LogBUN HGB Age LogWBC Frac LogPBM SCalc
        7     18.0223  LogBUN HGB Platelet Age LogWBC Frac SCalc

        8     18.3925  LogBUN HGB Platelet Age LogWBC Frac LogPBM SCalc
        8     18.1636  LogBUN HGB Platelet Age Frac LogPBM Protein SCalc
        8     18.1309  LogBUN HGB Platelet Age LogWBC Frac Protein SCalc

        9     18.4550  LogBUN HGB Platelet Age LogWBC Frac LogPBM Protein SCalc
```

Example 54.3. Conditional Logistic Regression for m:n Matching

Conditional logistic regression is used to investigate the relationship between an outcome and a set of prognostic factors in matched case-control studies. The outcome is whether the subject is a case or a control. If there is only one case and one control, the matching is 1:1. The *m:n* matching refers to the situation in which there is a varying number of cases and controls in the matched sets. You can perform conditional logistic regression with the PHREG procedure by using the discrete logistic model and forming a stratum for each matched set. In addition, you need to create dummy survival times so that all the cases in a matched set have the same event time value, and the corresponding controls are censored at later times.

Consider the following set of low infant birth-weight data extracted from Appendix 1 of Hosmer and Lemeshow (1989). These data represent 189 women, of whom 59 had low birth-weight babies and 130 had normal weight babies. Under investigation are the following risk factors: weight in pounds at the last menstrual period (LWT), presence of hypertension (HT), smoking status during pregnancy (Smoke), and presence of uterine irritability (UI). For HT, Smoke, and UI, a value of 1 indicates a "yes" and a value of 0 indicates a "no." The woman's age (Age) is used as the matching variable. The SAS data set LBW contains a subset of the data corresponding to women between the ages of 16 and 32.

```
data LBW;
   input id Age Low LWT Smoke HT UI @@;
   Time=2-Low;
   datalines;
 25  16   1   130   0  0  0    143  16   0   110   0  0  0
166  16   0   112   0  0  0    167  16   0   135   1  0  0
189  16   0   135   1  0  0    206  16   0   170   0  0  0
216  16   0    95   0  0  0     37  17   1   130   1  0  1
 45  17   1   110   1  0  0     68  17   1   120   1  0  0
 71  17   1   120   0  0  0     83  17   1   142   0  1  0
 93  17   0   103   0  0  0    113  17   0   122   1  0  0
116  17   0   113   0  0  0    117  17   0   113   0  0  0
147  17   0   119   0  0  0    148  17   0   119   0  0  0
180  17   0   120   1  0  0     49  18   1   148   0  0  0
 50  18   1   110   1  0  0     89  18   0   107   1  0  1
100  18   0   100   1  0  0    101  18   0   100   1  0  0
132  18   0    90   1  0  1    133  18   0    90   1  0  1
168  18   0   229   0  0  0    205  18   0   120   1  0  0
208  18   0   120   0  0  0     23  19   1    91   1  0  1
 33  19   1   102   0  0  0     34  19   1   112   1  0  1
 85  19   0   182   0  0  1     96  19   0    95   0  0  0
 97  19   0   150   0  0  0    124  19   0   138   1  0  0
129  19   0   189   0  0  0    135  19   0   132   0  0  0
142  19   0   115   0  0  0    181  19   0   105   0  0  0
187  19   0   235   1  1  0    192  19   0   147   1  0  0
193  19   0   147   1  0  0    197  19   0   184   1  1  0
224  19   0   120   1  0  0     27  20   1   150   1  0  0
 31  20   1   125   0  0  1     40  20   1   120   1  0  0
 44  20   1    80   1  0  1     47  20   1   109   0  0  0
 51  20   1   121   1  0  1     60  20   1   122   1  0  0
 76  20   1   105   0  0  0     87  20   0   105   1  0  0
104  20   0   120   0  0  1    146  20   0   103   0  0  0
155  20   0   169   0  0  1    160  20   0   141   0  0  1
172  20   0   121   1  0  0    177  20   0   127   0  0  0
201  20   0   120   0  0  0    211  20   0   170   1  0  0
217  20   0   158   0  0  0     20  21   1   165   1  1  0
 28  21   1   200   0  0  1     30  21   1   103   0  0  0
 52  21   1   100   0  0  0     84  21   1   130   1  1  0
 88  21   0   108   1  0  1     91  21   0   124   0  0  0
128  21   0   185   1  0  0    131  21   0   160   0  0  0
144  21   0   110   1  0  1    186  21   0   134   0  0  0
```

```
219  21   0   115   0  0  0    42  22   1   130   1  0  1
 67  22   1   130   1  0  0    92  22   0   118   0  0  0
 98  22   0    95   0  1  0   137  22   0    85   1  0  0
138  22   0   120   0  1  0   140  22   0   130   1  0  0
161  22   0   158   0  0  0   162  22   0   112   1  0  0
174  22   0   131   0  0  0   184  22   0   125   0  0  0
204  22   0   169   0  0  0   220  22   0   129   0  0  0
 17  23   1    97   0  0  1    59  23   1   187   1  0  0
 63  23   1   120   0  0  0    69  23   1   110   1  0  0
 82  23   1    94   1  0  0   130  23   0   130   0  0  0
139  23   0   128   0  0  0   149  23   0   119   0  0  0
164  23   0   115   1  0  0   173  23   0   190   0  0  0
179  23   0   123   0  0  0   182  23   0   130   0  0  0
200  23   0   110   0  0  0    18  24   1   128   0  0  0
 19  24   1   132   0  1  0    29  24   1   155   1  0  0
 36  24   1   138   0  0  0    61  24   1   105   1  0  0
118  24   0    90   1  0  0   136  24   0   115   0  0  0
150  24   0   110   0  0  0   156  24   0   115   0  0  0
185  24   0   133   0  0  0   196  24   0   110   0  0  0
199  24   0   110   0  0  0   225  24   0   116   0  0  0
 13  25   1   105   0  1  0    15  25   1    85   0  0  1
 24  25   1   115   0  0  0    26  25   1    92   1  0  0
 32  25   1    89   0  0  0    46  25   1   105   0  0  0
103  25   0   118   1  0  0   111  25   0   120   0  0  1
120  25   0   155   0  0  0   121  25   0   125   0  0  0
169  25   0   140   0  0  0   188  25   0    95   1  0  1
202  25   0   241   0  1  0   215  25   0   120   0  0  0
221  25   0   130   0  0  0    35  26   1   117   1  0  0
 54  26   1    96   0  0  0    75  26   1   154   0  1  0
 77  26   1   190   1  0  0    95  26   0   113   1  0  0
115  26   0   168   1  0  0   154  26   0   133   1  0  0
218  26   0   160   0  0  0    16  27   1   150   0  0  0
 43  27   1   130   0  0  1   125  27   0   124   1  0  0
  4  28   1   120   1  0  1    79  28   1    95   1  0  0
105  28   0   120   1  0  0   109  28   0   120   0  0  0
112  28   0   167   0  0  0   151  28   0   140   0  0  0
159  28   0   250   1  0  0   212  28   0   134   0  0  0
214  28   0   130   0  0  0    10  29   1   130   0  0  1
 94  29   0   123   1  0  0   114  29   0   150   0  0  0
123  29   0   140   1  0  0   190  29   0   135   0  0  0
191  29   0   154   0  0  0   209  29   0   130   1  0  0
 65  30   1   142   1  0  0    99  30   0   107   0  0  1
141  30   0    95   1  0  0   145  30   0   153   0  0  0
176  30   0   110   0  0  0   195  30   0   137   0  0  0
203  30   0   112   0  0  0    56  31   1   102   1  0  0
107  31   0   100   0  0  1   126  31   0   215   1  0  0
163  31   0   150   1  0  0   222  31   0   120   0  0  0
 22  32   1   105   1  0  0   106  32   0   121   0  0  0
134  32   0   132   0  0  0   170  32   0   134   1  0  0
175  32   0   170   0  0  0   207  32   0   186   0  0  0
;
```

The variable Low is used to determine whether the subject is a case (Low=1, low birth-weight baby) or a control (Low=0, normal weight baby). The dummy time variable Time takes the value 1 for cases and 2 for controls.

The following SAS statements produce a conditional logistic regression analysis of the data. The variable Time is the response, and Low is the censoring variable. Note that the data set is created so that all the cases have the same event time, and the controls have later censored times. The matching variable Age is used in the STRATA statement so each unique age value defines a stratum. The variables LWT, Smoke, HT, and UI are specified as explanatory variables. The TIES=DISCRETE option requests the discrete logistic model.

```
proc phreg data=LBW;
   model Time*Low(0)= LWT Smoke HT UI / ties=discrete;
   strata Age;
run;
```

The procedure displays a summary of the number of event and censored observations for each stratum. These are the number of cases and controls for each matched set shown in Output 54.3.1.

Output 54.3.1. Summary of Number of Case and Controls

```
                              The PHREG Procedure

                               Model Information

                       Data Set                    WORK.LBW
                       Dependent Variable          Time
                       Censoring Variable          Low
                       Censoring Value(s)          0
                       Ties Handling               DISCRETE


               Summary of the Number of Event and Censored Values

                                                                   Percent
    Stratum     Age              Total         Event    Censored    Censored

          1     16                   7             1           6       85.71
          2     17                  12             5           7       58.33
          3     18                  10             2           8       80.00
          4     19                  16             3          13       81.25
          5     20                  18             8          10       55.56
          6     21                  12             5           7       58.33
          7     22                  13             2          11       84.62
          8     23                  13             5           8       61.54
          9     24                  13             5           8       61.54
         10     25                  15             6           9       60.00
         11     26                   8             4           4       50.00
         12     27                   3             2           1       33.33
         13     28                   9             2           7       77.78
         14     29                   7             1           6       85.71
         15     30                   7             1           6       85.71
         16     31                   5             1           4       80.00
         17     32                   6             1           5       83.33
    -------------------------------------------------------------------
      Total                        174            54         120       68.97
```

Results of the conditional logistic regression analysis are shown in Output 54.3.2. Based on the Wald test for individual variables, the variables LWT, Smoke, and HT are statistically significant while UI is marginal.

The hazards ratios, computed by exponentiating the parameter estimates, are useful in interpreting the results of the analysis. If the hazards ratio of a prognostic factor is larger than 1, an increment in the factor increases the hazard rate. If the hazards ratio is less than 1, an increment in the factor decreases the hazard rate. Results indicate that women were more likely to have low birth-weight babies if they were underweight in the last menstrual cycle, were hypertensive, smoked during pregnancy, or suffered uterine irritability.

Output 54.3.2. Conditional Logistic Regression Analysis for the Low Birth-Weight Study

```
                              The PHREG Procedure

                               Convergence Status

                  Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                    Without           With
                    Criterion      Covariates      Covariates

                    -2 LOG L          159.069         141.108
                    AIC               159.069         149.108
                    SBC               159.069         157.064


                    Testing Global Null Hypothesis: BETA=0

            Test                    Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio           17.9613        4         0.0013
            Score                      17.3152        4         0.0017
            Wald                       15.5577        4         0.0037


                    Analysis of Maximum Likelihood Estimates

                        Parameter     Standard                                  Hazard
Variable     DF          Estimate        Error   Chi-Square   Pr > ChiSq         Ratio

LWT           1          -0.01498      0.00706       4.5001       0.0339         0.985
Smoke         1           0.80805      0.36797       4.8221       0.0281         2.244
HT            1           1.75143      0.73932       5.6120       0.0178         5.763
UI            1           0.88341      0.48032       3.3827       0.0659         2.419
```

For matched case-control studies with one case per matched set (1:*n* matching), the likelihood function for the conditional logistic regression reduces to that of the Cox model for the continuous time scale. For this situation, you can use the default TIES=BRESLOW.

Example 54.4. Model Using Time-Dependent Explanatory Variables

Time-dependent variables can be used to model the effects of subjects transferring from one treatment group to another. One example of the need for such strategies is the Stanford heart transplant program. Patients are accepted if physicians judge them suitable for heart transplant. Then, when a donor becomes available, physicians choose transplant recipients according to various medical criteria. A patient's status can be changed during the study from waiting for a transplant to being a transplant recipient. Transplant status can be defined by the time-dependent covariate function $z = z(t)$ as

$$z(t) = \begin{cases} 0 & \text{if the patient has not received the transplant at time } t \\ 1 & \text{if the patient has received the transplant at time } t \end{cases}$$

The Stanford heart transplant data that appear in Crowley and Hu (1977) consist of 103 patients, 69 of whom received transplants. The data are saved in a SAS data set called Heart. For each patient in the program, there is a birth date (Bir_Date), a date of acceptance (Acc_Date), and a date last seen (Ter_Date). The survival time (Time) in days is defined as Time = Ter_Date – Acc_Date. The survival time is said to be uncensored (Status=1) or censored (Status=0), depending on whether Ter_Date is the date of death or the closing date of the study. The age in years at acceptance into the program is Acc_Age = (Acc_Date – Bir_Date) / 365. Previous open-heart surgery for each patient is indicated by the variable PrevSurg. For each transplant recipient, there is a date of transplant (Xpl_Date) and three measures (NMismatch, Antigen, Mismatch) of tissue-type mismatching. The waiting period (WaitTime) in days for a transplant recipient is calculated as WaitTime = Xpl_Date – Acc_Date, and the age in years at transplant is Xpl_Age = (Xpl_Date – Bir_Date) / 365. For those who do not receive heart transplants, the WaitTime, Xpl_Age, NMismatch, Antigen, and Mismatch variables contain missing values.

The input data contains dates that have a two-digit year representation. The SAS option YEARCUTOFF=1900 is specified to ensure that a two-digit year xx is year 19xx.

The code is as follows:

```
options yearcutoff=1900;

data Heart;
   input ID
         @5  Bir_Date mmddyy8.
         @14 Acc_Date mmddyy8.
         @23 Xpl_Date mmddyy8.
         @32 Ter_Date mmddyy8.
         @41 Status 1.
         @43 PrevSurg 1.
         @45 NMismatch 1.
         @47 Antigen 1.
         @49 Mismatch 4.
         @54 Reject 1.
         @56 NotTyped $1.;
   label Bir_Date =’Date of birth’
         Acc_Date =’Date of acceptance’
         Xpl_Date =’Date of transplant’
         Ter_Date =’Date last seen’
         Status   =  ’Dead=1 Alive=0’
         PrevSurg =’Previous surgery’
         NMismatch= ’No of mismatches’
         Antigen  = ’HLA-A2 antigen’
         Mismatch =’Mismatch score’
         NotTyped = ’y=not tissue-typed’;
   Time= Ter_Date - Acc_Date;
   Acc_Age=int( (Acc_Date - Bir_Date)/365 );
   if ( Xpl_Date ne .) then do;
      WaitTime= Xpl_Date - Acc_Date;
      Xpl_Age= int( (Xpl_Date - Bir_Date)/365 );
   end;
   datalines;
  1 01 10 37 11 15 67          01 03 68 1 0
  2 03 02 16 01 02 68          01 07 68 1 0
  3 09 19 13 01 06 68 01 06 68 01 21 68 1 0 2 0 1.11 0
  4 12 23 27 03 28 68 05 02 68 05 05 68 1 0 3 0 1.66 0
  5 07 28 47 05 10 68          05 27 68 1 0
  6 11 18 13 06 13 68          06 15 68 1 0
  7 08 29 17 07 12 68 08 31 68 05 17 70 1 0 4 0 1.32 1
  8 03 27 23 08 01 68          09 09 68 1 0
  9 06 11 21 08 09 68          11 01 68 1 0
 10 02 09 26 08 11 68 08 22 68 10 07 68 1 0 2 0 0.61 1
 11 08 22 20 08 15 68 09 09 68 01 14 69 1 0 1 0 0.36 0
 12 07 09 15 09 17 68          09 24 68 1 0
 13 02 22 14 09 19 68 10 05 68 12 08 68 1 0 3 0 1.89 1
 14 09 16 14 09 20 68 10 26 68 07 07 72 1 0 1 0 0.87 1
 15 12 04 14 09 27 68          09 27 68 1 1
 16 05 16 19 10 26 68 11 22 68 08 29 69 1 0 2 0 1.12 1
 17 06 29 48 10 28 68          12 02 68 1 0
 18 12 27 11 11 01 68 11 20 68 12 13 68 1 0 3 0 2.05 0
 19 10 04 09 11 18 68          12 24 68 1 0
 20 10 19 13 01 29 69 02 15 69 02 25 69 1 0 3 1 2.76 1
 21 09 29 25 02 01 69 02 08 69 11 29 71 1 0 2 0 1.13 1
 22 06 05 26 03 18 69 03 29 69 05 07 69 1 0 3 0 1.38 1
 23 12 02 10 04 11 69 04 13 69 04 13 71 1 0 3 0 0.96 1
 24 07 07 17 04 25 69 07 16 69 11 29 69 1 0 3 1 1.62 1
```

```
25 02 06 36 04 28 69 05 22 69 04 01 74 0 0 2 0 1.06 0
26 10 18 38 05 01 69          03 01 73 0 0
27 07 21 60 05 04 69          01 21 70 1 0
28 05 30 15 06 07 69 08 16 69 08 17 69 1 0 2 0 0.47 0
29 02 06 19 07 14 69          08 17 69 1 0
30 09 20 24 08 19 69 09 03 69 12 18 71 1 0 4 0 1.58 1
31 10 04 14 08 23 69          09 07 69 1 0
32 04 02 05 08 29 69 09 14 69 11 13 69 1 0 4 0 0.69 1
33 01 01 21 11 27 69 01 16 70 04 01 74 0 0 3 0 0.91 0
34 05 24 29 12 12 69 01 03 70 04 01 74 0 0 2 0 0.38 0
35 08 04 26 01 21 70          02 01 70 1 0
36 05 01 21 04 04 70 05 19 70 07 12 70 1 0 2 0 2.09 1
37 10 24 08 04 25 70 05 13 70 06 29 70 1 0 3 1 0.87 1
38 11 14 28 05 05 70 05 09 70 05 09 70 1 0 3 0 0.87 0
39 11 12 19 05 20 70 05 21 70 07 11 70 1 0            y
40 11 30 21 05 25 70 07 04 70 04 01 74 0 1 4 0 0.75 0
41 04 30 25 08 19 70 10 15 70 04 01 74 0 1 2 0 0.98 0
42 03 13 34 08 21 70          08 23 70 1 0
43 06 01 27 10 22 70          10 23 70 1 1
44 05 02 28 11 30 70          01 08 71 1 1
45 10 30 34 01 05 71 01 05 71 02 18 71 1 0 1 0 0.0  0
46 06 01 22 01 10 71 01 11 71 10 01 73 1 1 2 0 0.81 1
47 12 28 23 02 02 71 02 22 71 04 14 71 1 0 3 0 1.38 1
48 01 23 15 02 05 71          02 13 71 1 0
49 06 21 34 02 15 71 03 22 71 04 01 74 0 1 4 0 1.35 0
50 03 28 25 02 15 71 05 08 71 10 21 73 1 1            y
51 06 29 22 03 24 71 04 24 71 01 02 72 1 0 4 1 1.08 1
52 01 24 30 04 25 71          08 04 71 1 0
53 02 27 24 07 02 71 08 11 71 01 05 72 1 0            y
54 09 16 23 07 02 71          07 04 71 1 0
55 02 24 19 08 09 71 08 18 71 10 08 71 1 0 2 0 1.51 1
56 12 05 32 09 03 71 11 08 71 04 01 74 0 0 4 0 0.98 0
57 06 08 30 09 13 71          02 08 72 1 0
58 09 17 23 09 23 71 10 13 71 08 30 72 1 1 2 1 1.82 1
59 05 12 30 09 29 71 12 15 71 04 01 74 0 1 2 0 0.19 0
60 10 29 22 11 18 71 11 20 71 01 09 72 1 0 3 0 0.66 1
61 05 12 19 12 04 71          12 05 71 1 0
62 08 01 32 12 09 71          02 15 72 1 0
63 04 15 39 12 12 71 01 07 72 04 01 74 0 0 3 1 1.93 0
64 04 09 23 02 01 72 03 04 72 09 06 73 1 1 1 0 0.12 0
65 11 19 20 03 06 72 03 17 72 05 22 72 1 0 2 0 1.12 1
66 01 02 19 03 20 72          04 20 72 1 0
67 09 03 52 03 23 72 05 18 72 01 01 73 1 0 3 0 1.02 0
68 01 10 27 04 07 72 04 09 72 06 13 72 1 0 3 1 1.68 1
69 06 05 24 06 01 72 06 10 72 04 01 74 0 0 2 0 1.20 0
70 06 17 19 06 17 72 06 21 72 07 16 72 1 0 3 1 1.68 1
71 02 22 25 07 21 72 08 20 72 04 01 74 0 0 3 0 0.97 0
72 11 22 45 08 14 72 08 17 72 04 01 74 0 0 3 1 1.46 0
73 05 13 16 09 11 72 10 07 72 12 09 72 1 0 3 1 2.16 1
74 07 20 43 09 18 72 09 22 72 10 04 72 1 0 1 0 0.61 0
75 07 25 20 09 29 72          09 30 72 1 0
76 09 03 20 10 04 72 11 18 72 04 01 74 0 1 3 1 1.70 0
77 08 27 31 10 06 72          10 26 72 1 0
78 02 20 24 11 03 72 05 31 73 04 01 74 0 0 3 0 0.81 0
79 02 18 19 11 30 72 02 04 73 03 05 73 1 0 2 0 1.08 1
80 06 27 26 12 06 72 12 31 72 04 01 74 0 1 3 0 1.41 0
81 02 21 20 01 12 73 01 17 73 04 01 74 0 0 4 1 1.94 0
82 09 19 42 11 01 71          01 01 73 0 0
83 10 04 19 01 24 73 02 24 73 04 13 73 1 0 4 0 3.05 0
```

```
 84 05 13 30 01 30 73 03 07 73 12 29 73 1 0 4 0 0.60 1
 85 02 13 25 02 06 73          02 10 73 1 0
 86 03 30 24 03 01 73 03 08 73 04 01 74 0 0 3 1 1.44 0
 87 12 19 26 03 21 73 05 19 73 07 08 73 1 0 2 0 2.25 1
 88 11 16 18 03 28 73 04 27 73 04 01 74 0 0 3 0 0.68 0
 89 03 19 22 04 05 73 08 21 73 10 28 73 1 0 4 1 1.33 1
 90 03 25 21 04 06 73 09 12 73 10 08 73 1 1 3 1 0.82 0
 91 09 08 25 04 13 73          03 18 74 1 0
 92 05 03 28 04 27 73 03 02 74 04 01 74 0 0 1 0 0.16 0
 93 10 10 25 07 11 73 08 07 73 04 01 74 0 0 2 0 0.33 0
 94 11 11 29 09 14 73 09 17 73 02 25 74 1 1 3 0 1.20 1
 95 06 11 33 09 22 73 09 23 73 10 07 73 1 0            y
 96 02 09 47 10 04 73 10 16 73 04 01 74 0 0 2 0 0.46 0
 97 04 11 50 11 22 73 12 12 73 04 01 74 0 0 3 1 1.78 0
 98 04 28 45 12 14 73 03 19 74 04 01 74 0 0 4 1 0.77 0
 99 02 24 24 12 25 73          01 14 74 1 0
100 01 31 39 02 22 74 03 31 74 04 01 74 0 1 3 0 0.67 0
101 08 25 24 03 02 74          04 01 74 0 0
102 10 30 33 03 22 74          04 01 74 0 0
103 05 20 28 09 13 67          09 18 67 1 0
;
```

Crowley and Hu (1977) have presented a number of analyses to assess the effects of various explanatory variables on the survival of patients. This example fits two of the models that they have considered.

The first model consists of two explanatory variables—the transplant status and the age at acceptance. The transplant status (XStatus) is a time-dependent variable defined by the programming statements between the MODEL statement and the RUN statement. The XStatus variable takes the value 1 or 0 at time t (measured from the date of acceptance), depending on whether or not the patient has received a transplant at that time. Note that the value of XStatus changes for subjects in each risk set (subjects still alive just before each distinct event time); therefore, the variable cannot be created in the DATA step. The variable Acc_Age, which is not time-dependent, accounts for the possibility that pretransplant risks vary with age.

```
proc phreg data= Heart;
   model Time*Status(0)= XStatus Acc_Age;
   if (WaitTime = . or Time < WaitTime) then XStatus=0.;
   else  XStatus= 1.0;
run;
```

Results of this analysis are shown in Output 54.4.1. Transplantation appears to be associated with a slight decrease in risk, although the effect is not significant ($p = 0.8432$). The age at acceptance as a pretransplant risk factor adds significantly to the model ($p = 0.0294$). The risk increases significantly with age at acceptance.

Output 54.4.1. Heart Transplant Study Analysis I

```
                              The PHREG Procedure

                               Model Information

             Data Set                 WORK.HEART
             Dependent Variable       Time
             Censoring Variable       Status          Dead=1 Alive=0
             Censoring Value(s)       0
             Ties Handling            BRESLOW


               Summary of the Number of Event and Censored Values

                                                        Percent
                     Total       Event     Censored    Censored

                       103          75           28       27.18


                               Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                  Without           With
                    Criterion     Covariates     Covariates

                    -2 LOG L         596.649        591.312
                    AIC              596.649        595.312
                    SBC              596.649        599.947


                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         5.3370        2         0.0694
            Score                    4.7900        2         0.0912
            Wald                     4.7812        2         0.0916
```

```
                                The PHREG Procedure

                      Analysis of Maximum Likelihood Estimates

                    Parameter     Standard                                  Hazard
 Variable    DF      Estimate        Error    Chi-Square    Pr > ChiSq       Ratio

 XStatus      1      -0.06046      0.30572        0.0391        0.8432       0.941
 Acc_Age      1       0.03147      0.01445        4.7443        0.0294       1.032
```

The second model consists of three explanatory variables—the transplant status, the transplant age, and the mismatch score. Four transplant recipients who were not typed have no Mismatch values; they are excluded from the analysis by the use of a WHERE clause. The transplant age (XAge) and the mismatch score (XScore) are also time-dependent and are defined in a fashion similar to that of XStatus. While the patient is waiting for a transplant, XAge and XScore have a value of 0. After the patient has migrated to the recipient population, XAge takes on the value of Xpl_Age (transplant age for the recipient), and XScore takes on the value of Mismatch (a measure of the degree of dissimilarity between donor and recipient).

```
proc phreg data= Heart;
   model Time*Status(0)= XStatus XAge XScore;
   where NotTyped ^= ’y’;
   if (WaitTime = . or Time < WaitTime) then do;
      XStatus=0.;
      XAge=0.;
      XScore= 0.;
   end;
   else do;
      XStatus= 1.0;
      XAge= Xpl_Age;
      XScore= Mismatch;
   end;
run;
```

Results of the analysis are shown in Output 54.4.2. Note that only 99 patients are included in this analysis, instead of 103 patients as in the previous analysis, since four transplant recipients who were not typed are excluded. The variable XAge is statistically significant ($p = 0.0146$) with a hazards ratio exceeding 1. Therefore, patients who had a transplant at younger ages lived longer than those who received a transplant later in their lives. The variable XScore has only minimal effect on the survival ($p = 0.1129$).

Output 54.4.2. Heart Transplant Study Analysis II

```
                               The PHREG Procedure

                                Model Information

             Data Set                    WORK.HEART
             Dependent Variable          Time
             Censoring Variable          Status          Dead=1 Alive=0
             Censoring Value(s)          0
             Ties Handling               BRESLOW


               Summary of the Number of Event and Censored Values

                                                          Percent
                     Total        Event     Censored     Censored

                        99           71           28        28.28


                                Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                              Model Fit Statistics

                                    Without           With
                    Criterion     Covariates     Covariates

                    -2 LOG L         561.646        551.911
                    AIC              561.646        557.911
                    SBC              561.646        564.699


                    Testing Global Null Hypothesis: BETA=0

            Test                    Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio            9.7350        3         0.0210
            Score                       9.0127        3         0.0291
            Wald                        9.0156        3         0.0291


                    Analysis of Maximum Likelihood Estimates

                     Parameter      Standard                                Hazard
  Variable    DF      Estimate         Error    Chi-Square    Pr > ChiSq     Ratio

  XStatus      1      -3.17799       1.18612        7.1787        0.0074     0.042
  XAge         1       0.05517       0.02259        5.9649        0.0146     1.057
  XScore       1       0.44424       0.28026        2.5125        0.1129     1.559
```

Example 54.5. Time-Dependent Repeated Measurements of a Covariate

Repeated determinations may be made during the course of a study of variables thought to be related to survival. Consider an experiment to study the dosing effect of a tumor-promoting agent. Forty-five rodents initially exposed to a carcinogen were randomly assigned to three dose groups. After the first death of an animal, the rodents were examined every week for the number of papillomas. Investigators were interested in determining the effects of dose on the carcinoma incidence after adjusting for the number of papillomas.

The input data set TUMOR consists of the following 19 variables:

- ID (subject identification)
- Time (survival time of the subject)
- Dead (censoring status where 1=dead and 0=censored)
- Dose (dose of the tumor-promoting agent)
- P1–P15 (number of papillomas at the 15 times that animals died. These 15 death times are weeks 27, 34, 37, 41, 43, 45, 46, 47, 49, 50, 51, 53, 65, 67, and 71. For instance, subject 1 died at week 47; it had no papilloma at week 27, five papillomas at week 34, six at week 37, eight at week 41, and 10 at weeks 43, 45, 46, and 47. For an animal that died before week 71, the number of papillomas is missing for those times beyond its death.)

The following SAS statements create the data set TUMOR:

```
data Tumor;
   infile datalines missover;
   input ID Time Dead Dose P1-P15;
   label ID='Subject ID';
   datalines;
 1 47 1  1.0  0  5  6  8 10 10 10 10
 2 71 1  1.0  0  0  0  0  0  0  0  0  1  1  1  1 1 1 1
 3 81 0  1.0  0  1  1  1  1  1  1  1  1  1  1  1 1 1 1
 4 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 5 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 6 65 1  1.0  0  0  0  1  1  1  1  1  1  1  1  1 1
 7 71 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 8 69 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0
 9 67 1  1.0  0  0  1  1  2  2  2  2  3  3  3  3 3 3
10 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
11 37 1  1.0  9  9  9
12 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
13 77 0  1.0  0  0  0  0  1  1  1  1  1  1  1  1 1 1 1
14 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
15 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
16 54 0  2.5  0  1  1  1  2  2  2  2  2  2  2  2
17 53 0  2.5  0  0  0  0  0  0  0  0  0  0  0  0
18 38 0  2.5  5 13 14
19 54 0  2.5  2  6  6  6  6  6  6  6  6  6  6  6
20 51 1  2.5 15 15 15 16 16 17 17 17 17 17 17
21 47 1  2.5 13 20 20 20 20 20 20 20
22 27 1  2.5 22
23 41 1  2.5  6 13 13 13
24 49 1  2.5  0  3  3  3  3  3  3  3  3
25 53 0  2.5  0  0  1  1  1  1  1  1  1  1  1  1
26 50 1  2.5  0  0  2  3  4  6  6  6  6  6
27 37 1  2.5  3 15 15
28 49 1  2.5  2  3  3  3  3  4  4  4  4
```

```
29 46 1  2.5  4  6  7  9  9  9  9
30 48 0  2.5 15 26 26 26 26 26 26 26
31 54 0 10.0 12 14 15 15 15 15 15 15 15 15 15 15
32 37 1 10.0 12 16 17
33 53 1 10.0  3  6  6  6  6  6  6  6  6  6  6  6
34 45 1 10.0  4 12 15 20 20 20
35 53 0 10.0  6 10 13 13 13 15 15 15 15 15 15 20
36 49 1 10.0  0  2  2  2  2  2  2  2  2
37 39 0 10.0  7  8  8
38 27 1 10.0 17
39 49 1 10.0  0  6  9 14 14 14 14 14 14
40 43 1 10.0 14 18 20 20 20
41 28 0 10.0  8
42 34 1 10.0 11 18
43 45 1 10.0 10 12 16 16 16 16
44 37 1 10.0  0  1  1
45 43 1 10.0  9 19 19 19 19
;
```

The number of papillomas (NPap) for each animal in the study was measured repeatedly over time. One way of handling time-dependent repeated measurements in the PHREG procedure is to use programming statements to capture the appropriate covariate values of the subjects in each risk set. In this example, NPap is a time-dependent explanatory variable with values that are calculated by means of the programming statements shown in the following SAS statements:

```
proc phreg data=Tumor;
   model Time*Dead(0)=Dose NPap;
   array pp{*} P1-P14;
   array tt{*} t1-t15;
   t1 = 27;
   t2 = 34;
   t3 = 37;
   t4 = 41;
   t5 = 43;
   t6 = 45;
   t7 = 46;
   t8 = 47;
   t9 = 49;
   t10= 50;
   t11= 51;
   t12= 53;
   t13= 65;
   t14= 67;
   t15= 71;
   if Time <  tt[1]  then NPap=0;
   else if time >= tt[15] then NPap=P15;
   else do i=1 to dim(pp);
      if tt[i] <= Time < tt[i+1] then NPap= pp[i];
   end;
run;
```

At each death time, the NPap value of each subject in the risk set is recalculated to reflect the actual number of papillomas at the given death time. For instance, subject one in the data set Tumor was in the risk sets at weeks 27 and 34; at week 27, the animal had no papilloma, while at week 34, it had five papillomas. Results of the analysis are shown in Output 54.5.1.

Output 54.5.1. Cox Regression Analysis on the Survival of Rodents

```
                            The PHREG Procedure

                             Model Information

                    Data Set                 WORK.TUMOR
                    Dependent Variable       Time
                    Censoring Variable       Dead
                    Censoring Value(s)       0
                    Ties Handling            BRESLOW


             Summary of the Number of Event and Censored Values

                                                     Percent
                   Total         Event     Censored    Censored

                      45            25           20       44.44


                             Convergence Status

               Convergence criterion (GCONV=1E-8) satisfied.


                           Model Fit Statistics

                                   Without           With
                  Criterion      Covariates     Covariates

                  -2 LOG L          166.793        143.269
                  AIC               166.793        147.269
                  SBC               166.793        149.707


                  Testing Global Null Hypothesis: BETA=0

          Test                    Chi-Square       DF     Pr > ChiSq

          Likelihood Ratio           23.5243        2         <.0001
          Score                      28.0498        2         <.0001
          Wald                       21.1646        2         <.0001


                  Analysis of Maximum Likelihood Estimates

                 Parameter     Standard                                  Hazard
Variable    DF    Estimate        Error    Chi-Square    Pr > ChiSq       Ratio

Dose         1     0.06885      0.05620        1.5010        0.2205       1.071
NPap         1     0.11714      0.02998       15.2705        <.0001       1.124
```

After the number of papillomas is adjusted for, the dose effect of the tumor-promoting agent is not statistically significant.

Another way to handle time-dependent repeated measurements in the PHREG procedure is to use the counting process style of input. Multiple records are created for each subject, one record for each distinct pattern of the time-dependent measurements. Each record contains a T1 value and a T2 value representing the time interval (T1,T2] during which the values of the explanatory variables remain unchanged. Each record also contains the censoring status at T2.

One advantage of using the counting process formulation is that you can easily obtain various residuals and influence statistics that are not available when programming statements are used to compute the values of the time-dependent variables. On the other hand, creating multiple records for the counting process formulation requires extra effort in data manipulation.

Consider a counting process style of input data set named Tumor1. It contains multiple observations for each subject in the data set Tumor. In addition to variables ID, Time, Dead, and Dose, four new variables are generated:

- T1 (left endpoint of the risk interval)
- T2 (right endpoint of the risk interval)
- NPap (number of papillomas in the time interval (T1,T2])
- Status (censoring status at T2)

For example, five observations are generated for the rodent that died at week 47 and that had no papilloma at week 27, five papillomas at week 34, six at week 37, eight at week 41, and 10 at weeks 43, 45, 46, and 47. The values of T1, T2, NPap, and Status for these five observations are (0,27,0,0), (27,34,5,0), (34,37,6,0), (37,41,8,0), and (41,47,10,1). Note that the variables ID, Time, and Dead are not needed for the estimation of the regression parameters, but they are useful for plotting the residuals.

The following SAS statements create the data set Tumor1:

```
data Tumor1(keep=ID Time Dead Dose T1 T2 NPap Status);
   array pp{*} P1-P14;
   array qq{*} P2-P15;
   array tt{1:15} _temporary_
      (27 34 37 41 43 45 46 47 49 50 51 53 65 67 71);
   set Tumor;
   T1 = 0;
   T2 = 0;
   Status = 0;
   if ( Time = tt[1] ) then do;
      T2 = tt[1];
      NPap = p1;
      Status = Dead;
      output;
   end;
   else do _i_=1 to dim(pp);
      if ( tt[_i_] = Time ) then do;
         T2= Time;
         NPap = pp[_i_] ;
         Status = Dead;
         output;
      end;
      else if (tt[_i_]  < Time ) then do;
         if (pp[_i_]  ^= qq[_i_] ) then do;
            if qq[_i_]  = . then T2= Time;
            else                 T2= tt[_i_] ;
            NPap= pp[_i_] ;
            Status= 0;
            output;
            T1 = T2;
         end;
      end;
   end;
   if ( Time >= tt[15] ) then do;
      T2 = Time;
      NPap = P15;
      Status = Dead;
      output;
   end;
   run;
```

In the following SAS statements, the counting process MODEL specification is used. The DFBETA statistics are output to a SAS data set named Out1. Note that Out1 contains multiple observations for each subject, that is, one observation for each risk interval (T1,T2].

```
proc phreg data=Tumor1;
   model (T1,T2)*Status(0)=Dose NPap;
   output out=Out1 resmart=mart dfbeta=db1-db2/order=data;
   id ID Time Dead;
run;
```

The output from PROC PHREG (not shown) is identical to Output 54.5.1 except for the “Summary of the Number of Event and Censored Values” table. The number of event observations remains unchanged between the two specifications of PROC PHREG, but the number of censored observations differs due to the splitting of each subject’s data into multiple observations for the counting process style of input.

Next, the MEANS procedure sums up the component statistics for each subject and outputs the results to a SAS data set named Out2.

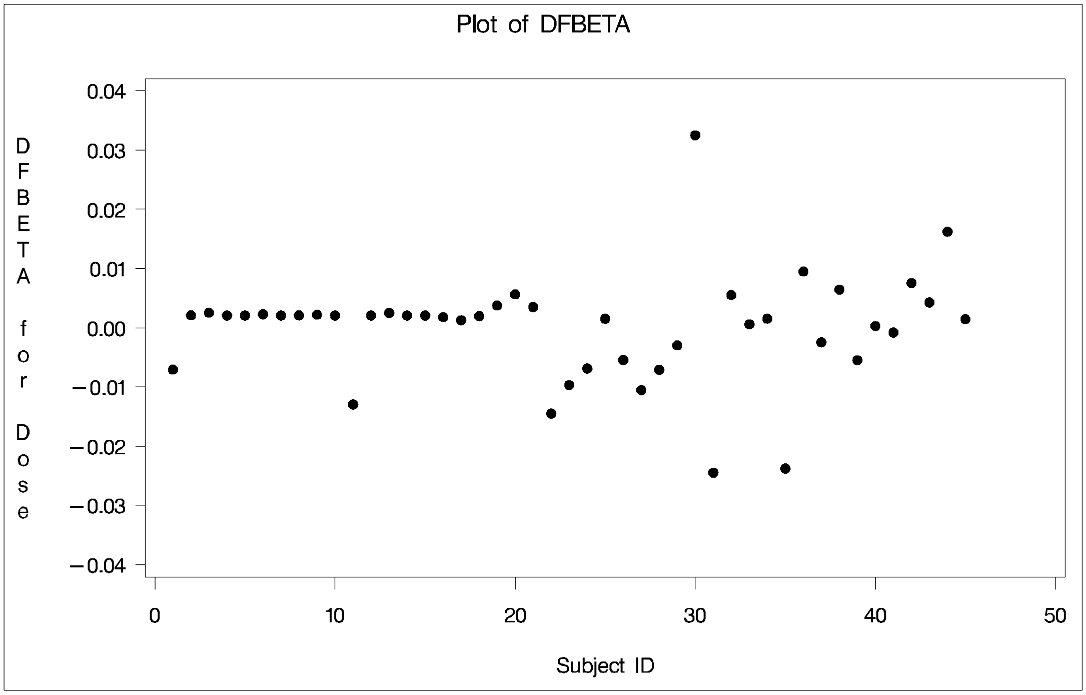
```
proc means data=Out1 noprint;
   by ID Time Dead;
   var mart db1-db2;
   output out=Out2 sum=mart db_dose db_npap;
run;
```

Finally, DFBETA statistics are plotted against subject ID for easy identification of influential points.

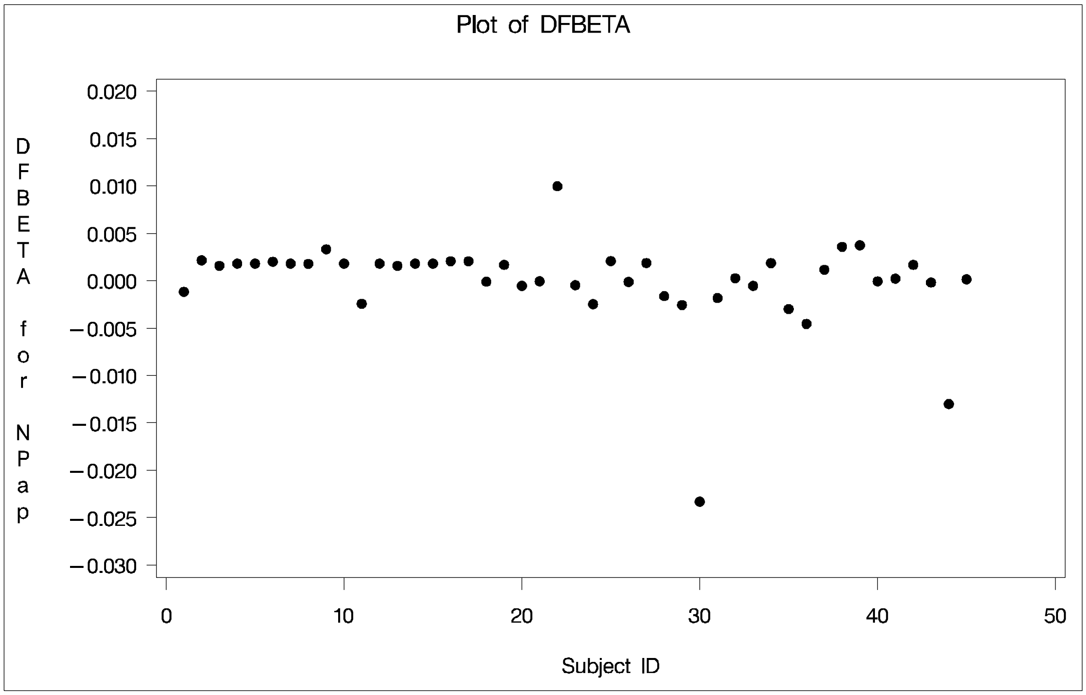
```
symbol1 v=dot h=0.8 c=blue;
axis1 label = (angle=-90 rotate=90 ’DFBETA for Dose’)
      minor = none
      order =  (-.04 to .04 by .01);
axis2 label = (angle=-90 rotate=90 ’DFBETA for NPap’)
      minor = none
      order =  (-.030 to .020 by .005);
title ’Plot of DFBETA’;
proc gplot data=Out2;
   plot db_dose * ID / frame hminor=0 vaxis=axis1 cframe=ligr;
   plot db_npap * ID / frame hminor=0 vaxis=axis2 cframe=ligr;
run;
```

The plots of the DFBETA statistics are shown in Output 54.5.2 and Output 54.5.3. Subject 30 appears to have a large influence on both the Dose and NPap coefficients. Subjects 31 and 35 have considerable influences on the DOSE coefficient, while subjects 22 and 44 have rather large influences on the NPap coefficient.

Output 54.5.2. Plot of DFBETA Statistic for DOSE versus Subject Number



Output 54.5.3. Plot of DFBETA Statistic for NPAP versus Subject Number



Example 54.6. Survivor Function Estimates for Specific Covariate Values

You may want to use your regression analysis results to generate predicted survival curves for subjects not in the study. This example illustrates how to use the BASELINE statement to obtain the survivor function for a new set of explanatory variable values. The various sets of explanatory variable values must be contained in a SAS data set.

In previous examples, LogBUN and HGB were identified as the most important prognostic factors for the myeloma data. Suppose you are interested in obtaining the survivor function estimates for the following two realizations of LogBUN and HGB, which are saved in a SAS data set called Inrisks.

```
data Inrisks;
   input LogBUN HGB;
   datalines;
1.00 10.0
1.80 12.0
;
```

In the BASELINE statement, you specify the name of the data set (COVARIATE=Inrisk) that contains the various sets of explanatory variable values and the name of the output SAS data set (OUT=Pred1) that contains the survivor function estimates. The option SURVIVAL=S puts the variable S containing the survivor function estimates in the output data set Pred1. Similarly, the options LOWER=S_lower and UPPER=S_upper put the variables S_lower and S_upper in Pred1; these variables contain, respectively, the lower and upper 95% confidence limits for the survival. The NOPRINT option in the PROC PHREG statement suppresses the displayed output (the analysis results are shown in Example 54.1). The PRINT procedure displays the observations in the data set Pred1.

```
proc phreg data=Myeloma noprint;
   model Time*VStatus(0)=LogBUN HGB;
   baseline covariates=Inrisks out=Pred1 survival=S
            lower=S_lower upper=S_upper;
run;
proc print data=Pred1;
run;
```

The first 32 observations of the data set Pred1 are shown in Output 54.6.1. They represent the survivor function for the realization LogBUN=1.00 and HGB=10.0. The first observation has survival time 0 and survivor function estimate 1.0. Each of the remaining 31 observations represents each unique event time in the input data set Myeloma. These observations are presented in ascending order of the event times. Likewise, the next 32 observations of the data set Pred1 (starting from the 33rd observation) represent the survivor function for the realization LogBUN=1.80 and HGB=12.0.

Output 54.6.1. Survivor Function Estimates for LogBUN=1.0 and HGB=10.0

```
                Log
          Obs   BUN    HGB     Time        S       S_lower     S_upper

            1    1      10      0.00    1.00000      .            .
            2    1      10      1.25    0.98622    0.96600     1.00000
            3    1      10      2.00    0.96438    0.92775     1.00000
            4    1      10      3.00    0.95687    0.91513     1.00000
            5    1      10      5.00    0.93966    0.88745     0.99494
            6    1      10      6.00    0.90211    0.83101     0.97929
            7    1      10      7.00    0.87192    0.78793     0.96487
            8    1      10      9.00    0.86073    0.77215     0.95947
            9    1      10     11.00    0.80252    0.69458     0.92725
           10    1      10     13.00    0.78969    0.67751     0.92044
           11    1      10     14.00    0.77554    0.65896     0.91274
           12    1      10     15.00    0.76116    0.64048     0.90458
           13    1      10     16.00    0.73142    0.60343     0.88654
           14    1      10     17.00    0.69988    0.56494     0.86706
           15    1      10     18.00    0.68345    0.54525     0.85667
           16    1      10     19.00    0.64951    0.50561     0.83438
           17    1      10     24.00    0.63105    0.48401     0.82278
           18    1      10     25.00    0.61267    0.46287     0.81096
           19    1      10     26.00    0.59428    0.44209     0.79887
           20    1      10     32.00    0.57437    0.41972     0.78601
           21    1      10     35.00    0.55400    0.39725     0.77258
           22    1      10     37.00    0.53276    0.37421     0.75849
           23    1      10     41.00    0.48783    0.32796     0.72564
           24    1      10     51.00    0.45964    0.29978     0.70476
           25    1      10     52.00    0.42933    0.27013     0.68234
           26    1      10     54.00    0.39588    0.23828     0.65773
           27    1      10     58.00    0.35744    0.20219     0.63191
           28    1      10     66.00    0.31314    0.16511     0.59386
           29    1      10     67.00    0.26060    0.12215     0.55597
           30    1      10     88.00    0.19554    0.07520     0.50849
           31    1      10     89.00    0.12708    0.03552     0.45460
           32    1      10     92.00    0.00000      .            .
```

By default, the procedure also outputs the set of survivor function estimates for LogBUN=1.3929 and HGB=10.2015, which are the sample means of LogBUN and HGB for the input data in Myeloma. (Note that in a stratified analysis, the sample means are calculated within each stratum.) The estimated survivor function estimates for these sample means are the last 32 observations in the data set Pred1. You can suppress this set of survival estimates by using the NOMEAN option in the BASELINE statement.

```
proc phreg data=Myeloma noprint;
   model Time*VStatus(0)=LogBUN HGB;
   baseline covariates=Inrisks out=Pred2 survival=S
            lower=S_lower upper=S_upper / nomean;
run;
```

The data set Pred2 consists of the first 64 observations of Pred1. If you are interested only in the survivor function estimates for the sample means of the explanatory variables, you can omit the COVARIATES= option in the BASELINE statement.

```
proc phreg data=Myeloma noprint;
   model Time*VStatus(0)=LogBUN HGB;
   baseline out=Pred3 survival=S lower=S_lower upper=S_upper;
run;
```

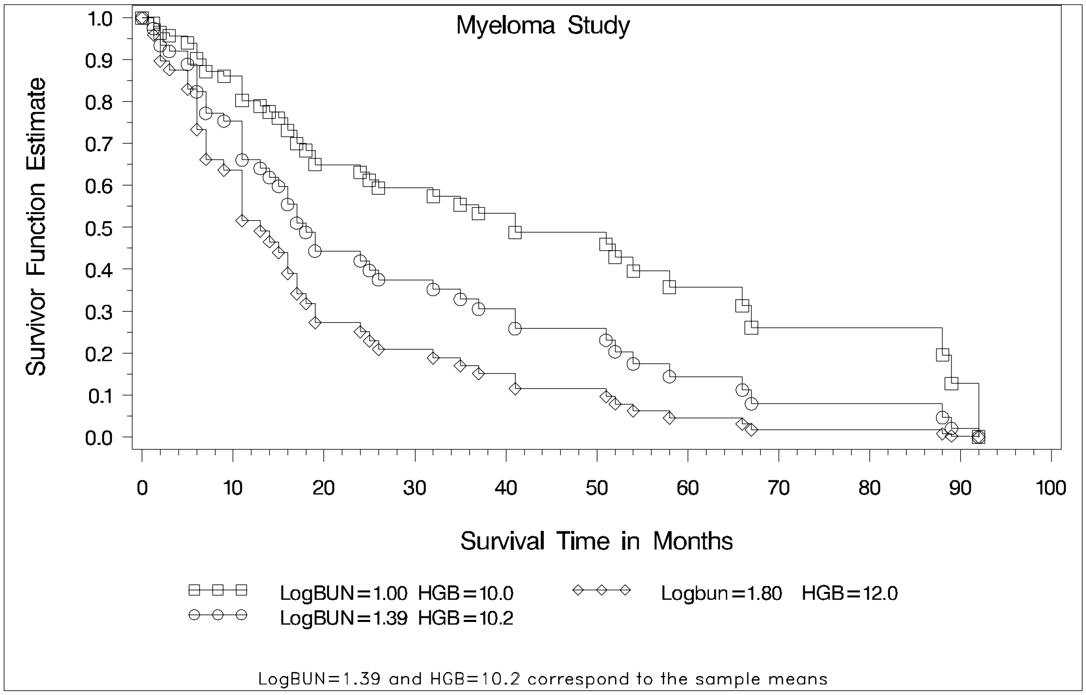
The data set Pred3 contains the last 32 observations of Pred1.

The following SAS statements are used to plot the survival curves in Pred1. For convenience, the variable Pattern is added to the data set Pred1 to identify the various patterns of explanatory variables.

```
data Pred1;
   set Pred1;
   if      LogBUN= 1.0 and HGB=10.0 then Pattern=1;
   else if LogBUN= 1.8 and HGB=12.0 then Pattern=2;
   else                                  Pattern=3;

legend1 label=none shape=symbol(3, .8)
   value=(f=swiss h=.8 ’LogBUN=1.00 HGB=10.0’
          ’LogBUN=1.80  HGB=12.0’ ’LogBUN=1.39 HGB=10.2’);
axis1 label=(h=1 f=swiss a=90) minor=(n=1);
axis2 label=(h=1 f=swiss ’Survival Time in Months’) minor=(n=4);


proc gplot data=Pred1;
   plot S*Time=Pattern / legend=legend1 vaxis=axis1
                         haxis=axis2    cframe=ligr;
   symbol1 interpol=stepLJ h=1 v=square  c=blue;
   symbol2 interpol=stepLJ h=1 v=diamond c=yellow;
   symbol3 interpol=stepLJ h=1 v=circle  c=red;
   note f=swiss h=1.5 j=c ’Myeloma Study’;
   footnote h=.8 f=duplex
      ’LogBUN=1.39 and HGB=10.2 correspond to the sample means’;
run;
```

The survivor function estimates for these three patterns of explanatory variables are displayed in Output 54.6.2. Note that these survivor functions are portrayed as right-continuous functions.

Output 54.6.2. Survival Curves for Specific Covariate Patterns



Example 54.7. Analysis of Residuals

Residuals are used to investigate the lack of fit of a model to a given subject. You can obtain martingale and deviance residuals for the Cox proportional hazards regression analysis by requesting that they be included in the OUTPUT data set. You can plot these statistics and look for outliers.

Consider the stepwise regression analysis performed in Example 54.1. The final model included variables LogBUN and HGB. You can generate residual statistics for this analysis by refitting the model containing those variables and including an OUTPUT statement. The keywords XBETA, RESMART, and RESDEV identify new variables that contain the linear predictor scores $\mathbf{z}_j'\widehat{\boldsymbol{\beta}}$, martingale residuals, and deviance residuals. These variables are xb, mart, and dev, respectively.

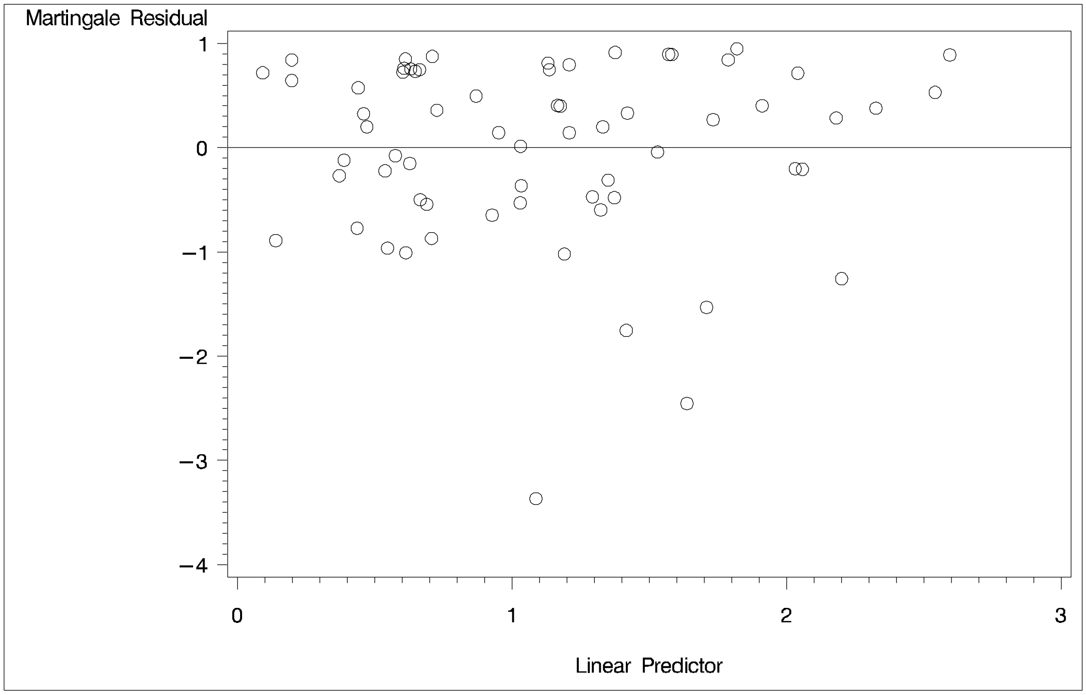
```
proc phreg data=Myeloma noprint;
   model Time*Vstatus(0)=LogBUN HGB;
   output out=Outp xbeta=xb resmart=mart resdev=dev;
run;
```

The following statements plot the residuals against the linear predictor scores:

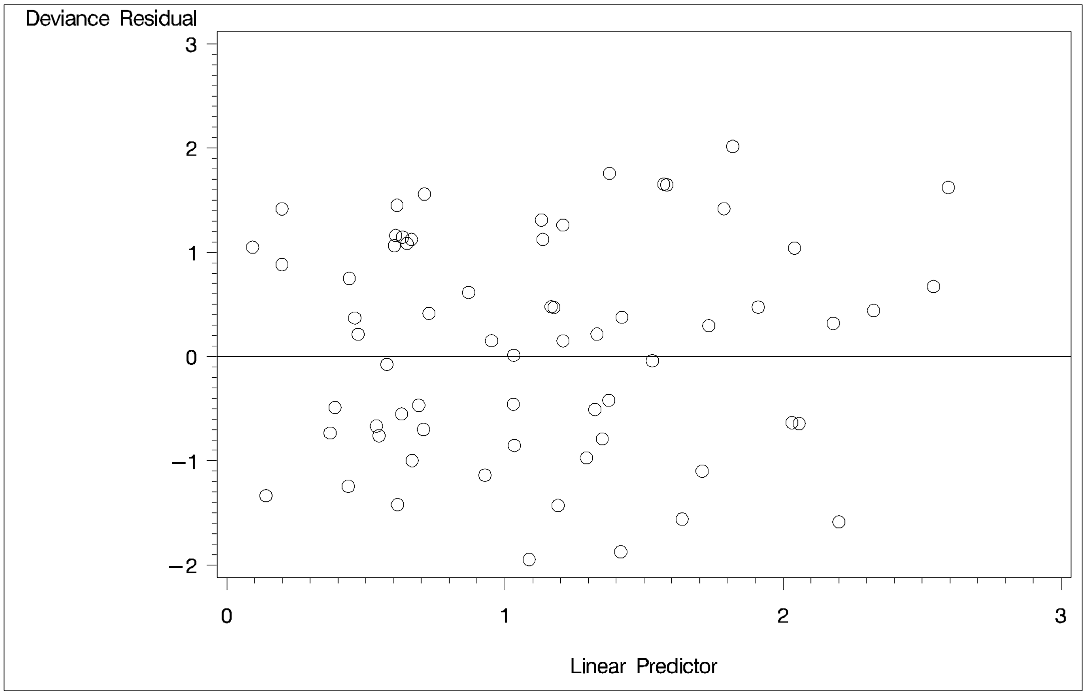
```
proc gplot data=Outp;
   plot (mart dev)*xb / vref=0 cframe=ligr;
   symbol1 value=circle c=blue;
run;
```

The resulting plots are shown in Output 54.7.1 and Output 54.7.2. The martingale residuals are skewed because of the single event setting of the Cox model. The martingale residual plot shows an isolation point (with linear predictor score 1.09 and martingale residual −3.37), but this observation is no longer distinguishable in the deviance residual plot. In conclusion, there is no indication of a lack of fit of the model to individual observations.

Output 54.7.1. Martingale Residual Plot



Output 54.7.2. Deviance Residual Plot



Example 54.8. Analysis of Recurrent Events Data

Recurrent events data consists of times to a number of repeated events for each sample unit; for example, times of recurrent episodes of a disease in patients. Various ways for analyzing recurrent events data are described in the section "Analysis of Multivariate Failure Time Data" on page 3247. The bladder cancer data listed in Wei, Lin, and Weissfeld (1989) is used here to illustrate these methods.

The data consist of 86 patients with superficial bladder tumors, which were removed when the patients entered the study. Of these patients, 48 were randomized into the placebo group, and 38 were randomized into the thiotepa group. Many patients had multiple recurrences of tumors during the study, and new tumors were removed at each visit. The data set contains the first four recurrences of the tumor for each patient, and each recurrence time was measured from the patient's entry time into the study.

The data consist of the following eight variables:

- Trt, treatment group (1=placebo and 2=thiotepa)
- Time, follow-up time
- Number, number of initial tumors
- Size, initial tumor size
- T1, T2, T3, and T4, times of the four potential recurrences of the bladder tumor. A patient with only two recurrences has missing values in T3 and T4.

In the data set Bladder, four observations are created for each patient, one for each of the four potential tumor recurrences. In addition to values of Trt, Number, and Size for the patient, each observation contains the following variables:

- ID, patient's identification (which is the sequence number of the subject)
- Visit, visit number (with value k for the kth potential tumor recurrence)
- TStart, time of the (k-1)th recurrence for Visit=k, or the entry time 0 if VISIT=1, or the follow-up time if the (k-1)th recurrence does not occur
- TStop, time of the kth recurrence if Visit=k or follow-up time if the kth recurrence does not occur
- Status, event status of TStop (1=recurrence and 0=censored)

For instance, a patient with only one recurrence time at month 6, who was followed until month 10, will have values for Visit, TStart, TStop, and Status of (1,0,6,1), (2,6,10,0), (3,10,10,0), and (4,10,10,0). The last two observations are redundant for the intensity model and the proportional means model, but they are important for the analysis of the marginal Cox models. If the follow-up time is beyond the time of the fourth turmor recurrence, it is tempting to create a fifth observation with the time of the fourth tumor recurrence as the TStart value, the follow-up time as the TStop value, and a Status value of 0. However, Therneau and Grambsch (2000), Section 8.5) have warned against incorporating such observations into the analysis.

The following SAS statements create the data set Bladder:

```
data Bladder;
   keep ID TStart TStop Status Trt Number Size Visit;
   retain ID TStart 0;
   array tt T1-T4;
   infile datalines missover;
   input Trt Time Number Size T1-T4;
   ID + 1;
   TStart=0;
   do over tt;
      Visit=_i_;
      if tt = . then do;
         TStop=Time;
         Status=0;
      end;
      else do;
         TStop=tt;
         Status=1;
      end;
      output;
      TStart=TStop;
   end;
   if (TStart < Time) then delete;
   datalines;
1       0       1     1
1       1       1     3
```

```
1       4       2     1
1       7       1     1
1       10      5     1
1       10      4     1   6
1       14      1     1
1       18      1     1
1       18      1     3   5
1       18      1     1   12  16
1       23      3     3
1       23      1     3   10  15
1       23      1     1   3   16  23
1       23      3     1   3   9   21
1       24      2     3   7   10  16  24
1       25      1     1   3   15  25
1       26      1     2
1       26      8     1   1
1       26      1     4   2   26
1       28      1     2   25
1       29      1     4
1       29      1     2
1       29      4     1
1       30      1     6   28  30
1       30      1     5   2   17  22
1       30      2     1   3   6   8   12
1       31      1     3   12  15  24
1       32      1     2
1       34      2     1
1       36      2     1
1       36      3     1   29
1       37      1     2
1       40      4     1   9   17  22  24
1       40      5     1   16  19  23  29
1       41      1     2
1       43      1     1   3
1       43      2     6   6
1       44      2     1   3   6   9
1       45      1     1   9   11  20  26
1       48      1     1   18
1       49      1     3
1       51      3     1   35
1       53      1     7   17
1       53      3     1   3   15  46  51
1       59      1     1
1       61      3     2   2   15  24  30
1       64      1     3   5   14  19  27
1       64      2     3   2   8   12  13
2       1       1     3
2       1       1     1
2       5       8     1   5
2       9       1     2
2       10      1     1
2       13      1     1
2       14      2     6   3
2       17      5     3   1   3   5   7
```

```
2       18      5     1
2       18      1     3   17
2       19      5     1   2
2       21      1     1   17   19
2       22      1     1
2       25      1     3
2       25      1     5
2       25      1     1
2       26      1     1   6    12   13
2       27      1     1   6
2       29      2     1   2
2       36      8     3   26   35
2       38      1     1
2       39      1     1   22   23   27   32
2       39      6     1   4    16   23   27
2       40      3     1   24   26   29   40
2       41      3     2
2       41      1     1
2       43      1     1   1    27
2       44      1     1
2       44      6     1   2    20   23   27
2       45      1     2
2       46      1     4   2
2       46      1     4
2       49      3     3
2       50      1     1
2       50      4     1   4    24   47
2       54      3     4
2       54      2     1   38
2       59      1     3
;
run;
```

First, consider fitting the intensity model (Andersen and Gill 1982) and the proportional means model (Lin et al. 2000). The counting process style of input is used in the PROC PHREG specification. For the proportional means model, inference is based on the robust sandwich covariance estimate, which is requested by the COVB(AGGREGATE) option in the PROC PHREG statement. The COVM option is specified for the analysis of the intensity model to use the model-based covariance estimate. Note that some of the observations in the data set Bladder have a degenerated interval of risk. The presence of these observations does not affect the results of the analysis since none of these observations are included in any of the risk sets. However, the procedure will run more efficiently without these observations; consequently, in the following SAS statements, the WHERE clause is used to eliminate these redundant observations.

```
title ’Intensity Model and Proportional Means Model’;
proc phreg data=Bladder covm covs(aggregate);
   model (TStart, TStop) * Status(0) = Trt Number Size;
   id id;
   where TStart < TStop;
run;
```

Results of fitting the intensity model and the proportional means model are shown in Output 54.8.1 and Output 54.8.2, respectively. The robust sandwich standard error estimate for Trt is larger than its model-based counterpart, rendering the effect of thiotepa less significant in the proportional means model (p=0.0747) than in the intensity model (p=0.0215).

Output 54.8.1. Analysis of the Intensity Model

```
                              The PHREG Procedure

                    Analysis of Maximum Likelihood Estimates
                       with Model-Based Variance Estimate

                      Parameter     Standard                                Hazard
Variable      DF       Estimate        Error    Chi-Square    Pr > ChiSq     Ratio

Trt            1       -0.45979      0.19996        5.2873        0.0215     0.631
Number         1        0.17165      0.04733       13.1541        0.0003     1.187
Size           1       -0.04256      0.06903        0.3801        0.5375     0.958
```

Output 54.8.2. Analysis of the Proportional Means Model

```
                    Analysis of Maximum Likelihood Estimates
                        with Sandwich Variance Estimate

                   Parameter    Standard   StdErr                                Hazard
Variable    DF      Estimate       Error    Ratio   Chi-Square   Pr > ChiSq     Ratio

Trt          1      -0.45979     0.25801    1.290       3.1757       0.0747     0.631
Number       1       0.17165     0.06131    1.296       7.8373       0.0051     1.187
Size         1      -0.04256     0.07555    1.094       0.3174       0.5732     0.958
```

Next, consider the conditional models of PWP (Prentice, Williams, and Peterson 1981). In the PWP models, the risk set for the (k+1)th recurrence is restricted to those patients who have experienced the first k recurrences. For example, a patient who experienced only one recurrence is an event observation for the first recurrence; this patient is a censored observation for the second recurrence and should not be included in the risk set for the third or fourth recurrence. The following DATA step eliminates those observations that should not be in the risk sets, forming a new input data set (named Bladder2) for fitting the PWP models. The variable Gaptime, represented the gap times between successive recurrences, is also created.

```
data Bladder2(drop=LastStatus);
    retain LastStatus;
    set Bladder;
    by ID;
    if first.id then LastStatus=1;
    if (Status=0 and LastStatus=0) then delete;
    LastStatus=Status;
    Gaptime=Tstop-Tstart;
run;
```

The following statements fit the PWP total time model. Variables Trt1, Trt2, Trt3, and Trt4 are visit-specific variables for Trt; variables Number1, Number2, Numvber3, and Number4 are visit-specific variables for Number; and variables Size1, Size2, Size3, and Size4 are visit-specific variables for Size.

```
title 'PWP Total Time Model with Noncommon Effects';
proc phreg data=Bladder2;
   model (TStart,Tstop) * Status(0) = Trt1-Trt4 Number1-Number4
                                      Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
   Size4= Size * (Visit=4);
   strata Visit;
run;
```

Results of the analysis of the PWP total time model are shown in Output 54.8.3. Note that patients who were at risk for a second recurrence are those who had a first recurrence. There is no significant treatment effect on the total time in any of the four tumor recurrences.

Output 54.8.3. Analysis of the PWP Total Time Model with Noncommon Effects

```
                              The PHREG Procedure

               Summary of the Number of Event and Censored Values

                                                                Percent
       Stratum    Visit         Total       Event    Censored   Censored

             1    1                85          47          38      44.71
             2    2                46          29          17      36.96
             3    3                27          22           5      18.52
             4    4                20          14           6      30.00
       -------------------------------------------------------------------
         Total                    178         112          66      37.08


                    Analysis of Maximum Likelihood Estimates

                    Parameter     Standard                                  Hazard
   Variable   DF     Estimate        Error   Chi-Square   Pr > ChiSq        Ratio

   Trt1        1     -0.51757      0.31576       2.6868       0.1012        0.596
   Trt2        1     -0.45967      0.40642       1.2792       0.2581        0.631
   Trt3        1      0.11700      0.67183       0.0303       0.8617        1.124
   Trt4        1     -0.04059      0.79251       0.0026       0.9592        0.960
   Number1     1      0.23605      0.07607       9.6287       0.0019        1.266
   Number2     1     -0.02044      0.09052       0.0510       0.8213        0.980
   Number3     1      0.01219      0.18208       0.0045       0.9466        1.012
   Number4     1      0.18915      0.24443       0.5989       0.4390        1.208
   Size1       1      0.06790      0.10125       0.4498       0.5024        1.070
   Size2       1     -0.15425      0.12300       1.5728       0.2098        0.857
   Size3       1      0.14891      0.26299       0.3206       0.5713        1.161
   Size4       1    0.0000732      0.34297       0.0000       0.9998        1.000
```

The following statements fit the PWP gap-time model.

```
title 'PWP Gap Time Model with Noncommon Effects';
proc phreg data=Bladder2;
   model Gaptime * Status(0) = Trt1-Trt4 Number1-Number4
                               Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
   Size4= Size * (Visit=4);
   strata Visit;
run;
```

Results of the analysis of the PWP gap-time model are shown in Output 54.8.4. Note that the regression coefficients for the first tumor recurrence are the same as those of the total time model, since the total time and and the gap time are the same for the first recurrence. There is no significant treatment effect on the gap times for any of the four tumor recurrences.

Output 54.8.4. Analysis of the PWP Gap Time Model with Noncommon Effects

```
                              The PHREG Procedure

                    Analysis of Maximum Likelihood Estimates

                         Parameter     Standard                                 Hazard
Variable      DF         Estimate         Error    Chi-Square    Pr > ChiSq      Ratio

Trt1           1         -0.51757       0.31576        2.6868        0.1012      0.596
Trt2           1         -0.25911       0.40511        0.4091        0.5224      0.772
Trt3           1          0.22105       0.54909        0.1621        0.6873      1.247
Trt4           1         -0.19498       0.64184        0.0923        0.7613      0.823
Number1        1          0.23605       0.07607        9.6287        0.0019      1.266
Number2        1         -0.00571       0.09667        0.0035        0.9529      0.994
Number3        1          0.12935       0.15970        0.6561        0.4180      1.138
Number4        1          0.42079       0.19816        4.5091        0.0337      1.523
Size1          1          0.06790       0.10125        0.4498        0.5024      1.070
Size2          1         -0.11636       0.11924        0.9524        0.3291      0.890
Size3          1          0.24995       0.23113        1.1695        0.2795      1.284
Size4          1          0.03557       0.29043        0.0150        0.9025      1.036
```

You can fit the PWP total time model with common effects with the following SAS statements. However, the analysis is not shown here.

```
title2 'PWP Total Time Model with Common Effects';
proc phreg data=Bladder2;
   model (tstart,tstop) * status(0) = Trt Number Size;
   strata Visit;
run;
```

You can fit the PWP gap time model with common effects with the following statements. Again, the analysis is not shown here.

```
title2 'PWP Gap Time Model with Common Effects';
proc phreg data=Bladder2;
   model Gaptime * Status(0) = Trt Number Vize;
   strata Visit;
run;
```

Recurrent events data are a special case of multiple events data in which the recurrence times are regarded as multivariate failure times and the marginal approach of WLW (Wei, Lin, and Weissfeld 1989) can be used. WLW fits a Cox model to each of the component times and makes statistical inference of the regression parameters based on a robust sandwich covariance matrix estimate. No specific correlation structure is imposed on the multivariate failure times. For the kth marginal model, let $\boldsymbol{\beta}_k$ denote the row vector of regression parameters, let $\hat{\boldsymbol{\beta}}_k$ denote the maximum likelihood estimate of $\boldsymbol{\beta}_k$, let $\hat{\mathbf{A}}_k$ denote the covariance matrix obtained by inverting the observed information matrix, and let $\mathbf{R}_k$ denote the matrix of score residuals. WLW showed that the joint distribution of $(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_4)'$ can be approximated by a multivariate normal distribution with mean vector $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_4)'$ and robust covariance matrix

$$\begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} & \mathbf{V}_{13} & \mathbf{V}_{14} \\ \mathbf{V}_{21} & \mathbf{V}_{22} & \mathbf{V}_{23} & \mathbf{V}_{24} \\ \mathbf{V}_{31} & \mathbf{V}_{32} & \mathbf{V}_{33} & \mathbf{V}_{34} \\ \mathbf{V}_{41} & \mathbf{V}_{42} & \mathbf{V}_{43} & \mathbf{V}_{44} \end{pmatrix}$$

with the submatrix $\mathbf{V}_{ij}$ given by

$$\mathbf{V}_{ij} = \hat{\mathbf{A}}_i(\mathbf{R}_i'\mathbf{R}_j)\hat{\mathbf{A}}_j$$

In this example, there are four marginal proportional hazards models, one for each potential recurrence time. Instead of fitting one model at a time, you can fit all four marginal models in one analysis by using the STRATA statement and model-specific covariates. Using the Visit as the STRATA variable on the input data set Bladder, PROC PHREG simultaneously fits all four marginal models, one for each Visit value. The COVS(AGGREGATE) option is specified to compute the robust sandwich variance estimate by summing up the score residuals for each distinct pattern of ID value. The TEST statement TREATMENT is used to perform the global test of no treatment effect for each tumor recurrence, the AVERAGE option is specified to estimate the parameter for the common treatment effect, and the E option displays the optimal weights for the common treatment effect.

```
proc phreg data=Bladder covs(aggregate);
   model TStop*Status(0)=Trt1-Trt4 Number1-Number4 Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
   Size4= Size * (Visit=4);
   strata Visit;
   id ID;
   TREATMENT: test trt1,trt2,trt3,trt4/average e;
   run;
```

Out of the 86 patients, 47 patients have only one tumor recurrence, 29 patients have two recurrences, 22 patients have three recurrences, and 14 patients have four recurrences (Output 54.8.5). Parameter estimates for the four marginal models are shown in Output 54.8.6. The 4 DF Wald test (Output 54.8.7) indicates a lack of evidence of a treatment effect in any of the four recurrences (p=0.4105). The optimal weights for estimating the parameter of the common treatment effect are 0.67684, 0.25723, -0.07547, and 0.14140 for Trt1, Trt2, Trt3, and Trt4, respectively, which gives a parameter estimate of -0.5489 with a standard error estimate of 0.2853. A more sensitive test for a treatment effect is the 1 DF test based on this common parameter; however, there is still insufficient evidence for such effect at the 0.05 level (p=0.0543).

Output 54.8.5. Summary of Bladder Tumor Recurrences in 86 Patients

```
                             The PHREG Procedure

                Summary of the Number of Event and Censored Values

                                                                 Percent
    Stratum     Visit          Total        Event     Censored    Censored

          1     1                 86           47           39       45.35
          2     2                 86           29           57       66.28
          3     3                 86           22           64       74.42
          4     4                 86           14           72       83.72
    -------------------------------------------------------------------
      Total                      344          112          232       67.44
```

Output 54.8.6. Analysis of Marginal Cox Models

```
                        Analysis of Maximum Likelihood Estimates

                     Parameter    Standard   StdErr                              Hazard
  Variable   DF       Estimate       Error    Ratio   Chi-Square   Pr > ChiSq     Ratio

  Trt1        1       -0.51762     0.30750    0.974       2.8336       0.0923     0.596
  Trt2        1       -0.61944     0.36391    0.926       2.8975       0.0887     0.538
  Trt3        1       -0.69988     0.41516    0.903       2.8419       0.0918     0.497
  Trt4        1       -0.65079     0.48971    0.848       1.7661       0.1839     0.522
  Number1     1        0.23599     0.07208    0.947      10.7204       0.0011     1.266
  Number2     1        0.13756     0.08690    0.946       2.5059       0.1134     1.147
  Number3     1        0.16984     0.10356    0.984       2.6896       0.1010     1.185
  Number4     1        0.32880     0.11382    0.909       8.3453       0.0039     1.389
  Size1       1        0.06789     0.08529    0.842       0.6336       0.4260     1.070
  Size2       1       -0.07612     0.11812    0.881       0.4153       0.5193     0.927
  Size3       1       -0.21131     0.17198    0.943       1.5097       0.2192     0.810
  Size4       1       -0.20317     0.19106    0.830       1.1308       0.2876     0.816
```

Output 54.8.7. Tests of Treatment Effects

```
                    Linear Coefficients for Test TREATMENT

                                                                      Average
    Parameter         Row1         Row2         Row3         Row4      Effect

    Trt1                 1            0            0            0     0.67684
    Trt2                 0            1            0            0     0.25723
    Trt3                 0            0            1            0    -0.07547
    Trt4                 0            0            0            1     0.14140
    Number1              0            0            0            0     0.00000
    Number2              0            0            0            0     0.00000
    Number3              0            0            0            0     0.00000
    Number4              0            0            0            0     0.00000
    Size1                0            0            0            0     0.00000
    Size2                0            0            0            0     0.00000
    Size3                0            0            0            0     0.00000
    Size4                0            0            0            0     0.00000
    CONSTANT             0            0            0            0     0.00000


                            Test TREATMENT Results

                              Wald
                        Chi-Square      DF    Pr > ChiSq

                            3.9668       4        0.4105


                        Average Effect for Test TREATMENT

                                 Standard
                 Estimate           Error       z-Score    Pr > |z|

                  -0.5489          0.2853       -1.9240      0.0543
```

Example 54.9. Analysis of Clustered Data

When experimental units are naturally or artificially clustered, failure times of experimental units within a cluster are correlated. Lee, Wei, and Amato (1992) estimate the regression parameters in the Cox model by the maximum partial likelihood estimates under an independent working assumption and use a robust sandwich covariance matrix estimate to account for the intracluster dependence. A subset of data from the Diabetic Retinopathy Study (DRS) is used to illustrate the methodology as in Lin (1994).

The data consist of 197 diabetic patients who have a high risk of experiencing blindness in both eyes as defined by DRS criteria. One eye of each patient is treated with laser photocoagulation. The hypothesis of interest is whether the laser treatment delays the occurrence of blindness. Since juvenile and adult diabetes have very different courses, it is also desirable to examine how the age of onset of diabetes may affect the time of blindness. Since there are no biological differences between the left eye and the right eye, it is natural to assume a common baseline hazard function for the failure times of the left and the right eyes.

Each patient is a cluster that contributes two observations to the input data set, one for each eye. The following variables are in the input data set Blind:

- ID, patient's identification
- Time, failure time
- Status, event indicator (0=censored and 1=uncensored)
- Treatment, treatment received (1=laser photocoagulation and 0=otherwise)
- DiabeticType, type of diabetes (0=juvenile onset with age of onset at 20 or under, and 1= adult onset with age of onset over 20)

```
data Blind;
   input ID Time Status DiabeticType Treatment @@;
   datalines;
    5 46.23 0 1 1    5 46.23 0 1 0   14 42.50 0 0 1   14 31.30 1 0 0
   16 42.27 0 0 1   16 42.27 0 0 0   25 20.60 0 0 1   25 20.60 0 0 0
   29 38.77 0 0 1   29  0.30 1 0 0   46 65.23 0 0 1   46 54.27 1 0 0
   49 63.50 0 0 1   49 10.80 1 0 0   56 23.17 0 0 1   56 23.17 0 0 0
   61  1.47 0 0 1   61  1.47 0 0 0   71 58.07 0 1 1   71 13.83 1 1 0
  100 46.43 1 1 1  100 48.53 0 1 0  112 44.40 0 1 1  112  7.90 1 1 0
  120 39.57 0 1 1  120 39.57 0 1 0  127 30.83 1 1 1  127 38.57 1 1 0
  133 66.27 0 1 1  133 14.10 1 1 0  150 20.17 1 0 1  150  6.90 1 0 0
  167 58.43 0 1 1  167 41.40 1 1 0  176 58.20 0 0 1  176 58.20 0 0 0
  185 57.43 0 1 1  185 57.43 0 1 0  190 56.03 0 0 1  190 56.03 0 0 0
  202 67.53 0 0 1  202 67.53 0 0 0  214 61.40 0 1 1  214  0.60 1 1 0
  220 10.27 1 0 1  220  1.63 1 0 0  243 66.20 0 0 1  243 66.20 0 0 0
  255  5.67 1 0 1  255 13.83 1 0 0  264 58.83 0 0 1  264 29.97 1 0 0
  266 60.27 0 1 1  266 26.37 1 1 0  284  5.77 1 1 1  284  1.33 1 1 0
  295  5.90 1 0 1  295 35.53 1 0 0  300 25.63 1 1 1  300 21.90 1 1 0
  302 33.90 1 0 1  302 14.80 1 0 0  315  1.73 1 0 1  315  6.20 1 0 0
  324 46.90 0 1 1  324 22.00 1 1 0  328 31.13 0 0 1  328 31.13 0 0 0
  335 30.20 1 0 1  335 22.00 1 0 0  342 70.90 0 0 1  342 70.90 0 0 0
  349 25.80 1 1 1  349 13.87 1 1 0  357  5.73 1 1 1  357 48.30 1 1 0
  368 53.43 0 0 1  368 53.43 0 0 0  385  1.90 1 0 1  385 51.10 0 0 0
  396  9.90 1 1 1  396  9.90 1 1 0  405 34.20 0 0 1  405 34.20 0 0 0
  409 46.73 0 1 1  409  2.67 1 1 0  419 18.73 0 1 1  419 13.83 1 1 0
  429 32.03 0 1 1  429  4.27 1 1 0  433 69.87 0 1 1  433 13.90 1 1 0
  445 66.80 0 0 1  445 66.80 0 0 0  454 64.73 0 0 1  454 64.73 0 0 0
  468  1.70 1 0 1  468  1.70 1 0 0  480  1.77 1 0 1  480 43.03 1 0 0
  485 29.03 0 0 1  485 29.03 0 0 0  491 56.57 0 1 1  491 56.57 0 1 0
  503  8.30 1 1 1  503  8.30 1 1 0  515 21.57 0 1 1  515 18.43 1 1 0
  522 31.57 0 0 1  522 31.57 0 0 0  538 31.63 0 1 1  538 31.63 1 1 0
  547 39.77 0 1 1  547 39.77 0 1 0  550 18.70 1 0 1  550  6.53 1 0 0
  554 18.90 0 0 1  554 18.90 0 0 0  557 56.80 0 0 1  557 22.23 1 0 0
  561 55.60 0 0 1  561 14.00 1 0 0  568 42.17 1 0 1  568 42.17 1 0 0
  572 10.70 0 0 1  572  5.33 1 0 0  576 66.33 0 0 1  576 59.80 1 0 0
  581 52.33 0 1 1  581  5.83 1 1 0  606 58.17 0 0 1  606  2.17 1 0 0
  610 14.30 1 0 1  610 48.43 1 0 0  615 25.83 0 0 1  615 25.83 0 0 0
  618 45.40 0 0 1  618 45.40 0 0 0  624 47.60 0 0 1  624 47.60 0 0 0
  631 13.33 1 0 1  631  9.60 1 0 0  636 42.10 0 0 1  636 42.10 0 0 0
  645 39.93 0 0 1  645 39.93 0 0 0  653 14.27 1 0 1  653  7.60 1 0 0
  662 34.57 1 0 1  662  1.80 1 0 0  664 65.80 0 0 1  664  4.30 1 0 0
  683  4.10 1 1 1  683 12.20 1 1 0  687 60.93 0 0 1  687 60.93 0 0 0
  701 57.20 0 0 1  701 57.20 0 0 0  706 38.07 0 1 1  706 12.73 1 1 0
  717 54.10 0 1 1  717 54.10 1 1 0  722 59.27 0 0 1  722  9.40 1 0 0
```

```
 731 21.57 1 0 1  731  9.90 1 0 0  740 54.10 0 0 1  740 54.10 0 0 0
 749 50.47 0 1 1  749 50.47 0 1 0  757 46.17 0 0 1  757 46.17 0 0 0
 760 46.30 0 0 1  760 46.30 0 0 0  766 38.83 0 1 1  766 38.83 0 1 0
 769 44.60 0 0 1  769 44.60 0 0 0  772 43.07 0 0 1  772 43.07 0 0 0
 778 26.23 1 1 1  778 40.03 0 1 0  780 41.60 0 0 1  780 18.03 1 0 0
 793 38.07 0 1 1  793 38.07 0 1 0  800 65.23 0 1 1  800 65.23 0 1 0
 804  7.07 1 1 1  804 66.77 0 1 0  810 13.77 1 0 1  810 13.77 1 0 0
 815  9.63 0 1 1  815  9.63 1 1 0  832 46.23 0 0 1  832 46.23 0 0 0
 834 45.73 0 0 1  834  1.50 1 0 0  838 33.63 1 1 1  838 33.63 1 1 0
 857 40.17 0 0 1  857 40.17 0 0 0  866 63.33 1 1 1  866 27.60 1 1 0
 887 38.47 1 1 1  887  1.63 1 1 0  903 55.23 0 1 1  903 55.23 0 1 0
 910 52.77 0 1 1  910 25.30 1 1 0  920 57.17 0 0 1  920 46.20 1 0 0
 925  9.87 0 1 1  925  1.70 1 1 0  931 57.90 0 0 1  931 57.90 0 0 0
 936  5.90 0 0 1  936  5.90 0 0 0  945 32.20 0 0 1  945 32.20 0 0 0
 949 10.33 1 0 1  949  0.83 1 0 0  952  6.13 1 0 1  952 50.90 0 0 0
 962 43.67 0 0 1  962 25.93 1 0 0  964 38.30 0 0 1  964 38.30 0 0 0
 971 38.77 0 1 1  971 19.40 1 1 0  978 38.07 0 0 1  978 21.97 1 0 0
 983 38.30 0 0 1  983 38.30 0 0 0  987 26.20 1 0 1  987 70.03 0 0 0
1002 62.57 0 0 1 1002 18.03 1 0 0 1017 13.83 1 1 1 1017  1.57 1 1 0
1029 46.50 0 1 1 1029 13.37 1 1 0 1034 11.07 1 0 1 1034  1.97 1 0 0
1037 42.47 0 1 1 1037 22.20 1 1 0 1042 38.73 0 1 1 1042 38.73 0 1 0
1069 51.13 0 1 1 1069 51.13 0 1 0 1074  6.10 1 0 1 1074 46.50 0 0 0
1098  2.10 1 0 1 1098 11.30 1 0 0 1102 17.73 1 0 1 1102 42.30 0 0 0
1112 26.47 0 0 1 1112 26.47 0 0 0 1117 10.77 0 0 1 1117 10.77 0 0 0
1126 55.33 0 1 1 1126 55.33 0 1 0 1135 58.67 0 0 1 1135 58.67 0 0 0
1145 12.93 1 1 1 1145  4.97 1 1 0 1148 54.20 0 1 1 1148 26.47 1 1 0
1167 49.57 0 0 1 1167 49.57 0 0 0 1184 24.43 1 1 1 1184  9.87 1 1 0
1191 50.23 0 1 1 1191 50.23 0 1 0 1205 13.97 1 0 1 1205 30.40 1 0 0
1213 43.33 0 0 1 1213 43.33 1 0 0 1228 42.23 0 1 1 1228 42.23 0 1 0
1247 74.93 0 0 1 1247 74.93 0 0 0 1250 66.93 0 1 1 1250 66.93 0 1 0
1253 73.43 0 0 1 1253 73.43 0 0 0 1267 67.47 0 1 1 1267 38.57 1 1 0
1281  3.67 0 1 1 1281  3.67 1 1 0 1287 48.87 1 0 1 1287 67.03 0 0 0
1293 65.60 0 0 1 1293 65.60 0 0 0 1296 15.83 0 0 1 1296 15.83 1 0 0
1309 20.07 0 1 1 1309  8.83 1 1 0 1312 67.43 0 0 1 1312 67.43 0 0 0
1317  1.47 0 0 1 1317  1.47 0 0 0 1321 62.93 0 0 1 1321 22.13 1 0 0
1333  6.30 1 0 1 1333 56.97 0 0 0 1347 59.70 0 0 1 1347 18.93 1 0 0
1361 13.80 1 0 1 1361 19.00 1 0 0 1366 55.13 0 1 1 1366 55.13 0 1 0
1373 13.57 1 0 1 1373  5.43 1 0 0 1397 42.20 0 1 1 1397 42.20 0 1 0
1410 38.27 0 1 1 1410 38.27 0 1 0 1413  7.10 0 0 1 1413  7.10 1 0 0
1425 63.63 0 1 1 1425 26.17 1 1 0 1447 59.00 0 0 1 1447 24.73 1 0 0
1461 54.37 0 1 1 1461 54.37 0 1 0 1469 54.60 0 1 1 1469 10.97 1 1 0
1480 63.87 0 1 1 1480 21.10 1 1 0 1487 62.37 0 1 1 1487 43.70 1 1 0
1491 62.80 0 1 1 1491 62.80 0 1 0 1499 63.33 0 1 1 1499 14.37 1 1 0
1503 58.53 0 1 1 1503 58.53 0 1 0 1513 58.07 0 1 1 1513 58.07 0 1 0
1524 58.50 0 1 1 1524 58.50 0 1 0 1533  1.50 1 1 1 1533 14.37 0 1 0
1537 54.73 0 0 1 1537 38.40 1 0 0 1552 50.63 0 0 1 1552  2.83 1 0 0
1554 51.10 0 1 1 1554 51.10 0 1 0 1562 49.93 0 1 1 1562  6.57 1 1 0
1572 46.27 0 1 1 1572 46.27 1 1 0 1581 10.60 0 1 1 1581 10.60 0 1 0
1585 42.77 0 1 1 1585 42.77 0 1 0 1596 34.37 1 0 1 1596 42.27 0 0 0
1600 42.07 0 0 1 1600 42.07 0 0 0 1603 38.77 0 0 1 1603 38.77 0 0 0
1619 74.97 0 1 1 1619 61.83 1 1 0 1627  6.57 1 0 1 1627 66.97 0 0 0
1636 38.87 1 0 1 1636 68.30 0 0 0 1640 42.43 1 0 1 1640 46.63 1 0 0
1643 67.07 0 0 1 1643 67.07 0 0 0 1649  2.70 1 0 1 1649  2.70 0 0 0
1666 63.80 0 0 1 1666 63.80 0 0 0 1672 32.63 0 0 1 1672 32.63 0 0 0
1683 62.00 0 1 1 1683 62.00 0 1 0 1688 13.10 1 0 1 1688 54.80 0 0 0
1705  8.00 0 0 1 1705  8.00 0 0 0 1717 51.60 0 1 1 1717 42.33 1 1 0
1727 49.97 0 1 1 1727  2.90 1 1 0 1746 45.90 0 0 1 1746  1.43 1 0 0
1749 41.93 0 1 1 1749 41.93 0 1 0
;
```

```
   run;
```

As a preliminary analysis, PROC FREQ is used to break down the numbers of blindness in the control and treated eyes.

```
   proc freq data=Blind;
      table Treatment*Status;
      run;
```

Output 54.9.1. Breakdown of Blindness in the Control and Treated Groups

```
                              The FREQ Procedure

                          Table of Treatment by Status

                      Treatment     Status

                      Frequency|
                      Percent  |
                      Row Pct  |
                      Col Pct  |       0|       1|  Total
                      ---------+--------+--------+
                             0 |     96 |    101 |    197
                               |  24.37 |  25.63 |  50.00
                               |  48.73 |  51.27 |
                               |  40.17 |  65.16 |
                      ---------+--------+--------+
                             1 |    143 |     54 |    197
                               |  36.29 |  13.71 |  50.00
                               |  72.59 |  27.41 |
                               |  59.83 |  34.84 |
                      ---------+--------+--------+
                      Total         239      155      394
                                  60.66    39.34   100.00
```

By the end of the study, 54 treated eyes and 101 untreated eyes have developed blindness (Output 54.9.1).

The analysis of Lee, Wei, and Amato (1992) can be carried out by the following PROC PHREG specification. The explanatory variables in this Cox model are Treatment, DiabeticType, and the Treatment × DiabeticType interaction. The COVS(AGGREGATE) is specified to compute the robust sandwich covariance matrix estimate.

```
   proc phreg data=Blind covs(aggregate);
      model Time*Status(0)=Treatment DiabeticType Interaction;
      Interaction= Treatment * DiabeticType;
      id ID;
      run;
```

The robust standard error estimates are smaller than the model-based counterparts (Output 54.9.2), since the ratio of the robust standard error estimate relative to the model-based estimate is less than 1 for each variable. Laser photocoagulation appears to be effective (p=0.0217) in delaying the occurrence of blindness. The effect is much more prominent for adult onset diabetes than for juvenile onset diabetes.

Output 54.9.2. Inference Based on the Robust Sandwich Covariance

```
                                The PHREG Procedure

                    Analysis of Maximum Likelihood Estimates

                    Parameter   Standard StdErr                         Hazard
Variable      DF     Estimate      Error  Ratio Chi-Square Pr > ChiSq    Ratio

Treatment      1     -0.42467    0.18497  0.850     5.2713     0.0217    0.654
DiabeticType   1      0.34084    0.19558  0.982     3.0371     0.0814    1.406
Interaction    1     -0.84566    0.30353  0.865     7.7622     0.0053    0.429
```

Example 54.10. Model Assessment Using Cumulative Sums of Martingale Residuals (Experimental)

The Mayo liver disease example of Lin, Wei, and Ying (1993) is reproduced here to illustrate the checking of the functional form of a covariate and the assessment of the proportional hazards assumption. The data consist of 418 patients with primary biliary cirrhosis (PBC), among which 161 had died as of the date of data listing. A subset of the variables are saved in the SAS data set Liver. The data set contains the following variables:

- Time, follow-up time in years
- Status, event indicator with value 1 for death time and value 0 for censored time
- Age, age in years from birth to study registration
- Albumin, serum albumin level in gm/dl
- Bilirubin, serum bilirubin level in mg/dl
- Edema, edema presence
- Protime, prothrombin time in seconds

```
data Liver;
   input Time Status Age Albumin Bilirubin Edema Protime @@;
   label Time="Follow-up Time in Years";
   Time= Time / 365.25;
   datalines;
  400 1 58.7652 2.60 14.5 1.0 12.2 4500 0 56.4463 4.14  1.1 0.0 10.6
 1012 1 70.0726 3.48  1.4 0.5 12.0 1925 1 54.7406 2.54  1.8 0.5 10.3
 1504 0 38.1054 3.53  3.4 0.0 10.9 2503 1 66.2587 3.98  0.8 0.0 11.0
 1832 0 55.5346 4.09  1.0 0.0  9.7 2466 1 53.0568 4.00  0.3 0.0 11.0
 2400 1 42.5079 3.08  3.2 0.0 11.0   51 1 70.5599 2.74 12.6 1.0 11.5
 3762 1 53.7139 4.16  1.4 0.0 12.0  304 1 59.1376 3.52  3.6 0.0 13.6
 3577 0 45.6893 3.85  0.7 0.0 10.6 1217 1 56.2218 2.27  0.8 1.0 11.0
 3584 1 64.6461 3.87  0.8 0.0 11.0 3672 0 40.4435 3.66  0.7 0.0 10.8
  769 1 52.1834 3.15  2.7 0.0 10.5  131 1 53.9302 2.80 11.4 1.0 12.4
 4232 0 49.5606 3.56  0.7 0.5 11.0 1356 1 59.9535 3.51  5.1 0.0 13.0
 3445 0 64.1889 3.83  0.6 0.0 11.4  673 1 56.2765 3.63  3.4 0.0 11.6
  264 1 55.9671 2.94 17.4 1.0 11.7 4079 1 44.5202 4.00  2.1 0.0  9.9
 4127 0 45.0732 4.10  0.7 0.0 11.3 1444 1 52.0246 3.68  5.2 0.0  9.9
   77 1 54.4394 3.31 21.6 0.5 12.0  549 1 44.9473 3.23 17.2 1.0 13.0
 4509 0 63.8768 3.78  0.7 0.0 10.6  321 1 41.3854 2.54  3.6 0.0 11.0
 3839 1 41.5524 3.44  4.7 0.0 10.3 4523 0 53.9959 3.34  1.8 0.0 10.6
 3170 1 51.2827 3.19  0.8 0.0 12.0 3933 0 52.0602 3.70  0.8 0.0 10.5
 2847 1 48.6188 3.20  1.2 0.0 10.6 3611 0 56.4107 3.39  0.3 0.0 10.6
  223 1 61.7276 3.01  7.1 1.0 12.0 3244 1 36.6270 3.53  3.3 0.0 11.0
 2297 1 55.3922 3.00  0.7 0.0 10.6 4467 0 46.6694 3.34  1.3 0.0 11.0
 1350 1 33.6345 3.26  6.8 0.0 11.7 4453 0 33.6947 3.54  2.1 0.0 11.0
 4556 0 48.8706 3.64  1.1 0.0 10.6 3428 1 37.5825 3.55  3.3 1.0 11.7
 4025 0 41.7933 3.93  0.6 0.0 10.9 2256 1 45.7988 2.84  5.7 0.0 12.7
 2576 0 47.4278 3.65  0.5 0.0  9.8 4427 0 49.1362 3.70  1.9 0.0 11.0
  708 1 61.1526 3.82  0.8 0.0 11.0 2598 1 53.5086 3.36  1.1 0.0 10.6
 3853 1 52.0876 3.60  0.8 0.0 10.6 2386 1 50.5407 3.70  6.0 0.0 10.6
 1000 1 67.4086 3.10  2.6 0.0 11.0 1434 1 39.1978 3.40  1.3 1.0 11.0
 1360 1 65.7632 3.94  1.8 0.0 11.0 1847 1 33.6181 3.80  1.1 0.0 10.6
 3282 1 53.5715 3.18  2.3 0.5 12.4 4459 0 44.5695 4.08  0.7 0.0 10.6
 2224 1 40.3943 3.50  0.8 0.0 10.6 4365 0 58.3819 3.40  0.9 0.0 10.3
 4256 0 43.8987 3.94  0.6 0.0 13.0 3090 1 60.7064 2.75  1.3 0.0 13.2
  859 1 46.6283 3.12 22.5 1.0 11.6 1487 1 62.9076 3.50  2.1 0.0 11.0
 3992 0 40.2026 3.60  1.2 0.0 10.0 4191 1 46.4531 3.70  1.4 0.0 11.0
 2769 1 51.2882 3.91  1.1 0.0 10.0 4039 0 32.6133 4.09  0.7 0.0 10.6
 1170 1 49.3388 3.46 20.0 0.5 12.4 3458 0 56.3997 4.64  0.6 0.0 10.6
 4196 0 48.8460 3.57  1.2 0.0 11.5 4184 0 32.4928 3.54  0.5 0.0 10.0
 4190 0 38.4942 3.60  0.7 0.0 11.0 1827 1 51.9206 3.99  8.4 0.0 11.0
 1191 1 43.5181 2.53 17.1 0.5 11.5   71 1 51.9425 3.08 12.2 0.5 11.6
  326 1 49.8261 3.41  6.6 0.5 12.1 1690 1 47.9452 3.02  6.3 0.0 10.6
 3707 0 46.5161 4.24  0.8 0.0 10.9  890 1 67.4114 3.72  7.2 0.0 11.2
 2540 1 63.2635 3.65 14.4 0.0 11.7 3574 1 67.3101 4.09  4.5 0.0 11.1
 4050 0 56.0137 3.50  1.3 0.5 12.9 4032 0 55.8303 3.76  0.4 0.0 11.2
 3358 1 47.2170 3.48  2.1 0.0 11.5 1657 1 52.7584 3.21  5.0 0.0 10.9
  198 1 37.2786 4.40  1.1 0.0 10.7 2452 0 41.3936 4.06  0.6 0.5 12.0
 1741 1 52.4435 3.65  2.0 0.0 11.4 2689 1 33.4757 4.22  1.6 0.0 11.0
  460 1 45.6071 3.47  5.0 0.5 11.9  388 1 76.7091 3.13  1.4 1.0 12.2
 3913 0 36.5339 3.67  1.3 0.0 11.1  750 1 53.9165 3.11  3.2 0.0 11.8
  130 1 46.3901 2.64 17.4 1.0 11.7 3850 0 48.8460 3.70  1.0 0.0 10.4
  611 1 71.8932 3.26  2.0 0.5 11.4 3823 0 28.8843 3.77  1.0 0.0 10.2
 3820 0 48.4682 3.35  1.8 0.0 10.2  552 1 51.4689 3.00  2.3 0.0 12.0
 3581 0 44.9500 3.60  0.9 0.0 10.4 3099 0 56.5695 3.97  0.9 0.0 10.1
  110 1 48.9637 3.67  2.5 1.0 11.1 3086 1 43.0171 3.64  1.1 0.0 11.1
 3092 0 34.0397 4.20  1.1 0.0 10.3 3222 1 68.5092 3.90  2.1 0.0 10.6
 3388 0 62.5216 4.03  0.6 0.0 17.1 2583 1 50.3573 3.50  0.4 0.0 10.3
 2504 0 44.0630 3.61  0.5 0.0 10.6 2105 1 38.9103 3.54  1.9 0.0 10.9
 2350 0 41.1526 4.18  5.5 0.0 10.7 3445 1 55.4579 3.67  2.0 0.0 11.8
  980 1 51.2334 3.74  6.7 0.0 11.1 3395 1 52.8268 4.30  3.2 0.0 11.7
 3422 0 42.6393 4.19  0.7 0.0 10.3 3336 0 61.0705 3.63  3.0 0.5  9.9
 1083 1 49.6564 3.11  6.5 0.0 11.0 2288 1 48.8542 3.30  3.5 0.0 10.2
  515 1 54.2560 3.83  0.6 0.0  9.5 2033 0 35.1513 3.98  3.5 0.0 10.6
  191 1 67.9069 3.08  1.3 1.0 13.2 3297 0 55.4360 4.13  0.6 0.0 10.7
  971 1 45.8207 3.23  5.1 1.0 13.0 3069 0 52.8898 3.90  0.6 0.0 10.8
 2468 0 47.1814 3.51  1.3 0.0 10.0  824 1 53.5989 3.12  1.2 0.0 11.1
 3255 0 44.1040 4.08  0.5 0.0 10.0 1037 1 41.9493 2.89 16.2 0.0 12.6
 3239 0 63.6140 3.87  0.9 0.0  9.7 1413 1 44.2272 3.43 17.4 0.0 11.5
  850 1 62.0014 3.80  2.8 0.0 13.2 2944 0 40.5530 3.83  1.9 0.0  9.8
 2796 1 62.6448 3.95  1.5 0.0 10.1 3149 0 42.3354 3.67  0.7 0.0 10.7
```

```
3150 0 42.9678 3.57  0.4 0.0 11.0 3098 0 55.9617 3.35  0.8 0.0  9.8
2990 0 62.8611 3.60  1.1 0.0 10.1 1297 1 51.2498 3.93  7.3 0.0 10.5
2106 0 46.7625 3.31  1.1 0.0 11.6 3059 0 54.0753 4.09  1.1 0.0 10.0
3050 0 47.0363 3.77  0.9 0.0 10.6 2419 1 55.7262 3.48  1.0 0.0  9.9
 786 1 46.1027 3.60  2.9 0.0 11.0  943 1 52.2875 3.26 28.0 0.5 10.0
2976 0 51.2005 3.84  0.7 0.0 11.4 2615 0 33.8645 3.89  1.2 0.5  9.4
2995 0 75.0116 3.37  1.2 0.5 10.7 1427 1 30.8638 3.26  7.2 0.0  9.8
 762 1 61.8042 3.79  3.0 0.5  9.9 2891 0 34.9870 3.63  1.0 0.0 10.0
2870 0 55.0418 3.03  0.9 0.0  9.4 1152 1 69.9411 3.01  2.3 0.0 10.9
2863 0 49.6044 3.85  0.5 0.0 11.1  140 1 69.3771 2.56  2.4 1.0 14.1
2666 0 43.5565 3.35  0.6 0.5 11.2  853 1 59.4086 3.52 25.5 0.0 11.5
2835 0 48.7584 3.42  0.6 0.0 10.0 2475 0 36.4928 3.37  3.4 0.0 11.2
1536 1 45.7604 3.46  2.5 0.0 10.1 2772 0 57.3717 3.62  0.6 0.0 10.5
2797 0 42.7433 3.56  2.3 0.0  9.6  186 1 58.8172 3.19  3.2 0.0 12.0
2055 1 53.4976 4.08  0.3 0.0  9.9  264 1 43.4141 3.34  8.5 0.5 13.3
1077 1 53.3060 3.45  4.0 0.0 11.3 2721 0 41.3552 3.26  5.7 0.0  9.5
1682 1 60.9582 3.86  0.9 0.0 10.3 2713 0 47.7536 3.80  0.4 0.0  9.2
1212 1 35.4908 4.22  1.3 0.0 10.1 2692 0 48.6626 3.61  1.2 0.0  9.0
2574 0 52.6680 4.52  0.5 0.0 10.1 2301 0 49.8700 3.34  1.3 0.0  9.8
2657 0 30.2752 3.42  3.0 0.0  9.8 2644 0 55.5674 3.85  0.5 0.0  9.7
2624 0 52.1533 3.80  0.8 0.0 10.1 1492 1 41.6099 3.56  3.2 0.0 10.1
2609 0 55.4524 4.01  0.9 0.0 10.4 2580 0 70.0041 4.08  0.6 0.0 10.2
2573 0 43.9425 3.83  1.8 0.0  9.9 2563 0 42.5681 4.38  4.7 0.0 10.4
2556 0 44.5695 3.58  1.4 0.0 10.3 2555 0 56.9446 3.69  0.6 0.0  9.9
2241 0 40.2601 3.73  0.5 0.0 10.1  974 1 37.6071 3.55 11.0 0.0  9.8
2527 0 48.3614 3.54  0.8 0.0 10.5 1576 1 70.8364 3.53  2.0 0.5 12.7
 733 1 35.7919 3.43 14.0 0.0 11.5 2332 0 62.6229 3.48  0.7 0.0 11.0
2456 0 50.6475 3.63  1.3 0.0  9.9 2504 0 54.5270 3.93  2.3 0.0 10.2
 216 1 52.6927 3.35 24.5 0.0 15.2 2443 0 52.7201 3.69  0.9 0.0  9.8
 797 1 56.7721 3.19 10.8 0.0 10.4 2449 0 44.3970 4.30  1.5 0.0  9.1
2330 0 29.5551 3.90  3.7 0.0 11.5 2363 0 57.0404 3.36  1.4 0.0 11.6
2365 0 44.6270 3.97  0.6 0.0 10.1 2357 0 35.7974 2.90  0.7 0.0  9.6
1592 0 40.7173 3.43  2.1 0.0 10.2 2318 0 32.2327 3.55  4.7 0.0  9.9
2294 0 41.0924 3.20  0.6 0.0 10.8 2272 0 61.6400 3.80  0.5 0.0 10.0
2221 0 37.0568 4.04  0.5 0.0  9.9 2090 1 62.5791 3.74  0.7 0.0 10.2
2081 1 48.9774 3.55  2.5 0.0 10.3 2255 0 61.9904 4.07  0.6 0.0 11.0
2171 0 72.7721 3.33  0.6 0.5 10.1  904 1 61.2950 3.20  3.9 0.0 10.0
2216 0 52.6242 4.01  0.7 0.0  9.5 2224 0 49.7632 3.37  0.9 0.0 10.0
2195 0 52.9144 3.76  1.3 0.0 10.3 2176 0 47.2635 3.98  1.2 0.0  9.9
2178 0 50.2040 3.40  0.5 0.0 10.2 1786 1 69.3470 3.43  0.9 0.0  9.9
1080 1 41.1691 3.85  5.9 0.0 10.7 2168 0 59.1650 3.68  0.5 0.0 10.4
 790 1 36.0794 3.31 11.4 0.0 10.8 2170 0 34.5955 3.89  0.5 0.0 10.1
2157 0 42.7132 4.17  1.6 0.0  9.6 1235 1 63.6304 3.22  3.8 0.0 10.6
2050 0 56.6297 3.65  0.9 0.0  9.7  597 1 46.2642 3.38  4.5 0.0 12.4
 334 1 61.2430 2.43 14.1 1.0 11.0 1945 0 38.6201 3.66  1.0 0.0  9.7
2022 0 38.7707 3.66  0.7 0.0 10.1 1978 0 56.6954 3.70  0.5 0.0  9.6
 999 1 58.9514 3.35  2.3 0.0  9.7 1967 0 36.9227 3.35  0.7 0.0  9.6
 348 1 62.4148 3.05  4.5 0.5 11.4 1979 0 34.6092 3.41  3.3 0.0 11.5
1165 1 58.3354 1.96  3.4 0.0 10.7 1951 0 50.1821 3.02  0.4 0.0 10.6
1932 0 42.6858 3.06  0.9 0.0  9.8 1776 0 34.3792 3.35  0.9 0.0 11.2
1882 0 33.1828 4.16 13.0 0.0 11.9 1908 0 38.3819 3.79  1.5 0.0  9.7
1882 0 59.7618 2.95  1.6 0.0 10.1 1874 0 66.4120 3.35  0.6 0.5  9.8
 694 1 46.7899 2.94  0.8 0.0 11.2 1831 0 56.0794 3.72  0.4 0.0 10.1
 837 0 41.3744 3.62  4.4 0.0  9.8 1810 0 64.5722 2.97  1.9 0.0  9.9
 930 1 67.4880 2.81  8.0 0.0 10.0 1690 1 44.8296 3.22  3.9 0.0  9.6
1790 0 45.7714 3.65  0.6 0.0  9.6 1435 0 32.9500 3.77  2.1 0.0 10.1
 732 0 41.2211 2.83  6.1 0.0 10.0 1785 0 55.4168 3.51  0.8 0.0 10.0
1783 0 47.9808 3.20  1.3 0.0 10.6 1769 0 40.7912 3.36  0.6 0.0 10.9
1457 0 56.9747 3.61  0.5 0.0  9.9 1770 0 68.4627 3.35  1.1 0.0 10.0
```

```
1765 0 78.4394 3.03  7.1 0.0 11.2  737 0 39.8576 3.75  3.1 0.0 10.0
1735 0 35.3101 3.85  0.7 0.0 10.3 1701 0 31.4442 3.74  1.1 0.0  9.7
1614 0 58.2642 4.23  0.5 0.0 10.6 1702 0 51.4880 3.44  1.1 0.0  9.6
1615 0 59.9699 2.97  3.1 0.0  9.8 1656 0 74.5243 3.59  5.6 0.0 10.9
1677 0 52.3641 3.14  3.2 0.0  9.5 1666 0 42.7871 3.06  2.8 0.0  9.5
1301 0 34.8747 3.57  1.1 0.5 11.4 1542 0 44.1396 3.12  3.4 0.0 11.2
1084 0 46.3819 3.20  3.5 0.0 10.0 1614 0 56.3094 3.32  0.5 0.0 10.2
 179 1 70.9076 2.33  6.6 1.0 12.1 1191 1 55.3949 2.75  6.4 0.5 11.0
1363 0 45.0842 3.50  3.6 0.0 10.1 1568 0 26.2779 3.74  1.0 0.0 10.2
1569 0 50.4723 3.50  1.0 0.0  9.7 1525 0 38.3984 2.93  0.5 0.0  9.8
1558 0 47.4196 3.46  2.2 0.0  9.6 1447 0 47.9808 3.07  1.6 0.0  9.6
1349 0 38.3162 3.77  2.2 0.0  9.5 1481 0 50.1081 3.85  1.0 0.0 10.7
1434 0 35.0883 3.56  1.0 0.5  9.8 1420 0 32.5038 3.70  5.6 0.0  9.9
1433 0 56.1533 3.77  0.5 0.0  9.8 1412 0 46.1547 3.69  1.6 0.0  9.6
  41 1 65.8836 2.10 17.9 1.0 12.9 1455 0 33.9439 3.52  1.3 0.0  9.5
1030 0 62.8611 3.99  1.1 0.0  9.6 1418 0 48.5640 3.44  1.3 0.0  9.5
1401 0 46.3491 3.48  0.8 0.0 10.0 1408 0 38.8528 3.36  2.0 0.0  9.8
1234 0 58.6475 3.46  6.4 0.0 10.1 1067 0 48.9363 3.89  8.7 0.5  9.6
 799 1 67.5729 3.99  4.0 0.5  9.8 1363 0 65.9849 3.57  1.4 0.0  9.8
 901 0 40.9008 3.18  3.2 0.0  9.9 1329 0 50.2450 3.73  8.6 0.0 11.2
1320 0 57.1964 2.98  8.5 1.0 12.3 1302 0 60.5366 3.07  6.6 0.0 10.9
 877 0 35.3511 3.83  2.4 0.0 10.3 1321 0 31.3812 3.31  0.8 0.0 10.9
 533 0 55.9863 3.43  1.2 0.0 11.3 1300 0 52.7255 3.37  1.1 0.0 10.2
1293 0 38.0917 3.76  2.4 0.0 10.8  207 1 58.1711 2.23  5.2 0.0 12.3
1295 0 45.2101 3.57  1.0 0.0 10.5 1271 0 37.7988 3.95  0.7 0.0 10.6
1250 0 60.6598 3.25  1.0 0.0 10.6 1230 0 35.5346 3.93  0.5 0.0 10.8
1216 0 43.0664 3.61  2.9 0.0 10.6 1216 0 56.3915 3.45  0.6 0.0 10.7
1149 0 30.5736 3.56  0.8 0.0 10.5 1153 0 61.1828 3.58  0.4 0.0 10.4
 994 0 58.2998 2.75  0.4 0.0 10.8  939 0 62.3326 3.35  1.7 0.0 10.2
 839 0 37.9986 3.16  2.0 0.0 10.5  788 0 33.1526 3.79  6.4 0.0 10.8
4062 0 60.0000 3.65  0.7 0.0 11.0 3561 1 65.0000 3.04  1.4 0.5 12.1
2844 0 54.0000 4.03  0.7 0.0  9.8 2071 1 75.0000 3.96  0.7 0.5 11.3
3030 0 62.0000 2.48  0.8 0.0 10.0 1680 0 43.0000 3.68  0.7 0.0  9.5
  41 1 46.0000 2.93  5.0 0.0 10.4 2403 0 44.0000 3.81  0.4 0.5 10.5
1170 0 61.0000 3.41  1.3 0.5 10.9 2011 1 64.0000 3.69  1.1 0.0 10.5
3523 0 40.0000 4.04  0.6 0.0 11.2 3468 0 63.0000 3.94  0.6 0.0 11.5
4795 0 34.0000 3.24  1.8 0.0 18.0 1236 0 52.0000 3.42  1.5 0.0 10.3
4214 0 49.0000 3.99  1.2 0.0 11.2 2111 1 54.0000 3.60  1.0 0.0 12.1
1462 1 63.0000 3.40  0.7 0.0 10.1 1746 1 54.0000 3.63  3.5 0.0 10.3
  94 1 46.0000 3.56  3.1 0.5 13.6  785 1 53.0000 2.87 12.6 0.0 11.8
1518 1 56.0000 3.92  2.8 0.0 10.6  466 1 56.0000 3.51  7.1 0.0 11.8
3527 0 55.0000 4.15  0.6 0.0 10.1 2635 0 65.0000 3.34  2.1 0.0 10.1
2286 1 56.0000 3.64  1.8 0.0 10.0  791 1 47.0000 3.42 16.0 0.0 13.8
3492 0 60.0000 4.38  0.6 0.0 10.6 3495 0 53.0000 4.19  5.4 0.0 11.2
 111 1 54.0000 3.29  9.0 0.0 13.1 3231 0 50.0000 4.01  0.9 0.0 10.5
 625 1 48.0000 2.84 11.1 0.0 12.2 3157 0 36.0000 3.76  8.9 0.0 10.6
3021 0 48.0000 3.76  0.5 0.0 10.1  559 1 70.0000 3.81  0.6 0.5 11.0
2812 1 51.0000 3.92  3.4 0.0  9.3 2834 0 52.0000 3.14  0.9 0.0 12.3
2855 0 54.0000 3.82  1.4 0.0 10.3  662 1 48.0000 4.10  2.1 0.0  9.0
 727 1 66.0000 3.40 15.0 0.0 11.1 2716 0 53.0000 4.19  0.6 0.0  9.9
2698 0 62.0000 3.40  1.3 0.0 10.6  990 1 59.0000 3.12  1.3 0.0  9.6
2338 0 39.0000 3.75  1.6 0.0 10.4 1616 1 67.0000 3.26  2.2 0.5 11.1
2563 0 58.0000 3.46  3.0 0.0 10.4 2537 0 64.0000 3.49  0.8 0.0 10.3
2534 0 46.0000 2.89  0.8 0.0 10.6  778 1 64.0000 3.15  1.8 0.0 10.4
 617 0 41.0000 2.31  5.5 0.0 10.4 2267 0 49.0000 3.04 18.0 0.0  9.7
2249 0 44.0000 3.50  0.6 0.0  9.9  359 1 59.0000 3.35  2.7 0.0 11.5
1925 0 63.0000 3.58  0.9 0.0 10.0  249 1 61.0000 3.01  1.3 0.0 10.7
2202 0 64.0000 3.49  1.1 0.0  9.8   43 1 49.0000 2.77 13.8 0.0 11.1
1197 1 42.0000 4.52  4.4 0.0 10.8 1095 1 50.0000 3.36 16.0 0.0 10.0
```

```
  489 1 51.0000 3.52  7.3 0.5 11.1 2149 0 37.0000 3.55  0.6 0.0 10.3
 2103 0 62.0000 3.29  0.7 0.0  9.8 1980 0 51.0000 3.10  0.7 0.0 10.6
 1347 0 52.0000 3.24  1.7 0.0 10.5 1478 1 44.0000 3.63  9.5 0.0 10.2
 1987 0 33.0000 3.76  2.2 0.0  9.9 1168 1 60.0000 3.62  1.8 0.5  9.9
  597 1 63.0000 2.73  3.3 0.5 11.1 1725 0 33.0000 4.08  2.9 0.0 10.5
 1899 0 41.0000 3.66  1.7 0.0 11.0  221 1 51.0000 2.58 14.0 0.0 11.6
 1022 0 37.0000 3.00  0.8 0.5 10.8 1639 0 59.0000 3.40  1.3 0.0  9.7
 1635 0 55.0000 2.93  0.7 0.0 10.6 1654 0 54.0000 2.38  1.7 0.0  9.8
 1653 0 49.0000 3.00 13.6 0.5  9.9 1560 0 40.0000 3.50  0.9 0.0 10.9
 1581 0 67.0000 3.06  0.7 0.0 10.0 1419 0 68.0000 3.15  3.0 0.0 10.0
 1443 0 41.0000 2.80  1.2 0.0 11.0 1368 0 69.0000 3.03  0.4 0.0 10.9
  193 1 52.0000 2.96  0.7 0.5  9.9 1367 0 57.0000 3.07  2.0 0.5 12.1
 1329 0 36.0000 3.98  1.4 0.0 11.0 1343 0 50.0000 3.48  1.6 0.0 10.2
 1328 0 64.0000 3.65  0.5 0.0 10.2 1375 0 62.0000 3.49  7.3 0.0 10.9
 1260 0 42.0000 2.82  8.1 0.0 10.4 1223 0 44.0000 3.34  0.5 0.0 10.6
  935 1 69.0000 3.19  4.2 0.0 11.1  943 0 52.0000 3.01  0.8 0.0 10.6
 1141 0 66.0000 3.33  2.5 0.0 10.8 1092 0 40.0000 3.60  4.6 0.0 10.4
 1150 0 52.0000 3.64  1.0 0.0 10.6  703 1 46.0000 2.68  4.5 0.0 11.5
 1129 0 54.0000 3.69  1.1 0.0 10.8 1086 0 51.0000 3.17  1.9 0.5 10.7
 1067 0 43.0000 3.73  0.7 0.0 10.8 1072 0 39.0000 3.81  1.5 0.0 10.8
 1119 0 51.0000 3.57  0.6 0.0 10.6 1097 0 67.0000 3.58  1.0 0.0 10.8
  989 0 35.0000 3.23  0.7 0.0 10.8  681 1 67.0000 2.96  1.2 0.0 10.9
 1103 0 39.0000 3.83  0.9 0.0 11.2 1055 0 57.0000 3.42  1.6 0.0  9.9
  691 0 58.0000 3.75  0.8 0.0 10.4  976 0 53.0000 3.29  0.7 0.0 10.6
;
run;
```

Consider fitting a Cox model for the survival time of the PCB patients with covariates Bilirubin, log(Protime), log(Albumin), Age and Edema. The log tranform, which is often applied to blood chemistry measurements, is deliberately not employed for Bilirubin. It is of interest to assess the functional form of the variable Bilirubin in the Cox model.

```
ods html;
ods graphics on;

proc phreg data=Liver;
   model Time*Status(0)=Bilirubin logProtime logAlbumin Age Edema;
   logProtime=log(Protime);
   logAlbumin=log(Albumin);
   assess var=(Bilirubin) /  resample;
   run;

ods graphics off;
ods html close;
```

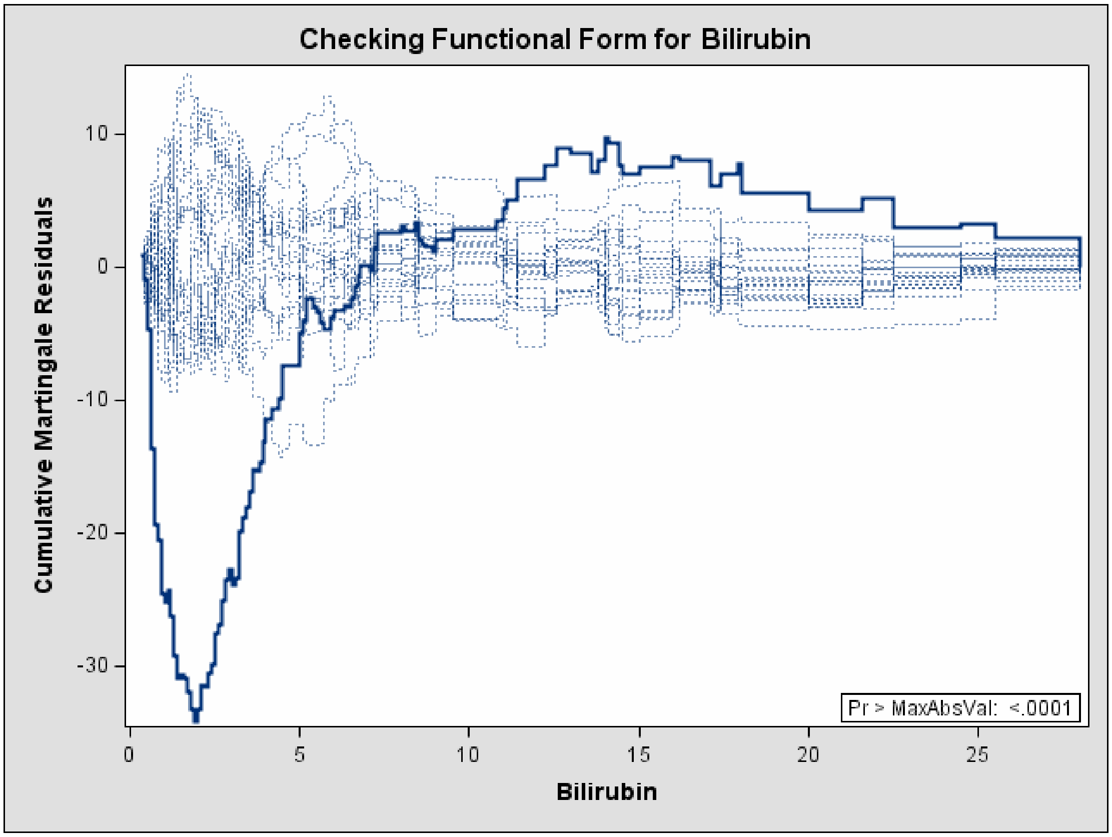
The ASSESS statement creates a plot of the cumulative martingale residuals against the values of the covariate Bilirubin, which is specified in the VAR= option. The RESAMPLE option computes the p-value of a Kolmogorov-type supremum test based on a sample of 1,000 simulated residual patterns.

Parameter estimates of the model fit are shown in Output 54.10.1. The plot in Output 54.10.2 displays the observed cumulative martingale residual process for Bilirubin together with 20 simulated realizations from the null distribution. This graphical display is requested by specifying the experimental ODS GRAPHICS statement and the experimental ASSESS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the PHREG procedure, see the section “ODS Graphics” on page 3271. It is obvious that the observed process is atypical compared to the simulated realizations. Also, none of the 1,000 simulated realizations has an absolute maximum exceeding that of the observed cumulative martingale residual process. Both the graphical and numerical results indicate that a transform is deemed necessary for Bilirubin in the model.

Output 54.10.1. Cox Model with Bilirubin as a Covariate

```
                              The PHREG Procedure

                    Analysis of Maximum Likelihood Estimates

                   Parameter     Standard                               Hazard
Variable      DF    Estimate        Error    Chi-Square    Pr > ChiSq     Ratio

Bilirubin      1     0.11733      0.01298       81.7567        <.0001     1.124
logProtime     1     2.77581      0.71482       15.0794        0.0001    16.052
logAlbumin     1    -3.17195      0.62945       25.3939        <.0001     0.042
Age            1     0.03779      0.00805       22.0288        <.0001     1.039
Edema          1     0.84772      0.28125        9.0850        0.0026     2.334
```

Output 54.10.2. Cumulative Martingale Residuals vs Bilirubin (Experimental)



The cumulative martingale residual plots in Output 54.10.3 provide guidance in suggesting a more appropriate functional form for a covariate. The four curves were created from simple forms of misspecification using 1,000 simulated times from a exponential model with 20% censoring. The true and fitted models are shown in Table 54.3.

Output 54.10.3. Typical Cumulative Residual Plot Patterns

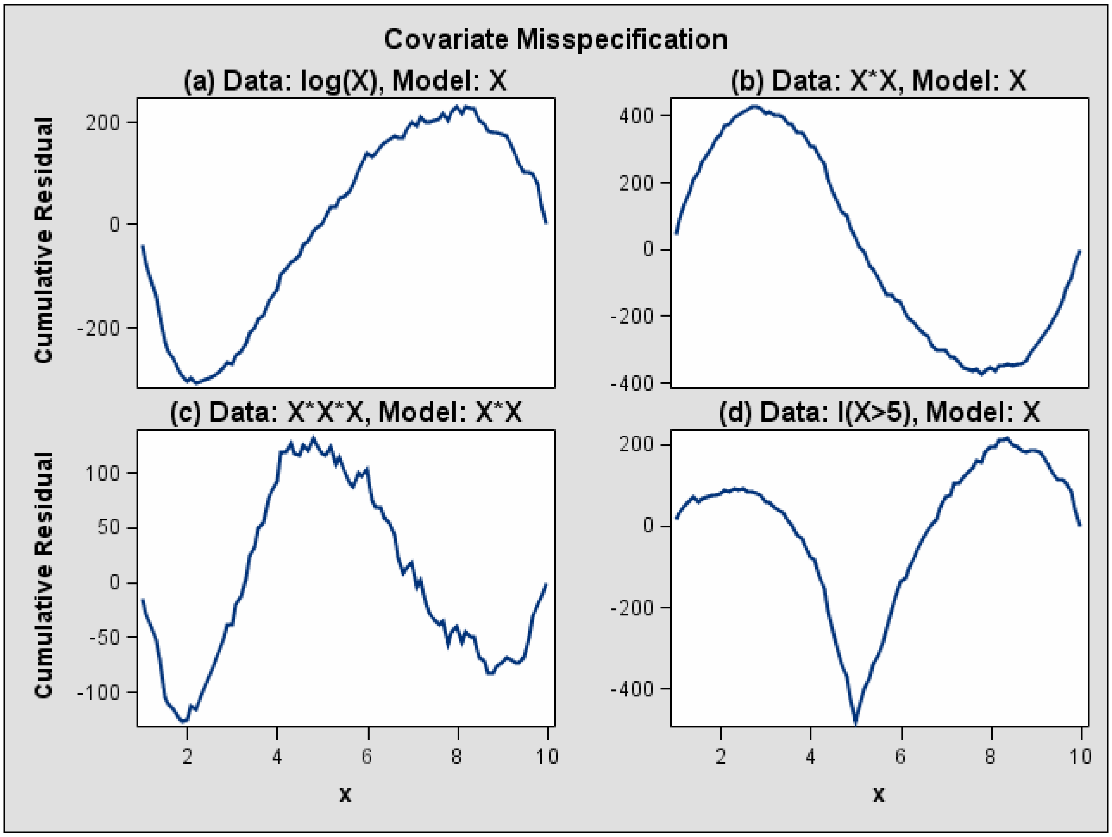


Table 54.3. Model Misspecifications

Plot	Data	Fitted Model
(a)	$\log(X)$	X
(b)	$\{X, X^2\}$	X
(c)	$\{X, X^2, X^3\}$	$\{X, X^2\}$
(d)	$I(X > 5)$	X

The curve of observed cumulative martingale residuals in Output 54.10.2 most resembles the behavior of the curve in plot (a) of Output 54.10.3, indicating that log(Bilirubin) might be a more appropriate term in the model than Bilirubin.

Next, the analysis of the natural history of the PBC is repeated with log(Bilirubin) replacing Bilirubin, and the functional form of log(Bilirubin) is assessed. Also assessed is the proportional hazards assumption for the Cox model.

```
ods html;
ods graphics on;

proc phreg data=Liver;
   model Time*Status(0)=logBilirubin logProtime logAlbumin Age Edema;
```

```
      logBilirubin=log(Bilirubin);
      logProtime=log(Protime);
      logAlbumin=log(Albumin);
      assess var=(logBilirubin) ph / crpanel resample seed=19;
      run;

   ods graphics off;
   ods html close;
```

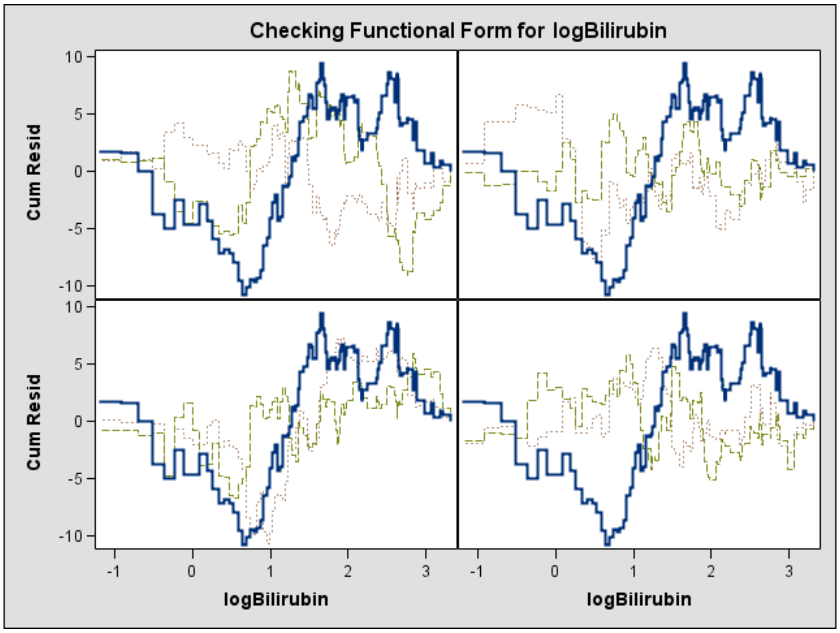
The SEED= option specifies a integer seed for generating random numbers. The CRPANEL option in the ASSESS statement requests a panel of four plots. Each plot displays the observed cumulative martingale residual process along with two simulated realizations. The PH option checks the proportional hazards assumption of the model by plotting the observed standardized score process with 20 simulated realizations for each covariate in the model.

Output 54.10.4 displays the parameter estimates of the fitted model. The cumulative martingale residual plots in Output 54.10.5 and Output 54.10.6 show that the observed martingale residual process is more typical of the simulated realizations. The p-value for the Kolmogorov-type supremum test based on 1,000 simulations is 0.052, indicating that the log transform is a much improved functional form for Bilirubin.

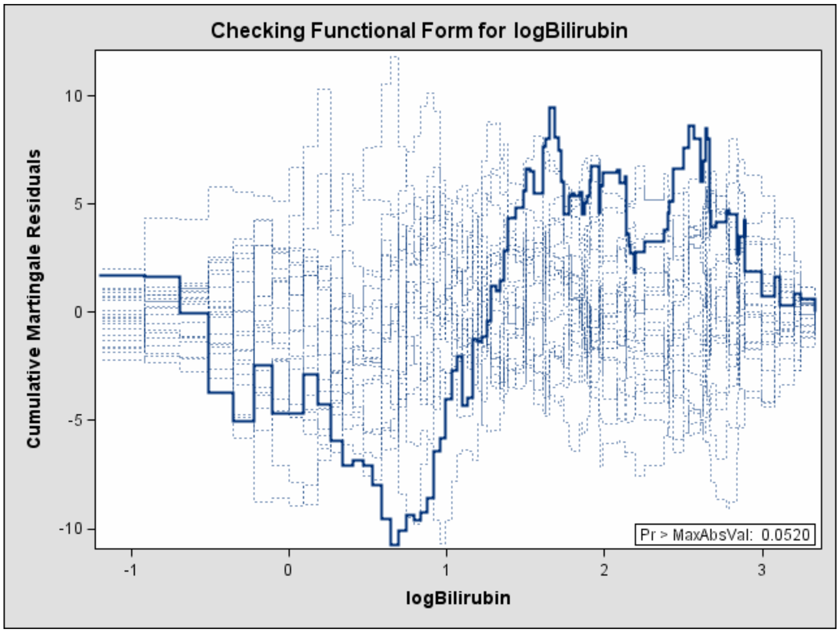
Output 54.10.4. Model with log(Bilirubin) as a Covariate

```
                              The PHREG Procedure

                    Analysis of Maximum Likelihood Estimates

                     Parameter     Standard                                 Hazard
Variable       DF     Estimate        Error   Chi-Square   Pr > ChiSq       Ratio

logBilirubin    1      0.87072      0.08263     111.0484       <.0001       2.389
logProtime      1      2.37789      0.76674       9.6181       0.0019      10.782
logAlbumin      1     -2.53264      0.64819      15.2664       <.0001       0.079
Age             1      0.03940      0.00765      26.5306       <.0001       1.040
Edema           1      0.85934      0.27114      10.0447       0.0015       2.362
```

Output 54.10.5. Panel Plot of Cumulative Martingale Residuals vs log(Bilirubin) (Experimental)

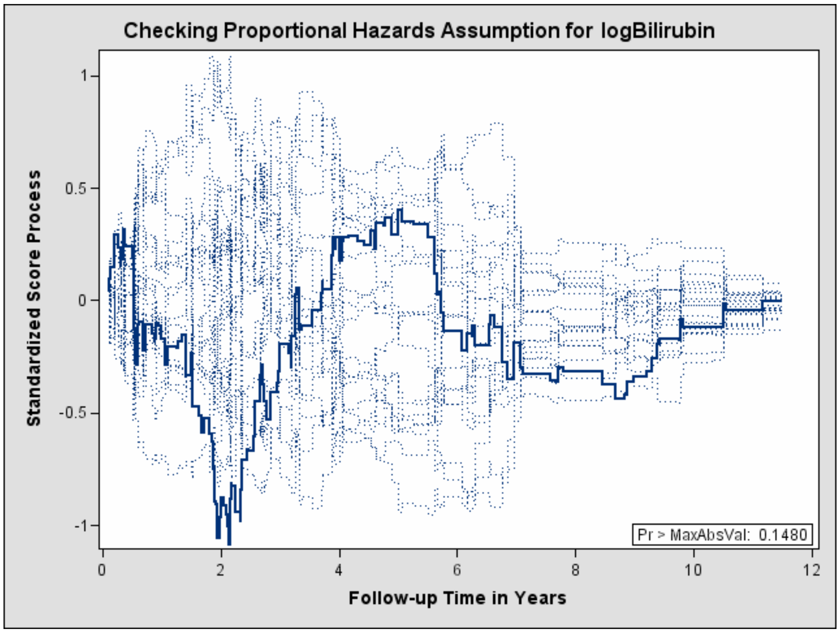


Output 54.10.6. Cumulative Martingale Residuals vs log(Bilirubin) (Experimental)

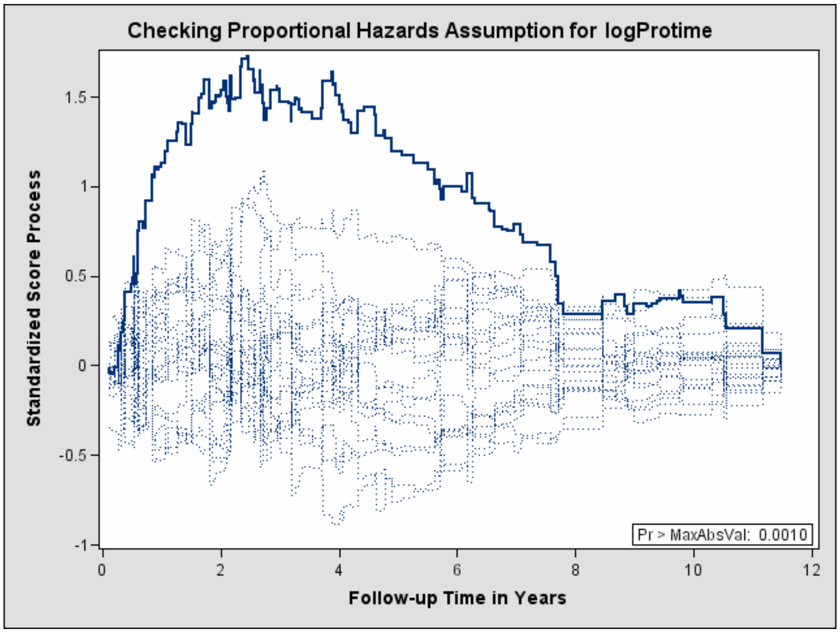


Output 54.10.7 and Output 54.10.8 display the results of proportional hazards assumption assessment for log(Bilirubin) and log(Protime) respectively. The latter plot reveals nonproportional hazards for log(Protime).

Output 54.10.7. Standardized Score Process for log(Bilirubin) (Experimental)



Output 54.10.8. Standardized Score Process for log(Protime) (Experimental)



Plots for log(Albumin), Age, and Edema are not shown here. The Kolmogorov-type supremum test results for all the covariates are shown in Output 54.10.9. In addition to log(Protime), the proportional hazards assumption appears to be violated for Edema.

Output 54.10.9. Kolmogorov-type Supremum Tests for Proportional Hazards Assumption

```
               Supremum Test for Proportionals Hazards Assumption

                       Maximum
                      Absolute                                   Pr >
   Variable              Value    Replications          Seed    MaxAbsVal

   logBilirubin         1.0880            1000            19       0.1480
   logProtime           1.7243            1000            19       0.0010
   logAlbumin           0.8443            1000            19       0.4390
   Age                  0.7387            1000            19       0.4780
   Edema                1.4350            1000            19       0.0310
```

References

Andersen, P. K., Borgan, Ø., Gill, R. D., and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, New York: Springer-Verlag.

Andersen, P. K. and Gill, R. D. (1982), "Cox's Regression Model Counting Process: A Large Sample Study," *Annals of Statistics*, 10, 1100–1120.

Binder, D. A. (1992), "Fitting Cox's Proportional Hazards Models from Survey Data," *Biometrika*, 79, 139–47.

Breslow, N. E. (1974), "Covariance Analysis of Censored Survival Data," *Biometrics*, 30, 89–99.

Cain, K. C. and Lange, N. T. (1984), "Approximate Case Influence for the Proportional Hazards Regression Model with Censored Data," *Biometrics*, 40, 493–499.

Cox, D. R. (1972), "Regression Models and Life Tables," *Journal of the Royal Statistical Society, Series B*, 20, 187–220 (with discussion).

Cox, D. R. (1975), "Partial Likelihood," *Biometrika*, 62, 269–276.

Crowley, J. and Hu, M. (1977), "Covariance Analysis of Heart Transplant Survival Data," *Journal of the American Statistical Association*, 72, 27–36.

DeLong, D. M., Guirguis, G. H., and So, Y. C. (1994), "Efficient Computation of Subset Selection Probabilities with Application to Cox Regression," *Biometrika*, 81, 607–611.

Efron, B. (1977), "The Efficiency of Cox's Likelihood Function for Censored Data," *Journal of the American Statistical Association*, 72, 557–565.

Fleming, T. R. and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: John Wiley & Sons, Inc.

Furnival, G. M. and Wilson, R. W. (1974), "Regression by Leaps and Bounds," *Technometrics*, 16, 499–511.

Gail, M. H., Lubin, J. H., and Rubinstein, L. V. (1981), "Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Death Times," *Biometrika*, 68, 703–707.

Grambsch, P. M. and Therneau, T. M. (1994), "Proportional Hazards Tests and Diagnostics Based on Weighted Residuals," *Biometrika*, 81, 515–526.

Hosmer, D. W. J. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons, Inc.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons, Inc.

Krall, J. M., Uthoff, V. A., and Harley, J. B. (1975), “A Step-up Procedure for Selecting Variables Associated with Survival,” *Biometrics*, 31, 49–57.

Lawless, J. and Nadeau, C. (1995), “Some Simple Robust Methods for the Analysis of Recurrent Events,” *Technometrics*, 37, 158–168.

Lee, E., Wei, L., and Amato, D. (1992), “Cox-Type Regression Analysis for Large Numbers of Small Groups of Correlated Failure Time Observations,” 237–247, Netherlands: Kluwer Academic Publishers.

Lin, D. (1994), “Cox Regression Analysis of Multivariate Failure Time Data: The Marginal Approach,” *Statistics in Medicine*, 13, 2233–2247.

Lin, D., Wei, L., Yang, I., and Ying, Z. (2000), “Semiparametric Regression for the Mean and Rate Functions of Recurrent Events,” *Journal of the Royal Statistical Society, Series B*, 62, 711–730.

Lin, D., Wei, L., and Ying, Z. (1993), “Checking the Cox Model with Cumulative Sums of Martingale-Based Residuals,” *Biometrika*, 80, 557–572.

Lin, D. Y. and Wei, L. J. (1989), “The Robust Inference for the Proportional Hazards Model,” *Journal of the American Statistical Association*, 84, 1074–1078.

Nelson, W. (2002), *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*, ASA-SIAM Series on Statistics and Applied Probability.

Pepe, M. and Cai, J. (1993), “Some Graphical Displays and Marginal Regression Analyses for Recurrent Failure Times and Time Dependent Covariates,” *Journal of the American Statistical Association*, 88, 881–820.

Pettitt, A. N. and Bin Daud, I. (1989), “Case-Weighted Measures of Influence for Proportional Hazards Regression,” *Applied Statistics*, 38, 313–329.

Prentice, R. L., Williams, B. J., and Peterson, A. V. (1981), “On the Regression Analysis of Multivariate Failure Time Data,” *Biometrika*, 68, 373–379.

Reid, N. and Crèpeau, H. (1985), “Influence Functions for Proportional Hazards Regression,” *Biometrika*, 72, 1–9.

Schoenfeld, D. (1982), “Partial Residuals for the Proportional Hazards Regression Model,” *Biometrika*, 69, 239–241.

Therneau, T. M. and Grambsch, P. M. (2000), *Modeling Survival Data: Extending the Cox Model*, New York: Springer-Verlag.

Therneau, T. M., Grambsch, P. M., and Fleming, T. R. (1990), “Martingale-Based Residuals and Survival Models,” *Biometrika*, 77, 147–160.

Tsiatis, A. (1981), “A Large Sample Study of the Estimates for the Integrated Hazard Function in Cox’s Regression Model for Survival Data,” *Annals of Statistics*, 9, 93–108.

Wei, L. J., Lin, D. Y., and Weissfeld, L. (1989), “Regression Analysis of Multivariate Incomplete Failure Time Data by Modeling Marginal Distribution,” *Journal of the American Statistical Association*, 84, 1065–1073.

Chapter 55
The PLAN Procedure

Chapter Contents

Chapter 55
The PLAN Procedure

Overview

The PLAN procedure constructs designs and randomizes plans for factorial experiments, especially nested and crossed experiments and randomized block designs. PROC PLAN can also be used for generating lists of permutations and combinations of numbers. The PLAN procedure can construct the following types of experimental designs:

- full factorials, with and without randomization
- certain balanced and partially balanced incomplete block designs
- generalized cyclic incomplete block designs
- Latin square designs

For other kinds of experimental designs, especially fractional factorial, response surface, and orthogonal array designs, refer to the FACTEX and OPTEX procedures and the ADX Interface in SAS/QC software.

PROC PLAN generates designs by first generating a selection of the levels for the first factor. Then, for the second factor, PROC PLAN generates a selection of its levels for each level of the first factor. In general, for a given factor, the PLAN procedure generates a selection of its levels for all combinations of levels for the factors that precede it. The selection can be done in five different ways:

- randomized selection, for which the levels are returned in a random order
- ordered selection, for which the levels are returned in a standard order every time a selection is generated
- cyclic selection, for which the levels returned are computed by cyclically permuting the levels of the previous selection
- permuted selection, for which the levels are a permutation of the integers $1, \ldots, n$
- combination selection, for which the m levels are selected as a combination of the integers $1, \ldots, n$ taken m at a time

The randomized selection method can be used to generate randomized plans. Also, by appropriate use of cyclic selection, any of the designs in the very wide class of generalized cyclic block designs (Jarrett and Hall 1978) can be generated.

There is no limit to the depth to which the different factors can be nested, and any number of randomized plans can be generated.

You can also declare a list of factors to be selected simultaneously with the lowest (that is, the most nested) factor. The levels of the factors in this list can be seen as constituting the treatment to be applied to the cells of the design. For this reason, factors in this list are called *treatments*. With this list, you can generate and randomize plans in one run of PROC PLAN.

Getting Started

Three Replications with Four Factors

Suppose you want to determine if the order in which four drugs are given affects the response of a subject. If you have only three subjects to test, you can use the following statements to design the experiment.

```
proc plan seed=27371;
   factors Replicate=3 ordered Drug=4;
run;
```

These statements produce a design with three replicates of the four levels of the factor Drug arranged in random order. The three levels of Replicate are arranged in order, as shown in Figure 55.1

```
                              The PLAN Procedure

                 Factor          Select       Levels     Order

                 Replicate            3            3    Ordered
                 Drug                 4            4    Random


                             Replicate      --Drug-

                                     1      3 2 4 1
                                     2      1 2 4 3
                                     3      4 1 2 3
```

Figure 55.1. Three Replications and Four Factors

You may also want to apply one of four different treatments to each cell of this plan (for example, applying different amounts of each drug). The following statements create the output shown in Figure 55.2

```
   factors Replicate=3 ordered Drug=4;
   treatments Treatment=4;
run;
```

```
                        The PLAN Procedure

                           Plot Factors

          Factor          Select      Levels     Order

          Replicate            3           3    Ordered
          Drug                 4           4    Random


                        Treatment Factors

          Factor          Select      Levels     Order

          Treatment            4           4    Random


           Replicate      --Drug-      --Treatment--

                   1      3 1 2 4      2   1   3   4
                   2      4 3 2 1      4   1   2   3
                   3      3 2 4 1      1   4   2   3
```

Figure 55.2. Using the TREATMENTS Statement

Randomly Assigning Subjects to Treatments

You can use the PLAN procedure to design a completely randomized design. Suppose you have 12 experimental units, and want to assign one of two treatments to each unit. Use a DATA step to store the unrandomized design in a SAS data set, then call PROC PLAN to randomize it by specifying one RANDOM factor of 12 levels. The following statements produce Figure 55.3 and Figure 55.4:

```
title 'Completely Randomized Design';
/* The unrandomized design */
data a;
   do unit=1 to 12;
      if (unit <= 6) then treat=1;
      else                treat=2;
      output;
   end;
run;

/* Randomize the design */
proc plan seed=27371;
   factors unit=12;
   output data=a out=b;
run;

proc sort data=b;
   by unit;
proc print;
run;
```

Figure 55.3 shows that the 12 levels of the unit factor have been randomly reordered and then lists the new ordering.

```
                    Completely Randomized Design

                         The PLAN Procedure

              Factor      Select      Levels     Order

              unit            12          12    Random


                ---------------unit---------------

                 8  5  1  4  6  2 12  7  3  9 10 11
```

Figure 55.3. A Completely Randomized Design for Two Treatments

After the data is sorted by the unit variable, the randomized design is displayed in Figure 55.4.

```
                    Completely Randomized Design

                        Obs    unit    treat

                          1       1       1
                          2       2       1
                          3       3       2
                          4       4       1
                          5       5       1
                          6       6       1
                          7       7       2
                          8       8       1
                          9       9       2
                         10      10       2
                         11      11       2
                         12      12       2
```

Figure 55.4. A Completely Randomized Design for Two Treatments

You can also generate the plan by using a TREATMENTS statement instead of a DATA step. The following statements generate the same plan.

```
proc plan seed=27371;
   factors unit=12;
   treatments treat=12 cyclic (1 1 1 1 1 1 2 2 2 2 2 2);
   output out=b;
run;
```

Syntax

The following statements are available in PROC PLAN.

> **PROC PLAN** < *options* > **;**
> **FACTORS** *factor-selections* < / **NOPRINT** > **;**
> **OUTPUT OUT=***SAS-data-set* < *factor-value-settings* > **;**
> **TREATMENTS** *factor-selections* **;**

To use PROC PLAN, you need to specify the PROC PLAN statement and at least one FACTORS statement before the first RUN statement. The TREATMENTS statement, OUTPUT statement, and additional FACTORS statements can appear either before the first RUN statement or after it. The rest of this section gives detailed syntax information for each of the statements, beginning with the PROC PLAN statement. The remaining statements are described in alphabetical order.

You can use PROC PLAN interactively by specifying multiple groups of statements, separated by RUN statements. For details, see the "Using PROC PLAN Interactively" section on page 3346.

PROC PLAN Statement

> **PROC PLAN** < *options* > **;**

The PROC PLAN statement starts the PLAN procedure and, optionally, specifies a random number seed or a default method for selecting levels of factors. By default, the procedure uses a random number seed generated from reading the time of day from the computer's clock and randomly selects levels of factors. These defaults can be modified with the SEED= and ORDERED options, respectively. Unlike many SAS/STAT procedures, the PLAN procedure does not have a DATA= option in the PROC statement; in this procedure, both the input and output data sets are specified in the OUTPUT statement.

You can specify the following options in the PROC PLAN statement:

SEED=*number*

specifies an integer used to start the pseudo-random number generator for selecting factor levels randomly. If you don't specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

ORDERED

selects the levels of the factor as the integers $1, 2, \ldots, m$, in order. For more detail, see the "Selection-Types" section on page 3340 and see the "Specifying Factor Structures" section on page 3348.

FACTORS Statement

FACTORS *factor-selections* < / **NOPRINT** > **;**

The FACTORS statement specifies the factors of the plan and generates the plan. Taken together, the *factor-selections* specify the plan to be generated; more than one *factor-selection* request can be used in a FACTORS statement. The form of a *factor-selection* is

name=m < **OF** *n* > < *selection-type* >

where

name is a valid SAS name. This gives the name of a factor in the design.

m is a positive integer that gives the number of values to be selected. If n is specified, the value of m must be less than or equal to n.

n is a positive integer that gives the number of values to be selected from.

selection-type specifies one of five methods for selecting m values. Possible values are COMB, CYCLIC, ORDERED, PERM or RANDOM. The CYCLIC *selection-type* has additional optional specifications that enable you to specify an initial block of numbers to be cyclically permuted and an increment used to permute the numbers. By default, the *selection-type* is RANDOM, unless you use the ORDERED option in the PROC PLAN statement. In this case, the default *selection-type* is ORDERED. For details, see the following section, "Selection-Types"; for examples, see the "Syntax Examples" section.

The following option can appear in the FACTORS statement after the slash:

NOPRINT

suppresses the display of the plan. This is particularly useful when you require only an output data set. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

Selection-Types

PROC PLAN interprets *selection-type* as follows:

RANDOM selects the m levels of the factor randomly without replacement from the integers $1, 2, \ldots, n$. Or, if n is not specified, RANDOM selects levels by randomly ordering the integers $1, 2, \ldots, m$.

ORDERED selects the levels of the factor as the integers $1, 2, \ldots, m$, in that order.

PERM selects the m levels of the factor as a permutation of the integers $1, \ldots\ m$ according to an algorithm that cycles through all $m!$ permutations. The permutations are produced in a sorted standard order; see Example 55.6 on page 3358.

COMB selects the m levels of the factor as a combination of the integers $1, \ldots, n$ taken m at a time, according to an algorithm that cycles through all $n!/(m!(n-m)!)$ combinations. The combinations are produced in a sorted standard order; see Example 55.6 on page 3358.

CYCLIC <**(***initial-block***)** >< *increment*>

selects the levels of the factor by cyclically permuting the integers $1, 2, \ldots, n$. Wrapping occurs at m if n is not specified, and at n if n is specified. Additional optional specifications are as follows:

With the *selection-type* CYCLIC, you can optionally specify an *initial-block* and an *increment*. The *initial-block* must be specified within parentheses, and it specifies the block of numbers to permute. The first permutation is the block you specify, the second is the block permuted by 1 (or by the *increment* you specify), and so on. By default, the *initial-block* is the integers $1, 2, \ldots, m$. If you specify an *initial-block*, it must have m values. Values specified in the *initial-block* do not have to be given in increasing order.

The *increment* specifies the increment by which to permute the block of numbers. By default, the *increment* is 1.

Syntax Examples

This section gives some simple syntax examples. For more complex examples and details on how to generate various designs, see the "Specifying Factor Structures" section on page 3348. The examples in this section assume that you use the default random selection method and do not use the ORDERED option in the PROC PLAN statement.

The following specification generates a random permutation of the numbers 1, 2, 3, 4, and 5.

```
factors A=5;
```

The following specification generates a random permutation of 5 of the integers from 1 to 8, selected without replacement.

```
factors A=5 of 8;
```

Adding the ORDERED *selection-type* to the two previous specifications generates an ordered list of the integers 1 to 5. The following specification cyclically permutes the integers 1, 2, 3, and 4.

```
factors A=4 cyclic;
```

Since this simple request generates only one permutation of the numbers, the procedure generates an ordered list of the integers 1 to 4. The following specification cyclically permutes the integers 5 to 8.

```
factors A=4 of 8 cyclic (5 6 7 8);
```

In this case, since only one permutation is performed, the procedure generates an ordered list of the integers 5 to 8. The following specification produces an ordered list for A, with values 1 and 2.

```
factors A=2 ordered B=4 of 8 cyclic (5 6 7 8) 2;
```

The associated factor levels for B are 5, 6, 7, 8 for level 1 of A; and 7, 8, 1, 2 for level 2 of A.

Handling More than One Factor-Selection

For cases with more than one *factor-selection* in the same FACTORS statement, PROC PLAN constructs the design as follows:

1. PROC PLAN first generates levels for the first *factor-selection*. These levels are permutations of integers (1, 2, and so on) appropriate for the selection type chosen. If you do not specify a selection type, PROC PLAN uses the default (RANDOM); if you specify the ORDERED option in the PROC PLAN statement, the procedure uses ORDERED as the default selection type.
2. For every integer generated for the first *factor-selection*, levels are generated for the second *factor-selection*. These levels are generated according to the specifications following the second equal sign.
3. This process is repeated until levels for all *factor-selections* have been generated.

The following statements give an example of generating a design with two random factors:

```
proc plan;
   factors One=4 Two=3;
run;
```

The procedure first generates a random permutation of the integers 1 to 4 and then, for each of these, generates a random permutation of the integers 1 to 3. You can think of factor Two as being nested within factor One, where the levels of factor One are to be randomly assigned to 4 units.

As another example, six random permutations of the numbers 1, 2, 3 can be generated by specifying

```
proc plan;
   factors a=6 ordered b=3;
run;
```

OUTPUT Statement

OUTPUT OUT=*SAS-data-set* < **DATA=***SAS-data-set* >
< *factor-value-settings* > **;**

The OUTPUT statement applies only to the last plan generated. If you use PROC PLAN interactively, the OUTPUT statement for a given plan must be immediately preceded by the FACTORS statement (and the TREATMENTS statement, if appropriate) for the plan. See the "Output Data Sets" section on page 3346 for more information on how output data sets are constructed. You can specify the following options in the OUTPUT statement:

OUT=*SAS-data-set*
DATA=*SAS-data-set*

You can use the OUTPUT statement both to output the last plan generated and to use the last plan generated to randomize another SAS data set.

When you specify only the OUT= option in the OUTPUT statement, PROC PLAN saves the last plan generated to the specified data set. The output data set contains one variable for each factor in the plan and one observation for each cell in the plan. The value of a variable in a given observation is the level of the corresponding factor for that cell. The OUT= option is required.

When you specify both the DATA= and OUT= options in the OUTPUT statement, then PROC PLAN uses the last plan generated to randomize the input data set (DATA=), saving the results to the output data set (OUT=). The output data set has the same form as the input data set but has modified values for the variables that correspond to factors (see the "Output Data Sets" section on page 3346 for details). Values for variables not corresponding to factors are transferred without change.

factor-value-settings

specify the values input or output for the factors in the design. The form for *factor-value-settings* is different when only an OUT= data set is specified and when both OUT= and DATA= data sets are specified. Both forms are discussed in the following section.

Factor-Value-Settings with Only an OUT= Data Set

When you specify only an OUT= data set, the form for each *factor-value-setting* specification is one of the following:

factor-name < **NVALS=***list-of-n-numbers* >
< **ORDERED | RANDOM** >

or

factor-name < **CVALS=***list-of-n-strings* >
< **ORDERED | RANDOM** >

where

factor-name is a factor in the last FACTORS statement preceding the OUTPUT statement.

NVALS= lists n numeric values for the factor. By default, the procedure uses NVALS=(1 2 3 $\cdots n$).

CVALS= lists n character strings for the factor. Each string can have up to 40 characters, and each string must be enclosed in quotes. **Warning:** When you use the CVALS= option, the variable created in the output data set has a length equal to the length of the longest string given as a value; shorter strings are padded with trailing blanks. For example, the values output for the first level of a two-level factor with the following two different specifications are not the same.

```
CVALS=('String 1' "String 2")

CVALS=('String 1' "A longer string")
```

The value output with the second specification is 'String 1' followed by seven blanks. In order to match two such values (for example, when merging two plans), you must use the TRIM function in the DATA step (refer to *SAS Language Reference: Dictionary*).

ORDERED | RANDOM specifies how values (those given with the NVALS= or CVALS= option, or the default values) are associated with the levels of a factor (the integers $1, 2, \ldots, n$). The default association type is ORDERED, for which the first value specified is output for a factor level setting of 1, the second value specified is output for a level of 2, and so on. You can also specify an association type of RANDOM, for which the levels are associated with the values in a random order. Specifying RANDOM is useful for randomizing crossed experiments (see the "Randomizing Designs" section on page 3351).

The following statements give an example of using the OUTPUT statement with only an OUT= data set and with both the NVALS= and CVALS= specifications.

```
proc plan;
   factors a=6 ordered b=3;
   output out=design a nvals=(10 to 60 by 10)
                     b cvals=('HSX' 'SB2' 'DNY');
run;
```

The DESIGN data set contains two variables, a and b. The values of the variable a are 10 when factor a equals 1, 20 when factor a equals 2, and so on. Values of the variable b are 'HSX' when factor b equals 1, 'SB2' when factor b equals 2, and 'DNY' when factor b equals 3.

Factor-Value-Settings with OUT= and DATA= Data Sets

If you specify an input data set with DATA=, then PROC PLAN assumes that each factor in the last plan generated corresponds to a variable in the input set. If the variable name is different from the name of the factor to which it corresponds, the two can be associated in the values specification by *input-variable-name = factor-name*

Then, the NVALS= or CVALS= specification can be used. The values given by NVALS= or CVALS= specify the input values as well as the output values for the corresponding variable.

Since the procedure assumes that the collection of input factor values constitutes a plan position description (see the "Output Data Sets" section on page 3346), the values must correspond to integers less than or equal to m, the number of values selected for the associated factor. If any input values do not correspond, then the collection does not define a plan position, and the corresponding observation is output without changing the values of any of the factor variables.

The following statements demonstrate the use of factor-value settings. The input SAS data set a contains variables Block and Plot, which are renamed Day and Hour, respectively.

```
proc plan;
   factors Day=7 Hour=6;
   output data=a out=b
      Block = Day  cvals=('Mon' 'Tue' 'Wed' 'Thu'
                          'Fri' 'Sat' 'Sun'      )
      Plot  = Hour;
run;
```

For another example of using both a DATA= and OUT= data set, see the "Randomly Assigning Subjects to Treatments" section on page 3337.

TREATMENTS Statement

TREATMENTS *factor-selections* **;**

The TREATMENTS statement specifies the *treatments* of the plan to generate, but it does not generate a plan. If you supply several FACTORS and TREATMENTS statements before the first RUN statement, the procedure uses only the last TREATMENTS specification and applies it to the plans generated by each of the FACTORS statements. The TREATMENTS statement has the same form as the FACTORS statement. The individual *factor-selections* also have the same form as in the FACTORS statement:

name=m $<$ **OF** *n* $>$ $<$ *selection-type* $>$

The procedure generates each *treatment* simultaneously with the lowest (that is, the most nested) factor in the last FACTORS statement. The m value for each *treatment* must be at least as large as the m for the most-nested factor.

The following statements give an example of using both a FACTORS and a TREATMENTS statement. First the FACTORS statement sets up the rows and columns of a 3×3 square (factors r and c). Then, the TREATMENTS statement augments the square with two cyclic treatments. The resulting design is a 3×3 Graeco-Latin square, a type of design useful in main-effects factorial experiments.

```
proc plan;
   factors r=3 ordered c=3 ordered;
   treatments a=3 cyclic
              b=3 cyclic 2;
run;
```

The resulting Graeco-Latin square design is reproduced below. Notice how the values of r and c are ordered (1, 2, 3) as requested.

```
r      --c--      --a--      --b--

1      1 2 3      1 2 3      1 2 3
2      1 2 3      2 3 1      3 1 2
3      1 2 3      3 1 2      2 3 1
```

Details

Using PROC PLAN Interactively

After specifying a design with a FACTORS statement and running PROC PLAN with a RUN statement, you can generate additional plans and output data sets without reinvoking PROC PLAN.

In PROC PLAN, all statements can be used interactively. You can execute statements singly or in groups by following the single statement or group of statements with a RUN statement.

If you use PROC PLAN interactively, you can end the procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of this statement is

```
quit;
```

When you use PROC PLAN interactively, additional RUN statements do not end the procedure but tell PROC PLAN to execute additional statements.

Output Data Sets

To understand how PROC PLAN creates output data sets, you need to look at how the procedure represents a plan. A plan is a list of values for all the factors, the values being chosen according to the factor-selection requests you specify. For example, consider the plan produced by the following statements:

```
proc plan seed=12345;
   factors a=3 b=2;
   run;
```

The plan as displayed by PROC PLAN is shown in Figure 55.5.

```
                          The PLAN Procedure

             Factor       Select       Levels      Order

             a                 3            3     Random
             b                 2            2     Random


                                 a      -b-

                                 2      2 1
                                 1      1 2
                                 3      2 1
```

Figure 55.5. A Simple Plan

The first cell of the plan has a=2 and b=2, the second a=2 and b=1, the third a=1 and b=1, and so on. If you output the plan to a data set with the OUTPUT statement, by default the output data set contains a numeric variable with that factor's name; the values of this numeric variable are the numbers of the successive levels selected for the factor in the plan. For example, the following statements produce Figure 55.6.

```
proc plan seed=12345;
   factors a=3 b=2;
   output out=out;
proc print data=out;
run;
```

```
                              Obs    a    b

                               1     2    2
                               2     2    1
                               3     1    1
                               4     1    2
                               5     3    2
                               6     3    1
```

Figure 55.6. Output Data Set from Simple Plan

Alternatively, you can specify the values that are output for a factor with the CVALS= or NVALS= option. Also, you can specify that the internal values be associated with the output values in a random order with the RANDOM option. See the "OUTPUT Statement" section on page 3343.

If you also specify an input data set (DATA=), each factor is associated with a variable in the DATA= data set. This occurs either implicitly by the factor and variable having the same name or explicitly as described in the specifications for the OUTPUT statement. In this case, the values of the variables corresponding to the factors are first read and then interpreted as describing the position of a cell in the plan. Then the respective values taken by the factors at that position are assigned to the variables in the OUT= data set. For example, consider the data set defined by the following statements.

```
data in;
   input a b;
   datalines;
1 1
2 1
3 1
;
```

Suppose you specify this data set as an input data set for the OUTPUT statement.

```
proc plan seed=12345;
   factors a=3 b=2;
   output out=out data=in;
proc print data=out;
   run;
```

PROC PLAN interprets the first observation as referring to the cell in the first row and column of the plan, since a=1 and b=1; likewise, the second observation is interpreted as the cell in the second row and first column, and the third observation as the cell in the third row and first column. In the output data set a and b have the values they have in the plan at these positions, as shown in Figure 55.7.

```
Obs    a    b

 1     2    2
 2     1    1
 3     3    2
```

Figure 55.7. Output Form of Input Data Set from Simple Plan

When the factors are random, this has the effect of randomizing the input data set in the same manner as the plan produced (see the "Randomizing Designs" section on page 3351 and the "Randomly Assigning Subjects to Treatments" section on page 3337).

Specifying Factor Structures

By appropriately combining features of the PLAN procedure, you can construct an extensive set of designs. The basic tools are the *factor-selections*, which are used in the FACTORS and TREATMENTS statements. Table 55.1 summarizes how the procedure interprets various *factor-selections* (assuming that the ORDERED option is not specified in the PROC PLAN statement).

Table 55.1. Factor Selection Interpretation

Form of Request	**Interpretation**	**Example**	**Results**
name=m	produce a random permutation of the integers $1, 2, \ldots, m$.	`t=15`	lists a random ordering of the numbers $1, 2, \ldots, 15$.
name=m cyclic	cyclically permute the integers $1, 2, \ldots, m$.	`t=5 cyclic`	selects the integers 1 to 5. On the next iteration, selects 2,3,4,5,1; then 3,4,5,1,2; and so on.
name=m of n	choose a random sample of m integers (without replacement) from the set of integers $1, 2, \ldots, n$.	`t=5 of 15`	lists a random selection of 5 numbers from 1 to 15. First, the procedure selects 5 numbers and then arranges them in random order.
name=m of n ordered	has the same effect as *name*=m ordered.	`t=5 of 15 ordered`	lists the integers 1 to 5 in increasing order (same as t=5 ordered).
name=m of n cyclic	permute m of the n integers.	`t=5 of 30 cyclic`	selects the integers 1 to 5. On the next iteration, selects 2,3,4,5,6; then 3,4,5,6,7; and so on. The 30th iteration 30,1,2,3,4; the 31st iteration produces 1,2,3,4,5; and so on.
name=m perm	produce a list of all permutations of m integers.	`t=5 perm`	lists the integers 1,2,3,4,5 on the first iteration; on the second lists 1,2,3,5,4; and on the 119th iteration lists 5,4,3,1,2; and on the last (120th) lists 5,4,3,2,1.
name=m of n comb	choose combinations of m integers from n integers.	`t=3 of 5 comb`	lists all combinations of 5 choose 3 integers. The first iteration is 1,2,3; the second is 1,2,4; the third is 1,2,5; and so on until the last iteration 3,4,5.

Table 55.1. (continued)

Form of Request	**Interpretation**	**Example**	**Results**
name=m of n cyclic (*initial-block*)	permute m of the n integers, starting with the values specified in the *initial-block*.	`t=4 of 30 cyclic (2 10 15 18)`	selects the integers 2,10,15,18. On the next iteration, selects 3,11,16,19; then 4,12,17,20; and so on. The thirteenth iteration is 14,22,27,30; the fourteenth iteration is 15,23,28,1; and so on.
name=m of n cyclic (*initial-block*) *increment*	permute m of the n integers. Start with the values specified in the *initial-block*, then add the *increment* to each value.	`t=4 of 30 cyclic (2 10 15 18) 2`	selects the integers 2,10,15,18. On the next iteration, selects 4,12,17,20; then 6,14,19,22; and so on. The wrap occurs at the eighth iteration. The eighth iteration is 16,24,29,2; and so on.

In Table 55.1, in order for more than one iteration to appear in the plan, another *name*=*j* factor selection (with $j > 1$) must precede the example factor selection. For example, the following statements produce six of the iterations described in the last entry of Table 55.1.

```
proc plan;
   factors c=6 ordered t=4 of 30 cyclic (2 10 15 18) 2;
run;
```

The following statements create a randomized complete block design and output the design to a data set.

```
proc plan ordered;
   factors blocks=3 cell=5;
   treatments t=5 random;
   output out=rcdb;
run;
```

Table 55.2 lists other kinds of experiment designs that can be constructed by PROC PLAN, along with section and page references for them in this chapter.

Table 55.2. Experimental Design Examples

Design	Page Number
Completely randomized design	page 3337
Split-plot design	page 3352
Nested design	page 3353
Latin square design	page 3356
Generalized cyclic incomplete block design	page 3357

Randomizing Designs

In many situations, proper randomization is crucial for the validity of any conclusions to be drawn from an experiment. Randomization is used both to neutralize the effect of any unknown systematic biases that may be involved in the design as well as to provide a basis for the assumptions underlying the analysis.

You can use PROC PLAN to randomize an already-existing design: one produced by a previous call to PROC PLAN, perhaps, or a more specialized design taken from a standard reference such as Cochran and Cox (1957). The method is to specify the appropriate block structure in the FACTORS statement and then to specify the data set where the design is stored with the DATA= option in the OUTPUT statement. For an illustration of this method, see the "Randomly Assigning Subjects to Treatments" section on page 3337).

Two sorts of randomization are provided for, corresponding to the RANDOM factor selection and association types in the FACTORS and OUTPUT statements, respectively. Designs in which factors are completely nested (for example, block designs) should be randomized by specifying that the selection type of each factor is RANDOM in the FACTORS statement, which is the default (see Example 55.3 on page 3354). On the other hand, if the factors are crossed (for example, row-and-column designs), they should be randomized by one random reassignment of their values for the whole design. To do this, specify that the association type of each factor is RANDOM in the OUTPUT statement (see Example 55.4 on page 3356).

Displayed Output

The PLAN procedure displays

- the m value for each factor, which is the number of values to be selected
- the n value for each factor, which is the number of values to be selected from
- the selection type for each factor, as specified in the FACTORS statement
- the initial block and increment number for cyclic factors
- the factor value selections making up each plan

In addition, notes are written to the log giving the starting and ending values of the random number seed for each call to PROC PLAN.

ODS Table Names

PROC PLAN assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 55.3. ODS Tables Produced by PROC PLAN

ODS Table Name	Description	Statement
FInfo	General factor information	FACTOR & no TREATMENT
PFInfo	Plot factor information	FACTOR & TREATMENT
Plan	Computed plan	default
TFInfo	Treatment factor information	FACTOR & TREATMENT

Examples

Example 55.1. A Split-Plot Design

This plan is appropriate for a split-plot design with main plots forming a randomized complete block design. In this example, there are three blocks, four main plots per block, and two subplots per main plot. First, three random permutations (one for each of the blocks) of the integers 1, 2, 3, and 4 are produced. The four integers correspond to the four levels of the main plot factor a; the permutation determines how the levels of a are assigned to the main plots within a block. For each of these twelve numbers (four numbers per block for three blocks), a random permutation of the integers 1 and 2 is produced. Each two-integer permutation determines the assignment of the two levels of the subplot factor b within a main plot. The following statements produce Output 55.1.1:

```
title ’Split Plot Design’;
proc plan seed=37277;
   factors block=3 ordered a=4 b=2;
run;
```

Output 55.1.1. A Split-Plot Design

```
            Split Plot Design

            The PLAN Procedure

Factor      Select      Levels     Order

block            3           3    Ordered
a                4           4    Random
b                2           2    Random


           block        a      -b-

               1        4      2 1
                        3      2 1
                        1      2 1
                        2      2 1
               2        4      1 2
                        3      1 2
                        1      2 1
                        2      1 2
               3        4      2 1
                        2      2 1
                        3      2 1
                        1      2 1
```

Example 55.2. A Hierarchical Design

In this example, three plants are nested within four pots, which are nested within three houses. The FACTORS statement requests a random permutation of the numbers 1, 2, and 3 to choose Houses randomly. The second step requests a random permutation of the numbers 1, 2, 3, and 4 for each of those first three numbers to randomly assign Pots to Houses. Finally, the FACTORS statement requests a random permutation of 1, 2, and 3 for each of the twelve integers in the second set of permutations. This last step randomly assigns Plants to Pots. The following statements produce Output 55.2.1:

```
title 'Hierarchical Design';
proc plan seed=17431;
   factors Houses=3 Pots=4 Plants=3 / noprint;
   output out=nested;
run;

proc print data=nested;
run;
```

Output 55.2.1. A Hierarchical Design

```
                     Hierarchical Design

                 Obs    Houses    Pots    Plants

                   1       1        3        2
                   2       1        3        3
                   3       1        3        1
                   4       1        1        3
                   5       1        1        1
                   6       1        1        2
                   7       1        2        2
                   8       1        2        3
                   9       1        2        1
                  10       1        4        3
                  11       1        4        2
                  12       1        4        1
                  13       2        4        1
                  14       2        4        3
                  15       2        4        2
                  16       2        2        2
                  17       2        2        1
                  18       2        2        3
                  19       2        3        2
                  20       2        3        3
                  21       2        3        1
                  22       2        1        2
                  23       2        1        3
                  24       2        1        1
                  25       3        4        1
                  26       3        4        3
                  27       3        4        2
                  28       3        1        3
                  29       3        1        2
                  30       3        1        1
                  31       3        2        1
                  32       3        2        2
                  33       3        2        3
                  34       3        3        3
                  35       3        3        2
                  36       3        3        1
```

Example 55.3. An Incomplete Block Design

Jarrett and Hall (1978) give an example of a generalized cyclic design with good efficiency characteristics. The design consists of two replicates of 52 treatments in 13 blocks of size 8. The following statements use the PLAN procedure to generate this design in an appropriately randomized form and store it in a SAS data set. Then, the TABULATE procedure is used to display the randomized plan. The following statements produce Output 55.3.1 and Output 55.3.2:

```
title 'Generalized Cyclic Block Design';
proc plan seed=33373;
   treatments trtmts=8 of 52 cyclic (1 2 3 4 32 43 46 49) 4;
   factors blocks=13 plots=8;
   output out=c;
quit;

proc tabulate;
   class blocks plots;
   var trtmts;
   table blocks, plots*(trtmts*f=8.) / rts=8;
run;
```

Output 55.3.1. A Generalized Cyclic Block Design

```
                     Generalized Cyclic Block Design

                          The PLAN Procedure

                             Plot Factors

                 Factor      Select      Levels     Order

                 blocks          13          13    Random
                 plots            8           8    Random


                           Treatment Factors

   Factor      Select      Levels     Order     Initial Block / Increment

   trtmts           8          52    Cyclic     (1 2 3 4 32 43 46 49) / 4


           blocks      -----plots-----      ---------trtmts--------

               10      7 4 8 1 2 3 5 6       1  2  3  4 32 43 46 49
                8      1 2 4 3 8 6 5 7       5  6  7  8 36 47 50  1
                9      2 5 4 7 3 1 8 6       9 10 11 12 40 51  2  5
                6      4 2 6 8 3 7 1 5      13 14 15 16 44  3  6  9
                7      4 7 6 3 1 2 8 5      17 18 19 20 48  7 10 13
                4      4 8 1 5 3 6 7 2      21 22 23 24 52 11 14 17
                2      6 2 3 8 7 5 1 4      25 26 27 28  4 15 18 21
                3      6 2 3 1 7 4 5 8      29 30 31 32  8 19 22 25
                1      1 2 7 8 5 6 3 4      33 34 35 36 12 23 26 29
                5      5 7 6 8 4 3 1 2      37 38 39 40 16 27 30 33
               12      5 8 1 4 7 3 6 2      41 42 43 44 20 31 34 37
               13      3 5 1 8 4 2 6 7      45 46 47 48 24 35 38 41
               11      4 1 5 2 3 8 6 7      49 50 51 52 28 39 42 45
```

Output 55.3.2. A Generalized Cyclic Block Design

```
                        Generalized Cyclic Block Design

--------------------------------------------------------------------------------
|      |                                plots                                  |
|      |-----------------------------------------------------------------------|
|      |   1    |   2    |   3    |   4    |   5    |   6    |   7    |   8    |
|      |--------+--------+--------+--------+--------+--------+--------+--------|
|      | trtmts | trtmts | trtmts | trtmts | trtmts | trtmts | trtmts | trtmts |
|      |--------+--------+--------+--------+--------+--------+--------+--------|
|      |  Sum   |  Sum   |  Sum   |  Sum   |  Sum   |  Sum   |  Sum   |  Sum   |
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|blocks|        |        |        |        |        |        |        |        |
|------|        |        |        |        |        |        |        |        |
|1     |      33|      34|      26|      29|      12|      23|      35|      36|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|2     |      18|      26|      27|      21|      15|      25|       4|      28|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|3     |      32|      30|      31|      19|      22|      29|       8|      25|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|4     |      23|      17|      52|      21|      24|      11|      14|      22|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|5     |      30|      33|      27|      16|      37|      39|      38|      40|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|6     |       6|      14|      44|      13|       9|      15|       3|      16|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|7     |      48|       7|      20|      17|      13|      19|      18|      10|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|8     |       5|       6|       8|       7|      50|      47|       1|      36|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|9     |      51|       9|      40|      11|      10|       5|      12|       2|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|10    |       4|      32|      43|       2|      46|      49|       1|       3|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|11    |      50|      52|      28|      49|      51|      42|      45|      39|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|12    |      43|      37|      31|      44|      41|      34|      20|      42|
|------+--------+--------+--------+--------+--------+--------+--------+--------|
|13    |      47|      35|      45|      24|      46|      38|      41|      48|
--------------------------------------------------------------------------------
```

Example 55.4. A Latin Square Design

All of the preceding examples involve designs with completely nested block structures, for which PROC PLAN was especially designed. However, by appropriate coordination of its facilities, a much wider class of designs can be accommodated. A Latin square design is based on experimental units that have a row-and-column block structure. The following example uses the CYCLIC option for a treatment factor tmts to generate a simple 4×4 Latin square. Randomizing a Latin square design involves randomly permuting the row, column, and treatment values independently. In order to do this, use the RANDOM option in the OUTPUT statement of PROC PLAN. The example also uses *factor-value-settings* in the OUTPUT statement. The following statements produce Output 55.4.1:

```
title 'Latin Square Design';
proc plan seed=37430;
   factors rows=4 ordered cols=4 ordered / noprint;
   treatments tmts=4 cyclic;
   output out=g
          rows cvals=('Day 1' 'Day 2' 'Day 3' 'Day 4') random
          cols cvals=('Lab 1' 'Lab 2' 'Lab 3' 'Lab 4') random
          tmts nvals=(   0      100     250     450  ) random;
quit;

proc tabulate;
   class rows cols;
   var tmts;
   table rows, cols*(tmts*f=6.) / rts=8;
run;
```

Output 55.4.1. A Randomized Latin Square Design

```
                     Latin Square Design

            ------------------------------------
            |      |           cols            |
            |      |---------------------------|
            |      |Lab 1 |Lab 2 |Lab 3 |Lab 4 |
            |      |------+------+------+------|
            |      | tmts | tmts | tmts | tmts |
            |      |------+------+------+------|
            |      | Sum  | Sum  | Sum  | Sum  |
            |------+------+------+------+------|
            |rows  |      |      |      |      |
            |------|      |      |      |      |
            |Day 1 |     0|   250|   100|   450|
            |------+------+------+------+------|
            |Day 2 |   250|   450|     0|   100|
            |------+------+------+------+------|
            |Day 3 |   100|     0|   450|   250|
            |------+------+------+------+------|
            |Day 4 |   450|   100|   250|     0|
            ------------------------------------
```

Example 55.5. A Generalized Cyclic Incomplete Block Design

The following statements depict how to create an appropriately randomized generalized cyclic incomplete block design for v treatments (given by the value of t) in b blocks (given by the value of b) of size k (with values of p indexing the cells within a block) with initial block $(e_1\ e_2\ \cdots\ e_k)$ and increment number i.

```
factors b=b p=k ;
treatments t=k of v cyclic (e1 e2 ··· ek ) i ;
```

For example, the specification

```
proc plan seed=37430;
   factors b=10 p=4;
   treatments t=4 of 30 cyclic (1 3 4 26) 2;
run;
```

generates the generalized cyclic incomplete block design given in Example 1 of Jarrett and Hall (1978), which is given by the rows and columns of the plan associated with the treatment factor t in Output 55.5.1.

Output 55.5.1. A Generalized Cyclic Incomplete Block Design

```
                           The PLAN Procedure

                             Plot Factors

                 Factor      Select      Levels     Order

                 b               10          10    Random
                 p                4           4    Random


                           Treatment Factors

                                                         Initial Block
        Factor      Select      Levels     Order         / Increment

        t                4          30    Cyclic         (1 3 4 26) / 2


                         b      ---p---      -----t-----

                         2      2 3 1 4       1  3  4 26
                         1      3 2 4 1       3  5  6 28
                         3      2 3 4 1       5  7  8 30
                        10      4 2 3 1       7  9 10  2
                         9      4 1 2 3       9 11 12  4
                         4      1 3 2 4      11 13 14  6
                         5      1 2 4 3      13 15 16  8
                         8      3 2 4 1      15 17 18 10
                         7      2 4 1 3      17 19 20 12
                         6      2 1 4 3      19 21 22 14
```

Example 55.6. Permutations and Combinations

Occasionally, you may need to generate all possible permutations of n things, or all possible combinations of n things taken m at a time.

For example, suppose you are planning an experiment in cognitive psychology where you want to present four successive stimuli to each subject. You want to observe each permutation of the four stimuli. The following statements use PROC PLAN to create a data set containing all possible permutations of 4 numbers in random order.

```
title 'All Permutations of 1,2,3,4';
proc plan seed=60359;
   factors    Subject  = 24
              Order    = 4  ordered;
   treatments Stimulus = 4  perm;
   output out=Psych;
proc sort data=Psych out=Psych;
   by Subject Order;
proc tabulate formchar='           ' noseps;
   class Subject Order;
   var Stimulus;
   table Subject, Order*(Stimulus*f=8.)*sum=' ' / rts=9;
run;
```

The variable Subject is set at 24 levels because there are $4! = 24$ total permutations to be listed. If Subject> 24, the list repeats. Output 55.6.1 and Output 55.6.2 display the PROC PLAN output. Note that the variable Subject is listed in random order.

Output 55.6.1. List of Permutations

```
                      All Permutations of 1,2,3,4

                          The PLAN Procedure

                             Plot Factors

               Factor       Select      Levels     Order

               Subject          24          24    Random
               Order             4           4    Ordered


                          Treatment Factors

              Factor        Select      Levels     Order

              Stimulus           4           4    Perm
```

Output 55.6.2. List of Permutations

```
                        All Permutations of 1,2,3,4

                             The PLAN Procedure

                    Subject      -Order-      -Stimulus-

                          4      1 2 3 4      1  2  3  4
                         15      1 2 3 4      1  2  4  3
                         24      1 2 3 4      1  3  2  4
                          1      1 2 3 4      1  3  4  2
                          5      1 2 3 4      1  4  2  3
                         17      1 2 3 4      1  4  3  2
                         19      1 2 3 4      2  1  3  4
                         14      1 2 3 4      2  1  4  3
                          6      1 2 3 4      2  3  1  4
                         23      1 2 3 4      2  3  4  1
                          8      1 2 3 4      2  4  1  3
                          2      1 2 3 4      2  4  3  1
                         13      1 2 3 4      3  1  2  4
                         16      1 2 3 4      3  1  4  2
                         12      1 2 3 4      3  2  1  4
                         18      1 2 3 4      3  2  4  1
                         21      1 2 3 4      3  4  1  2
                          9      1 2 3 4      3  4  2  1
                         22      1 2 3 4      4  1  2  3
                         10      1 2 3 4      4  1  3  2
                          7      1 2 3 4      4  2  1  3
                         11      1 2 3 4      4  2  3  1
                          3      1 2 3 4      4  3  1  2
                         20      1 2 3 4      4  3  2  1
```

The output data set Psych contains 96 observations of the 3 variables (Subject, Order, and Stimulus). Sorting the output data set by Subject and by Order within Subject results in all possible permutations of Stimulus in random order. PROC TABULATE displays these permutations in Output 55.6.3.

Output 55.6.3. Randomized Permutations

```
        All Permutations of 1,2,3,4


                       Order

           1        2        3        4

        Stimulus Stimulus Stimulus Stimulus

Subject
1              1        3        4        2
2              2        4        3        1
3              4        3        1        2
4              1        2        3        4
5              1        4        2        3
6              2        3        1        4
7              4        2        1        3
8              2        4        1        3
9              3        4        2        1
10             4        1        3        2
11             4        2        3        1
12             3        2        1        4
13             3        1        2        4
14             2        1        4        3
15             1        2        4        3
16             3        1        4        2
17             1        4        3        2
18             3        2        4        1
19             2        1        3        4
20             4        3        2        1
21             3        4        1        2
22             4        1        2        3
23             2        3        4        1
24             1        3        2        4
```

As another example, suppose you have six alternative treatments, any four of which can occur together in a block (in no particular order). The following statements use PROC PLAN to create a data set containing all possible combinations of six numbers taken four at a time. In this case, you use ODS to create the data set.

```
title 'All Combinations of (6 Choose 4) Integers';
ods output Plan=Combinations;
proc plan;
   factors Block=15 ordered
           Treat= 4 of 6 comb;
run;
proc print data=Combinations noobs;
run;
```

The variable Block has 15 levels since there are a total of $6!/(4!2!) = 15$ combinations of four integers chosen from six integers. The data set formed by ODS from the displayed plan has one row for each block, with the four values of Treat corresponding to four different variables, as shown in Output 55.6.4.

Output 55.6.4. List of Combinations

```
                    All Combinations of (6 Choose 4) Integers

                                The PLAN Procedure

                    Factor      Select      Levels     Order

                    Block           15          15    Ordered
                    Treat            4           6    Comb


                                 Block      -Treat-

                                     1      1 2 3 4
                                     2      1 2 3 5
                                     3      1 2 3 6
                                     4      1 2 4 5
                                     5      1 2 4 6
                                     6      1 2 5 6
                                     7      1 3 4 5
                                     8      1 3 4 6
                                     9      1 3 5 6
                                    10      1 4 5 6
                                    11      2 3 4 5
                                    12      2 3 4 6
                                    13      2 3 5 6
                                    14      2 4 5 6
                                    15      3 4 5 6
```

Output 55.6.5. Combinations Data Set Created by ODS

```
                    All Combinations of (6 Choose 4) Integers

                    Block    Treat1    Treat2    Treat3    Treat4

                        1      1         2         3         4
                        2      1         2         3         5
                        3      1         2         3         6
                        4      1         2         4         5
                        5      1         2         4         6
                        6      1         2         5         6
                        7      1         3         4         5
                        8      1         3         4         6
                        9      1         3         5         6
                       10      1         4         5         6
                       11      2         3         4         5
                       12      2         3         4         6
                       13      2         3         5         6
                       14      2         4         5         6
                       15      3         4         5         6
```

References

Cochran, W. G. and Cox, G. M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons, Inc.

Jarrett, R. G. and Hall, W. B. (1978), "Generalized Cyclic Incomplete Block Designs," *Biometrika*, 65, 397–401.

Chapter 56
The PLS Procedure

Chapter Contents

Chapter 56
The PLS Procedure

Overview

The PLS procedure fits models using any one of a number of linear predictive methods, including *partial least squares* (PLS). Ordinary least squares regression, as implemented in SAS/STAT procedures such as PROC GLM and PROC REG, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for *new* observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called *factors* (also called *components*, *latent vectors*, or *latent variables*), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation.

Note that the name “partial least squares” also applies to a more general statistical method that is *not* implemented in this procedure. The partial least squares method was originally developed in the 1960s by the econometrician Herman Wold (1966) for modeling “paths” of causal relation between any number of “blocks” of variables. However, the PLS procedure fits only *predictive* partial least squares models, with one “block” of predictors and one “block” of responses. If you are interested in fitting more general path models, you should consider using the CALIS procedure.

Basic Features

The techniques implemented by the PLS procedure are

- principal components regression, which extracts factors to explain as much predictor sample variation as possible.
- reduced rank regression, which extracts factors to explain as much response variation as possible. This technique, also known as (maximum) redundancy analysis, differs from multivariate linear regression only when there are multiple responses.
- partial least squares regression, which balances the two objectives of explaining response variation and explaining predictor variation. Two different formulations for partial least squares are available: the original predictive method of Wold (1966) and the SIMPLS method of de Jong (1993).

The number of factors to extract depends on the data. Basing the model on more extracted factors improves the model fit to the observed data, but extracting too many factors can cause *over-fitting*, that is, tailoring the model too much to the current data, to the detriment of future predictions. The PLS procedure enables you to choose the number of extracted factors by *cross validation*, that is, fitting the model to part of the data, minimizing the prediction error for the unfitted part, and iterating with different portions of the data in the roles of fitted and unfitted. Various methods of cross validation are available, including one-at-a-time validation, and splitting the data into blocks. The PLS procedure also offers test set validation, where the model is fit to the entire primary input data set and the fit is evaluated over a distinct test data set.

You can use the general linear modeling approach of the GLM procedure to specify a model for your design, allowing for general polynomial effects as well as classification or ANOVA effects. You can save the model fit by the PLS procedure in a data set and apply it to new data by using the SCORE procedure.

Getting Started

Spectrometric Calibration

The example in this section illustrates basic features of the PLS procedure. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose that you are researching pollution in the Baltic Sea, and you would like to use the spectra of samples of sea water to determine the amounts of three compounds present in samples from the Baltic Sea: lignin sulfonate (ls: pulp industry pollution), humic acids (ha: natural forest products), and optical whitener from detergent (dt). Spectrometric calibration is a type of problem in which partial least squares can be very effective. The predictors are the spectra emission intensities at different frequencies in sample spectrum, and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples with known compositions are used. The calibration data consist of 16 samples of known concentrations of ls, ha, and dt, with spectra based on 27 frequencies (or, equivalently, wavelengths). The following statements create a SAS data set named Sample for these data.

```
data Sample;
   input obsnam $ v1-v27 ls ha dt @@;
   datalines;
EM1   2766 2610 3306 3630 3600 3438 3213 3051 2907 2844 2796
      2787 2760 2754 2670 2520 2310 2100 1917 1755 1602 1467
      1353 1260 1167 1101 1017         3.0110  0.0000   0.00
EM2   1492 1419 1369 1158  958  887  905  929  920  887  800
       710  617  535  451  368  296  241  190  157  128  106
        89   70   65   56   50         0.0000  0.4005   0.00
EM3   2450 2379 2400 2055 1689 1355 1109  908  750  673  644
       640  630  618  571  512  440  368  305  247  196  156
       120   98   80   61   50         0.0000  0.0000  90.63
EM4   2751 2883 3492 3570 3282 2937 2634 2370 2187 2070 2007
```

```
        1974 1950 1890 1824 1680 1527 1350 1206 1080  984  888
         810  732  669  630  582           1.4820  0.1580  40.00
EM5     2652 2691 3225 3285 3033 2784 2520 2340 2235 2148 2094
        2049 2007 1917 1800 1650 1464 1299 1140 1020  909  810
         726  657  594  549  507           1.1160  0.4104  30.45
EM6     3993 4722 6147 6720 6531 5970 5382 4842 4470 4200 4077
        4008 3948 3864 3663 3390 3090 2787 2481 2241 2028 1830
        1680 1533 1440 1314 1227           3.3970  0.3032  50.82
EM7     4032 4350 5430 5763 5490 4974 4452 3990 3690 3474 3357
        3300 3213 3147 3000 2772 2490 2220 1980 1779 1599 1440
        1320 1200 1119 1032  957           2.4280  0.2981  70.59
EM8     4530 5190 6910 7580 7510 6930 6150 5490 4990 4670 4490
        4370 4300 4210 4000 3770 3420 3060 2760 2490 2230 2060
        1860 1700 1590 1490 1380           4.0240  0.1153  89.39
EM9     4077 4410 5460 5857 5607 5097 4605 4170 3864 3708 3588
        3537 3480 3330 3192 2910 2610 2325 2064 1830 1638 1476
        1350 1236 1122 1044  963           2.2750  0.5040  81.75
EM10    3450 3432 3969 4020 3678 3237 2814 2487 2205 2061 2001
        1965 1947 1890 1776 1635 1452 1278 1128  981  867  753
         663  600  552  507  468           0.9588  0.1450 101.10
EM11    4989 5301 6807 7425 7155 6525 5784 5166 4695 4380 4197
        4131 4077 3972 3777 3531 3168 2835 2517 2244 2004 1809
        1620 1470 1359 1266 1167           3.1900  0.2530 120.00
EM12    5340 5790 7590 8390 8310 7670 6890 6190 5700 5380 5200
        5110 5040 4900 4700 4390 3970 3540 3170 2810 2490 2240
        2060 1870 1700 1590 1470           4.1320  0.5691 117.70
EM13    3162 3477 4365 4650 4470 4107 3717 3432 3228 3093 3009
        2964 2916 2838 2694 2490 2253 2013 1788 1599 1431 1305
        1194 1077  990  927  855           2.1600  0.4360  27.59
EM14    4380 4695 6018 6510 6342 5760 5151 4596 4200 3948 3807
        3720 3672 3567 3438 3171 2880 2571 2280 2046 1857 1680
        1548 1413 1314 1200 1119           3.0940  0.2471  61.71
EM15    4587 4200 5040 5289 4965 4449 3939 3507 3174 2970 2850
        2814 2748 2670 2529 2328 2088 1851 1641 1431 1284 1134
        1020  918  840  756  714           1.6040  0.2856 108.80
EM16    4017 4725 6090 6570 6354 5895 5346 4911 4611 4422 4314
        4287 4224 4110 3915 3600 3240 2913 2598 2325 2088 1917
        1734 1587 1452 1356 1257           3.1620  0.7012  60.00
;
```

Fitting a PLS Model

To isolate a few underlying spectral factors that provide a good predictive model, you can fit a PLS model to the 16 samples using the following SAS statements:

```
proc pls data=sample;
   model ls ha dt = v1-v27;
run;
```

By default, the PLS procedure extracts at most 15 factors. The procedure lists the amount of variation accounted for by each of these factors, both individual and cumulative; this listing is shown in Figure 56.1.

```
                                The PLS Procedure

                         Percent Variation Accounted for
                         by Partial Least Squares Factors

            Number of
            Extracted       Model Effects        Dependent Variables
              Factors     Current        Total     Current        Total

                    1     97.4607      97.4607     41.9155      41.9155
                    2      2.1830      99.6436     24.2435      66.1590
                    3      0.1781      99.8217     24.5339      90.6929
                    4      0.1197      99.9414      3.7898      94.4827
                    5      0.0415      99.9829      1.0045      95.4873
                    6      0.0106      99.9935      2.2808      97.7681
                    7      0.0017      99.9952      1.1693      98.9374
                    8      0.0010      99.9961      0.5041      99.4415
                    9      0.0014      99.9975      0.1229      99.5645
                   10      0.0010      99.9985      0.1103      99.6747
                   11      0.0003      99.9988      0.1523      99.8270
                   12      0.0003      99.9991      0.1291      99.9561
                   13      0.0002      99.9994      0.0312      99.9873
                   14      0.0004      99.9998      0.0065      99.9938
                   15      0.0002     100.0000      0.0062     100.0000
```

Figure 56.1. PLS Variation Summary

Note that all of the variation in both the predictors and the responses is accounted for by only 15 factors; this is because there are only 16 sample observations. More importantly, almost all of the variation is accounted for with even fewer factors—one or two for the predictors and three to eight for the responses.

Selecting the Number of Factors by Cross Validation

A PLS model is not complete until you choose the number of factors. You can choose the number of factors by using cross validation, in which the data set is divided into two or more groups. You fit the model to all groups except one, then you check the capability of the model to predict responses for the group omitted. Repeating this for each group, you then can measure the overall capability of a given form of the model. The Predicted REsidual Sum of Squares (PRESS) statistic is based on the residuals generated by this process.

To select the number of extracted factors by cross validation, you specify the CV= option with an argument that says which cross validation method to use. For example, a common method is split-sample validation, in which the different groups are comprised of every nth observation beginning with the first, every nth observation beginning with the second, and so on. You can use the CV=SPLIT option to specify split-sample validation with $n = 7$ by default, as in the following SAS statements:

```
proc pls data=sample cv=split;
   model ls ha dt = v1-v27;
run;
```

The resulting output is shown in Figure 56.2 and Figure 56.3.

```
                                The PLS Procedure

          Split-sample Validation for the Number of Extracted Factors

                            Number of        Root
                            Extracted        Mean
                              Factors       PRESS

                                    0    1.107747
                                    1    0.957983
                                    2    0.931314
                                    3    0.520222
                                    4    0.530501
                                    5    0.586786
                                    6    0.475047
                                    7    0.477595
                                    8    0.483138
                                    9    0.485739
                                   10     0.48946
                                   11    0.521445
                                   12    0.525653
                                   13    0.531049
                                   14    0.531049
                                   15    0.531049


                    Minimum root mean PRESS           0.4750
                    Minimizing number of factors           6
```

Figure 56.2. Split-Sample Validated PRESS Statistics for Number of Factors

```
                                The PLS Procedure

                       Percent Variation Accounted for
                       by Partial Least Squares Factors

           Number of
           Extracted        Model Effects        Dependent Variables
             Factors     Current        Total     Current        Total

                   1     97.4607      97.4607     41.9155      41.9155
                   2      2.1830      99.6436     24.2435      66.1590
                   3      0.1781      99.8217     24.5339      90.6929
                   4      0.1197      99.9414      3.7898      94.4827
                   5      0.0415      99.9829      1.0045      95.4873
                   6      0.0106      99.9935      2.2808      97.7681
```

Figure 56.3. PLS Variation Summary for Split-Sample Validated Model

The absolute minimum PRESS is achieved with six extracted factors. Notice, however, that this is not much smaller than the PRESS for three factors. By using the CVTEST option, you can perform a statistical model comparison suggested by van der Voet (1994) to test whether this difference is significant, as shown in the following SAS statements:

```
proc pls data=sample cv=split cvtest(seed=12345);
   model ls ha dt = v1-v27;
run;
```

The model comparison test is based on a rerandomization of the data. By default, the seed for this randomization is based on the system clock, but it is specified here. The resulting output is shown in Figure 56.4 and Figure 56.5.

```
                             The PLS Procedure

          Split-sample Validation for the Number of Extracted Factors

                  Number of        Root
                  Extracted        Mean                   Prob >
                    Factors       PRESS         T**2        T**2

                          0    1.107747     9.272858      0.0010
                          1    0.957983     10.62305      <.0001
                          2    0.931314     8.950878      <.0001
                          3    0.520222     5.133259      0.1430
                          4    0.530501     5.168427      0.1330
                          5    0.586786     6.437266      0.0150
                          6    0.475047            0      1.0000
                          7    0.477595     2.809763      0.4750
                          8    0.483138     7.189526      0.0110
                          9    0.485739     7.931726      0.0060
                         10     0.48946     6.612597      0.0140
                         11    0.521445     6.666235      0.0130
                         12    0.525653     7.092861      0.0070
                         13    0.531049     7.538298      0.0020
                         14    0.531049     7.538298      0.0020
                         15    0.531049     7.538298      0.0020


              Minimum root mean PRESS                          0.4750
              Minimizing number of factors                          6
              Smallest number of factors with p > 0.1               3
```

Figure 56.4. Testing Split-Sample Validation for Number of Factors

```
                             The PLS Procedure

                         Percent Variation Accounted for
                         by Partial Least Squares Factors

           Number of
           Extracted      Model Effects         Dependent Variables
             Factors    Current        Total    Current        Total

                   1    97.4607      97.4607    41.9155      41.9155
                   2     2.1830      99.6436    24.2435      66.1590
                   3     0.1781      99.8217    24.5339      90.6929
```

Figure 56.5. PLS Variation Summary for Tested Split-Sample Validated Model

The p-value of 0.1430 in comparing the cross-validated residuals from models with 6 and 3 factors indicates that the difference between the two models is insignificant;

therefore, the model with fewer factors is preferred. The variation summary shows that over 99% of the predictor variation and over 90% of the response variation are accounted for by the three factors.

Predicting New Observations

Now that you have chosen a three-factor PLS model for predicting pollutant concentrations based on sample spectra, suppose that you have two new samples. The following SAS statements create a data set containing the spectra for the new samples:

```
data newobs;
   input obsnam $ v1-v27 @@;
   datalines;
EM17  3933 4518 5637 6006 5721 5187 4641 4149 3789
      3579 3447 3381 3327 3234 3078 2832 2571 2274
      2040 1818 1629 1470 1350 1245 1134 1050  987
EM25  2904 2997 3255 3150 2922 2778 2700 2646 2571
      2487 2370 2250 2127 2052 1713 1419 1200  984
       795  648  525  426  351  291  240  204  162
;
```

You can apply the PLS model to these samples to estimate pollutant concentration. To do so, append the new samples to the original 16, and specify that the predicted values for all 18 be output to a data set, as shown in the following statements:

```
data all; set sample newobs;
proc pls data=all nfac=3;
   model ls ha dt = v1-v27;
   output out=pred p=p_ls p_ha p_dt;
proc print data=pred;
   where (obsnam in ('EM17','EM25'));
   var obsnam p_ls p_ha p_dt;
run;
```

The new observations are not used in calculating the PLS model, since they have no response values. Their predicted concentrations are shown in Figure 56.6.

```
Obs    obsnam        p_ls        p_ha        p_dt

 17     EM17      2.54261     0.31877     81.4174
 18     EM25     -0.24716     1.37892     46.3212
```

Figure 56.6. Predicted Concentrations for New Observations

Syntax

The following statements are available in PROC PLS. Items within the brackets $<>$ are optional.

PROC PLS $<$ *options* $>$ **;**
  BY *variables* **;**
  CLASS *variables* $<$ */ option* $>$ **;**
  MODEL *dependent-variables = effects* $<$ */ options* $>$ **;**
  OUTPUT OUT= *SAS-data-set* $<$ *options* $>$ **;**

To analyze a data set, you must use the PROC PLS and MODEL statements. You can use the other statements as needed.

PROC PLS Statement

PROC PLS $<$ *options* $>$ **;**

You use the PROC PLS statement to invoke the PLS procedure and, optionally, to indicate the analysis data and method. The following options are available.

CENSCALE
lists the centering and scaling information for each response and predictor.

CV=ONE
CV=SPLIT $<$ **(***n***)** $>$
CV=BLOCK $<$ **(***n***)** $>$
CV=RANDOM $<$ **(***cv-random-opts***)** $>$
CV=TESTSET(*SAS-data-set***)**
specifies the cross validation method to be used. By default, no cross validation is performed. The method CV=ONE requests one-at-a-time cross validation, CV=SPLIT requests that every nth observation be excluded, CV=BLOCK requests that n blocks of consecutive observations be excluded, CV=RANDOM requests that observations be excluded at random, and CV=TESTSET(*SAS-data-set*) specifies a test set of observations to be used for validation (formally, this is called “test set validation” rather than “cross validation”). You can, optionally, specify n for CV=SPLIT and CV=BLOCK; the default is $n = 7$. You can also specify the following optional *cv-random-options* in parentheses after the CV=RANDOM option:

NITER=*n* specifies the number of random subsets to exclude. The default value is 10.

NTEST=*n* specifies the number of observations in each random subset chosen for exclusion. The default value is one-tenth of the total number of observations.

SEED=*n* specifies an integer used to start the pseudo-random number generator for selecting the random test set. If you don’t specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer’s clock.

CVTEST $<$ **(***cvtest-options***)** $>$

specifies that van der Voet's (1994) randomization-based model comparison test be performed to test models with different numbers of extracted factors against the model that minimizes the predicted residual sum of squares; see the "Cross Validation" section on page 3384 for more information. You can also specify the following *cv-test-options* in parentheses after the CVTEST option:

PVAL=*n* specifies the cut-off probability for declaring an insignificant difference. The default value is 0.10.

STAT=*test-statistic* specifies the test statistic for the model comparison. You can specify either T2, for Hotelling's T^2 statistic, or PRESS, for the predicted residual sum of squares. The default value is T2.

NSAMP=*n* specifies the number of randomizations to perform. The default value is 1000.

SEED=*n* specifies the seed value for randomization generation (the clock time is used by default).

DATA=*SAS-data-set*

names the SAS data set to be used by PROC PLS. The default is the most recently created data set.

DETAILS

lists the details of the fitted model for each successive factor. The details listed are different for different extraction methods: see the "Displayed Output" section on page 3387 for more information.

METHOD=PLS $<$ **(** *PLS-options* **)** $>$
METHOD=SIMPLS
METHOD=PCR
METHOD=RRR

specifies the general factor extraction method to be used. The value PLS requests partial least squares, SIMPLS requests the SIMPLS method of de Jong (1993), PCR requests principal components regression, and RRR requests reduced rank regression. The default is METHOD=PLS. You can also specify the following optional *PLS-options* in parentheses after METHOD=PLS:

ALGORITHM=NIPALS | SVD | EIG | RLGW names the specific algorithm used to compute extracted PLS factors. NIPALS requests the usual iterative NIPALS algorithm, SVD bases the extraction on the singular value decomposition of $X'Y$, EIG bases the extraction on the eigenvalue decomposition of $Y'XX'Y$, and RLGW is an iterative approach that is efficient when there are many predictors (Ränner et al. 1994). ALGORITHM=SVD is the most accurate but least efficient approach; the default is ALGORITHM=NIPALS.

MAXITER=*n* specifies the maximum number of iterations for the NIPALS and RLGW algorithms. The default value is 200.

EPSILON=*n* specifies the convergence criterion for the NIPALS and RLGW algorithms. The default value is 10^{-12}.

Experimental

MISSING=NONE
MISSING=AVG
MISSING=EM $<$ **(** *EM-options* **)** $>$

specifies how observations with missing values are to be handled in computing the fit. The default is MISSING=NONE, for which observations with any missing variables (dependent or independent) are excluded from the analysis. MISSING=AVG specifies that the fit be computed by filling in missing values with the average of the nonmissing values for the corresponding variable. If you specify MISSING=EM then the procedure first computes the model with MISSING=AVG, then fills in missing values by their predicted values based on that model and computes the model again. You can also specify the following optional *EM-options* in parentheses after MISSING=EM:

MAXITER=*n* specifies the maximum number of iterations for the imputation/fit loop. The default value is 1. If you specify a large value of MAXITER= then the loop will iterate until it converges (as controlled by the EPSILON= option).

EPSILON=*n* specifies the convergence criterion for the imputation/fit loop. The default value for is 10^{-8}. This option is only effective if you specify a large value for the MAXITER= option.

NFAC=*n*

specifies the number of factors to extract. The default is $\min\{15, p, N\}$, where p is the number of predictors (the number of dependent variables for METHOD=RRR) and N is the number of runs (observations). This is probably more than you need for most applications. Extracting too many factors can lead to an over-fit model, one that matches the training data too well, sacrificing predictive ability. Thus, if you use the default NFAC= specification, you should also either use the CV= option to select the appropriate number of factors for the final model or consider the analysis to be preliminary and examine the results to determine the appropriate number of factors for a subsequent analysis.

NOCENTER

suppresses centering of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. See the "Centering and Scaling" section on page 3386 for more information.

NOCVSTDIZE

suppresses re-centering and re-scaling of the responses and predictors before each model is fit in the cross validation. See the "Centering and Scaling" section on page 3386 for more information.

NOPRINT

suppresses the normal display of results. This is useful when you want only the output statistics saved in a data set. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, “Using the Output Delivery System,” for more information.

NOSCALE

suppresses scaling of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. See the “Centering and Scaling” section on page 3386 for more information.

VARSCALE

specifies that continuous model variables should be centered and scaled prior to centering and scaling the model effects in which they are involved. The rescaling specified by the VARSCALE option may be more appropriate if the model involves cross products between model variables; however, the VARSCALE option still may not produce the model you expect. See the “Centering and Scaling” section on page 3386 for more information.

VARSS

lists, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PLS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PLS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < */ option* > **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

Any variable in the model that is not listed in the CLASS statement is assumed to be continuous. Continuous variables must be numeric.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

MODEL Statement

MODEL *response-variables* **=** *predictor-effects* < **/** *options* >**;**

The MODEL statement names the responses and the predictors, which determine the $\mathbf{Y}$ and $\mathbf{X}$ matrices of the model, respectively. Usually you simply list the names of the predictor variables as the model effects, but you can also use the effects notation of PROC GLM to specify polynomial effects and interactions; see the "Specification of Effects" section on page 1784 in Chapter 32, "The GLM Procedure," for further details. The MODEL statement is required. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run).

You can specify the following options in the MODEL statement after a slash (/).

INTERCEPT

By default, the responses and predictors are centered; thus, no intercept is required in the model. You can specify the INTERCEPT option to override the default.

SOLUTION

lists the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set keyword***=***names* < . . . *keyword***=***names* >**;**

You use the OUTPUT statement to specify a data set to receive quantities that can be computed for every input observation, such as extracted factors and predicted values. The following *keywords* are available:

PREDICTED	predicted values for responses
YRESIDUAL	residuals for responses
XRESIDUAL	residuals for predictors
XSCORE	extracted factors (X-scores, latent vectors, latent variables, T)
YSCORE	extracted responses (Y-scores, U)
STDY	standardized (centered and scaled) responses
STDX	standardized (centered and scaled) predictors
H	approximate leverage
PRESS	approximate predicted residuals
TSQUARE	scaled sum of squares of score values
STDXSSE	sum of squares of residuals for standardized predictors
STDYSSE	sum of squares of residuals for standardized responses

Suppose that there are N_x predictors and N_y responses and that the model has N_f selected factors.

- The keywords XRESIDUAL and STDX define an output variable for each predictor, so N_x names are required after each one.
- The keywords PREDICTED, YRESIDUAL, STDY, and PRESS define an output variable for each response, so N_y names are required after each of these keywords.
- The keywords XSCORE and YSCORE specify an output variable for each selected model factor. For these keywords, you provide only one base name, and the variables corresponding to each successive factor are named by appending the factor number to the base name. For example, if $N_f = 3$ then a specification of XSCORE=T would produce the variables T1, T2, and T3.
- Finally, the keywords H, TSQUARE, STDXSSE, and STDYSSE each specify a single output variable, so only one name is required after each of these keywords.

Details

Regression Methods

All of the predictive methods implemented in PROC PLS work essentially by finding linear combinations of the predictors (factors) to use to predict the responses linearly. The methods differ only in how the factors are derived, as explained in the following sections.

Partial Least Squares

Partial least squares (PLS) works by extracting one factor at a time. Let $X = X_0$ be the centered and scaled matrix of predictors and $Y = Y_0$ the centered and scaled matrix of response values. The PLS method starts with a linear combination $\mathbf{t} = X_0\mathbf{w}$ of the predictors, where $\mathbf{t}$ is called a *score* vector and $\mathbf{w}$ is its associated *weight* vector. The PLS method predicts both X_0 and Y_0 by regression on $\mathbf{t}$:

$$\begin{aligned} \hat{X}_0 &= \mathbf{t}\mathbf{p}', \text{ where } \mathbf{p}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'X_0 \\ \hat{Y}_0 &= \mathbf{t}\mathbf{c}', \text{ where } \mathbf{c}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'Y_0 \end{aligned}$$

The vectors $\mathbf{p}$ and $\mathbf{c}$ are called the X- and Y-*loadings*, respectively.

The specific linear combination $\mathbf{t} = X_0\mathbf{w}$ is the one that has maximum covariance $\mathbf{t}'\mathbf{u}$ with some response linear combination $\mathbf{u} = Y_0\mathbf{q}$. Another characterization is that the X- and Y-weights $\mathbf{w}$ and $\mathbf{q}$ are proportional to the first left and right singular vectors of the covariance matrix $X_0'Y_0$ or, equivalently, the first eigenvectors of $X_0'Y_0Y_0'X_0$ and $Y_0'X_0X_0'Y_0$, respectively.

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing X_0 and Y_0 with the X- and Y-residuals from the first factor

$$\begin{aligned} X_1 &= X_0 - \hat{X}_0 \\ Y_1 &= Y_0 - \hat{Y}_0 \end{aligned}$$

These residuals are also called the *deflated* X and Y blocks. The process of extracting a score vector and deflating the data matrices is repeated for as many extracted factors as are desired.

SIMPLS

Note that each extracted PLS factor is defined in terms of different X-variables X_i. This leads to difficulties in comparing different scores, weights, and so forth. The SIMPLS method of de Jong (1993) overcomes these difficulties by computing each score $\mathbf{t}_i = X\mathbf{r}_i$ in terms of the original (centered and scaled) predictors X. The SIMPLS X-weight vectors r_i are similar to the eigenvectors of $SS' = X'YY'X$, but they satisfy a different orthogonality condition. The $\mathbf{r}_1$ vector is just the first eigenvector $\mathbf{e}_1$ (so that the first SIMPLS score is the same as the first PLS score), but whereas the second eigenvector maximizes

$$\mathbf{e}_1' SS' \mathbf{e}_2 \text{ subject to } \mathbf{e}_1' \mathbf{e}_2 = 0$$

the second SIMPLS weight $\mathbf{r}_2$ maximizes

$$\mathbf{r}_1' SS' \mathbf{r}_2 \text{ subject to } \mathbf{r}_1' X' X \mathbf{r}_2 = \mathbf{t}_1' \mathbf{t}_2 = 0$$

The SIMPLS scores are identical to the PLS scores for one response but slightly different for more than one response; refer to de Jong (1993) for details. The X- and Y-loadings are defined as in PLS, but since the scores are all defined in terms of X, it is easy to compute the overall model coefficients B:

$$\begin{aligned}\hat{Y} &= \sum_i \mathbf{t_i} \mathbf{c_i}' \\ &= \sum_i X \mathbf{r_i} \mathbf{c_i}' \\ &= XB, \text{ where } B = RC'\end{aligned}$$

Principal Components Regression

Like the SIMPLS method, principal components regression (PCR) defines all the scores in terms of the original (centered and scaled) predictors X. However, unlike both the PLS and SIMPLS methods, the PCR method chooses the X-weights/X-scores without regard to the response data. The X-scores are chosen to explain as much variation in X as possible; equivalently, the X-weights for the PCR method are the eigenvectors of the predictor covariance matrix $X'X$. Again, the X- and Y-loadings are defined as in PLS; but, as in SIMPLS, it is easy to compute overall model coefficients for the original (centered and scaled) responses Y in terms of the original predictors X.

Reduced Rank Regression

As discussed in the preceding sections, partial least squares depends on selecting factors $\mathbf{t} = X\mathbf{w}$ of the predictors and $\mathbf{u} = Y\mathbf{q}$ of the responses that have maximum covariance, whereas principal components regression effectively ignores $\mathbf{u}$ and selects $\mathbf{t}$ to have maximum variance, subject to orthogonality constraints. In contrast, reduced rank regression selects $\mathbf{u}$ to account for as much variation in the *predicted* responses as possible, effectively ignoring the predictors for the purposes of factor extraction. In reduced rank regression, the Y-weights $\mathbf{q}_i$ are the eigenvectors of the covariance matrix $\hat{Y}_{\mathrm{LS}}' \hat{Y}_{\mathrm{LS}}$ of the responses predicted by ordinary least squares regression; the X-scores are the projections of the Y-scores $Y\mathbf{q}_i$ onto the X space.

Relationships Between Methods

When you develop a predictive model, it is important to consider not only the explanatory power of the model for current responses, but also how well sampled the predictive functions are, since this impacts how well the model can extrapolate to future observations. All of the techniques implemented in the PLS procedure work by extracting successive factors, or linear combinations of the predictors, that optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, principal components regression selects factors that explain as much predictor variation as possible, reduced rank regression selects factors that explain as much response variation as possible, and partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation.

To see the relationships between these methods, consider how each one extracts a single factor from the following artificial data set consisting of two predictors and one response:

```
data data;
   input x1 x2 y;
   datalines;
    3.37651  2.30716        0.75615
    0.74193 -0.88845        1.15285
    4.18747  2.17373        1.42392
    0.96097  0.57301        0.27433
   -1.11161 -0.75225       -0.25410
   -1.38029 -1.31343       -0.04728
    1.28153 -0.13751        1.00341
   -1.39242 -2.03615        0.45518
    0.63741  0.06183        0.40699
   -2.52533 -1.23726       -0.91080
    2.44277  3.61077       -0.82590
   ;

proc pls data=data nfac=1 method=rrr;
   title "Reduced Rank Regression";
   model y = x1 x2;
proc pls data=data nfac=1 method=pcr;
   title "Principal Components Regression";
   model y = x1 x2;
proc pls data=data nfac=1 method=pls;
   title "Partial Least Squares Regression";
   model y = x1 x2;
run;
```

The amount of model and response variation explained by the first factor for each method is shown in Figure 56.7 through Figure 56.9.

```
                          Reduced Rank Regression

                             The PLS Procedure

                      Percent Variation Accounted for by
                       Reduced Rank Regression Factors

            Number of
            Extracted       Model Effects        Dependent Variables
              Factors     Current        Total    Current        Total

                    1     15.0661      15.0661   100.0000     100.0000
```

Figure 56.7. Variation Explained by First Reduced Rank Regression Factor

```
                      Principal Components Regression

                             The PLS Procedure

             Percent Variation Accounted for by Principal Components

            Number of
            Extracted       Model Effects        Dependent Variables
              Factors     Current        Total    Current        Total

                    1     92.9996      92.9996     9.3787       9.3787
```

Figure 56.8. Variation Explained by First Principal Components Regression Factor

```
                      Partial Least Squares Regression

                             The PLS Procedure

                      Percent Variation Accounted for
                      by Partial Least Squares Factors

            Number of
            Extracted       Model Effects        Dependent Variables
              Factors     Current        Total    Current        Total

                    1     88.5357      88.5357    26.5304      26.5304
```

Figure 56.9. Variation Explained by First Partial Least Squares Regression Factor

Notice that, while the first reduced rank regression factor explains *all* of the response variation, it accounts for only about 15% of the predictor variation. In contrast, the first principal components regression factor accounts for most of the predictor variation (93%) but only 9% of the response variation. The first partial least squares factor accounts for only slightly less predictor variation than principal components but about three times as much response variation.

Figure 56.10 illustrates how partial least squares balances the goals of explaining response and predictor variation in this case.

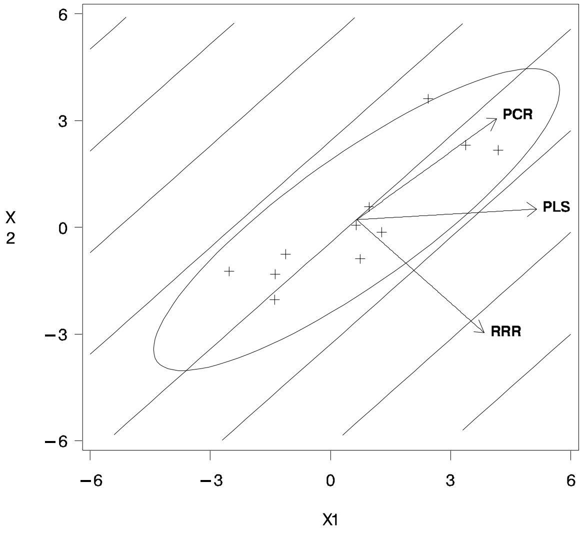


Figure 56.10. Depiction of First Factors for Three Different Regression Methods

The ellipse shows the general shape of the 11 observations in the predictor space, with the contours of increasing y overlaid. Also shown are the directions of the first factor for each of the three methods. Notice that, while the predictors vary most in the x1 = x2 direction, the response changes most in the orthogonal x1 = -x2 direction. This explains why the first principal component accounts for little variation in the response and why the first reduced rank regression factor accounts for little variation in the predictors. The direction of the first partial least squares factor represents a compromise between the other two directions.

Cross Validation

None of the regression methods implemented in the PLS procedure fit the observed data any better than ordinary least squares (OLS) regression; in fact, all of the methods approach OLS as more factors are extracted. The crucial point is that, when there are many predictors, OLS can *over-fit* the observed data; biased regression methods with fewer extracted factors can provide better predictability of *future* observations. However, as the preceding observations imply, the quality of the observed data fit cannot be used to choose the number of factors to extract; the number of extracted factors must be chosen on the basis of how well the model fits observations not involved in the modeling procedure itself.

One method of choosing the number of extracted factors is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted factors fit the other part of the data (the *test set*). This is called *test set validation*. However, it is rare that you have enough data to make both parts large enough for pure test set validation to be useful. Alternatively, you can make several different divisions of the observed data into training set and test set. This is called *cross validation*, and there are several different types. In *one-at-a-time* cross validation, the first observation is held out as a single-element test set, with all other observations as the training set; next, the second observation is held out, then the third, and so on. Another method is to hold out successive blocks of observations as test sets, for example, observations 1 through 7, then observations 8 through 14, and so on; this is known as *blocked* validation. A similar method is *split-sample* cross validation, in which successive groups of widely separated observations are held out as the test set, for example, observations {1, 11, 21, . . . }, then observations {2, 12, 22, . . . }, and so on. Finally, test sets can be selected from the observed data randomly; this is known as *random sample* cross validation.

Which validation you should use depends on your data. Test set validation is preferred when you have enough data to make a division into a sizable training set and test set that represent the predictive population well. You can specify that the number of extracted factors be selected by test set validation by using the CV=TESTSET(*data set*) option, where *data set* is the name of the data set containing the test set. If you do not have enough data for test set validation, you can use one of the cross validation techniques. The most common technique is one-at-a-time validation (which you can specify with the CV=ONE option or just the CV option), unless the observed data is serially correlated, in which case either blocked or split-sample validation may be more appropriate (CV=BLOCK or CV=SPLIT); you can specify the number of test sets in blocked or split-sample validation with a number in parentheses after the CV= option. Note that CV=ONE is the most computationally intensive of the cross validation methods, since it requires a recomputation of the PLS model for every input observation. Also, note that using random subset selection with CV=RANDOM may lead two different researchers to produce different PLS models on the same data (unless the same seed is used).

Whichever validation method you use, the number of factors chosen is usually the one that minimizes the predicted residual sum of squares (PRESS); this is the default choice if you specify any of the CV methods with PROC PLS. However, often models with fewer factors have PRESS statistics that are only marginally larger than the absolute minimum. To address this, van der Voet (1994) has proposed a statistical test for comparing the predicted residuals from different models; when you apply van der Voet's test, the number of factors chosen is the fewest with residuals that are insignificantly larger than the residuals of the model with minimum PRESS.

To see how van der Voet's test works, let $R_{i,jk}$ be the jth predicted residual for response k for the model with i extracted factors; the PRESS statistic is $\sum_{jk} R^2_{i,jk}$. Also, let $i_{\min}$ be the number of factors for which PRESS is minimized. The critical value for van der Voet's test is based on the differences between squared predicted residuals

$$D_{i,jk} \quad = \quad R^2_{i,jk} - R^2_{i_{\min},jk}$$

One alternative for the critical value is $C_i = \sum_{jk} D_{i,jk}$, which is just the difference between the PRESS statistics for i and $i_{\min}$ factors; alternatively, van der Voet suggests Hotelling's T^2 statistic $C_i = \mathbf{d}'_{i,\cdot} S_i^{-1} \mathbf{d}_{i,\cdot}$ where $\mathbf{d}_{i,\cdot}$ is the sum of the vectors $\mathbf{d}_{i,j} = \{D_{i,j1}, \ldots, D_{i,jN_y}\}'$ and S_i is the sum of squares and crossproducts matrix

$$S_i \quad = \quad \sum_j \mathbf{d}_{i,j} \mathbf{d}'_{i,j}$$

Virtually, the significance level for van der Voet's test is obtained by comparing C_i with the distribution of values that result from randomly exchanging $R^2_{i,jk}$ and $R^2_{i_{\min},jk}$. In practice, a Monte Carlo sample of such values is simulated and the significance level is approximated as the proportion of simulated critical values that are greater than C_i. If you apply van der Voet's test by specifying the CVTEST option, then, by default, the number of extracted factors chosen is the least number with an approximate significance level that is greater than 0.10.

Centering and Scaling

By default, the predictors and the responses are centered and scaled to have mean 0 and standard deviation 1. Centering the predictors and the responses ensures that the criterion for choosing successive factors is based on how much *variation* they explain, in either the predictors or the responses or both. (See the "Regression Methods" section on page 3380 for more details on how different methods explain variation.) Without centering, both the mean variable value and the variation around that mean are involved in selecting factors. Scaling serves to place all predictors and responses on an equal footing relative to their variation in the data. For example, if Time and Temp are two of the predictors, then scaling says that a change of $\mathrm{std}(\text{Time})$ in Time is roughly equivalent to a change of $\mathrm{std}(\text{Temp})$ in Temp.

Usually, both the predictors and responses should be centered and scaled. However, if their values already represent variation around a nominal or target value, then you can use the NOCENTER option in the PROC PLS statement to suppress centering. Likewise, if the predictors or responses are already all on comparable scales, then you can use the NOSCALE option to suppress scaling.

Note that, if the predictors involve crossproduct terms, then, by default, the variables are *not* standardized before standardizing the cross product. That is, if the ith values of two predictors are denoted x^1_i and x^2_i, then the default standardized ith value of the cross product is

$$\frac{x^1_i x^2_i - \mathrm{mean}(x^1_j x^2_j)}{\mathrm{std}(x^1_j x^2_j)}$$

If you want the cross product to be based instead on standardized variables

$$\frac{x^1_i - m^1}{s^1} \times \frac{x^2_i - m^2}{s^2}$$

where $m^k = \mathrm{mean}(x^k_j)$ and $s^k = \mathrm{std}(x^k_j)$ for $k = 1, 2$, then you should use the VARSCALE option in the PROC PLS statement. Standardizing the variables separately is usually a good idea, but unless the model also contains all cross products nested within each term, the resulting model may not be equivalent to a simple linear model in the same terms. To see this, note that a model involving the cross product of two standardized variables

$$\frac{x^1_i - m^1}{s^1} \times \frac{x^2_i - m^2}{s^2} = x^1_i x^2_i \frac{1}{s^1 s^2} - x^1_i \frac{m^2}{s^1 s^2} - x^2_i \frac{m^1}{s^1 s^2} + \frac{m^1 m^2}{s^1 s^2}$$

involves both the crossproduct term and the linear terms for the unstandardized variables.

When cross validation is performed for the number of effects, there is some disagreement among practitioners as to whether each cross validation training set should be retransformed. By default, PROC PLS does so, but you can suppress this behavior by specifying the NOCVSTDIZE option in the PROC PLS statement.

Missing Values

By default, PROC PLS handles missing values very simply. Observations with any missing independent variables (including all class variables) are excluded from the analysis, and no predictions are computed for such observations. Observations with no missing independent variables but any missing dependent variables are also excluded from the analysis, but predictions are computed.

However, the experimental MISSING= option on the PROC PLS statement provides more sophisticated ways of modeling in the presence of missing values. If you specify MISSING=AVG or MISSING=EM, then all observations in the input data set contribute to both the analysis and the OUTPUT OUT= data set. With MISSING=AVG, the fit is computed by filling in missing values with the average of the nonmissing values for the corresponding variable. With MISSING=EM, the procedure first computes the model with MISSING=AVG, then fills in missing values by their predicted values based on that model and computes the model again. Alternatively, you can specify MISSING=EM(MAXITER=*n*) with a large value of *n* in order to perform this imputation/fit loop until convergence.

Displayed Output

By default, PROC PLS displays just the amount of predictor and response variation accounted for by each factor.

If you perform a cross validation for the number of factors by specifying the CV option on the PROC PLS statement, then the procedure displays a summary of the cross validation for each number of factors, along with information about the optimal number of factors.

If you specify the DETAILS option on the PROC PLS statement, then details of the fitted model are displayed for each successive factor. These details include for each number of factors

- the predictor loadings
- the predictor weights
- the response weights
- the coded regression coefficients (for METHOD = SIMPLS, PCR, or RRR)

If you specify the CENSCALE option on the PROC PLS statement, then centering and scaling information for each response and predictor is displayed.

If you specify the VARSS option on the PROC PLS statement, the procedure displays, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

If you specify the SOLUTION option on the MODEL statement, then PROC PLS displays the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

ODS Table Names

PROC PLS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 56.1. ODS Tables Produced in PROC PLS

ODS Table Name	Description	Statement	Option
CVResults	Results of cross validation	PROC	CV
CenScaleParms	Parameter estimates for centered and scaled data	MODEL	SOLUTION
CodedCoef	Coded coefficients	PROC	DETAILS
ParameterEstimates	Parameter estimates for raw data	MODEL	SOLUTION
PercentVariation	Variation accounted for by each factor	PROC	default
ResidualSummary	Residual summary from cross validation	PROC	CV
XEffectCenScale	Centering and scaling information for predictor effects	PROC	CENSCALE
XLoadings	Loadings for independents	PROC	DETAILS
XVariableCenScale	Centering and scaling information for predictor variables	PROC	CENSCALE and VARSCALE
XWeights	Weights for independents	PROC	DETAILS
YVariableCenScale	Centering and scaling information for responses	PROC	CENSCALE
YWeights	Weights for dependents	PROC	DETAILS

Examples

Example 56.1. Examining Model Details

The following example, from Umetrics (1995), demonstrates different ways to examine a PLS model. The data come from the field of drug discovery. New drugs are developed from chemicals that are biologically active. Testing a compound for biological activity is an expensive procedure, so it is useful to be able to predict biological activity from cheaper chemical measurements. In fact, computational chemistry makes it possible to calculate certain chemical measurements without even making the compound. These measurements include size, lipophilicity, and polarity at various sites on the molecule. The following statements create a data set named penta, which contains these data.

```
data penta;
   input obsnam $ S1 L1 P1 S2 L2 P2
                  S3 L3 P3 S4 L4 P4
                  S5 L5 P5  log_RAI @@;
   n = _n_;
   datalines;
VESSK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          1.9607 -1.6324  0.5746  1.9607 -1.6324  0.5746
          2.8369  1.4092 -3.1398                    0.00
VESAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          1.9607 -1.6324  0.5746  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.28
VEASK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  1.9607 -1.6324  0.5746
          2.8369  1.4092 -3.1398                    0.20
VEAAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.51
VKAAK    -2.6931 -2.5271 -1.2871  2.8369  1.4092 -3.1398
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.11
VEWAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    2.73
VEAAP    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
         -1.2201  0.8829  2.2253                    0.18
VEHAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          2.4064  1.7438  1.1057  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    1.53
VAAAK    -2.6931 -2.5271 -1.2871  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   -0.10
GEAAK     2.2261 -5.3648  0.3049  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   -0.52
LEAAK    -4.1921 -1.0285 -0.9801  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.40
FEAAK    -4.9217  1.2977  0.4473  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.30
VEGGK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          2.2261 -5.3648  0.3049  2.2261 -5.3648  0.3049
          2.8369  1.4092 -3.1398                   -1.00
VEFAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.9217  1.2977  0.4473  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    1.57
VELAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.1921 -1.0285 -0.9801  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.59
AAAAA     0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902                   -0.10
```

```
AAYAA     0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
         -1.3944  2.3230  0.0139  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902                    0.46
AAWAA     0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902                    0.75
VAWAA    -2.6931 -2.5271 -1.2871  0.0744 -1.7333  0.0902
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902                    1.43
VAWAK    -2.6931 -2.5271 -1.2871  0.0744 -1.7333  0.0902
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    1.45
VKWAA    -2.6931 -2.5271 -1.2871  2.8369  1.4092 -3.1398
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902                    1.71
VWAAK    -2.6931 -2.5271 -1.2871 -4.7548  3.6521  0.8524
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                    0.04
VAAWK    -2.6931 -2.5271 -1.2871  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902 -4.7548  3.6521  0.8524
          2.8369  1.4092 -3.1398                    0.23
EKWAP     3.0777  0.3891 -0.0701  2.8369  1.4092 -3.1398
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
         -1.2201  0.8829  2.2253                    1.30
VKWAP    -2.6931 -2.5271 -1.2871  2.8369  1.4092 -3.1398
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
         -1.2201  0.8829  2.2253                    2.35
RKWAP     2.8827  2.5215 -3.4435  2.8369  1.4092 -3.1398
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
         -1.2201  0.8829  2.2253                    1.98
VEWVK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.7548  3.6521  0.8524 -2.6931 -2.5271 -1.2871
          2.8369  1.4092 -3.1398                    1.71
PGFSP    -1.2201  0.8829  2.2253  2.2261 -5.3648  0.3049
         -4.9217  1.2977  0.4473  1.9607 -1.6324  0.5746
         -1.2201  0.8829  2.2253                    0.90
FSPFR    -4.9217  1.2977  0.4473  1.9607 -1.6324  0.5746
         -1.2201  0.8829  2.2253 -4.9217  1.2977  0.4473
          2.8827  2.5215 -3.4435                    0.64
RYLPT     2.8827  2.5215 -3.4435 -1.3944  2.3230  0.0139
         -4.1921 -1.0285 -0.9801 -1.2201  0.8829  2.2253
          0.9243 -2.0921 -1.3996                    0.40
GGGGG     2.2261 -5.3648  0.3049  2.2261 -5.3648  0.3049
          2.2261 -5.3648  0.3049  2.2261 -5.3648  0.3049
          2.2261 -5.3648  0.3049                     .
;
data ptrain; set penta; if (n <= 15);
data ptest ; set penta; if (n >  15);
run;
```

You would like to study the relationship between these measurements and the activity of the compound, represented by the logarithm of the relative Bradykinin activating activity (log_RAI). Notice that these data consist of many predictors relative to the number of observations. Partial least squares is especially appropriate in this situation as a useful tool for finding a few underlying predictive factors that account for most of the variation in the response. Typically, the model is fit for part of the data (the “training” or “work” set), and the quality of the fit is judged by how well it predicts the other part of the data (the “test” or “prediction” set). For this example, the first 15 observations serve as the training set and the rest constitute the test set (refer to Ufkes et al. 1978, 1982).

When you fit a PLS model, you hope to find a few PLS factors that explain most of the variation in both predictors and responses. Factors that explain response variation provide good predictive models for new responses, and factors that explain predictor variation are well represented by the observed values of the predictors. The following statements fit a PLS model with two factors and save predicted values, residuals, and other information for each data point in a data set named outpls.

```
proc pls data=ptrain nfac=2;
   model log_RAI = S1-S5 L1-L5 P1-P5;
   output out=outpls predicted = yhat1
                     yresidual = yres1
                     xresidual = xres1-xres15
                     xscore    = xscr
                     yscore    = yscr;
run;
```

The PLS procedure displays a table, shown in Output 56.1.1, showing how much predictor and response variation is explained by each PLS factor.

Output 56.1.1. Amount of Training Set Variation Explained

```
                                The PLS Procedure

                          Percent Variation Accounted for
                          by Partial Least Squares Factors

         Number of
         Extracted         Model Effects         Dependent Variables
           Factors       Current        Total     Current        Total

                 1       16.9014      16.9014     89.6399      89.6399
                 2       12.7721      29.6735      7.8368      97.4767
```

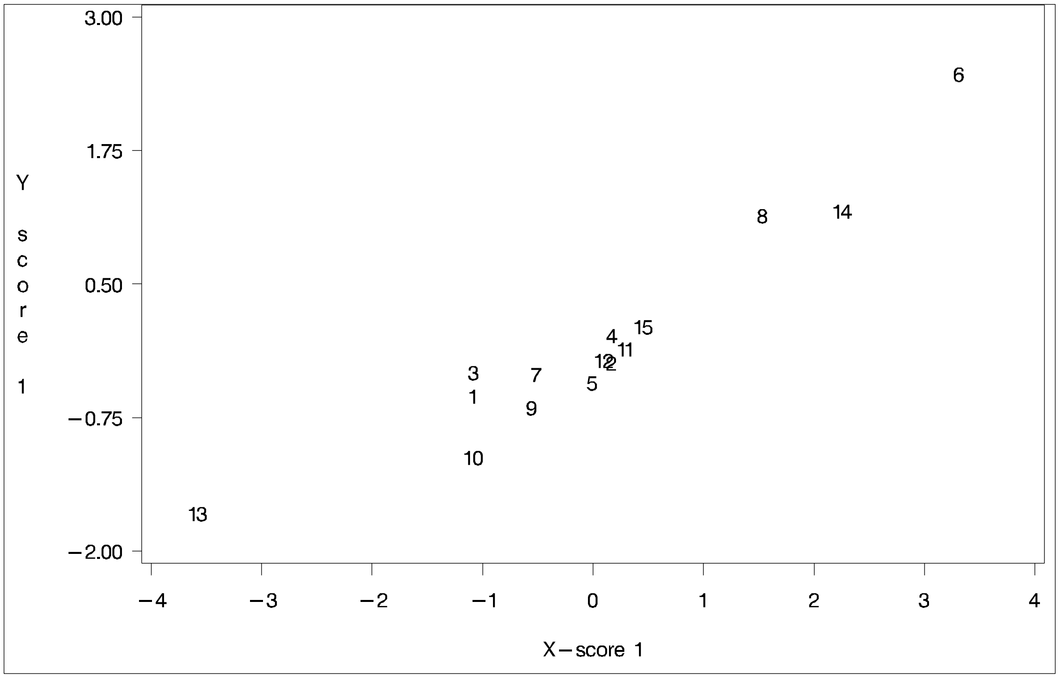
From Output 56.1.1, note that 97% of the response variation is already explained, but only 29% of the predictor variation is explained.

Partial least squares algorithms choose successive orthogonal factors that maximize the covariance between each X-score and the corresponding Y-score. For a good PLS model, the first few factors show a high correlation between the X- and Y-scores. The correlation usually decreases from one factor to the next. You can plot the X-scores versus the Y-scores for the first PLS factor using the following SAS statements.

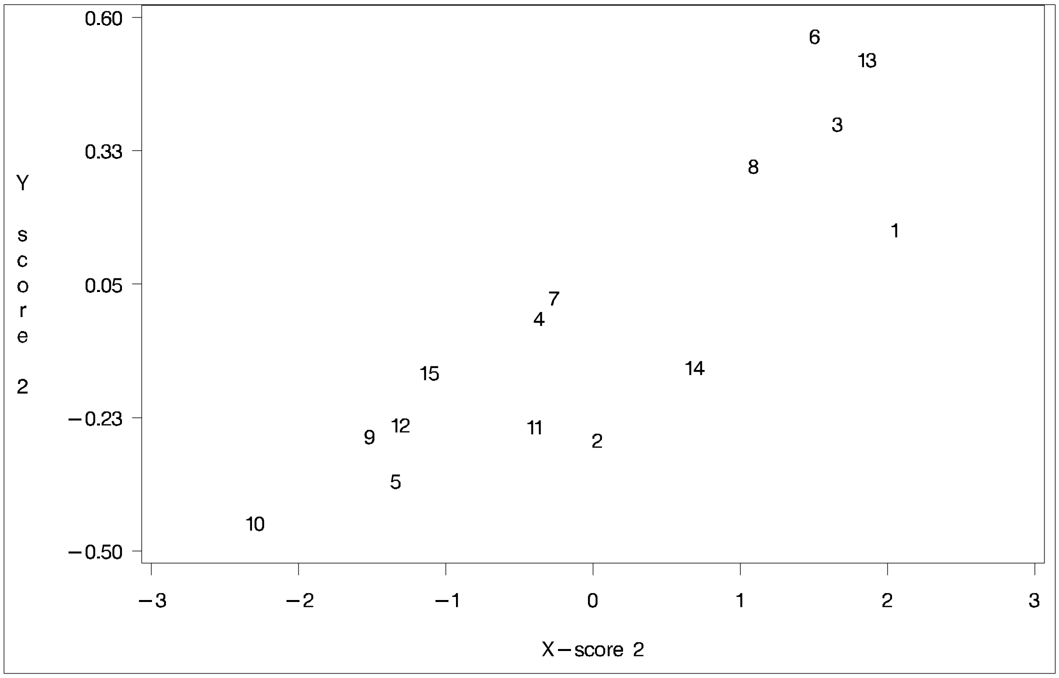
```
%let ifac = 1;
data pltanno; set outpls;
   length text $ 2;
   retain function 'label' position '5' hsys '3' xsys '2' ysys '2'
          color 'blue' style 'swissb';
   text=%str(n); x=xscr&ifac; y=yscr&ifac;
axis1 label=(angle=270 rotate=90 "Y score &ifac")
      major=(number=5) minor=none;
axis2 label=("X-score &ifac") minor=none;
symbol1 v=none i=none;
proc gplot data=outpls;
   plot yscr&ifac*xscr&ifac=1
      / anno=pltanno vaxis=axis1 haxis=axis2 frame cframe=ligr;
run;
```

By changing the macro variable ifac to 2 instead of 1, you can use the same statements to plot the X-scores versus the Y-scores for the second PLS factor. The resulting plots are shown in Output 56.1.2 and Output 56.1.3. The numbers on the plot represent the observation number in the penta data set.

Output 56.1.2. First X- and Y-scores for Penta-Peptide Model 1



Output 56.1.3. Second X- and Y-scores for Penta-Peptide Model 1



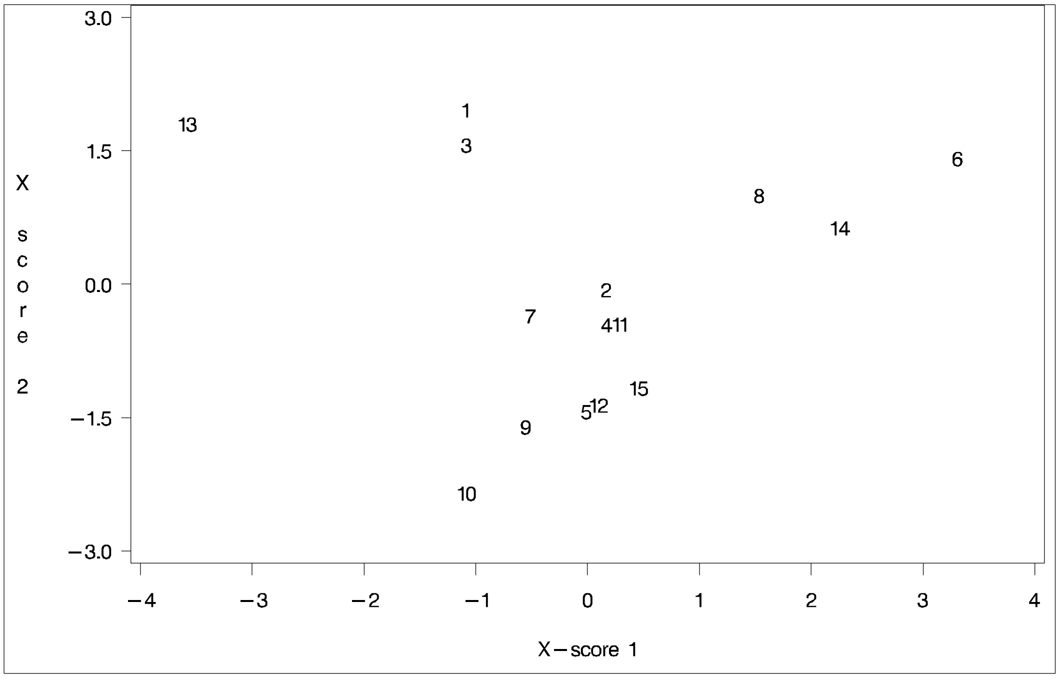
For this example, the figures show high correlation between X- and Y-scores for the first factor but somewhat lower correlation for the second factor.

You can also plot the X-scores against each other to look for irregularities in the data. You should look for patterns or clearly grouped observations. If you see a curved pattern, for example, you may want to add a quadratic term. Two or more groupings of observations indicate that it might be better to analyze the groups separately, perhaps by including classification effects in the model. The following SAS statements produce a plot of the first and second X-scores:

```
data pltanno; set outpls;
   length text $ 2;
   retain function ’label’ position ’5’ hsys ’3’ xsys ’2’ ysys ’2’
          color ’blue’ style ’swissb’;
   text=%str(n); x=xscr1; y=xscr2;
axis1 label=(angle=270 rotate=90 "X score 2")
      major=(number=5) minor=none;
axis2 label=("X-score 1") minor=none;
symbol1 v=none i=none;
proc gplot data=outpls;
   plot xscr2*xscr1=1
      / anno=pltanno vaxis=axis1 haxis=axis2 frame cframe=ligr;
run;
```

The plot is shown in Output 56.1.4.

Output 56.1.4. First and Second X-scores for Penta-Peptide Model 1



This plot appears to show most of the observations close together, with a few being more spread out with larger positive X-scores for factor 2. There are no clear grouping patterns, but observation 13 stands out; note that this observation is the most extreme on all three plots so far. This run may be overly influential in the PLS analysis; thus, you should check to make sure it is reliable.

Plots of the weights give the directions toward which each PLS factor projects. They show which predictors are most represented in each factor. Those predictors with small weights in absolute value are less important than those with large weights.

You can use the DETAILS option in the PROC PLS statement to display various model details, including the X-weights. You can then use the ODS statement to send the weights to an output data set, as follows:

```
ods output XWeights=xweights;
proc pls data=ptrain nfac=2 details;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;
```

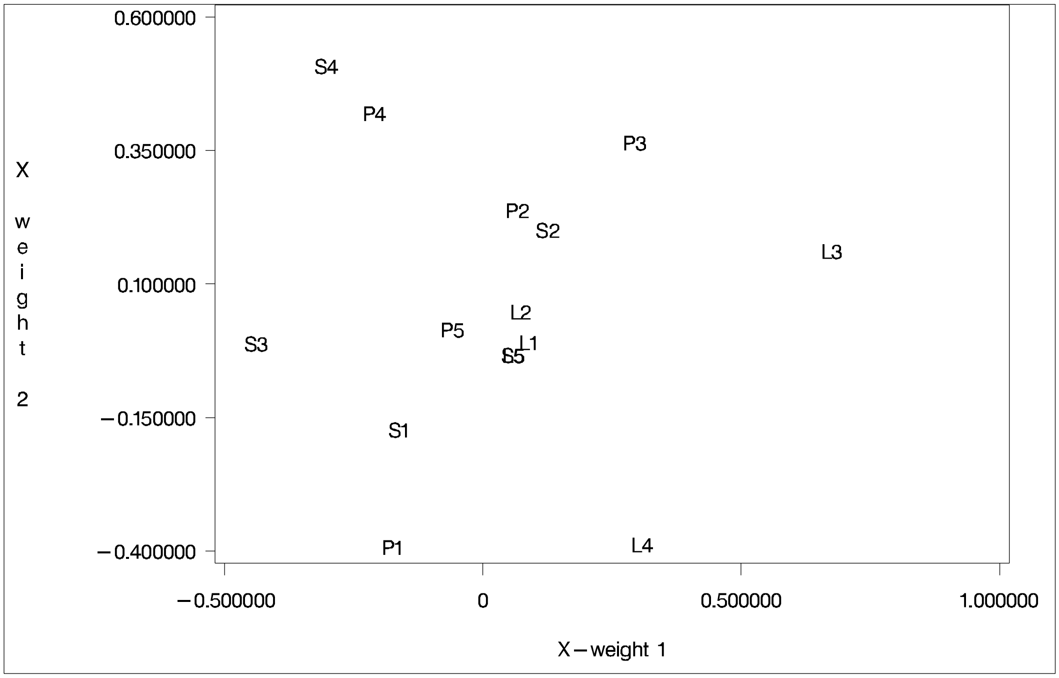
Once the X-weights are in a data set, you can use the following statements to plot the weights for the first two PLS factors against one another:

```
proc transpose data=xweights(drop=NumberOfFactors InnerRegCoef)
               out =xweights;
data xweights; set xweights;
```

```
   rename col1=w1 col2=w2;
data wt_anno; set xweights;
   length text $ 2;
   retain function 'label'
          position '5'
          hsys     '3'
          xsys     '2'
          ysys     '2'
          color    'blue'
          style    'swissb';
   text=%str(_name_); x=w1; y=w2;
run;

axis1 label=(angle=270 rotate=90 "X weight 2")
      major=(number=5) minor=none;
axis2 label=("X-weight 1") minor=none;
symbol1 v=none i=none;
proc gplot data=xweights;
   plot w2*w1=1 / anno=wt_anno vaxis=axis1
                  haxis=axis2 frame cframe=ligr;
run; quit;
```

The plot of the X-weights is shown in Output 56.1.5.

Output 56.1.5. First and Second X-weights for Penta-Peptide Model 1



The weights plot shows a cluster of X-variables that are weighted at nearly zero for both factors. These variables add little to the model fit, and removing them may improve the model's predictive capability.

To explore further which predictors can be eliminated from the analysis, you can look at the regression coefficients for the standardized data. Predictors with small coefficients (in absolute value) make a small contribution to the response prediction. Another statistic summarizing the contribution a variable makes to the model is the *Variable Importance for Projection* (VIP) of Wold (1994). Whereas the regression coefficients represent the importance each predictor has in the prediction of just the response, the VIP represents the value of each predictor in fitting the PLS model for both predictors and response. If a predictor has a relatively small coefficient (in absolute value) *and* a small value of VIP, then it is a prime candidate for deletion. Wold in Umetrics (1995) considers a value less than 0.8 to be “small” for the VIP. The following statements produce coefficients and the VIP.

```
/*
/  Put coefficients, weights, and R**2’s into data sets.
/-------------------------------------------------------*/
ods listing close;
ods output PercentVariation  = pctvar
           XWeights          = xweights
           CenScaleParms     = solution;
proc pls data=ptrain nfac=2 details;
   model log_RAI = S1 L1 P1
                   S2 L2 P2
                   S3 L3 P3
                   S4 L4 P4
                   S5 L5 P5 / solution;
run;
ods listing;

/*
/  Just reformat the coefficients.
/-------------------------------------------------------*/
data solution; set solution;
   format log_RAI 8.5;
   if (RowName = ’Intercept’) then delete;
   rename RowName = Predictor log_RAI = B;
run;

/*
/  Transpose weights and R**2’s.
/-------------------------------------------------------*/
data xweights; set xweights; _name_=’W’||trim(left(_n_));
data pctvar  ; set pctvar  ; _name_=’R’||trim(left(_n_));
proc transpose data=xweights(drop=NumberOfFactors InnerRegCoef)
               out =xweights;
proc transpose data=pctvar(keep=_name_ CurrentYVariation)
               out =pctvar;
run;

/*
/  Sum the normalized squared weights times the
/  normalized R**2’s.  The VIP is defined as the square
/  root of this weighted average times the number of
/  predictors.
/---------------------------------------------------------*/
proc sql;
   create table vip as
      select *,
          w1 /sqrt(uss(w1)) as wnorm1,
          w2 /sqrt(uss(w2)) as wnorm2
         from xweights left join pctvar(drop=_name_) on 1;
data vip; set vip; keep _name_ vip;
   array wnorm{2};
   array r{2};
   VIP = 0;
   do i = 1 to 2;
      VIP = VIP + r{i}*(wnorm{i}**2)/sum(of r1-r2);
      end;
   VIP = sqrt(VIP * 15);
data vipbpls; merge solution vip(drop=_name_);
proc print data=vipbpls;
run;
```

The output appears in Output 56.1.6.

Output 56.1.6. Estimated PLS Regression Coefficients and VIP (Model 1)

```
Obs    Predictor           B       VIP

  1       S1        -0.13831    0.61108
  2       L1         0.05720    0.31822
  3       P1        -0.19064    0.75127
  4       S2         0.12383    0.50482
  5       L2         0.05909    0.27123
  6       P2         0.09361    0.35927
  7       S3        -0.28415    1.57775
  8       L3         0.47131    2.43480
  9       P3         0.26613    1.13222
 10       S4        -0.09145    1.22255
 11       L4         0.12265    1.17994
 12       P4        -0.04878    0.88380
 13       S5         0.03320    0.21288
 14       L5         0.03320    0.21288
 15       P5        -0.03320    0.21288
```

For this data set, the variables L1, L2, P2, S5, L5, and P5 have small absolute coefficients and small VIP. Looking back at the weights plot in Output 56.1.5, you can see that these variables tend to be the ones near zero for both PLS factors. You should consider dropping these variables from the model.

Example 56.2. Examining Outliers

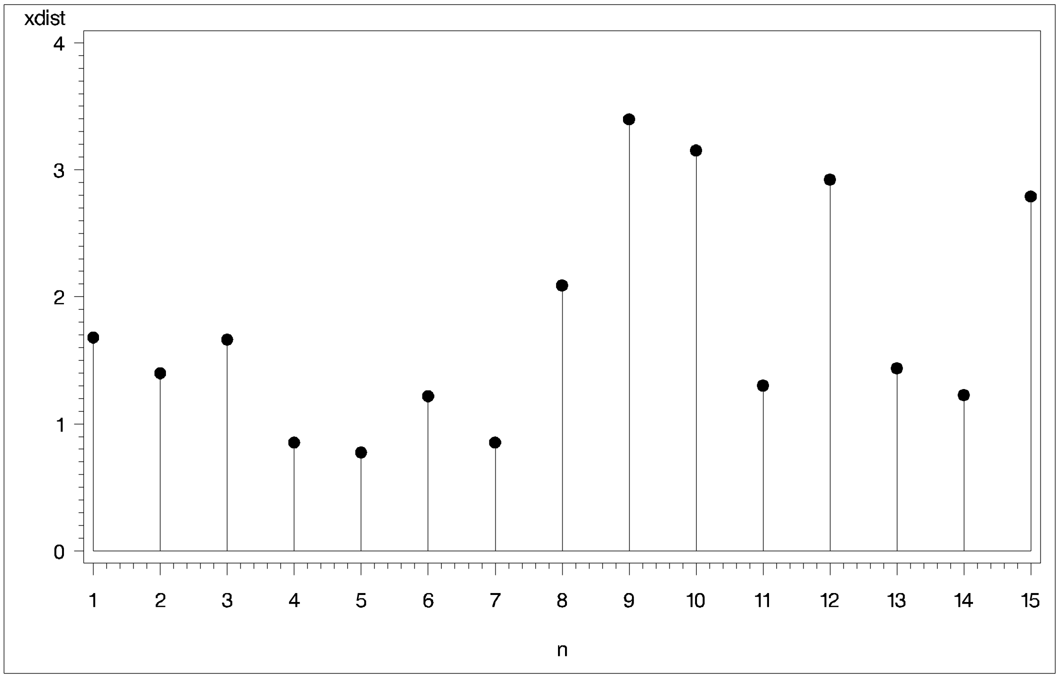
This example is a continuation of Example 56.1 on page 3388.

A PLS model effectively models both the predictors and the responses. In order to check for outliers, you should, therefore, look at the Euclidean distance from each point to the PLS model in both the standardized predictors and the standardized responses. No point should be dramatically farther from the model than the rest. If there is a group of points that are all farther from the model than the rest, they may have something in common, in which case they should be analyzed separately. The following statements compute and plot these distances to the reduced model, dropping variables L1, L2, P2, P4, S5, L5, and P5:

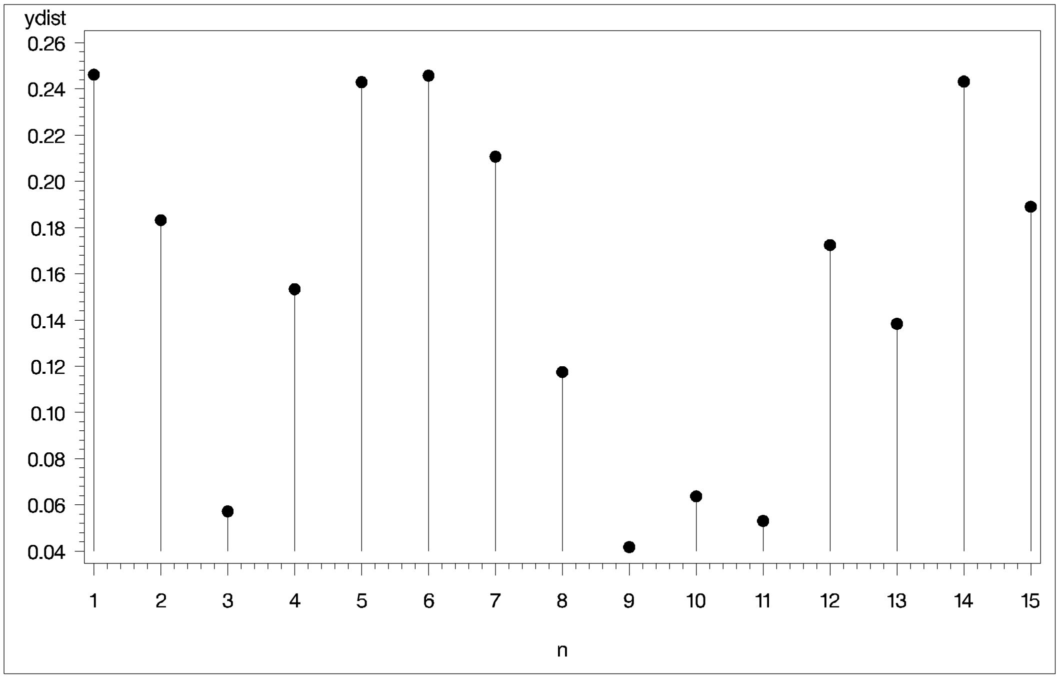
```
proc pls data=ptrain nfac=2 noprint;
   model log_RAI = S1    P1
                   S2
                   S3 L3 P3
                   S4 L4   ;
   output out=stdres stdxsse=stdxsse
                     stdysse=stdysse;
data stdres; set stdres;
   xdist = sqrt(stdxsse);
   ydist = sqrt(stdysse);
run;

symbol1 i=needles v=dot c=blue;
proc gplot data=stdres;
   plot xdist*n=1 / cframe=ligr;
proc gplot data=stdres;
   plot ydist*n=1 / cframe=ligr;
run;
```

The plots are shown in Output 56.2.1 and Output 56.2.2.

Output 56.2.1. Distances from the X-variables to the Model (Training Set)



Output 56.2.2. Distances from the Y-variables to the Model (Training Set)

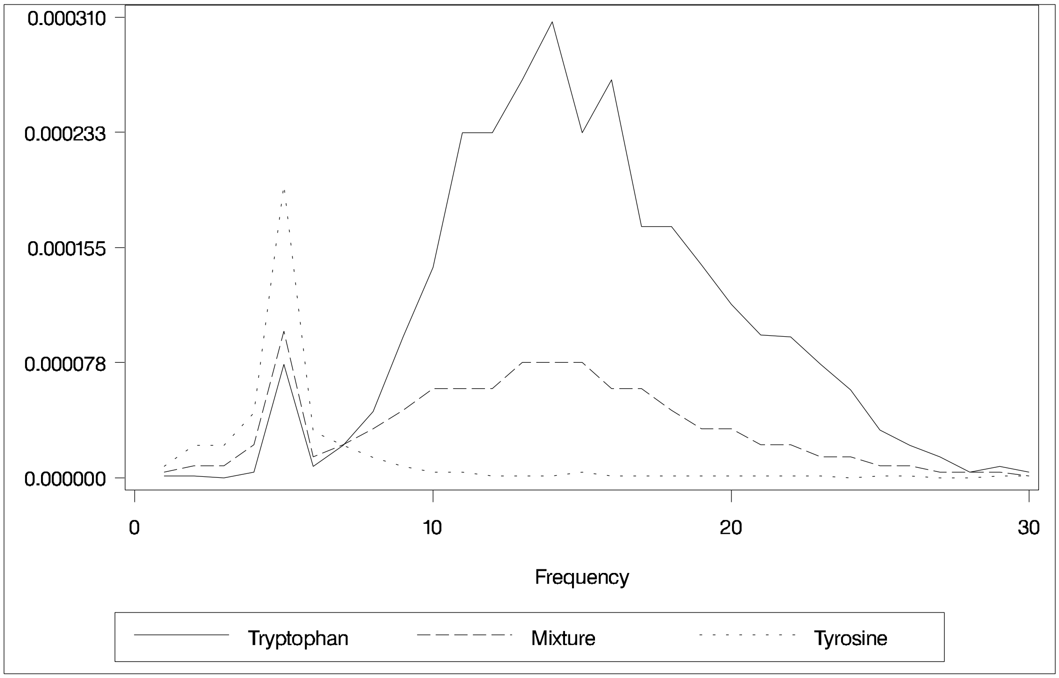


There appear to be no profound outliers in either the predictor space or the response space.

Example 56.3. Choosing a PLS Model by Test Set Validation

The following example demonstrates issues in spectrometric calibration. The data (Umetrics 1995) consist of spectrographic readings on 33 samples containing known concentrations of two amino acids, tyrosine and tryptophan. The spectra are measured at 30 frequencies across the overall range of frequencies. For example, Output 56.3.1 shows the observed spectra for three samples, one with only tryptophan, one with only tyrosine, and one with a mixture of the two, all at a total concentration of 10^{-6}.

Output 56.3.1. Spectra for Three Samples of Tyrosine and Tryptophan



Of the 33 samples, 18 are used as a training set and 15 as a test set. The data originally appear in McAvoy et al. (1989).

These data were created in a lab, with the concentrations fixed in order to provide a wide range of applicability for the model. You want to use a linear function of the logarithms of the spectra to predict the logarithms of tyrosine and tryptophan concentration, as well as the logarithm of the total concentration. Actually, because of the possibility of zeros in both the responses and the predictors, slightly different transformations are used. The following statements create SAS data sets containing the training and test data, named ftrain and ftest, respectively:

```
data ftrain;
   input obsnam $ tot tyr f1-f30 @@;
   try = tot - tyr;
   if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
   if (try) then try_log = log10(try); else try_log = -8;
   tot_log = log10(tot);
   datalines;
17mix35 0.00003 0
 -6.215 -5.809 -5.114 -3.963 -2.897 -2.269 -1.675 -1.235
 -0.900 -0.659 -0.497 -0.395 -0.335 -0.315 -0.333 -0.377
 -0.453 -0.549 -0.658 -0.797 -0.878 -0.954 -1.060 -1.266
 -1.520 -1.804 -2.044 -2.269 -2.496 -2.714
19mix35 0.00003 3E-7
 -5.516 -5.294 -4.823 -3.858 -2.827 -2.249 -1.683 -1.218
 -0.907 -0.658 -0.501 -0.400 -0.345 -0.323 -0.342 -0.387
 -0.461 -0.554 -0.665 -0.803 -0.887 -0.960 -1.072 -1.272
 -1.541 -1.814 -2.058 -2.289 -2.496 -2.712
21mix35 0.00003 7.5E-7
 -5.519 -5.294 -4.501 -3.863 -2.827 -2.280 -1.716 -1.262
 -0.939 -0.694 -0.536 -0.444 -0.384 -0.369 -0.377 -0.421
 -0.495 -0.596 -0.706 -0.824 -0.917 -0.988 -1.103 -1.294
 -1.565 -1.841 -2.084 -2.320 -2.521 -2.729
23mix35 0.00003 1.5E-6
 -5.294 -4.705 -4.262 -3.605 -2.726 -2.239 -1.681 -1.250
 -0.925 -0.697 -0.534 -0.437 -0.381 -0.359 -0.369 -0.426
 -0.499 -0.591 -0.701 -0.843 -0.925 -0.989 -1.109 -1.310
 -1.579 -1.852 -2.090 -2.316 -2.521 -2.743
25mix35 0.00003 3E-6
 -4.600 -4.069 -3.764 -3.262 -2.598 -2.191 -1.680 -1.273
 -0.958 -0.729 -0.573 -0.470 -0.422 -0.407 -0.422 -0.468
 -0.538 -0.639 -0.753 -0.887 -0.968 -1.037 -1.147 -1.357
 -1.619 -1.886 -2.141 -2.359 -2.585 -2.792
27mix35 0.00003 7.5E-6
 -3.812 -3.376 -3.026 -2.726 -2.249 -1.919 -1.541 -1.198
 -0.951 -0.764 -0.639 -0.570 -0.528 -0.525 -0.550 -0.606
 -0.689 -0.781 -0.909 -1.031 -1.126 -1.191 -1.303 -1.503
 -1.784 -2.058 -2.297 -2.507 -2.727 -2.970
29mix35 0.00003 0.000015
 -3.053 -2.641 -2.382 -2.194 -1.977 -1.913 -1.728 -1.516
 -1.317 -1.158 -1.029 -0.963 -0.919 -0.915 -0.933 -0.981
 -1.055 -1.157 -1.271 -1.409 -1.505 -1.546 -1.675 -1.880
 -2.140 -2.415 -2.655 -2.879 -3.075 -3.319
28mix35 0.00003 0.0000225
 -2.626 -2.248 -2.004 -1.839 -1.742 -1.791 -1.786 -1.772
 -1.728 -1.666 -1.619 -1.591 -1.575 -1.580 -1.619 -1.671
 -1.754 -1.857 -1.982 -2.114 -2.210 -2.258 -2.379 -2.570
 -2.858 -3.117 -3.347 -3.568 -3.764 -4.012
26mix35 0.00003 0.000027
 -2.370 -1.990 -1.754 -1.624 -1.560 -1.655 -1.772 -1.899
 -1.982 -2.074 -2.157 -2.211 -2.267 -2.317 -2.369 -2.460
 -2.545 -2.668 -2.807 -2.951 -3.030 -3.075 -3.214 -3.376
 -3.685 -3.907 -4.129 -4.335 -4.501 -4.599
24mix35 0.00003 0.0000285
 -2.326 -1.952 -1.702 -1.583 -1.507 -1.629 -1.771 -1.945
 -2.115 -2.297 -2.448 -2.585 -2.696 -2.808 -2.913 -3.030
 -3.163 -3.265 -3.376 -3.534 -3.642 -3.721 -3.858 -4.012
 -4.262 -4.501 -4.704 -4.822 -4.956 -5.292
22mix35 0.00003 0.00002925
 -2.277 -1.912 -1.677 -1.556 -1.487 -1.630 -1.791 -1.969
 -2.203 -2.437 -2.655 -2.844 -3.032 -3.214 -3.378 -3.503
 -3.646 -3.812 -3.958 -4.129 -4.193 -4.262 -4.415 -4.501
 -4.823 -5.111 -5.113 -5.294 -5.290 -5.294
20mix35 0.00003 0.0000297
 -2.266 -1.912 -1.688 -1.546 -1.500 -1.640 -1.801 -2.011
 -2.277 -2.545 -2.823 -3.094 -3.376 -3.572 -3.812 -4.012
 -4.262 -4.415 -4.501 -4.705 -4.823 -4.823 -4.956 -5.111
 -5.111 -5.516 -5.524 -5.806 -5.806 -5.806
18mix35 0.00003 0.00003
 -2.258 -1.900 -1.666 -1.524 -1.479 -1.621 -1.803 -2.043
 -2.308 -2.626 -2.895 -3.214 -3.568 -3.907 -4.193 -4.423
 -4.825 -5.111 -5.111 -5.516 -5.516 -5.516 -5.516 -5.806
 -5.806 -5.806 -5.806 -5.806 -6.210 -6.215
trp2    0.0001 0
 -5.922 -5.435 -4.366 -3.149 -2.124 -1.392 -0.780 -0.336
 -0.002  0.233  0.391  0.490  0.540  0.563  0.541  0.488
  0.414  0.313  0.203  0.063 -0.028 -0.097 -0.215 -0.411
 -0.678 -0.953 -1.208 -1.418 -1.651 -1.855
mix5    0.0001 0.00001
 -3.932 -3.411 -2.964 -2.462 -1.836 -1.308 -0.796 -0.390
 -0.076  0.147  0.294  0.394  0.446  0.460  0.443  0.389
  0.314  0.220  0.099 -0.033 -0.128 -0.197 -0.308 -0.506
 -0.785 -1.050 -1.313 -1.529 -1.745 -1.970
mix4    0.0001 0.000025
 -2.996 -2.479 -2.099 -1.803 -1.459 -1.126 -0.761 -0.424
 -0.144  0.060  0.195  0.288  0.337  0.354  0.330  0.274
  0.206  0.105 -0.009 -0.148 -0.242 -0.306 -0.424 -0.626
 -0.892 -1.172 -1.425 -1.633 -1.877 -2.071
mix3    0.0001 0.00005
 -2.128 -1.661 -1.344 -1.160 -0.996 -0.877 -0.696 -0.495
 -0.313 -0.165 -0.042  0.032  0.069  0.079  0.050 -0.006
 -0.082 -0.179 -0.295 -0.436 -0.523 -0.584 -0.706 -0.898
 -1.178 -1.446 -1.696 -1.922 -2.128 -2.350
mix6    0.0001 0.00009
 -1.140 -0.757 -0.497 -0.362 -0.329 -0.412 -0.513 -0.647
 -0.772 -0.877 -0.958 -1.040 -1.104 -1.162 -1.233 -1.317
 -1.425 -1.543 -1.661 -1.804 -1.877 -1.959 -2.034 -2.249
 -2.502 -2.732 -2.964 -3.142 -3.313 -3.576
;

data ftest;
   input obsnam $ tot tyr f1-f30 @@;
   try = tot - tyr;
   if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
   if (try) then try_log = log10(try); else try_log = -8;
   tot_log = log10(tot);
   datalines;
43trp6  1E-6 0
 -5.915 -5.918 -6.908 -5.428 -4.117 -5.103 -4.660 -4.351
 -4.023 -3.849 -3.634 -3.634 -3.572 -3.513 -3.634 -3.572
 -3.772 -3.772 -3.844 -3.932 -4.017 -4.023 -4.117 -4.227
 -4.492 -4.660 -4.855 -5.428 -5.103 -5.428
59mix6  1E-6 1E-7
 -5.903 -5.903 -5.903 -5.082 -4.213 -5.083 -4.838 -4.639
 -4.474 -4.213 -4.001 -4.098 -4.001 -4.001 -3.907 -4.001
 -4.098 -4.098 -4.206 -4.098 -4.213 -4.213 -4.335 -4.474
 -4.639 -4.838 -4.837 -5.085 -5.410 -5.410
51mix6  1E-6 2.5E-7
 -5.907 -5.907 -5.415 -4.843 -4.213 -4.843 -4.843 -4.483
 -4.343 -4.006 -4.006 -3.912 -3.830 -3.830 -3.755 -3.912
 -4.006 -4.001 -4.213 -4.213 -4.335 -4.483 -4.483 -4.642
 -4.841 -5.088 -5.088 -5.415 -5.415 -5.415
49mix6  1E-6 5E-7
 -5.419 -5.091 -5.091 -4.648 -4.006 -4.846 -4.648 -4.483
 -4.343 -4.220 -4.220 -4.220 -4.110 -4.110 -4.110 -4.220
 -4.220 -4.343 -4.483 -4.483 -4.650 -4.650 -4.846 -4.846
 -5.093 -5.091 -5.419 -5.417 -5.417 -5.907
53mix6  1E-6 7.5E-7
 -5.083 -4.837 -4.837 -4.474 -3.826 -4.474 -4.639 -4.838
 -4.837 -4.639 -4.639 -4.641 -4.641 -4.639 -4.639 -4.837
 -4.838 -4.838 -5.083 -5.082 -5.083 -5.410 -5.410 -5.408
 -5.408 -5.900 -5.410 -5.903 -5.900 -6.908
57mix6  1E-6 9E-7
 -5.082 -4.836 -4.639 -4.474 -3.826 -4.636 -4.638 -4.638
 -4.837 -5.082 -5.082 -5.408 -5.082 -5.080 -5.408 -5.408
 -5.408 -5.408 -5.408 -5.408 -5.408 -5.900 -5.900 -5.900
 -5.900 -5.900 -5.900 -5.900 -6.908 -6.908
41tyro6 1E-6 1E-6
 -5.104 -4.662 -4.662 -4.358 -3.705 -4.501 -4.662 -4.859
 -5.104 -5.431 -5.433 -5.918 -5.918 -5.918 -5.431 -5.918
 -5.918 -5.918 -5.918 -5.918 -5.918 -5.918 -5.918 -6.908
 -5.918 -5.918 -6.908 -6.908 -5.918 -5.918
28trp5  0.00001 0
 -5.937 -5.937 -5.937 -4.526 -3.544 -3.170 -2.573 -2.115
 -1.792 -1.564 -1.400 -1.304 -1.244 -1.213 -1.240 -1.292
 -1.373 -1.453 -1.571 -1.697 -1.801 -1.873 -2.008 -2.198
 -2.469 -2.706 -2.990 -3.209 -3.384 -3.601
37mix5  0.00001 1E-6
 -5.109 -4.865 -4.501 -4.029 -3.319 -3.070 -2.569 -2.207
 -1.895 -1.684 -1.516 -1.423 -1.367 -1.348 -1.374 -1.415
 -1.503 -1.596 -1.718 -1.839 -1.927 -1.997 -2.118 -2.333
 -2.567 -2.874 -3.106 -3.313 -3.579 -3.781
33mix5  0.00001 2.5E-6
 -4.366 -4.129 -3.781 -3.467 -3.037 -2.939 -2.593 -2.268
 -1.988 -1.791 -1.649 -1.565 -1.520 -1.509 -1.524 -1.580
 -1.665 -1.758 -1.882 -2.037 -2.090 -2.162 -2.284 -2.465
 -2.761 -3.037 -3.270 -3.520 -3.709 -3.937
31mix5  0.00001 5E-6
 -3.790 -3.373 -3.119 -2.915 -2.671 -2.718 -2.555 -2.398
 -2.229 -2.085 -1.971 -1.902 -1.860 -1.837 -1.881 -1.949
 -2.009 -2.127 -2.230 -2.381 -2.455 -2.513 -2.624 -2.827
 -3.117 -3.373 -3.586 -3.785 -4.040 -4.366
35mix5  0.00001 7.5E-6
 -3.321 -2.970 -2.765 -2.594 -2.446 -2.548 -2.616 -2.617
 -2.572 -2.550 -2.508 -2.487 -2.488 -2.487 -2.529 -2.593
 -2.688 -2.792 -2.908 -3.037 -3.149 -3.189 -3.273 -3.467
 -3.781 -4.029 -4.241 -4.501 -4.669 -4.865
39mix5  0.00001 9E-6
 -3.142 -2.812 -2.564 -2.404 -2.281 -2.502 -2.589 -2.706
 -2.842 -2.964 -3.068 -3.103 -3.182 -3.268 -3.361 -3.411
 -3.517 -3.576 -3.705 -3.849 -3.932 -3.932 -4.029 -4.234
 -4.501 -4.664 -4.860 -5.104 -5.431 -5.433
26tyro5 0.00001 0.00001
 -3.037 -2.696 -2.464 -2.321 -2.239 -2.444 -2.602 -2.823
 -3.144 -3.396 -3.742 -4.063 -4.398 -4.699 -4.893 -5.138
 -5.140 -5.461 -5.463 -5.945 -5.461 -5.138 -5.140 -5.138
 -5.138 -5.463 -5.461 -5.461 -5.461 -5.461
tyro2   0.0001 0.0001
```

```
    -1.081 -0.710 -0.470 -0.337 -0.327 -0.433 -0.602 -0.841
    -1.119 -1.423 -1.750 -2.121 -2.449 -2.818 -3.110 -3.467
    -3.781 -4.029 -4.241 -4.366 -4.501 -4.366 -4.501 -4.501
    -4.668 -4.668 -4.865 -4.865 -5.109 -5.111
   ;
```

The following statements fit a PLS model with 10 factors.

```
   proc pls data=ftrain nfac=10;
      model tot_log tyr_log try_log = f1-f30;
   run;
```

The table shown in Output 56.3.2 indicates that only three or four factors are required to explain almost all of the variation in both the predictors and the responses.

Output 56.3.2. Amount of Training Set Variation Explained

```
                                The PLS Procedure

                          Percent Variation Accounted for
                          by Partial Least Squares Factors

            Number of
            Extracted         Model Effects          Dependent Variables
              Factors       Current        Total       Current        Total

                    1       81.1654      81.1654       48.3385      48.3385
                    2       16.8113      97.9768       32.5465      80.8851
                    3        1.7639      99.7407       11.4438      92.3289
                    4        0.1951      99.9357        3.8363      96.1652
                    5        0.0276      99.9634        1.6880      97.8532
                    6        0.0132      99.9765        0.7247      98.5779
                    7        0.0052      99.9817        0.2926      98.8705
                    8        0.0053      99.9870        0.1252      98.9956
                    9        0.0049      99.9918        0.1067      99.1023
                   10        0.0034      99.9952        0.1684      99.2707
```

In order to choose the optimal number of PLS factors, you can explore how well models based on the training data with different numbers of factors fit the test data. To do so, use the CV=TESTSET option, with an argument pointing to the test data set ftest, as in the following statements:

```
   proc pls data=ftrain nfac=10 cv=testset(ftest)
                                cvtest(stat=press seed=12345);
      model tot_log tyr_log try_log = f1-f30;
   run;
```

The results of the test set validation are shown in Output 56.3.3. They indicate that, although five PLS factors give the minimum predicted residual sum of squares, the residuals for four factors are insignificantly different from those for five. Thus, the smaller model is preferred.

Output 56.3.3. Test Set Validation for the Number of PLS Factors

```
                               The PLS Procedure

            Test Set Validation for the Number of Extracted Factors

                        Number of        Root
                        Extracted        Mean    Prob >
                          Factors       PRESS     PRESS

                                0    3.056797    <.0001
                                1    2.630561    <.0001
                                2     1.00706    0.0070
                                3    0.664603    0.0020
                                4    0.521578    0.3800
                                5    0.500034    1.0000
                                6    0.513561    0.5100
                                7    0.501431    0.6870
                                8    1.055791    0.1530
                                9    1.435085    0.1010
                               10    1.720389    0.0320


              Minimum root mean PRESS                      0.5000
              Minimizing number of factors                      5
              Smallest number of factors with p > 0.1           4



                               The PLS Procedure

                        Percent Variation Accounted for
                        by Partial Least Squares Factors

           Number of
           Extracted        Model Effects        Dependent Variables
             Factors     Current        Total     Current        Total

                   1     81.1654     81.1654     48.3385     48.3385
                   2     16.8113     97.9768     32.5465     80.8851
                   3      1.7639     99.7407     11.4438     92.3289
                   4      0.1951     99.9357      3.8363     96.1652
```

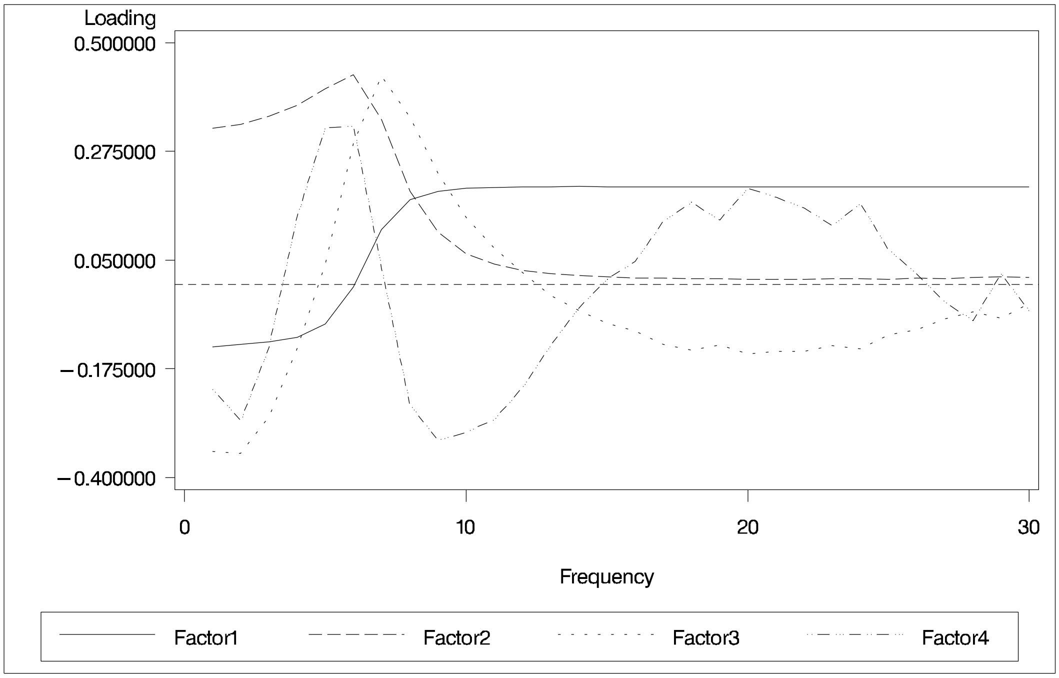
The factor loadings show how the PLS factors are constructed from the centered and scaled predictors. For spectral calibration, it is useful to plot the loadings against the frequency. In many cases, the physical meanings that can be attached to factor loadings help to validate the scientific interpretation of the PLS model. You can use the following statements to plot the loadings for the four PLS factors against frequency.

```
ods listing close;
ods output XLoadings=xloadings;
proc pls data=ftrain nfac=4 details method=pls;
   model tot_log tyr_log try_log = f1-f30;
run;
ods listing;
proc transpose data=xloadings(drop=NumberOfFactors)
               out =xloadings;
```

```
data xloadings; set xloadings;
   n = _n_;
   rename col1=Factor1 col2=Factor2
          col3=Factor3 col4=Factor4;
run;
goptions border;
axis1 label=("Loading"  ) major=(number=5) minor=none;
axis2 label=("Frequency")                  minor=none;
symbol1 v=none i=join c=red    l=1;
symbol2 v=none i=join c=green  l=1 /*l= 3*/;
symbol3 v=none i=join c=blue   l=1 /*l=34*/;
symbol4 v=none i=join c=yellow l=1 /*l=46*/;
legend1 label=none cborder=black;
proc gplot data=xloadings;
   plot (Factor1 Factor2 Factor3 Factor4)*n
      / overlay legend=legend1 vaxis=axis1
        haxis=axis2 vref=0 lvref=2 frame cframe=ligr;
run; quit;
```

The resulting plot is shown in Output 56.3.4.

Output 56.3.4. Predictor Loadings Across Frequencies



Notice that all four factors handle frequencies below and above about 7 or 8 differently. For example, the first factor is very nearly a simple contrast between the averages of the two sets of frequencies, and the second factor appears to be a weighted sum of only the frequencies in the first set.

References

Dijkstra, T. (1983), "Some Comments on Maximum Likelihood and Partial Least Squares Methods," *Journal of Econometrics*, 22, 67–90.

Dijkstra, T. (1985), *Latent Variables in Linear Stochastic Models: Reflections on Maximum Likelihood and Partial Least Squares Methods.* Second Edition, Amsterdam, The Netherlands: Sociometric Research Foundation.

Geladi, P, and Kowalski, B. (1986), "Partial Least-Squares Regression: A Tutorial," *Analytica Chimica Acta*, 185, 1–17.

Frank, I. and Friedman, J. (1993), "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109–135.

Haykin, S. (1994), *Neural Networks, a Comprehensive Foundation*, New York: Macmillan.

Helland, I. (1988), "On the Structure of Partial Least Squares Regression," *Communications in Statistics, Simulation and Computation*, 17(2), 581–607.

Hoerl, A. and Kennard, R. (1970), "Ridge Regression: Biased Estimation for Non-orthogonal Problems," *Technometrics*, 12, 55–67.

de Jong, S. and Kiers, H. (1992), "Principal Covariates Regression," *Chemometrics and Intelligent Laboratory Systems*, 14, 155–164.

de Jong, S. (1993), "SIMPLS: An Alternative Approach to Partial Least Squares Regression," *Chemometrics and Intelligent Laboratory Systems*, 18, 251–263.

Lindberg, W., Persson, J-A., and Wold, S. (1983), "Partial Least-Squares Method for Spectrofluorimetric Analysis of Mixtures of Humic Acid and Ligninsulfonate," *Analytical Chemistry*, 55, 643–648.

McAvoy, T. J., Wang, N. S., Naidu, S., Bhat, N., Gunter, J., and Simmons, M. (1989), "Interpreting Biosensor Data via Backpropagation," *International Joint Conference on Neural Networks*, 1, 227–233.

Naes, T. and Martens, H. (1985), "Comparison of Prediction Methods for Multicollinear Data," *Communications in Statistics, Simulation and Computation*, 14(3), 545–576.

Ränner, S., Lindgren, F., Geladi, P., and Wold, S. (1994), "A PLS Kernel Algorithm for Data Sets with Many Variables and Fewer Objects," *Journal of Chemometrics*, 8, 111–125.

Sarle, W.S. (1994), "Neural Networks and Statistical Models," in *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute, 1538–1550.

Shao, J. (1993), “Linear Model Selection by Cross-Validation,” *Journal of the American Statistical Association*, 88, 486–494.

Tobias, R. (1995), “An Introduction to Partial Least Squares Regression,” in *Proceedings of the Twentieth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 1250–1257.

Ufkes, J. G. R., Visser, B. J., Heuver, G., and Van Der Meer, C. (1978), “Structure-Activity Relationships of Bradykinin-Potentiating Peptides,” *European Journal of Pharmacology*, 50, 119.

Ufkes, J. G. R., Visser, B. J., Heuver, G., Wynne, H. J., and Van Der Meer, C. (1982), “Further Studies on the Structure-Activity Relationships of Bradykinin-Potentiating Peptides,” *European Journal of Pharmacology*, 79, 155.

Umetrics, Inc. (1995), *Multivariate Analysis (3-day course)*, Winchester, MA.

van den Wollenberg, A.L. (1977), “Redundancy Analysis—An Alternative to Canonical Correlation Analysis,” *Psychometrika*, 42, 207–219.

van der Voet, H. (1994), “Comparing the Predictive Accuracy of Models Using a Simple Randomization Test,” *Chemometrics and Intelligent Laboratory Systems*, 25, 313–323.

Wold, H. (1966), “Estimation of Principal Components and Related Models by Iterative Least Squares,” in *Multivariate Analysis*, ed. P. R. Krishnaiah, New York: Academic Press, 391–420.

Wold, S. (1994), “PLS for Multivariate Linear Modeling,” *QSAR: Chemometric Methods in Molecular Design. Methods and Principles in Medicinal Chemistry*, ed. H. van de Waterbeemd, Weinheim, Germany: Verlag-Chemie.

Chapter 57
The POWER Procedure

Chapter Contents

Chapter 57
The POWER Procedure

Overview

Power and sample size analysis enables you to optimize the resource usage and design of a study, improving chances of conclusive results with maximum efficiency. The POWER procedure performs prospective power and sample size analyses for a variety of goals, such as the following:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

Here *prospective* indicates that the analysis pertains to planning for a future study. This is in contrast to *retrospective* power analysis for a past study, which is not supported by the procedure.

A variety of statistical analyses are covered:

- t tests for means
- equivalence tests for means
- confidence intervals for means
- tests of binomial proportions
- multiple regression
- tests of correlation and partial correlation
- one-way analysis of variance
- rank tests for comparing two survival curves

For more complex linear models, see Chapter 34, "The GLMPOWER Procedure."

Input for PROC POWER includes the components considered in study planning:

- design
- statistical model and test
- significance level (alpha)
- surmised effects and variability
- power
- sample size

You designate one of these components by a missing value in the input, in order to identify it as the result parameter. The procedure calculates this result value over one or more scenarios of input values for all other components. Power and sample size are the most common result values, but for some analyses the result can be something else. For example, you can solve for the sample size of a single group for a two-sample t test.

In addition to tabular results, PROC POWER produces graphs. You can produce the most common types of plots easily with default settings and use a variety of options for more customized graphics. For example, you can control the choice of axis variables, axis ranges, number of plotted points, mapping of graphical features (such as color, line style, symbol and panel) to analysis parameters, and legend appearance.

The POWER procedure is one of several tools available in SAS/STAT software for power and sample size analysis. PROC GLMPOWER supports more complex linear models. The Power and Sample Size application provides a user interface and implements many of the analyses supported in the procedures.

The following sections of this chapter describe how to use PROC POWER and discuss the underlying statistical methodology. The "Getting Started" section on page 3412 introduces PROC POWER with simple examples of power computation for a one-sample t test and sample size determination for a two-sample t test. The "Syntax" section on page 3420 describes the syntax of the procedure. The "Details" section on page 3488 summarizes the methods employed by PROC POWER and provides details on several special topics. The "Examples" section on page 3536 illustrates the use of the POWER procedure with several applications.

For more discussion and examples on the main concepts in power and sample size analysis, refer to Castelloe (2000), Castelloe and O'Brien (2001), Muller and Benignus (1992), O'Brien and Muller (1993), and Lenth (2001).

Getting Started

Computing Power for a One-Sample t Test

Suppose you want to improve the accuracy of a machine used to print logos on sports jerseys. The machine has an inherently high variability, but its horizontal alignment can be adjusted. The operator agrees to pay for a costly adjustment if you can establish a non-zero mean horizontal displacement in either direction with high confidence. You have 150 jerseys at your disposal to measure, and you want to determine your chances of a significant result (power) using a one-sample t test with a 2-sided $\alpha = 0.05$.

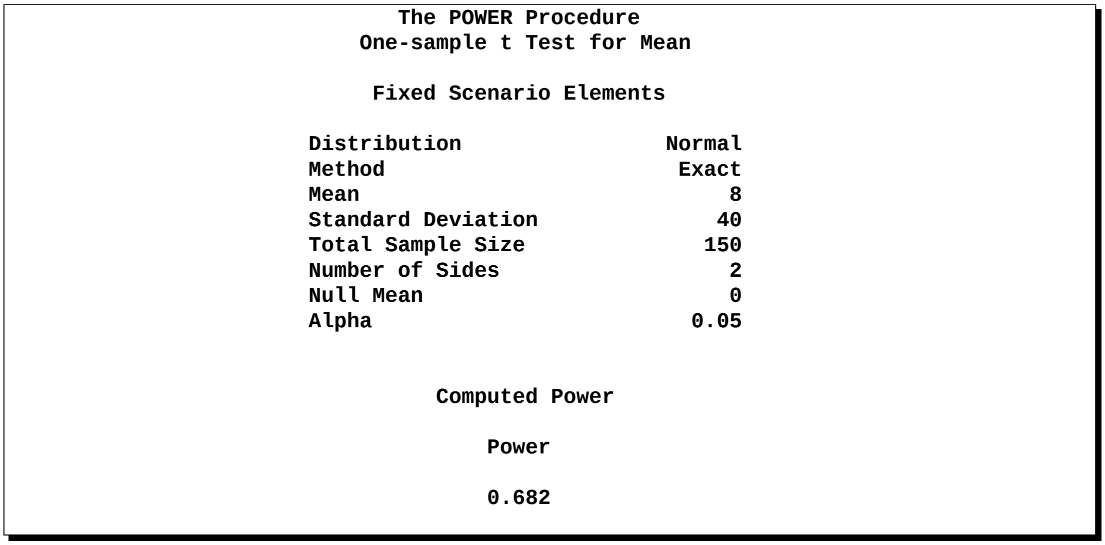
You decide that 8 mm is the smallest displacement worth addressing. Hence, you will assume a true mean of 8 in the power computation. Experience indicates that the standard deviation is about 40.

Use the ONESAMPLEMEANS statement in the POWER procedure to compute the power. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify your conjectures for the mean and standard deviation using the MEAN= and STDDEV= options and the sample size using the NTOTAL= option. The statements required to perform this analysis are as follows:

```
proc power;
   onesamplemeans
      mean   = 8
      ntotal = 150
      stddev = 40
      power  = .;
run;
```

Default values for the TEST=, DIST=, ALPHA=, NULL=, and SIDES= options specify a 2-sided t test for a mean of 0, assuming a normal distribution with a significance level of α = 0.05.

Figure 57.1 shows the output.



```
                    The POWER Procedure
                 One-sample t Test for Mean

                  Fixed Scenario Elements

             Distribution                Normal
             Method                       Exact
             Mean                             8
             Standard Deviation              40
             Total Sample Size              150
             Number of Sides                  2
             Null Mean                        0
             Alpha                         0.05


                      Computed Power

                          Power

                          0.682
```

Figure 57.1. Sample Size Analysis for One-Sample t Test

The power is about 0.68. In other words, there is about a 2/3 chance that the t test will produce a significant result demonstrating the machine's average off-center displacement. This probability depends on the assumptions for the mean and standard deviation.

Now, suppose you want to account for some of your uncertainty in conjecturing the true mean and standard deviation by evaluating the power for four scenarios using reasonable low and high values, 5 and 10 for the mean, and 30 and 50 for the standard deviation. Also, you may be able to measure more than 150 jerseys, and you would like to know under what circumstances you could get by with fewer. You want to plot power for sample sizes between 100 and 200 to visualize how sensitive the power is to changes in sample size for these four scenarios of means and standard deviations. The following statements perform this analysis:

```
proc power;
   onesamplemeans
      mean   = 5 10
      ntotal = 150
      stddev =  30 50
      power  = .;
   plot x=n min=100 max=200;
run;
```

The new mean and standard deviation values are specified using the MEAN= and STDDEV= options in the ONESAMPLEMEANS statement. The PLOT statement with X=N produces a plot with sample size on the x-axis. (The result parameter, in this case the power, is always plotted on the other axis.) The MIN= and MAX= options in the PLOT statement determine the sample size range.

Figure 57.2 shows the output, and Figure 57.3 shows the plot.

```
                        The POWER Procedure
                     One-sample t Test for Mean

                      Fixed Scenario Elements

               Distribution                Normal
               Method                       Exact
               Total Sample Size              150
               Number of Sides                  2
               Null Mean                        0
               Alpha                         0.05


                          Computed Power

                                   Std
                   Index    Mean    Dev    Power

                       1       5     30    0.527
                       2       5     50    0.229
                       3      10     30    0.982
                       4      10     50    0.682
```

Figure 57.2. Sample Size Analysis for One-Sample t Test with Input Ranges

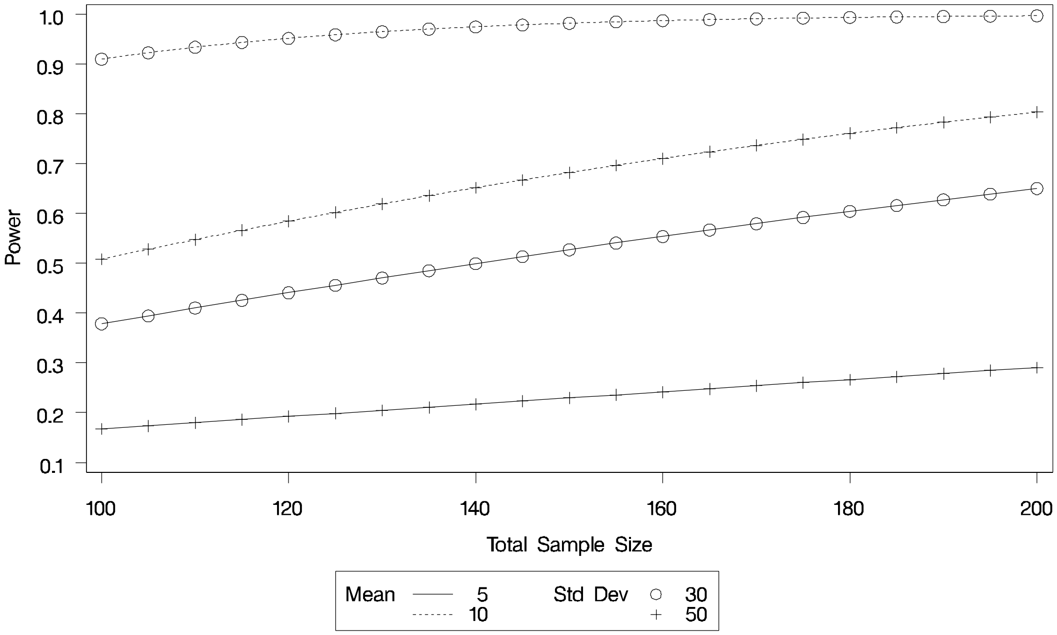


Figure 57.3. Plot of Power versus Sample Size for One-Sample t Test with Input Ranges

The power ranges from about 0.23 to 0.98 for a sample size of 150 depending on the mean and standard deviation. In Figure 57.3, the line style identifies the mean, and the plotting symbol identifies the standard deviation. The locations of plotting symbols indicate computed powers; the curves are linear interpolations of these points. The plot suggests sufficient power for a mean of 10 and standard deviation of 30 (for any of the sample sizes) but insufficient power for the other three scenarios.

Determining Required Sample Size for a Two-Sample t Test

In this example you want to compare two physical therapy treatments designed to increase muscle flexibility. You need to determine the number of patients required to achieve a power of at least 0.9 to detect a group mean difference in a two-sample t test. You will use $\alpha = 0.05$ (two-tailed).

The mean flexibility with the standard treatment (as measured on a scale of 1 to 20) is well known to be about 13 and is thought to be between 14 and 15 with the new treatment. You conjecture three alternative scenarios for the means,

1. $\mu_1 = 13, \mu_2 = 14$
2. $\mu_1 = 13, \mu_2 = 14.5$
3. $\mu_1 = 13, \mu_2 = 15$

You conjecture two scenarios for the common group standard deviation:

1. $\sigma = 1.2$
2. $\sigma = 1.7$

You also want to try three weighting schemes:

1. equal group sizes (balanced, or 1:1)
2. twice as many patients with the new treatment (1:2)
3. three times as many patients with the new treatment (1:3)

This makes $3 \times 2 \times 3 = 18$ scenarios in all.

Use the TWOSAMPLEMEANS statement in the POWER procedure to determine the sample sizes required to give 90% power for each of these 18 scenarios. Indicate total sample size as the result parameter by specifying the NTOTAL= option with a missing value (.). Specify your conjectures for the means using the GROUPMEANS= option. Using the "matched" notation (discussed in the "Specifying Value Lists in Analysis Statements" section on page 3490), enclose the two group means for each scenario in parentheses. Use the STDDEV= option to specify scenarios for the common standard deviation. Specify the weighting schemes using the GROUPWEIGHTS= option. You could again use the matched notation. But for illustrative purposes, specify the scenarios for each group weight separately using the "crossed" notation, with scenarios for each group weight separated by a vertical bar (|). The statements that perform the analysis are as follows:

```
proc power;
   twosamplemeans
      groupmeans   = (13 14) (13 14.5) (13 15)
      stddev       = 1.2 1.7
      groupweights = 1 | 1 2 3
      power        = 0.9
      ntotal       = .;
run;
```

Default values for the TEST=, DIST=, NULLDIFF=, ALPHA=, and SIDES= options specify a 2-sided t test of group mean difference equal to 0, assuming a normal distribution with a significance level of α = 0.05. The results are shown in Figure 57.4.

```
                       The POWER Procedure
              Two-sample t Test for Mean Difference

                     Fixed Scenario Elements

                  Distribution              Normal
                  Method                     Exact
                  Group 1 Weight                 1
                  Nominal Power                0.9
                  Number of Sides                2
                  Null Difference                0
                  Alpha                       0.05


                        Computed N Total

                               Std                Actual         N
   Index     Mean1     Mean2     Dev     Weight2     Power     Total

       1        13      14.0     1.2           1     0.907        64
       2        13      14.0     1.2           2     0.908        72
       3        13      14.0     1.2           3     0.905        84
       4        13      14.0     1.7           1     0.901       124
       5        13      14.0     1.7           2     0.905       141
       6        13      14.0     1.7           3     0.900       164
       7        13      14.5     1.2           1     0.910        30
       8        13      14.5     1.2           2     0.906        33
       9        13      14.5     1.2           3     0.916        40
      10        13      14.5     1.7           1     0.900        56
      11        13      14.5     1.7           2     0.901        63
      12        13      14.5     1.7           3     0.908        76
      13        13      15.0     1.2           1     0.913        18
      14        13      15.0     1.2           2     0.927        21
      15        13      15.0     1.2           3     0.922        24
      16        13      15.0     1.7           1     0.914        34
      17        13      15.0     1.7           2     0.921        39
      18        13      15.0     1.7           3     0.910        44
```

Figure 57.4. Sample Size Analysis for Two-Sample t Test Using Group Means

The interpretation is that in the best-case scenario (large mean difference of 2, small standard deviation of 1.2, and balanced design), a sample size of $N = 18$ ($n_1 = n_2 = 9$) patients is sufficient to achieve a power of at least 0.9. In the worst-case scenario (small mean difference of 1, large standard deviation of 1.7, and a 1:3 unbalanced design), a sample size of $N = 164$ ($n_1 = 41, n_2 = 123$) patients is necessary. The Nominal Power of 0.9 in the Fixed Scenario Elements table represents the input target power, and the Actual Power column in the Computed N Total table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting exactly.

Note the following characteristics of the analysis, and ways you can modify them if you wish.

- The total sample sizes are rounded up to multiples of the weight sums (2 for the 1:1 design, 3 for the 1:2 design, and 4 for the 1:3 design) to ensure that each group size is an integer. To request raw fractional sample size solutions, use the NFRACTIONAL option.
- Only the group weight that varies (the one for group 2) is displayed as an

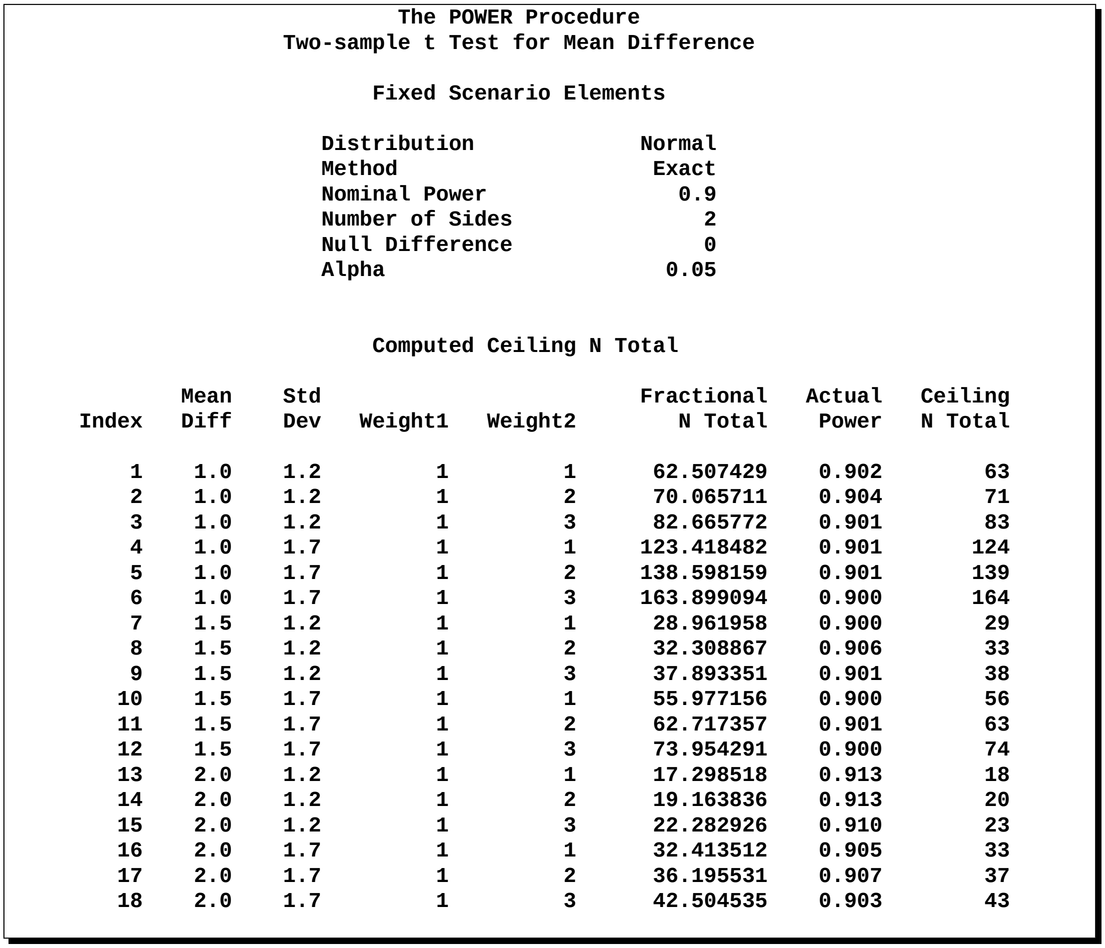
output column, while the weight for group 1 appears in the Fixed Scenario Elements table. To display the group weights together in output columns, use the matched version of the value list rather than the crossed version.

- If you can only specify differences between group means (instead of their individual values), or if you want to display the mean differences instead of the individual means, use the MEANDIFF= option instead of the GROUPMEANS= option.

The following statements implement all of these modifications.

```
proc power;
   twosamplemeans
      nfractional
      meandiff     = 1 to 2 by 0.5
      stddev       = 1.2 1.7
      groupweights = (1 1) (1 2) (1 3)
      power        = 0.9
      ntotal       = .;
run;
```

Figure 57.5 shows the new results.



```
                          The POWER Procedure
                  Two-sample t Test for Mean Difference

                        Fixed Scenario Elements

                      Distribution            Normal
                      Method                   Exact
                      Nominal Power              0.9
                      Number of Sides              2
                      Null Difference              0
                      Alpha                     0.05


                        Computed Ceiling N Total

          Mean    Std                            Fractional   Actual   Ceiling
   Index  Diff    Dev   Weight1    Weight2          N Total    Power   N Total

       1   1.0    1.2         1          1        62.507429    0.902        63
       2   1.0    1.2         1          2        70.065711    0.904        71
       3   1.0    1.2         1          3        82.665772    0.901        83
       4   1.0    1.7         1          1       123.418482    0.901       124
       5   1.0    1.7         1          2       138.598159    0.901       139
       6   1.0    1.7         1          3       163.899094    0.900       164
       7   1.5    1.2         1          1        28.961958    0.900        29
       8   1.5    1.2         1          2        32.308867    0.906        33
       9   1.5    1.2         1          3        37.893351    0.901        38
      10   1.5    1.7         1          1        55.977156    0.900        56
      11   1.5    1.7         1          2        62.717357    0.901        63
      12   1.5    1.7         1          3        73.954291    0.900        74
      13   2.0    1.2         1          1        17.298518    0.913        18
      14   2.0    1.2         1          2        19.163836    0.913        20
      15   2.0    1.2         1          3        22.282926    0.910        23
      16   2.0    1.7         1          1        32.413512    0.905        33
      17   2.0    1.7         1          2        36.195531    0.907        37
      18   2.0    1.7         1          3        42.504535    0.903        43
```

Figure 57.5. Sample Size Analysis for Two-Sample t Test Using Mean Differences

Note that the Nominal Power of 0.9 applies to the raw computed sample size (Fractional N Total), and the Actual Power column applies to the rounded sample size (Ceiling N Total). Some of the adjusted sample sizes in Figure 57.5 are lower than those in Figure 57.4 because underlying group sample sizes are allowed to be fractional (for example, the first Ceiling N Total of 63 corresponding to equal group sizes of 31.5).

Syntax

The following statements are available in PROC POWER.

PROC POWER < *options* > **;**

MULTREG < *options* > **;**
ONECORR < *options* > **;**
ONESAMPLEFREQ < *options* > **;**
ONESAMPLEMEANS < *options* > **;**
ONEWAYANOVA < *options* > **;**
PAIREDFREQ < *options* > **;**
PAIREDMEANS < *options* > **;**
TWOSAMPLEFREQ < *options* > **;**
TWOSAMPLEMEANS < *options* > **;**
TWOSAMPLESURVIVAL < *options* > **;**

PLOT < *plot-options* > < **/** *graph-options* > **;**

The statements in the POWER procedure consist of the PROC POWER statement, a set of *analysis statements* (for requesting specific power and sample size analyses), and the PLOT statement (for producing graphs). The PROC POWER statement and at least one of the analysis statements are required. The analysis statements are MULTREG, ONECORR, ONESAMPLEFREQ, ONESAMPLEMEANS, ONEWAYANOVA, PAIREDFREQ, PAIREDMEANS, TWOSAMPLEFREQ, TWOSAMPLEMEANS, and TWOSAMPLESURVIVAL.

You can use multiple analysis statements and multiple PLOT statements. Each analysis statement produces a separate sample size analysis. Each PLOT statement refers to the previous analysis statement and generates a separate graph (or set of graphs).

The name of an analysis statement describes the framework of the statistical analysis for which sample size calculations are desired. You use options in the analysis statements to identify the result parameter to compute, to specify the statistical test and computational options, and to provide one or more scenarios for the values of relevant analysis parameters.

Table 57.1 summarizes the basic functions of each statement in PROC POWER. The syntax of each statement in Table 57.1 is described in the following pages.

Table 57.1. Statements in the POWER Procedure

Statement	**Description**
PROC POWER	invokes the procedure
MULTREG	tests of one or more coefficients in multiple linear regression
ONECORR	Fisher's z test and t test of (partial) correlation
ONESAMPLEFREQ	tests of a single binomial proportion
ONESAMPLEMEANS	one-sample t test, confidence interval precision, or equivalence test

Table 57.1. (continued)

Statement	Description
ONEWAYANOVA	one-way ANOVA including single-degree-of-freedom contrasts
PAIREDFREQ	McNemar's test for paired proportions
PAIREDMEANS	paired t test, confidence interval precision, or equivalence test
TWOSAMPLEFREQ	chi-square, likelihood ratio, and Fisher's exact tests for two independent proportions
TWOSAMPLEMEANS	two-sample t test, confidence interval precision, or equivalence test
TWOSAMPLESURVIVAL	log-rank, Gehan, and Tarone-Ware tests for comparing two survival curves
PLOT	displays plots for previous sample size analysis

See the "Summary of Analyses" section on page 3488 for a summary of the analyses available and the syntax required for them.

PROC POWER Statement

PROC POWER < *options* > **;**

The PROC POWER statement invokes the POWER procedure. You can specify the following option.

PLOTONLY
specifies that only graphical results from the PLOT statement should be produced.

MULTREG Statement

MULTREG < *options* > **;**

The MULTREG statement performs power and sample size analyses for Type III F tests of sets of predictors in multiple linear regression, assuming either fixed or normally distributed predictors.

Summary of Options

Table 57.2 summarizes categories of options available in the MULTREG statement.

Table 57.2. Summary of Options in the MULTREG Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= MODEL= NFULLPREDICTORS= NOINT NREDUCEDPREDICTORS= NTESTPREDICTORS=

Table 57.2. (continued)

Task	Options
Specify effects	PARTIALCORR= RSQUAREDIFF= RSQUAREFULL= RSQUAREREDUCED=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.3 summarizes the valid result parameters in the MULTREG statement.

Table 57.3. Summary of Result Parameters in the MULTREG Statement

Analyses	Solve for	Syntax
TEST=TYPE3	Power	POWER = .
	Sample size	NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MODEL=*keyword-list*

specifies the assumed distribution of the tested predictors. MODEL=FIXED indicates a fixed predictor distribution. MODEL=RANDOM (the default) indicates a joint multivariate normal distribution for the response and tested predictors. You may use the aliases CONDITIONAL for FIXED and UNCONDITIONAL for RANDOM. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*.

FIXED	fixed predictors
RANDOM	random (multivariate normal) predictors

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NFULLPREDICTORS=*number-list*
NFULLPRED=*number-list*

specifies the number of predictors in the full model, not counting the intercept. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NOINT

specifies a no-intercept model (for both full and reduced models). By default, the intercept is included in the model. If you wish to test the intercept, you can specify the NOINT option and simply consider the intercept to be one of the predictors being tested. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NREDUCEDPREDICTORS=*number-list*
NREDUCEDPRED=*number-list*
NREDPRED=*number-list*

specifies the number of predictors in the reduced model, not counting the intercept. This is the same as the difference between values of the NFULLPREDICTORS= and NTESTPREDICTORS= options. Note that supplying a value of 0 is the same as specifying an F test of a Pearson correlation. This option cannot be used at the same time as the NTESTPREDICTORS= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTESTPREDICTORS=*number-list*
NTESTPRED=*number-list*

specifies the number of predictors being tested. This is the same as the difference between values of the NFULLPREDICTORS= and NREDUCEDPREDICTORS= options. Note that supplying identical values for the NTESTPREDICTORS= and NFULLPREDICTORS= options is the same as specifying an F test of a Pearson correlation. This option cannot be used at the same time as the NREDUCEDPREDICTORS= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). The minimum acceptable value for the sample size depends on the MODEL=, NOINT, NFULLPREDICTORS=, NTESTPREDICTORS=, and NREDUCEDPREDICTORS= options. It ranges from $p + 1$ to $p + 3$, where p is the value of the NFULLPREDICTORS option. See Table 57.26 on page 3501 for further information on minimum NTOTAL values, and see the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- MODEL
- NFULLPREDICTORS
- NTESTPREDICTORS
- NREDUCEDPREDICTORS
- ALPHA
- PARTIALCORR
- RSQUAREFULL
- RSQUAREREDUCED
- RSQUAREDIFF
- NTOTAL
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the MULTREG statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the MULTREG statement.

PARTIALCORR=*number-list*
PCORR=*number-list*

specifies the partial correlation between the tested predictors and the response, adjusting for any other predictors in the model. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

RSQUAREDIFF=*number-list*
RSQDIFF=*number-list*

specifies the difference in R^2 between the full and reduced models. This is equivalent to the proportion of variation explained by the predictors you are testing. It is also equivalent to the squared semipartial correlation of the tested predictors with the response. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

RSQUAREFULL=*number-list*
RSQFULL=*number-list*

specifies the R^2 of the full model, where R^2 is the proportion of variation explained by the model. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

RSQUAREREDUCED=*number-list*
RSQREDUCED=*number-list*
RSQRED=*number-list*

specifies the R^2 of the reduced model, where R^2 is the proportion of variation explained by the model. If the reduced model is an empty or intercept-only model (in other words, if NREDUCEDPREDICTORS = 0 or NTESTPREDICTORS = NFULLPREDICTORS), then RSQUAREREDUCED = 0 is assumed. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

TEST= TYPE3

specifies a Type III F test of a set of predictors adjusting for any other predictors in the model. This is the default test option.

Restrictions on Option Combinations

To specify the number of predictors, use any two of these three options:

- the number of predictors in the full model (NFULLPREDICTORS=)
- the number of predictors in the reduced model (NREDUCEDPREDICTORS=)
- the number of predictors being tested (NTESTPREDICTORS=)

To specify the effect, choose one of the following parameterizations:

- partial correlation (using the PARTIALCORR= option)
- R^2 for the full and reduced models (using any two of RSQUAREDIFF=, RSQUAREFULL=, and RSQUAREREDUCED=)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the MULTREG statement.

Type III F Test of a Set of Predictors

You can express effects in terms of partial correlation. Default values of the TEST=, MODEL=, and ALPHA= options specify a Type III F test with a significance level of 0.05, assuming normally distributed predictors.

```
proc power;
   multreg
      model = random
      nfullpredictors = 7
      ntestpredictors = 3
      partialcorr = 0.35
      ntotal = 100
      power = .;
run;
```

You can also express effects in terms of R^2.

```
proc power;
   multreg
      model = fixed
      nfullpredictors = 7
      ntestpredictors = 3
      rsquarefull = 0.9
      rsquarediff = 0.1
      ntotal = .
      power = 0.9;
run;
```

ONECORR Statement

ONECORR < *options* > **;**

The ONECORR statement performs power and sample size analyses for tests of simple and partial Pearson correlation between two variables. Both Fisher's z test and the t test are supported.

Summary of Options

Table 57.4 summarizes categories of options available in the ONECORR statement.

Table 57.4. Summary of Options in the ONECORR Statement

Task	Options
Define analysis	DIST= TEST=
Specify analysis information	ALPHA= MODEL= NPARTIALVARS= NULL= SIDES=
Specify effects	CORR=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.5 summarizes the valid result parameters in the ONECORR statement.

Table 57.5. Summary of Result Parameters in the ONECORR Statement

Analyses	Solve for	Syntax
TEST=PEARSON	Power	POWER = .
	Sample size	NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CORR=*number-list*

specifies the correlation between two variables, possibly adjusting for other variables as determined by the NPARTIALVARS= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DIST=FISHERZ
DIST=T

specifies the underlying distribution assumed for the test statistic. FISHERZ corresponds to Fisher's z normalizing transformation of the correlation coefficient. T corresponds to the t transformation of the correlation coefficient. Note that DIST=T is equivalent to analyses in the MULTREG statement with NTESTPREDICTORS=1. The default value is FISHERZ.

MODEL=*keyword-list*

specifies the assumed distribution of the first variable when DIST=T. The second variable is assumed to have a normal distribution. MODEL=FIXED indicates a fixed distribution. MODEL=RANDOM (the default) indicates a joint bivariate normal distribution with the second variable. You may use the aliases CONDITIONAL for FIXED and UNCONDITIONAL for RANDOM. This option can only be used for DIST=T. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*.

FIXED	fixed variables
RANDOM	random (bivariate normal) variables

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPARTIALVARS=*number-list*
NPVARS=*number-list*

specifies the number of variables adjusted for in the correlation between the two primary variables. The default value is 0, corresponding to a simple correlation. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). Values for the sample size must be at least $p + 3$ when DIST=T and MODEL=CONDITIONAL, and at least $p + 4$ when either DIST=FISHER or when DIST=T and MODEL=UNCONDITIONAL, where p is the value of the NPARTIALVARS option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLCORR=*number-list*
NULLC=*number-list*

specifies the null value of the correlation. The default value is 0. This option can only be used with the DIST=FISHERZ analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- MODEL
- SIDES
- NULL
- ALPHA
- NPARTIALVARS
- CORR
- NTOTAL
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the ONECORR statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the ONECORR statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

TEST= PEARSON

specifies a test of the Pearson correlation coefficient between two variables, possibly adjusting for other variables. This is the default test option.

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONECORR statement.

Fisher's z test for Pearson Correlation

Default values of TEST=PEARSON, ALPHA=.05, SIDES=2, and NPARTIALVARS=0 are assumed.

```
proc power;
   onecorr dist=fisherz
   null = 0.15
   corr = 0.35
   ntotal = 180
   power = .
run;
```

t test for Pearson Correlation

Default values of TEST=PEARSON, MODEL=RANDOM, ALPHA=.05, and SIDES=2 are assumed.

```
proc power;
   onecorr dist=t
   npartialvars = 4
   corr = 0.45
   ntotal = .
   power = 0.85
run;
```

ONESAMPLEFREQ Statement

ONESAMPLEFREQ < *options* > **;**

The ONESAMPLEFREQ statement performs power and sample size analyses for exact and approximate tests of a single binomial proportion.

Summary of Options

Table 57.6 summarizes categories of options available in the ONESAMPLEFREQ statement.

Table 57.6. Summary of Options in the ONESAMPLEFREQ Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= NULLPROPORTION= SIDES=
Specify effect	PROPORTION=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Choose computational method	METHOD=
Control ordering in output	OUTPUTORDER=

Table 57.7 summarizes the valid result parameters for different analyses in the ONESAMPLEFREQ statement.

Table 57.7. Summary of Result Parameters in the ONESAMPLEFREQ Statement

Analyses	Solve for	Syntax
TEST=ADJZ METHOD=EXACT	Power	POWER = .
TEST=ADJZ METHOD=NORMAL	Power	POWER = .
	Sample size	NTOTAL = .
TEST=EXACT	Power	POWER = .
TEST=Z METHOD=EXACT	Power	POWER = .
TEST=Z METHOD=NORMAL	Power	POWER = .
	Sample size	NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

METHOD=EXACT
METHOD=NORMAL

specifies the computational method. METHOD=EXACT (the default) computes exact results using the binomial distribution. METHOD=NORMAL computes approximate results using the normal approximation to the binomial distribution.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLPROPORTION=*number-list*
NULLP=*number-list*

specifies the null proportion. A value of 0.5 corresponds to the sign test. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLPROPORTION
- ALPHA
- PROPORTION
- NTOTAL
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the ONESAMPLEFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the ONESAMPLEFREQ statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

PROPORTION=*number-list*
P=*number-list*

specifies the binomial proportion, that is, the expected proportion of successes in the hypothetical binomial trial. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

The default value is 2.

TEST= ADJZ
TEST= EXACT
TEST= Z

specifies the statistical analysis. TEST=ADJZ specifies a normal-approximate z test with continuity adjustment. TEST=EXACT (the default) specifies the exact binomial test. TEST=Z specifies a normal-approximate z test without any continuity adjustment, which is the same as the chi-square test when SIDES=2.

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONESAMPLEFREQ statement.

Exact Test of a Binomial Proportion

Defaults for the SIDES= and ALPHA= options specify a 2-sided test with a 0.05 significance level.

```
proc power;
   onesamplefreq test=binomial
      nullproportion = 0.2
      proportion = 0.3
      ntotal = 100
      power = .;
run;
```

z Test

Defaults for the SIDES= and ALPHA= options specify a 2-sided test with a 0.05 significance level.

```
proc power;
   onesamplefreq test=z
      nullproportion = 0.8
      proportion = 0.85
      sides = u
      ntotal = .
      power = .9;
run;
```

z Test with Continuity Adjustment

Defaults for the SIDES= and ALPHA= options specify a 2-sided test with a 0.05 significance level.

```
proc power;
   onesamplefreq test=adjz
      nullproportion = 0.15
      proportion = 0.1
      sides = l
      ntotal = .
      power = .9;
run;
```

ONESAMPLEMEANS Statement

ONESAMPLEMEANS < *options* > **;**

The ONESAMPLEMEANS statement performs power and sample size analyses for t tests, equivalence tests, and confidence interval precision involving one sample.

Summary of Options

Table 57.8 summarizes categories of options available in the ONESAMPLEMEANS statement.

Table 57.8. Summary of Options in the ONESAMPLEMEANS Statement

Task	Options
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULL= SIDES= UPPER=
Specify effects	HALFWIDTH= MEAN=
Specify variability	CV= STDDEV=
Specify sample size	NTOTAL=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.9 summarizes the valid result parameters for different analyses in the ONESAMPLEMEANS statement.

Table 57.9. Summary of Result Parameters in the ONESAMPLEMEANS Statement

Analyses	Solve for	Syntax

TEST=T DIST=NORMAL	Power
	Sample size
	Alpha
	Mean
	Standard Deviation
		POWER = .
		NTOTAL = .
		ALPHA = .
		MEAN = .
		STDDEV = .

TEST=T DIST=LOGNORMAL	Power
	Sample size
		POWER = .
		NTOTAL = .

TEST=EQUIV	Power
	Sample size
		POWER = .
		NTOTAL = .

CI=T	Prob(width)
	Sample size
		PROBWIDTH = .
		NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CI
CI= T

specifies an analysis of precision of the confidence interval for the mean. Instead of power, the relevant probability for this analysis is the probability of achieving a desired precision. Specifically, it is the probability that the half-width of the confidence interval will be at most the value specified by the HALFWIDTH= option. If neither the CI= option nor the TEST= option is used, the default is TEST=T.

CV=*number-list*

specifies the coefficient of variation, defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DIST=LOGNORMAL
DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value is NORMAL.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can only be used with the CI=T analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean. This option can only be used with the TEST=EQUIV analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MEAN=*number-list*

specifies the mean, in the original scale. The mean is arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option can only be used with the TEST=T and TEST=EQUIV analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLMEAN=*number-list*
NULLM=*number-list*

specifies the null mean, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). The default value is 0 when DIST=NORMAL and 1 when DIST=LOGNORMAL. This option can only be used with the TEST=T analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLMEAN
- LOWER
- UPPER
- ALPHA
- MEAN
- HALFWIDTH
- STDDEV
- CV
- NTOTAL
- POWER
- PROBTYPE
- PROBWIDTH

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the ONESAMPLEMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the ONESAMPLEMEANS statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. This option can only be used with the TEST=T and TEST=EQUIV analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option. You may use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can only be used with the CI=T analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*.

CONDITIONAL	width probability conditional on interval containing the mean
UNCONDITIONAL	unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can only be used with the CI=T analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between negative infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option can only be used with the TEST=T and CI=T analyses. The default value is 2.

STDDEV=*number-list*
STD=*number-list*

specifies the standard deviation. You can use this option only with DIST=NORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

TEST
TEST=EQUIV
TEST=T

specifies the statistical analysis. TEST=EQUIV specifies an equivalence test of the mean using a two one-sided tests (TOST) analysis (Schuirmann 1987). TEST or TEST=T (the default) specifies a t test on the mean. If neither the TEST= option nor the CI= option is used, the default is TEST=T.

UPPER=*number-list*

specifies the upper equivalence bound for the mean, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can only be used with the TEST=EQUIV analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (using the TEST= option)
- confidence interval precision (using the CI= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONESAMPLEMEANS statement.

One-sample t Test

Default values for the DIST=, SIDES=, NULLMEAN=, and ALPHA= options specify a 2-sided test for zero mean with a normal distribution and a significance level of 0.05.

```
proc power;
   onesamplemeans test=t
      mean = 7
      stddev = 3
      ntotal = 50
      power = .;
run;
```

One-sample t Test with Lognormal Data

Default values for the SIDES=, NULLMEAN=, and ALPHA= options specify a 2-sided test for unit mean with a significance level of 0.05.

```
proc power;
   onesamplemeans test=t dist=lognormal
      mean = 7
      cv = 0.8
      ntotal = .
      power = 0.9;
run;
```

Equivalence Test for Mean of Normal Data

Default values for the DIST= and ALPHA= options specify a normal distribution and a significance level of 0.05.

```
proc power;
   onesamplemeans test=equiv
      lower = 2
      upper = 7
      mean = 4
      stddev = 3
      ntotal = 100
      power = .;
run;
```

Equivalence Test for Mean of Lognormal Data

The default of ALPHA=0.05 specifies a significance level of 0.05.

```
proc power;
   onesamplemeans test=equiv dist=lognormal
      lower = 1
      upper = 5
      mean = 3
      cv = 0.6
      ntotal = .
      power = 0.85;
run;
```

Confidence Interval for Mean

By default CI=T analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean. The defaults of SIDES=2 and ALPHA=0.05 specify a 2-sided interval with a confidence level of 0.95.

```
proc power;
   onesamplemeans ci = t
      halfwidth = 14
      stddev = 8
      ntotal = 50
      probwidth = .;
run;
```

ONEWAYANOVA Statement

ONEWAYANOVA < *options* > **;**

The ONEWAYANOVA statement performs power and sample size analyses for one-degree-of-freedom contrasts and the overall F test in one-way analysis of variance.

Summary of Options

Table 57.10 summarizes categories of options available in the ONEWAYANOVA statement.

Table 57.10. Summary of Options in the ONEWAYANOVA Statement

Task	**Options**
Define analysis	TEST=
Specify analysis information	ALPHA= CONTRAST= SIDES= NULLCONTRAST=
Specify effects	GROUPMEANS=
Specify variability	STDDEV=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.11 summarizes the valid result parameters for different analyses in the ONEWAYANOVA statement.

Table 57.11. Summary of Result Parameters in the ONEWAYANOVA Statement

Analyses	**Solve for**	**Syntax**
TEST=CONTRAST	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

TEST=OVERALL	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CONTRAST= (*values* **)** < **(...** *values* **)** >

specifies coefficients for single-degree-of-freedom hypothesis tests. You must provide a coefficient for every mean appearing in the GROUPMEANS= option. Specify multiple contrasts either with additional sets of coefficients or with additional CONTRAST= options. For example, you can specify two different contrasts of five means using

```
CONTRAST = (1 -1 0 0 0) (1 0 -1 0 0)
```

GROUPMEANS=*grouped-number-list*
GMEANS=*grouped-number-list*

specifies the group means. This option is used to implicitly set the number of groups. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPNS= *grouped-number-list*
GNS= *grouped-number-list*

specifies the group sample sizes. The number of groups represented must be the same as with the GROUPMEANS= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPWEIGHTS= *grouped-number-list*
GWEIGHTS= *grouped-number-list*

specifies the sample size allocation weights for the groups. This option controls how the total sample size is divided between the groups. Each set of values across all groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). The number of groups represented must be the same as with the GROUPMEANS= option. Values must be integers unless the NFRACTIONAL option is used. The default value is 1 for each group, amounting to a balanced design. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP= *number-list*
NPERG= *number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLCONTRAST=*number-list*
NULLC=*number-list*

specifies the null value of the contrast. The default value is 0. This option can only be used with the TEST=CONTRAST analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- CONTRAST
- SIDES
- NULLCONTRAST
- ALPHA
- GROUPMEANS
- STDDEV
- GROUPWEIGHTS
- NTOTAL
- NPERGROUP
- GROUPNS
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the ONEWAYANOVA statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the ONEWAYANOVA statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value

L lower 1-sided with alternative less than null value

This option can only be used with the TEST=CONTRAST analysis. The default value is 2.

STDDEV=*number-list*
STD=*number-list*

specifies the error standard deviation. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

TEST= CONTRAST
TEST= OVERALL

specifies the statistical analysis. TEST=CONTRAST specifies a one-degree-of-freedom test of a contrast of means. The test is the usual F test for the 2-sided case and the usual t test for the 1-sided case. TEST=OVERALL specifies the overall F test of equality of all means. The default is TEST=CONTRAST if the CONTRAST= option is used, and TEST=OVERALL otherwise.

Restrictions on Option Combinations

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (using the NPERGROUP= option)
- total sample size and allocation weights (using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONEWAYANOVA statement.

One-Degree-of-Freedom Contrast

You can use the NPERGROUP= option in a balanced design. Default values for the SIDES=, NULLCONTRAST=, and ALPHA= options specify a 2-sided test for a contrast value of 0 with a significance level of 0.05.

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design with the NTOTAL= and GROUPWEIGHTS= options.

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      groupweights = (1 2 2)
      ntotal = .
      power = 0.9;
run;
```

Another way to specify the sample sizes is with the GROUPN= option.

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      groupns = (20 40 40)
      power = .;
run;
```

Overall F Test

The default of ALPHA=0.05 specifies a significance level of 0.05.

```
proc power;
   onewayanova test=overall
      groupmeans = 3 | 7 | 8
      stddev = 4
      npergroup = 50
      power = .;
run;
```

PAIREDFREQ Statement

PAIREDFREQ < *options* > **;**

The PAIREDFREQ statement performs power and sample size analyses for McNemar's test for paired proportions.

Summary of Options

Table 57.12 summarizes categories of options available in the PAIREDFREQ statement.

Table 57.12. Summary of Options in the PAIREDFREQ Statement

Task	Options
Define analysis	DIST=
	TEST=
Specify analysis information	ALPHA=
	NULLDISCPROPRATIO=

Table 57.12. (continued)

Task	Options
	SIDES=
Specify effects	DISCPROPDIFF= DISCPROPORTIONS= DISCPROPRATIO= REFPROPORTION= TOTALPROPDISC=
Specify sample size	NPAIRS=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Choose computational method	METHOD=
Control ordering in output	OUTPUTORDER=

Table 57.13 summarizes the valid result parameters in the PAIREDFREQ statement.

Table 57.13. Summary of Result Parameters in the PAIREDFREQ Statement

Analyses	Solve for	Syntax
TEST=MCNEMAR METHOD=CONNOR	Power	POWER = .
	Sample size	NPAIRS = .
TEST=MCNEMAR METHOD=EXACT	Power	POWER = .
TEST=MCNEMAR METHOD=MIETTINEN	Power	POWER = .
	Sample size	NPAIRS = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DISCPROPORTIONS=*grouped-number-list*
DISCPS=*grouped-number-list*

specifies the two discordant proportions, p_{10} and p_{01}. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

DISCPROPDIFF=*number-list*
DISCPDIFF=*number-list*

specifies the difference $p_{01} - p_{10}$ between discordant proportions. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DISCPROPRATIO=*number-list*
DISCPRATIO=*number-list*

specifies the ratio p_{01}/p_{10} of discordant proportions. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DIST=EXACT_COND
DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. EXACT_COND corresponds to the exact conditional test, based on the exact binomial distribution of the two types of discordant pairs given the total number of discordant pairs. NORMAL corresponds to the conditional test based on the normal approximation to the binomial distribution of the two types of discordant pairs given the total number of discordant pairs. The default value is EXACT_COND.

METHOD=CONNOR
METHOD=EXACT
METHOD=MIETTINEN

specifies the computational method. METHOD=EXACT (the default) uses the exact binomial distributions of the total number of discordant pairs and the two types of discordant pairs. METHOD=CONNOR uses an approximation from Connor (1987), and METHOD=MIETTINEN uses an approximation from Miettinen (1968). The CONNOR and MIETTINEN methods are valid only for DIST=NORMAL.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option. This option cannot be used with METHOD=EXACT.

NPAIRS= *number-list*

specifies the total number of proportion pairs (concordant and discordant) or requests a solution for the number of pairs with a missing value (NPAIRS=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLDISCPROPRATIO=*number-list*
NULLDISCPRATIO=*number-list*
NULLRATIO=*number-list*
NULLR=*number-list*

specifies the null value of the ratio of discordant proportions. The default value is 1. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLDISCPROPRATIO
- ALPHA
- DISCPROPORTIONS
- DISCPROPDIFF
- TOTALPROPDISC
- REFPROPORTION
- DISCPROPRATIO
- NPAIRS
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the PAIREDFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the PAIREDFREQ statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

REFPROPORTION=*number-list*
REFP=*number-list*

specifies the reference discordant proportion p_{10}. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

The default value is 2.

TEST=MCNEMAR

specifies the McNemar test of paired proportions. This is the default test option.

TOTALPROPDISC=*number-list*
TOTALPDISC=*number-list*
PDISC=*number-list*

specifies the sum of the two discordant proportions, $p_{10} + p_{01}$. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Restrictions on Option Combinations

To specify the proportions, choose one of the following parameterizations:

- discordant proportions (using the DISCPROPORTIONS= option)
- difference and sum of discordant proportions (using the DISCPROPORTIONDIFF= and TOTALPROPDISC= options)
- ratio of discordant proportions and reference discordant proportion (using the DISCPROPRATIO= and REFPROPORTION= options)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the PAIREDFREQ statement.

McNemar Exact Conditional Test

You can express effects in terms of the individual discordant proportions. Default values for the TEST=, SIDES=, ALPHA=, and NULLDISCPROPRATIO= options specify a 2-sided McNemar test for no effect with a significance level of 0.05.

```
proc power;
   pairedfreq dist=exact_cond
      discproportions = 0.15 | 0.45
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the difference and sum of discordant proportions.

```
proc power;
   pairedfreq dist=exact_cond
      discpropdiff = 0.3
      totalpropdisc = 0.6
      npairs = .
      power = 0.9;
run;
```

You can also express effects in terms of the ratio of discordant proportions and the denominator of the ratio.

```
proc power;
   pairedfreq dist=exact_cond
      discpropratio = 3
      refproportion = 0.15
      npairs = 80
      power = .;
run;
```

McNemar Normal Approximation Test

The default value for the METHOD= option specifies an exact sample size computation. Default values for the TEST=, SIDES=, ALPHA=, and NULLDISCPROPRATIO= options specify a 2-sided McNemar test for no effect with a significance level of 0.05.

```
proc power;
   pairedfreq dist=normal
      discproportions = 0.15 | 0.45
      npairs = .
      power = .9;
run;
```

PAIREDMEANS Statement

PAIREDMEANS < *options* > **;**

The PAIREDMEANS statement performs power and sample size analyses for t tests, equivalence tests, and confidence interval precision involving paired samples.

Summary of Options

Table 57.14 summarizes categories of options available in the PAIREDMEANS statement.

Table 57.14. Summary of Options in the PAIREDMEANS Statement

Task	**Options**
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULLDIFF= NULLRATIO= SIDES= UPPER=
Specify effects	HALFWIDTH= MEANDIFF= MEANRATIO= PAIREDMEANS=
Specify variability	CORR=

Table 57.14. (continued)

Task	Options
	CV= PAIREDCVS= PAIREDSTDDEVS= STDDEV=
Specify sample size	NPAIRS=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.15 summarizes the valid result parameters for different analyses in the PAIREDMEANS statement.

Table 57.15. Summary of Result Parameters in the PAIREDMEANS Statement

Analyses	Solve for	Syntax

TEST=DIFF	Power
	Sample size
		POWER = .
		NPAIRS = .

TEST=RATIO	Power
	Sample size
		POWER = .
		NPAIRS = .

TEST=EQUIV_DIFF	Power
	Sample size
		POWER = .
		NPAIRS = .

TEST=EQUIV_RATIO	Power
	Sample size
		POWER = .
		NPAIRS = .

CI=DIFF	Prob(width)
	Sample size
		PROBWIDTH = .
		NPAIRS = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CI
CI=DIFF

specifies an analysis of precision of the confidence interval for the mean difference. Instead of power, the relevant probability for this analysis is the probability of achieving a desired precision. Specifically, it is the probability that the half-width of the observed confidence interval will be at most the value specified by the

HALFWIDTH= option. If neither the CI= option nor the TEST= option is used, the default is TEST=DIFF.

CORR=*number-list*

specifies the correlation between members of a pair. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CV=*number-list*

specifies the coefficient of variation assumed to be common to both members of a pair. The coefficient of variation is defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DIST=LOGNORMAL
DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value (also the only acceptable value in each case) is NORMAL for TEST=DIFF, TEST=EQUIV_DIFF, and CI=DIFF; and LOGNORMAL for TEST=RATIO and TEST=EQUIV_RATIO.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can only be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can only be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MEANDIFF=*number-list*

specifies the mean difference, defined as the mean of the difference between the second and first members of a pair, $\mu_2 - \mu_1$. This option can only be used with the TEST=DIFF and TEST=EQUIV_DIFF analyses. When TEST=EQUIV_DIFF, the mean difference is interpreted as the treatment mean minus the reference mean. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MEANRATIO=*number-list*

specifies the geometric mean ratio, defined as γ_2/γ_1. This option can only be used with the TEST=RATIO and TEST=EQUIV_RATIO analyses. When TEST=EQUIV_RATIO, the mean ratio is interpreted as the treatment mean divided by the reference mean. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPAIRS= *number-list*

specifies the number of pairs or requests a solution for the number of pairs with a missing value (NPAIRS=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLDIFF=*number-list*
NULLD=*number-list*

specifies the null mean difference. The default value is 0. This option can only be used with the TEST=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLRATIO=*number-list*
NULLR=*number-list*

specifies the null mean ratio. The default value is 1. This option can only be used with the TEST=RATIO analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLDIFF
- NULLRATIO
- LOWER
- UPPER
- ALPHA
- PAIREDFREQ
- MEANDIFF
- MEANRATIO
- HALFWIDTH
- STDDEV
- PAIREDSTDDEVS
- CV
- PAIREDCVS
- CORR
- NPAIRS

- POWER
- PROBTYPE
- PROBWIDTH

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the PAIREDMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the PAIREDMEANS statement.

PAIREDCVS=*grouped-number-list*

specifies the coefficient of variation for each member of a pair. Unlike the CV= option, the PAIREDCVS= option supports different values for each member of a pair. This option can only be used with DIST=LOGNORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

PAIREDMEANS=*grouped-number-list*
PMEANS=*grouped-number-list*

specifies the two paired means, in the original scale. The means are arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option cannot be used with the CI=DIFF analysis. When TEST=EQUIV_DIFF, the means are interpreted as the reference mean (first) and the treatment mean (second). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

PAIREDSTDDEVS=*grouped-number-list*
PAIREDSTDS=*grouped-number-list*
PSTDDEVS=*grouped-number-list*
PSTDS=*grouped-number-list*

specifies the standard deviation of each member of a pair. Unlike the STDDEV= option, the PAIREDSTDDEVS= option supports different values for each member of a pair. This option can only be used with DIST=NORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. This option cannot be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean difference is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option.

You may use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can only be used with the CI=DIFF analysis. See the “Specifying Value Lists in Analysis Statements” section on page 3490 for information on specifying the *keyword-list*.

CONDITIONAL	width probability conditional on interval containing the mean
UNCONDITIONAL	unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can only be used with the CI=DIFF analysis. See the “Specifying Value Lists in Analysis Statements” section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the “Specifying Value Lists in Analysis Statements” section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between negative infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option cannot be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. The default value is 2.

STDDEV=*number-list*
STD=*number-list*

specifies the standard deviation assumed to be common to both members of a pair. This option can only be used with DIST=NORMAL. See the “Specifying Value Lists in Analysis Statements” section on page 3490 for information on specifying the *number-list*.

TEST
TEST=DIFF
TEST=EQUIV_DIFF
TEST=EQUIV_RATIO
TEST=RATIO

specifies the statistical analysis. TEST or TEST=DIFF (the default) specifies a paired t test on the mean difference. TEST=EQUIV_DIFF specifies an additive equivalence test of the mean difference using a two one-sided tests (TOST) analysis (Schuirmann 1987). TEST=EQUIV_RATIO specifies a multiplicative equivalence test of the mean ratio using a TOST analysis. TEST=RATIO specifies a paired t test on the geometric mean ratio. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

UPPER=*number-list*

specifies the upper equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can only be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (using the TEST= option)
- confidence interval precision (using the CI= option)

To specify the means, choose one of the following parameterizations:

- individual means (using the PAIREDMEANS= option)
- mean difference (using the MEANDIFF= option)
- mean ratio (using the MEANRATIO= option)

To specify the coefficient of variation, choose one of the following parameterizations:

- common coefficient of variation (using the CV= option)
- individual coefficients of variation (using the PAIREDCVS= option)

To specify the standard deviation, choose one of the following parameterizations:

- common standard deviation (using the STDDEV= option)
- individual standard deviations (using the PAIREDSTDDEVS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the PAIREDMEANS statement.

Paired t Test

You can express effects in terms of the mean difference and variability in terms of a correlation and common standard deviation. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a 2-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=diff
      meandiff = 7
      corr = 0.4
      stddev = 12
      npairs = 50
      power = .;
run;
```

You can also express effects in terms of individual means and variability in terms of correlation and individual standard deviations.

```
proc power;
   pairedmeans test=diff
      pairedmeans = 8 | 15
      corr = 0.4
      pairedstddevs = (7 12)
      npairs = .
      power = 0.9;
run;
```

Paired t Test of Mean Ratio with Lognormal Data

You can express variability in terms of correlation and a common coefficient of variation. Defaults for the DIST=, SIDES=, NULLRATIO= and ALPHA= options specify a 2-sided test of mean ratio = 1 assuming a lognormal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=ratio
      meanratio = 7
      corr = 0.3
      cv = 1.2
      npairs = 30
      power = .;
run;
```

You can also express variability in terms of correlation and individual coefficients of variation.

```
proc power;
   pairedmeans test=ratio
      meanratio = 7
      corr = 0.3
      pairedcvs = 0.8 | 0.9
      npairs = 30
      power = .;
run;
```

Additive Equivalence Test for Mean Difference with Normal Data

Default values for the DIST= and ALPHA= options specify a normal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=equiv_diff
      lower = 2
      upper = 5
      meandiff = 4
      corr = 0.2
      stddev = 8
      npairs = .
      power = 0.9;
run;
```

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data

Default values for the DIST= and ALPHA= options specify a lognormal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=equiv_ratio
      lower = 3
      upper = 7
      meanratio = 5
      corr = 0.2
      cv = 1.1
      npairs = 50
      power = .;
run;
```

Confidence Interval for Mean Difference

By default CI=DIFF analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean difference. The defaults of SIDES=2 and ALPHA=0.05 specify a 2-sided interval with a confidence level of 0.95.

```
proc power;
   pairedmeans ci = diff
      halfwidth = 4
      corr = 0.35
      stddev = 8
      npairs = 30
      probwidth = .;
run;
```

TWOSAMPLEFREQ Statement

TWOSAMPLEFREQ < *options* > **;**

The TWOSAMPLEFREQ statement performs power and sample size analyses for tests of two independent proportions. Pearson's chi-square, Fisher's exact, and likelihood ratio chi-square tests are supported.

Summary of Options

Table 57.16 summarizes categories of options available in the TWOSAMPLEFREQ statement.

Table 57.16. Summary of Options in the TWOSAMPLEFREQ Statement

Task	**Options**
Define analysis	TEST=
Specify analysis information	ALPHA= NULLPROPORTIONDIFF= NULLODDSRATIO= NULLRELATIVERISK= SIDES=
Specify effects	GROUPPROPORTIONS= ODDSRATIO= PROPORTIONDIFF= REFPROPORTION= RELATIVERISK=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.17 summarizes the valid result parameters for different analyses in the TWOSAMPLEFREQ statement.

Table 57.17. Summary of Result Parameters in the TWOSAMPLEFREQ Statement

Analyses	Solve for	Syntax
TEST=FISHER	Power	POWER = .
	Sample size	NTOTAL = .
		NPERGROUP = .
TEST=LRCHI	Power	POWER = .
	Sample size	NTOTAL = .
		NPERGROUP = .
TEST=PCHI	Power	POWER = .
	Sample size	NTOTAL = .
		NPERGROUP = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

GROUPPROPORTIONS=*grouped-number-list*
GPROPORTIONS=*grouped-number-list*
GROUPPS=*grouped-number-list*
GPS=*grouped-number-list*

specifies the two independent proportions, p_1 and p_2. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPNS= *grouped-number-list*
GNS= *grouped-number-list*

specifies the two group sample sizes or requests a solution for one group sample size given the other. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPWEIGHTS= *grouped-number-list*
GWEIGHTS= *grouped-number-list*

specifies the sample size allocation weights for the two groups, or requests a solution for one group weight given the other. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP= *number-list*
NPERG= *number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLODDSRATIO=*number-list*
NULLOR=*number-list*

specifies the null odds ratio. The default value is 1. This option can only be used along with the ODDSRATIO= option in the TEST=PCHI analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLPROPORTIONDIFF=*number-list*
NULLPDIFF=*number-list*

specifies the null proportion difference. The default value is 0. This option can only be used along with the GROUPPROPORTIONS= or PROPORTIONDIFF= option in the TEST=PCHI analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLRELATIVERISK=*number-list*
NULLRR=*number-list*

specifies the null relative risk. The default value is 1. This option can only be used along with the RELATIVERISK= option in the TEST=PCHI analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

ODDSRATIO=*number-list*
OR=*number-list*

specifies the odds ratio $[p_2/(1-p_2)] \, / \, [p_1/(1-p_1)]$. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLDIFF
- NULLODDSRATIO
- NULLRELATIVERISK
- ALPHA
- GROUPPROPORTIONS
- REFPROPORTION
- PROPORTIONDIFF
- ODDSRATIO
- RELATIVERISK
- GROUPWEIGHTS
- NTOTAL
- NPERGROUP
- GROUPNS
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the TWOSAMPLEFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the TWOSAMPLEFREQ statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

PROPORTIONDIFF=*number-list*
PDIFF=*number-list*

specifies the proportion difference $p_2 - p_1$. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

REFPROPORTION=*number-list*
REFP=*number-list*

specifies the reference proportion p_1. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

RELATIVERISK=*number-list*
RR=*number-list*

specifies the relative risk p_2/p_1. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

The default value is 2.

TEST=FISHER
TEST=LRCHI
TEST=PCHI

specifies the statistical analysis. TEST=FISHER specifies Fisher's exact test. TEST=LRCHI specifies the likelihood ratio chi-square test. TEST=PCHI (the default) specifies Pearson's chi-square test.

Restrictions on Option Combinations

To specify the proportions, choose one of the following parameterizations:

- individual proportions (using the GROUPPROPORTIONS= option)
- difference between proportions and reference proportion (using the PROPORTIONDIFF and REFPROPORTION= options)
- odds ratio and reference proportion (using the ODDSRATIO= and REFPROPORTION= options)
- relative risk and reference proportion (using the RELATIVERISK= and REFPROPORTION= options)

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (using the NPERGROUP= option)
- total sample size and allocation weights (using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLEFREQ statement.

Pearson Chi-square Test for Two Proportions

You can use the NPERGROUP= option in a balanced design and express effects in terms of the individual proportions. Default values for the SIDES= and ALPHA= options specify a 2-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=pchi
      groupproportions = (.15 .25)
      nullproportiondiff = .03
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design using the NTOTAL= and GROUPWEIGHTS= options and express effects in terms of the odds ratio. The default value of the NULLODDSRATIO= option specifies a test of no effect.

```
proc power;
   twosamplefreq test=pchi
      oddsratio = 2.5
      refproportion = 0.3
      groupweights = (1 2)
      ntotal = .
      power = 0.8;
run;
```

You can also specify sample sizes with the GROUPNS= option and express effects in terms of relative risks. The default value of the NULLRELATIVERISK= option specifies a test of no effect.

```
proc power;
   twosamplefreq test=pchi
      relativerisk = 1.5
      refproportion = 0.2
      groupns = 40 | 60
      power = .;
run;
```

You can also express effects in terms of the proportion difference. The default value of the NULLPROPORTIONDIFF= option specifies a test of no effect, and the default value of the GROUPWEIGHTS= option specifies a balanced design.

```
proc power;
   twosamplefreq test=pchi
      proportiondiff = 0.15
      refproportion = 0.4
      ntotal = 100
      power = .;
run;
```

Fisher's Exact Conditional Test for Two Proportions

Default values for the SIDES= and ALPHA= options specify a 2-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=fisher
      groupproportions = (.35 .15)
      npergroup = 50
      power = .;
run;
```

Likelihood Ratio Chi-square Test for Two Proportions

Default values for the SIDES= and ALPHA= options specify a 2-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=lrchi
      oddsratio = 2
      refproportion = 0.4
      npergroup = .
      power = 0.9;
run;
```

TWOSAMPLEMEANS Statement

TWOSAMPLEMEANS < *options* > **;**

The TWOSAMPLEMEANS statement performs power and sample size analyses for pooled and unpooled t tests, equivalence tests, and confidence interval precision involving two independent samples.

Summary of Options

Table 57.18 summarizes categories of options available in the TWOSAMPLEMEANS statement.

Table 57.18. Summary of Options in the TWOSAMPLEMEANS Statement

Task	**Options**
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULLDIFF= NULLRATIO= SIDES= UPPER=
Specify effects	HALFWIDTH= GROUPMEANS=

Table 57.18. (continued)

Task	Options
	MEANDIFF= MEANRATIO=
Specify variability	CV= GROUPSTDDEVS= STDDEV=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 57.19 summarizes the valid result parameters for different analyses in the TWOSAMPLEMEANS statement.

Table 57.19. Summary of Result Parameters in the TWOSAMPLEMEANS Statement

Analyses	Solve for	Syntax
TEST=DIFF	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

	Group sample size	GROUPNS = *n1* | .
		GROUPNS = . | *n2*
		GROUPNS = (*n1* .)
		GROUPNS = (. *n2*)

	Group weight	GROUPWEIGHTS = *w1* | .
		GROUPWEIGHTS = . | *w2*
		GROUPWEIGHTS = (*w1* .)
		GROUPWEIGHTS = (. *w2*)

	Alpha	ALPHA = .

	Group mean	GROUPMEANS = *mean1* | .
		GROUPMEANS = . | *mean2*
		GROUPMEANS = (*mean1* .)
		GROUPMEANS = (. *mean2*)

	Mean difference	MEANDIFF = .
	Standard deviation	STDDEV = .
TEST=DIFF_SATT	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

Table 57.19. (continued)

Analyses	Solve for	Syntax

TEST=RATIO	Power
	Sample size
		POWER = .
		NTOTAL = .
		NPERGROUP = .

TEST=EQUIV_DIFF	Power
	Sample size
		POWER = .
		NTOTAL = .
		NPERGROUP = .

TEST=EQUIV_RATIO	Power
	Sample size
		POWER = .
		NTOTAL = .
		NPERGROUP = .

CI=DIFF	Prob(width)
	Sample size
		PROBWIDTH = .
		NTOTAL = .
		NPERGROUP = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test or requests a solution for alpha with a missing value (ALPHA=.). The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CI
CI=DIFF

specifies an analysis of precision of the confidence interval for the mean difference, assuming equal variances. Instead of power, the relevant probability for this analysis is the probability that the interval half-width is at most the value specified by the HALFWIDTH= option. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

CV=*number-list*

specifies the coefficient of variation assumed to be common to both groups. The coefficient of variation is defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

DIST=LOGNORMAL
DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value (also the only acceptable value in each case) is NORMAL for TEST=DIFF, TEST=DIFF_SATT, TEST=EQUIV_DIFF, and CI=DIFF; and LOGNORMAL for TEST=RATIO and TEST=EQUIV_RATIO.

GROUPMEANS=*grouped-number-list*
GMEANS=*grouped-number-list*

specifies the two group means or requests a solution for one group mean given the other. Means are in the original scale. They are arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option cannot be used with the CI=DIFF analysis. When TEST=EQUIV_DIFF, the means are interpreted as the reference mean (first) and the treatment mean (second). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPNS= *grouped-number-list*
GNS= *grouped-number-list*

specifies the two group sample sizes or requests a solution for one group sample size given the other. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPSTDDEVS=*grouped-number-list*
GSTDDEVS=*grouped-number-list*
GROUPSTDS=*grouped-number-list*
GSTDS=*grouped-number-list*

specifies the standard deviation of each group. Unlike the STDDEV= option, the GROUPSTDDEVS= option supports different values for each group. It is valid only for the Satterthwaite t test (TEST=DIFF_SATT DIST=NORMAL). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPWEIGHTS= *grouped-number-list*
GWEIGHTS= *grouped-number-list*

specifies the sample size allocation weights for the two groups, or requests a solution for one group weight given the other. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can only be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). Values must be greater than 0 when DIST=LOGNORMAL. This option can only be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the "Specifying

Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MEANDIFF=*number-list*

specifies the mean difference, defined as $\mu_2 - \mu_1$, or requests a solution for the mean difference with a missing value (MEANDIFF=.). This option can only be used with the TEST=DIFF, TEST=DIFF_SATT, and TEST=EQUIV_DIFF analyses. When TEST=EQUIV_DIFF, the mean difference is interpreted as the treatment mean minus the reference mean. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

MEANRATIO=*number-list*

specifies the geometric mean ratio, defined as γ_2/γ_1. This option can only be used with the TEST=RATIO and TEST=EQUIV_RATIO analyses. When TEST=EQUIV_RATIO, the mean ratio is interpreted as the treatment mean divided by the reference mean. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP= *number-list*
NPERG= *number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLDIFF=*number-list*
NULLD=*number-list*

specifies the null mean difference. The default value is 0. This option can only be used with the TEST=DIFF and TEST=DIFF_SATT analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NULLRATIO=*number-list*
NULLR=*number-list*

specifies the null mean ratio. The default value is 1. This option can only be used with the TEST=RATIO analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- NULLDIFF
- NULLRATIO
- LOWER
- UPPER
- ALPHA
- GROUPMEANS
- MEANDIFF
- MEANRATIO
- HALFWIDTH
- STDDEV
- GROUPSTDDEVS
- CV
- GROUPWEIGHTS
- NTOTAL
- NPERGROUP
- GROUPNS
- POWER
- PROBTYPE
- PROBWIDTH

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the TWOSAMPLEMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the TWOSAMPLEMEANS statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. This option cannot be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of

CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean difference is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option. You may use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can only be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*.

CONDITIONAL	width probability conditional on interval containing the mean
UNCONDITIONAL	unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can only be used with the CI=DIFF analysis. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are

1	1-sided with alternative hypothesis in same direction as effect
2	2-sided
U	upper 1-sided with alternative greater than null value
L	lower 1-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between negative infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option cannot be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. The default value is 2.

STDDEV=*number-list*

specifies the standard deviation assumed to be common to both groups, or requests a solution for the common standard deviation with a missing value (STDDEV=.). This option can only be used with DIST=NORMAL. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

TEST
TEST=DIFF
TEST=DIFF_SATT
TEST=EQUIV_DIFF
TEST=EQUIV_RATIO
TEST=RATIO

specifies the statistical analysis. TEST or TEST=DIFF (the default) specifies a pooled t test on the mean difference, assuming equal variances. TEST=DIFF_SATT specifies a Satterthwaite unpooled t test on the mean difference, assuming unequal variances. TEST=EQUIV_DIFF specifies an additive equivalence test of the mean difference using a two one-sided tests (TOST) analysis (Schuirman 1987). TEST=EQUIV_RATIO specifies a multiplicative equivalence test of the mean ratio using a TOST analysis. TEST=RATIO specifies a pooled t test on the mean ratio, assuming equal coefficients of variation. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

UPPER=*number-list*

specifies the upper equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can only be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (using the TEST= option)
- confidence interval precision (using the CI= option)

To specify the means, choose one of the following parameterizations:

- individual group means (using the GROUPMEANS= option)
- mean difference (using the MEANDIFF= option)
- mean ratio (using the MEANRATIO= option)

To specify standard deviations in the Satterthwaite t test (TEST=DIFF_SATT), choose one of the following parameterizations:

- common standard deviation (using the STDDEV= option)
- individual group standard deviations (using the GROUPSTDDEVS= option)

To specify the sample sizes and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (using the NPERGROUP= option)
- total sample size and allocation weights (using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLEMEANS statement.

Two-sample t Test Assuming Equal Variances

You can use the NPERGROUP= option in a balanced design and express effects in terms of the mean difference. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a 2-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff
      meandiff = 7
      stddev = 12
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design using the NTOTAL= and GROUPWEIGHTS= options and express effects in terms of individual group means.

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupweights = (2 3)
      ntotal = .
      power = 0.9;
run;
```

Another way to specify the sample sizes is with the GROUPNS= option.

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupns = (25 40)
      power = .;
run;
```

Two-sample Satterthwaite t Test Assuming Unequal Variances

Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a 2-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff_satt
      meandiff = 3
      groupstddevs = 5 | 8
      groupweights = (1 2)
      ntotal = 60
      power = .;
run;
```

Two-sample Pooled t Test of Mean Ratio with Lognormal Data

Default values for the DIST=, SIDES=, NULLRATIO=, and ALPHA= options specify a 2-sided test of mean ratio = 1 assuming a lognormal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=ratio
      meanratio = 7
      cv = 0.8
      groupns = 50 | 70
      power = .;
run;
```

Additive Equivalence Test for Mean Difference with Normal Data

A default value of GROUPWEIGHTS=(1 1) specifies a balanced design. Default values for the DIST= and ALPHA= options specify a significance level of 0.05 and an assumption of normally distributed data.

```
proc power;
   twosamplemeans test=equiv_diff
      lower = 2
      upper = 5
      meandiff = 4
      stddev = 8
      ntotal = .
      power = 0.9;
run;
```

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data

Default values for the DIST= and ALPHA= options specify a significance level of 0.05 and an assumption of lognormally distributed data.

```
proc power;
   twosamplemeans test=equiv_ratio
      lower = 3
      upper = 7
      meanratio = 5
      cv = 0.75
      npergroup = 50
      power = .;
run;
```

Confidence Interval for Mean Difference

By default CI=DIFF analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean difference. The defaults of SIDES=2 and ALPHA=0.05 specify a 2-sided interval with a confidence level of 0.95.

```
proc power;
   twosamplemeans ci = diff
      halfwidth = 4
      stddev = 8
      groupns = (30 35)
      probwidth = .;
run;
```

TWOSAMPLESURVIVAL Statement

TWOSAMPLESURVIVAL < *options* > **;**

The TWOSAMPLESURVIVAL statement performs power and sample size analyses for comparing two survival curves. The log-rank, Gehan, and Tarone-Ware rank tests are supported.

Summary of Options

Table 57.20 summarizes categories of options available in the TWOSAMPLESURVIVAL statement.

Table 57.20. Summary of Options in the TWOSAMPLESURVIVAL Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= ACCRUALTIME= FOLLOWUPTIME= TOTALTIME= SIDES=
Specify effects	CURVE= GROUPMEDSURVTIMES= GROUPSURVEXPHAZARDS= GROUPSURVIVAL= HAZARDRATIO=

Table 57.20. (continued)

Task	Options
	REFSURVEXPHAZARD= REFSURVIVAL=
Specify loss information	GROUPLOSS= GROUPLOSSEXPHAZARDS= GROUPMEDLOSSTIMES=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Specify computational method	NSUBINTERVAL=
Control ordering in output	OUTPUTORDER=

Table 57.21 summarizes the valid result parameters for different analyses in the TWOSAMPLESURVIVAL statement.

Table 57.21. Summary of Result Parameters in the TWOSAMPLESURVIVAL Statement

Analyses	Solve for	Syntax
TEST=GEHAN	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

TEST=LOGRANK	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

TEST=TARONEWARE	Power	POWER = .

	Sample size	NTOTAL = .
		NPERGROUP = .

Dictionary of Options

ACCRUALTIME=*number-list*
ACCTIME=*number-list*
ACCRUALT=*number-list*
ACCT=*number-list*

specifies the accrual time. Accrual is assumed to occur uniformly from time 0 to the time specified by the ACCRUALTIME= option. If the GROUPSURVIVAL= or REFSURVIVAL= options are used, then the value of the total time (the sum of accrual and follow-up times) must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

CURVE("label")=points

defines a survival curve.

For the CURVE= option,

label identifies the curve in the output and with the GROUPLOSS=, GROUPSURVIVAL=, and REFSURVIVAL= options.

points specifies one or more (time, survival) pairs on the curve, where the survival value denotes the probability of surviving until at least the specified time.

A single-point curve is interpreted as exponential, and a multipoint curve is interpreted as piecewise linear. Points can be expressed in either of two forms:

- a series of time:survival pairs separated by spaces. For example,

  ```
  1:0.9 2:0.7 3:0.6
  ```

- a DOLIST of times enclosed in parentheses, followed by a colon (:), followed by a DOLIST of survival values enclosed in parentheses. For example,

  ```
  (1 to 3 by 1):(0.9 0.7 0.6)
  ```

  The DOLIST format is the same as in the DATA step language.

Points can also be expressed as combinations of the two forms, for example,

```
1:0.9 2:0.8 (3 to 6 by 1):(0.7 0.65 0.6 0.55)
```

The points have the following restrictions:

- the time values must be nonnegative and strictly increasing
- the survival values must be strictly decreasing
- the survival value at a time of 0 must be equal to 1
- if there is only one point, then the time must be greater than 0, and the survival value cannot be 0 or 1

FOLLOWUPTIME=*number-list*
FUTIME=*number-list*
FOLLOWUPT=*number-list*
FUT=*number-list*

specifies the follow-up time, the amount of time in the study past the accrual time. If the GROUPSURVIVAL= or REFSURVIVAL= options are used, then the value of the total time (the sum of accrual and follow-up times) must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

GROUPLOSS= *grouped-name-list*
GLOSS= *grouped-name-list*

specifies the exponential loss survival curve for each group, using labels specified with the CURVE= option. Loss is assumed to follow an exponential curve, indicating the expected rate of loss to follow-up over time. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-name-list*.

GROUPLOSSEXPHAZARDS= *grouped-number-list*
GLOSSEXPHAZARDS= *grouped-number-list*
GROUPLOSSEXPHS= *grouped-number-list*
GLOSSEXPHS= *grouped-number-list*

specifies the exponential hazards of the loss in each group. Loss is assumed to follow an exponential curve, indicating the expected rate of loss to follow-up over time. If none of the GROUPLOSSEXPHAZARDS=, GROUPLOSS=, and GROUPMEDLOSSTIMES= options are used, the default of GROUPLOSSEXPHAZARDS=(0 0) indicates no loss to follow-up. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPMEDLOSSTIMES= *grouped-number-list*
GMEDLOSSTIMES= *grouped-number-list*
GROUPMEDLOSSTS= *grouped-number-list*
GMEDLOSSTS= *grouped-number-list*

specifies the median times of the loss in each group. Loss is assumed to follow an exponential curve, indicating the expected rate of loss to follow-up over time. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPMEDSURVTIMES= *grouped-number-list*
GMEDSURVTIMES= *grouped-number-list*
GROUPMEDSURVTS= *grouped-number-list*
GMEDSURVTS= *grouped-number-list*

specifies the median survival times in each group. When the GROUPMEDSURVTIMES= option is used, the survival curve in each group is assumed to be exponential. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPNS= *grouped-number-list*
GNS= *grouped-number-list*

specifies the two group sample sizes or requests a solution for one group sample size given the other. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPSURVEXPHAZARDS= *grouped-number-list*
GSURVEXPHAZARDS= *grouped-number-list*
GROUPSURVEXPHS= *grouped-number-list*
GEXPHS= *grouped-number-list*

specifies exponential hazard rates of the survival curve for each group. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

GROUPSURVIVAL= *grouped-name-list*
GSURVIVAL= *grouped-name-list*
GROUPSURV= *grouped-name-list*
GSURV= *grouped-name-list*

specifies the survival curve for each group, using labels specified with the CURVE= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-name-list*.

GROUPWEIGHTS= *grouped-number-list*
GWEIGHTS= *grouped-number-list*

specifies the sample size allocation weights for the two groups, or requests a solution for one group weight given the other. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *grouped-number-list*.

HAZARDRATIO= *number-list*
HR= *number-list*

specifies the hazard ratio of the second group's survival curve to the first group's survival curve. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the "Sample Size Adjustment Options" section on page 3494 for information on the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP= *number-list*
NPERG= *number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NSUBINTERVAL= *number-list*
NSUBINTERVALS= *number-list*
NSUB= *number-list*
NSUBS= *number-list*

specifies the number of subintervals per unit time to use in internal calculations. Higher values increase computational time and memory requirements but generally lead to more accurate results. The default value is 12. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

NTOTAL= *number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) produces output sorted respectively by

- SIDES
- ACCRUALTIME
- FOLLOWUPTIME
- TOTALTIME
- NSUBINTERVAL
- ALPHA
- REFSURVIVAL
- GROUPSURVIVAL
- REFSURVEXPHAZARD
- HAZARDRATIO
- GROUPSURVEXPHAZARDS
- GROUPMEDSURVTIMES
- GROUPLOSSEXPHAZARDS
- GROUPLOSS
- GROUPMEDLOSSTIMES
- GROUPWEIGHTS

- NTOTAL
- NPERGROUP
- GROUPNS
- POWER

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order that their corresponding options are specified in the TWOSAMPLESURVIVAL statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order that their corresponding options are specified in the TWOSAMPLESURVIVAL statement.

POWER= *number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

REFSURVEXPHAZARD= *number-list*
REFSURVEXPH= *number-list*

specifies the exponential hazard rate of the survival curve for the first (reference) group. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

REFSURVIVAL= *name-list*
REFSURV= *name-list*

specifies the survival curve for the first (reference) group, using labels specified with the CURVE= option. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *name-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and direction of the statistical test or confidence interval. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *keyword-list*. Valid keywords and their interpretation are

1 1-sided with alternative hypothesis in same direction as effect

2 2-sided

U upper 1-sided with the alternative hypothesis favoring better survival in the second group

L lower 1-sided with the alternative hypothesis favoring better survival in the first (reference) group

The default value is 2.

TEST=GEHAN
TEST=LOGRANK
TEST=TARONEWARE

specifies the statistical analysis. TEST=GEHAN specifies the Gehan rank test. TEST=LOGRANK (the default) specifies the log-rank test. TEST=TARONEWARE specifies the Tarone-ware rank test.

TOTALTIME=*number-list*
TOTALT=*number-list*

specifies the total time, which is equal to the sum of accrual and follow-up times. If the GROUPSURVIVAL= or REFSURVIVAL= options are used, then the value of the total time must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Restrictions on Option Combinations

To specify the survival curves, choose one of the following parameterizations:

- arbitrary piecewise linear or exponential curves (using the CURVE= and GROUPSURVIVAL= options)
- curves with proportional hazards (using the CURVE=, REFSURVIVAL=, and HAZARDRATIO= options)
- exponential curves, using one of the following parameterizations:
  - median survival times (using the GROUPMEDSURVTIMES= option)
  - the hazard ratio and the hazard of the reference curve (using the HAZARDRATIO= and REFSURVEXPHAZARD= options)
  - the individual hazards (using the GROUPSURVEXPHAZARDS= option)

To specify the study time, use any two of the following three options:

- accrual time (using the ACCRUALTIME= option)
- follow-up time (using the FOLLOWUPTIME= option)
- total time, the sum of accrual and follow-up times (using the TOTALTIME= option)

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (using the NPERGROUP= option)
- total sample size and allocation weights (using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (using the GROUPNS= option)

To specify the exponential loss curves, choose one of the following parameterizations:

- a point on the loss curve of each group (using the CURVE= and GROUPLOSS= options)
- median loss times (using the GROUPMEDLOSSTIMES= option)
- the individual loss hazards (using the GROUPLOSSEXPHAZARDS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLESURVIVAL statement.

Log-Rank Test for Two Survival Curves

You can use the NPERGROUP= option in a balanced design and specify piecewise linear or exponential survival curves using the CURVE= and GROUPSURVIVAL= options. Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a 2-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=logrank
      curve("Control")   = (1 2 3):(0.8 0.7 0.6)
      curve("Treatment") = (5):(.6)
      groupsurvival = "Control" | "Treatment"
      accrualtime = 2
      followuptime = 1
      npergroup = 50
      power = .;
run;
```

In the preceding example, the “Control” curve is piecewise linear (since it has more than one point), and the “Treatment” curve is exponential (since it has only one point).

You can also specify an unbalanced design using the NTOTAL= and GROUPWEIGHTS= options and specify piecewise linear or exponential survival curves with proportional hazards using the CURVE=, REFSURVIVAL=, and HAZARDRATIO= options.

```
proc power;
   twosamplesurvival test=logrank
      curve("Control")   = (1 2 3):(0.8 0.7 0.6)
      refsurvival = "Control"
      hazardratio = 1.5
      accrualtime = 2
      followuptime = 1
      groupweights = (1 2)
      ntotal = .
      power = 0.8;
run;
```

You can also specify sample sizes with the GROUPNS= option and specify exponential survival curves in terms of median survival times.

```
proc power;
   twosamplesurvival test=logrank
      groupmedsurvtimes = (16 22)
      accrualtime = 6
      totaltime = 18
      groupns = 40 | 60
      power = .;
run;
```

You can also specify exponential survival curves in terms of the hazard ratio and reference hazard. The default value of the GROUPWEIGHTS= option specifies a balanced design.

```
proc power;
   twosamplesurvival test=logrank
      hazardratio = 1.2
      refsurvexphazard = 0.7
      accrualtime = 2
      totaltime = 4
      ntotal = 100
      power = .;
run;
```

You can also specify exponential survival curves in terms of the individual hazards.

```
proc power;
   twosamplesurvival test=logrank
      groupsurvexphazards = 0.7 | 0.84
      accrualtime = 2
      totaltime = 4
      ntotal = .
      power = 0.9;
run;
```

Gehan Rank Test for Two Survival Curves

Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a 2-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=gehan
      groupmedsurvtimes = 5 | 7
      accrualtime = 3
      totaltime = 6
      npergroup = .
      power = 0.8;
run;
```

Tarone-Ware Rank Test for Two Survival Curves

Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a 2-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=taroneware
      groupmedsurvtimes = 5 | 7
      accrualtime = 3
      totaltime = 6
      npergroup = 100
      power = .;
run;
```

PLOT Statement

PLOT < *plot-options* > < **/** *graph-options* > **;**

The PLOT statement produces a graph or set of graphs for the sample size analysis defined by the previous analysis statement. The *plot-options* define the plot characteristics, and the *graph-options* are SAS/GRAPH-style options.

Options

You can specify the following *plot-options* in the PLOT statement.

INTERPOL=JOIN
INTERPOL=NONE

specifies the type of curve to draw through the computed points. The INTERPOL=JOIN option connects computed points by straight lines. The INTERPOL=NONE option leaves computed points unconnected.

KEY= BYCURVE < **(** *bycurve-options* **)** >
KEY= BYFEATURE < **(** *byfeature-options* **)** >
KEY= ONCURVES

specifies the style of key (or “legend”) for the plot. The default is KEY=BYFEATURE, which specifies a key with a column of entries for each plot feature (line style, color, and/or symbol). Each entry shows the mapping between a value of the feature and the value(s) of the analysis parameter(s) linked to that feature. The KEY=BYCURVE option specifies a key with each row identifying a distinct curve in the plot. The KEY=ONCURVES option places a curve-specific label adjacent to each curve.

You can specify the following *byfeature-options* in parentheses after the KEY=BYCURVE option.

NUMBERS=OFF

NUMBERS=ON specifies how the key should identify curves. If NUMBERS=OFF, then the key includes symbol, color, and line style samples to identify the curves. If NUMBERS=ON, then the key includes numbers matching numeric labels placed adjacent to the curves. The default is NUMBERS=ON.

POS=BOTTOM

POS=INSET specifies the position of the key. The POS=BOTTOM option places the key below the x-axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

You can specify the following *byfeature-options* in parentheses after the KEY=BYFEATURE option.

POS=BOTTOM

POS=INSET specifies the position of the key. The POS=BOTTOM option places the key below the x-axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

MARKERS=ANALYSIS
MARKERS=COMPUTED
MARKERS=NICE
MARKERS=NONE

specifies the locations for plotting symbols.

The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the values of the relevant input parameter from the analysis statement preceding the PLOT statement.

The MARKERS=COMPUTED option (the default) places plotting symbols at the locations of actual computed points from the sample size analysis.

The MARKERS=NICE option places plotting symbols at tick mark locations (corresponding to the argument axis).

The MARKERS=NONE option disables plotting symbols.

MAX=*number*

specifies the maximum of the range of values for the parameter associated with the “argument” axis (the axis that is *not* representing the parameter being solved for). The default is the maximum value occurring for this parameter in the analysis statement preceding the PLOT statement.

MIN=*number*

specifies the minimum of the range of values for the parameter associated with the “argument” axis (the axis that is *not* representing the parameter being solved for). The default is the minimum value occurring for this parameter in the analysis statement preceding the PLOT statement.

NPOINTS=*number*
NPTS=*number*

specifies the number of values for the parameter associated with the “argument” axis (the axis that is *not* representing the parameter being solved for). You cannot use the NPOINTS= and STEP= options simultaneously. The default value for typical situations is 20.

STEP=*number*

specifies the increment between values of the parameter associated with the “argument” axis (the axis that is *not* representing the parameter being solved for). You cannot use the STEP= and NPOINTS= options simultaneously. By default, the NPOINTS= option is used instead of the STEP= option.

VARY (*feature* < **BY** *parameter-list* > **...** *feature* < **BY** *parameter-list* > **)**

specifies how plot features should be linked to varying analysis parameters. Available plot *features* are COLOR, LINESTYLE, PANEL, and SYMBOL. A “panel” refers to a separate plot with a heading identifying the subset of values represented in the plot.

The *parameter-list* is a list of one or more names separated by spaces. Each name must match the name of an analysis option used in the analysis statement preceding the PLOT statement. Also, the name must be the *primary* name for the analysis option, that is, the one listed first in the syntax description.

If you omit the < BY *parameter-list* > portion for a feature, then one or more multivalued parameters from the analysis will be automatically selected for you.

X=EFFECT
X=N
X=POWER

specifies a plot with the requested type of parameter on the x-axis and the parameter being solved for on the y-axis. When X=EFFECT, the parameter assigned to the x-axis is the one most representative of “effect size.” When X=N, the parameter assigned to the x-axis is the sample size. When X=POWER, the parameter assigned to the x-axis is the one most representative of “power” (either power itself or a similar probability, such as Prob(Width) for confidence interval analyses). You cannot use the X= and Y= options simultaneously. The default is X=POWER, unless the result parameter is power or Prob(Width), in which case the default is X=N.

You can only use the X=N option when a scalar sample size parameter is used as input in the analysis. For example, X=N can be used with total sample size or sample size per group, or with two group sample sizes when one is being solved for.

Table 57.22 summarizes the parameters representing effect size in different analyses.

Table 57.22. Effect Size Parameters For Different Analyses

Analysis Statement and Options	Effect Size Parameters
MULTREG	Partial correlation or R^2 difference
ONECORR	Correlation
ONESAMPLEFREQ	Proportion
ONESAMPLEMEANS TEST=T, ONESAMPLEMEANS TEST=EQUIV	Mean
ONESAMPLEMEANS CI=T	CI half-width
ONEWAYANOVA	none
PAIREDFREQ	Discordant proportion difference or ratio
PAIREDMEANS TEST=DIFF, PAIREDMEANS TEST=EQUIV_DIFF	Mean difference
PAIREDMEANS TEST=RATIO, PAIREDMEANS TEST=EQUIV_RATIO	Mean ratio
PAIREDMEANS CI=DIFF	CI half-width
TWOSAMPLEFREQ	Proportion difference, odds ratio, or relative risk
TWOSAMPLEMEANS TEST=DIFF, TWOSAMPLEMEANS TEST=DIFF_SATT, TWOSAMPLEMEANS TEST=EQUIV_DIFF	Mean difference
TWOSAMPLEMEANS TEST=RATIO, TWOSAMPLEMEANS TEST=EQUIV_RATIO	Mean ratio
TWOSAMPLEMEANS CI=DIFF	CI half-width
TWOSAMPLESURVIVAL	Hazard ratio if used, else none

XOPTS= (*x-options* **)**

specifies plot characteristics pertaining to the x-axis.

You can specify the following *x-options* in parentheses.

CROSSREF=NO

CROSSREF=YES specifies whether the reference lines defined by the REF= *x-option* should be crossed with a reference line on the y-axis that indicates the solution point on the curve.

REF=*number-list* specifies locations for reference lines extending from the x-axis across the entire plotting region. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

Y=EFFECT
Y=N
Y=POWER

specifies a plot with the requested type of parameter on the y-axis and the parameter being solved for on the x-axis. When Y=EFFECT, the parameter assigned to the y-axis is the one most representative of "effect size." When Y=N, the parameter assigned to the y-axis is the sample size. When Y=POWER, the parameter assigned to the y-axis is the one most representative of "power" (either power itself or a similar probability, such as Prob(Width) for confidence interval analyses). You cannot use the Y= and X= options simultaneously. By default, the X= option is used instead of the Y= option.

YOPTS= (*y-options* **)**

specifies plot characteristics pertaining to the y-axis.

You can specify the following *y-options* in parentheses.

CROSSREF=NO

CROSSREF=YES specifies whether the reference lines defined by the REF= *y-option* should be crossed with a reference line on the x-axis that indicates the solution point on the curve.

REF=*number-list* specifies locations for reference lines extending from the y-axis across the entire plotting region. See the "Specifying Value Lists in Analysis Statements" section on page 3490 for information on specifying the *number-list*.

You can specify the following *graph-options* in the PLOT statement after a slash (/).

DESCRIPTION='*string* **'**

specifies a descriptive string of up to 40 characters that appears in the "Description" field of the graphics catalog. The description does not appear on the plots. By default, PROC POWER assigns a description either of the form "*Y* versus *X*" (for a single-panel plot) or of the form "*Y* versus *X* (*S*)," where *Y* is the parameter on the y-axis, *X* is the parameter on the x-axis, and *S* is a description of the subset represented on the current panel of a multipanel plot.

NAME='*string* **'**

specifies a name of up to eight characters for the catalog entry for the plot. The default name is PLOT*n*, where *n* is the number of the plot statement within the current invocation of PROC POWER. If the name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry; for example, PLOT11 and PLOT12 for the second and third panels of a multipanel plot generated in the first PLOT statement in an invocation of PROC POWER.

Details

Overview of Power Concepts

In statistical hypothesis testing, you typically express the belief that some effect exists in a population by specifying an alternative hypothesis H_1. You state a null hypothesis H_0 as the assertion that the effect does *not* exist and attempt to gather evidence to reject H_0 in favor of H_1. Evidence is gathered in the form of sample data, and a statistical test is used to assess H_0. If H_0 is rejected but there really is *no* effect, this is called a *Type 1 error*. The probability of a Type 1 error is usually designated "alpha" or α, and statistical tests are designed to ensure that α is suitably small (for example, less than 0.05).

If there really is an effect in the population but H_0 is *not* rejected in the statistical test, then a *Type 2 error* has been made. The probability of a Type 2 error is usually designated "beta" or β. The probability $1 - \beta$ of avoiding a Type 2 error, that is, correctly rejecting H_0 and achieving statistical significance, is called the *power*. (**Note:** Another more general definition of power is the probability of rejecting H_0 for any given set of circumstances, even those corresponding to H_0 being true. The POWER procedure uses this more general definition.)

An important goal in study planning is to ensure an acceptably high level of power. Sample size plays a prominent role in power computations because the focus is often on determining a sufficient sample size to achieve a certain power, or assessing the power for a range of different sample sizes.

Some of the analyses in the POWER procedure focus on *precision* rather than power. An analysis of confidence interval precision is analogous to a traditional power analysis, with "CI Half-Width" taking the place of effect size and "Prob(Width)" taking the place of power. The *CI Half-Width* is the margin of error associated with the confidence interval, the distance between the point estimate and an endpoint. The *Prob(Width)* is the probability of obtaining a confidence interval with *at most* a target half-width.

Summary of Analyses

Table 57.23 gives a summary of the analyses supported in the POWER procedure. The name of the analysis statement reflects the type of data and design. The TEST=, CI=, and DIST= options specify the focus of the statistical hypothesis (in other words, the criterion on which the research question is based) and the test statistic to be used in data analysis.

Table 57.23. Summary of Analyses

Statement	**Options**	
Multiple linear regression: Type III F test	MULTREG	
Correlation: Fisher's z test	ONECORR	DIST=FISHERZ
Correlation: t test	ONECORR	DIST=T

Table 57.23. (continued)

Statement	Options	
Binomial proportion: Exact test	ONESAMPLEFREQ	TEST=EXACT
Binomial proportion: z test	ONESAMPLEFREQ	TEST=Z
Binomial proportion: z test with continuity adjustment	ONESAMPLEFREQ	TEST=ADJZ
One-sample t test	ONESAMPLEMEANS	TEST=T

One-sample t test with lognormal data	ONESAMPLEMEANS	TEST=T
		DIST=LOGNORMAL

One-sample equivalence test for mean of normal data	ONESAMPLEMEANS	TEST=EQUIV

One-sample equivalence test for mean of lognormal data	ONESAMPLEMEANS	TEST=EQUIV
		DIST=LOGNORMAL

Confidence interval for a mean	ONESAMPLEMEANS	CI=T
One-way ANOVA: One-degree-of-freedom contrast	ONEWAYANOVA	TEST=CONTRAST
One-way ANOVA: Overall F test	ONEWAYANOVA	TEST=OVERALL
McNemar exact conditional test	PAIREDFREQ	
McNemar normal approximation test	PAIREDFREQ	DIST=NORMAL
Paired t test	PAIREDMEANS	TEST=DIFF
Paired t test of mean ratio with lognormal data	PAIREDMEANS	TEST=RATIO
Paired additive equivalence of mean difference with normal data	PAIREDMEANS	TEST=EQUIV_DIFF
Paired multiplicative equivalence of mean ratio with lognormal data	PAIREDMEANS	TEST=EQUIV_RATIO
Confidence interval for mean of paired differences	PAIREDMEANS	CI=DIFF
Pearson chi-square test for two independent proportions	TWOSAMPLEFREQ	TEST=PCHI
Fisher’s exact test for two independent proportions	TWOSAMPLEFREQ	TEST=FISHER
Likelihood ratio chi-square test for two independent proportions	TWOSAMPLEFREQ	TEST=LRCHI
Two-sample t test assuming equal variances	TWOSAMPLEMEANS	TEST=DIFF

Table 57.23. (continued)

Statement	Options
Two-sample Satterthwaite t test assuming unequal variances	TWOSAMPLEMEANS TEST=DIFF_SATT
Two-sample pooled t test of mean ratio with lognormal data	TWOSAMPLEMEANS TEST=RATIO
Two-sample additive equivalence of mean difference with normal data	TWOSAMPLEMEANS TEST=EQUIV_DIFF
Two-sample multiplicative equivalence of mean ratio with lognormal data	TWOSAMPLEMEANS TEST=EQUIV_RATIO
Two-sample confidence interval for mean difference	TWOSAMPLEMEANS CI=DIFF
Log-rank test for comparing two survival curves	TWOSAMPLESURVIVAL TEST=LOGRANK
Gehan rank test for comparing two survival curves	TWOSAMPLESURVIVAL TEST=GEHAN
Tarone-Ware rank test for comparing two survival curves	TWOSAMPLESURVIVAL TEST=TARONEWARE

Specifying Value Lists in Analysis Statements

To specify one or more scenarios for an analysis parameter (or set of parameters), you provide a list of values for the statement option that corresponds to the parameter(s). To identify the parameter you wish to solve for, you place missing values in the appropriate list.

There are five basic types of such lists: *keyword-lists*, *number-lists*, *grouped-number-lists*, *name-lists*, and *grouped-name-lists*. Some parameters, such as the direction of a test, have values represented by one or more keywords in a *keyword-list*. Scenarios for scalar-valued parameters, such as power, are represented by a *number-list*. Scenarios for groups of scalar-valued parameters, such as group sample sizes in a multigroup design, are represented by a *grouped-number-list*. Scenarios for named parameters, such as reference survival curves, are represented by a *name-list*. Scenarios for groups of named parameters, such as group survival curves, are represented by a *grouped-name-list*.

The following subsections explain these five basic types of lists.

Keyword-lists

A *keyword-list* is a list of one or more keywords separated by spaces. For example, you can specify both 2-sided and upper-tailed versions of a one-sample t test:

```
SIDES = 2 U
```

Number-lists

A *number-list* can be one of two things: a series of one or more numbers expressed in the form of one or more DOLISTs, or a missing value indicator (.).

The DOLIST format is the same as in the DATA step language. For example, for the one-sample t test you can specify four scenarios (30, 50, 70, and 100) for a total sample size in any of the following ways.

```
NTOTAL = 30 50 70 100
NTOTAL = 30 to 70 by 20 100
```

A missing value identifies a parameter as the result parameter; it is valid only with options representing parameters you can solve for in a given analysis. For example, you can request a solution for NTOTAL:

```
NTOTAL = .
```

Grouped-number-lists

A *grouped-number-list* specifies multiple scenarios for numeric values in two or more groups, possibly including missing value indicators to solve for a specific group. The list can assume one of two general forms, a "crossed" version and a "matched" version.

Crossed Grouped-number-lists

The crossed version of a grouped number list consists of a series of *number-lists* (see the "Number-lists" section on page 3491), one representing each group, each separated by a vertical bar (|). The values for each group represent multiple scenarios for that group, and the scenarios for each individual group are crossed to produce the set of all scenarios for the analysis option. For example, you can specify the following six scenarios for the sizes (n_1, n_2) of two groups

$$\begin{array}{l}(20,30)(20,40)(20,50)\\(25,30)(25,40)(25,50)\end{array}$$

as follows:

```
GROUPNS = 20 25 | 30 40 50
```

If the analysis can solve for a value in one group given the other groups, then one of the *number-lists* in a *crossed grouped-number-list* can be a missing value indicator (.). For example, in a two-sample t test you can posit three scenarios for the group 2 sample size while solving for the group 1 sample size:

```
GROUPNS = . | 30 40 50
```

Some analyses can involve more than two groups. For example, you can specify 2 × 3 × 1 = 6 scenarios for the means of three groups in a one-way ANOVA as follows:

```
GROUPMEANS = 10 12 | 10 to 20 by 5 | 24
```

Matched Grouped-number-lists

The matched version of a grouped number list consists of a series of numeric lists each enclosed in parentheses. Each list consists of a value for each group and represents a single scenario for the analysis option. Multiple scenarios for the analysis option are represented by multiple lists. For example, you can express the crossed grouped-number-list

```
GROUPNS = 20 25 | 30 40 50
```

alternatively in a matched format:

```
GROUPNS = (20 30) (20 40) (20 50) (25 30) (25 40) (25 50)
```

The matched version is particularly useful when you wish to include only a subset of all combinations of individual group values. For example, you may want to pair 20 only with 50, and 25 only with 30 and 40:

```
GROUPNS = (20 50) (25 30) (25 40)
```

If the analysis can solve for a value in one group given the other groups, then you can replace the value for that group with a missing value indicator (.). If used, the missing value indicator must occur in the same group in every scenario. For example, you can solve for the group 1 sample size (as in the "Crossed Grouped-number-lists" section on page 3491) using a matched format:

```
GROUPNS = (. 30) (. 40) (. 50)
```

Some analyses can involve more than two groups. For example, you can specify two scenarios for the means of three groups in a one-way ANOVA:

```
GROUPMEANS = (15 24 32) (12 25 36)
```

Name-lists

A *name-list* is a list of one or more names in single or double quotes separated by spaces. For example, you can specify two scenarios for the reference survival curve in a log-rank test:

```
REFSURVIVAL = "Curve A" "Curve B"
```

Grouped-name-lists

A *grouped-name-list* specifies multiple scenarios for names in two or more groups. The list can assume one of two general forms, a "crossed" version and a "matched" version.

Crossed Grouped-name-lists

The crossed version of a grouped name list consists of a series of *name-lists* (see the “Name-lists” section on page 3492), one representing each group, each separated by a vertical bar (|). The values for each group represent multiple scenarios for that group, and the scenarios for each individual group are crossed to produce the set of all scenarios for the analysis option. For example, you can specify the following six scenarios for the survival curves (c_1, c_2) of two groups

$$(\text{“Curve A”}, \text{“Curve C”})(\text{“Curve A”}, \text{“Curve D”})(\text{“Curve A”}, \text{“Curve E”})$$
$$(\text{“Curve B”}, \text{“Curve C”})(\text{“Curve B”}, \text{“Curve D”})(\text{“Curve B”}, \text{“Curve E”})$$

as follows:

```
GROUPSURVIVAL = "Curve A" "Curve B" | "Curve C" "Curve D"
                                          "Curve E"
```

Matched Grouped-name-lists

The matched version of a grouped name list consists of a series of name lists each enclosed in parentheses. Each list consists of a name for each group and represents a single scenario for the analysis option. Multiple scenarios for the analysis option are represented by multiple lists. For example, you can express the crossed grouped-name-list

```
GROUPSURVIVAL = "Curve A" "Curve B" | "Curve C" "Curve D"
                                          "Curve E"
```

alternatively in a matched format:

```
GROUPSURVIVAL = ("Curve A" "Curve C")
                ("Curve A" "Curve D")
                ("Curve A" "Curve E")
                ("Curve B" "Curve C")
                ("Curve B" "Curve D")
                ("Curve B" "Curve E")
```

The matched version is particularly useful when you wish to include only a subset of all combinations of individual group values. For example, you may want to pair “Curve A” only with “Curve C”, and “Curve B” only with “Curve D” and “Curve E”:

```
GROUPSURVIVAL = ("Curve A" "Curve C")
                ("Curve B" "Curve D")
                ("Curve B" "Curve E")
```

Sample Size Adjustment Options

By default, PROC POWER rounds sample sizes conservatively (down in the input, up in the output) so that all total sizes (and individual group sample sizes, if a multigroup design) are integers. This is generally considered conservative because it selects the closest realistic design providing *at most* the power of the (possibly fractional) input or mathematically optimized design. In addition, in a multigroup design, all group sizes are adjusted to be multiples of the corresponding group weights. For example, if GROUPWEIGHTS = (2 6), then all group 1 sample sizes become multiples of 2, and all group 2 sample sizes become multiples of 6 (and all total sample sizes become multiples of 8).

With the NFRACTIONAL option, sample size input is not rounded, and sample size output (whether total or group-wise) are reported in two versions, a raw “fractional” version and a “ceiling” version rounded up to the nearest integer.

Whenever an input sample size is adjusted, both the original (“nominal”) and adjusted (“actual”) sample sizes are reported. Whenever computed output sample sizes are adjusted, both the original input (“nominal”) power and the achieved (“actual”) power at the adjusted sample size are reported.

Error and Information Output

The Error column in the main output table explains reasons for missing results and flags numerical results that are bounds rather than exact answers. For example, consider the sample size analysis implemented by the following statements:

```
proc power;
   twosamplefreq test=pchi
     oddsratio= 1.0001
     refproportion=.4
     nulloddsratio=1
     power=.9
     ntotal=.;
run;
```

```
                          The POWER Procedure
               Pearson Chi-square Test for Two Proportions

                        Fixed Scenario Elements

        Distribution                             Asymptotic normal
        Method                                Normal approximation
        Null Odds Ratio                                          1
        Reference (Group 1) Proportion                         0.4
        Odds Ratio                                          1.0001
        Nominal Power                                          0.9
        Number of Sides                                          2
        Alpha                                                 0.05
        Group 1 Weight                                           1
        Group 2 Weight                                           1


                           Computed N Total

           Actual
            Power      N Total    Error

            0.206     2.15E+09    Solution is a lower bound
```

Figure 57.6. Error Column

The output in Figure 57.6 reveals that the sample size to achieve a power of 0.9 could not be computed, but that the sample size 2.15E+09 achieves a power of 0.206.

The Information column provides further details about Error entries, warnings about any boundary conditions detected, and notes about any adjustments to input. Note that the Information column is hidden by default in the main output. You can view it by using the ODS OUTPUT statement to save the output as a data set and the PRINT procedure. For example, the following SAS statements print both the Error and Info columns for a power computation in a two-sample t test.

```
proc power;
   twosamplemeans
      meandiff= 0 7
      stddev=2
      ntotal=2 5
      power=.;
   ods output output=Power;
proc print noobs data=Power;
   var MeanDiff NominalNTotal NTotal Power Error Info;
run;
```

The output is shown in Figure 57.7.

```
Mean   Nominal
Diff   NTotal    NTotal   Power       Error       Info

  0       2         2       .     Invalid input   N too small / No effect
  0       5         4     0.050                   Input N adjusted / No effect
  7       2         2       .     Invalid input   N too small
  7       5         4     0.477                   Input N adjusted
```

Figure 57.7. Error and Information Columns

The mean difference of 0 specified with the MEANDIFF= option leads to a “No effect” message to appear in the Info column. The sample size of 2 specified with the NTOTAL= option leads to an “Invalid input” message in the Error column and an “NTotal too small” message in the Info column. The sample size of 5 leads to an “Input N adjusted” message in the Info column because it is rounded down to 4 to produce integer group sizes of 2 per group.

Displayed Output

If you use the PLOTONLY option in the PROC POWER statement, the procedure only displays graphical output. Otherwise, the displayed output of the POWER procedure includes the following:

- the “Fixed Scenario Elements” table, which shows all applicable single-valued analysis parameters, in the following order: distribution, method, parameters input explicitly, and parameters supplied with defaults
- an output table showing the following when applicable (in order): the index of the scenario, all multivalued input, ancillary results, the primary computed result, and error descriptions
- plots (if requested)

For each input parameter, the order of the input values is preserved in the output.

Ancillary results include the following:

- Actual Power, the achieved power, if it differs from the input (Nominal) power value
- Actual Prob(Width), the achieved precision probability, if it differs from the input (Nominal) probability value
- Actual Alpha, the achieved significance level, if it differs from the input (Nominal) alpha value
- fractional sample size, if the NFRACTIONAL option is used in the analysis statement

If sample size is the result parameter and the NFRACTIONAL option is used in the analysis statement, then both “Fractional” and “Ceiling” sample size results are displayed. Fractional sample sizes correspond to the “Nominal” values of power or precision probability. Ceiling sample sizes are simply the fractional sample sizes rounded up to the nearest integer; they correspond to “Actual” values of power or precision probability.

ODS Table Names

PROC POWER assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 57.24. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 57.24. ODS Tables Produced in PROC POWER

ODS Table Name	Description	Statement
FixedElements	factoid with single-valued analysis parameters	default*
Output	all input and computed analysis parameters, error messages, and information messages for each scenario	default
PlotContent	data contained in plots, including analysis parameters and indices identifying plot features. (**Note:** this table is saved as a data set and not displayed in PROC POWER output.)	PLOT

*Depends on input.

The ODS path names are created as follows:

- Power.<analysis statement name>$< n >$.FixedElements
- Power.<analysis statement name>$< n >$.Output
- Power.<analysis statement name>$< n >$.PlotContent
- Power.<analysis statement name>$< n >$.Plot$< m >$

where

- The Plot$< m >$ objects are the graphs.
- The $< n >$ indexing the analysis statement name is only used if there is more than one instance.
- The $< n >$ indexing the plots increases with every panel in every plot statement, resetting to 1 only at new analysis statements.

Computational Resources

Memory

In the TWOSAMPLESURVIVAL statement, the amount of required memory is roughly proportional to the product of the number of subintervals (specified by the NSUBINTERVAL= option) and the total time of the study (specified by the ACCRUALTIME=, FOLLOWUPTIME=, and TOTALTIME= options).

CPU Time

In the Satterthwaite t test analysis (TWOSAMPLEMEANS TEST=DIFF_SATT), the required CPU time grows as the mean difference decreases relative to the standard deviations. In the PAIREDFREQ statement, the required CPU time for the exact power computation (METHOD=EXACT) grows with the sample size.

Computational Methods and Formulas

This section describes the approaches used in PROC POWER to compute power for each analysis. The first subsection defines some common notation. The following subsections describe the various power analyses, including discussions of the data, statistical test, and power formula for each analysis. Unless otherwise indicated, computed values for parameters besides power (for example, sample size) are obtained by solving power formulas for the desired parameters.

Common Notation

Table 57.25 displays notation for some of the more common parameters across analyses. The Associated Syntax column shows examples of relevant analysis statement options, where applicable.

Table 57.25. Common Notation

Symbol	Description	Associated Syntax
α	significance level	ALPHA=
N	total sample size	NTOTAL=, NPAIRS=
n_i	sample size in ith group	NPERGROUP=, GROUPNS=
w_i	allocation weight for ith group (standardized to sum to 1)	GROUPWEIGHTS=
μ	(arithmetic) mean	MEAN=
μ_i	(arithmetic) mean in ith group	GROUPMEANS=, PAIREDMEANS=
μ_{diff}	(arithmetic) mean difference, $\mu_2 - \mu_1$ or $\mu_T - \mu_R$	MEANDIFF=
μ_0	null mean or mean difference (arithmetic)	NULL=, NULLDIFF=
γ	geometric mean	MEAN=
γ_i	geometric mean in ith group	GROUPMEANS=, PAIREDMEANS=
γ_0	null mean or mean ratio (geometric)	NULL=, NULLRATIO=

Table 57.25. (continued)

Symbol	Description	Associated Syntax
σ	standard deviation (or common standard deviation per group)	STDDEV=
σ_i	standard deviation in ith group	GROUPSTDDEVS=, PAIREDSTDDEVS=
σ_{diff}	standard deviation of differences	
CV	coefficient of variation, defined as the ratio of the standard deviation to the (arithmetic) mean	CV=, PAIREDCVS=
ρ	correlation	CORR=
μ_T, μ_R	treatment and reference (arithmetic) means for equivalence test	GROUPMEANS=, PAIREDMEANS=
γ_T, γ_R	treatment and reference geometric means for equivalence test	GROUPMEANS=, PAIREDMEANS=
θ_L	lower equivalence bound	LOWER=
θ_U	upper equivalence bound	UPPER=
$t(\nu, \delta)$	t distribution with d.f. ν and noncentrality δ	
$F(\nu_1, \nu_2, \lambda)$	F distribution with numerator d.f. ν_1, denominator d.f. ν_2, and noncentrality λ	
$t_{p;\nu}$	pth percentile of t distribution with d.f. ν	
$F_{p;\nu_1,\nu_2}$	pth percentile of F distribution with numerator d.f. ν_1 and denominator d.f. ν_2	
$\mathrm{Bin}(N, p)$	binomial distribution with sample size N and proportion p	

A "lower 1-sided" test is associated with SIDES=L (or SIDES=1 with the effect smaller than the null value), and an "upper 1-sided" test is associated with SIDES=U (or SIDES=1 with the effect larger than the null value).

Owen (1965) defines a function, known as Owen's Q, that is convenient for representing terms in power formulas for confidence intervals and equivalence tests:

$$Q_\nu(t, \delta; a, b) = \frac{\sqrt{2\pi}}{\Gamma(\frac{\nu}{2})2^{\frac{\nu-2}{2}}} \int_a^b \Phi\left(\frac{tx}{\sqrt{\nu}} - \delta\right) x^{\nu-1}\phi(x)\mathrm{d}x$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the density and cumulative distribution function of the standard normal distribution, respectively.

Analyses in the MULTREG Statement

Type III F Test in Multiple Regression (TEST=TYPE3)

Maxwell (2000) discusses a number of different ways to represent effect sizes (and to compute exact power based on them) in multiple regression. PROC POWER supports two of these, multiple partial correlation and R^2 in full and reduced models.

Let p denote the total number of predictors in the full model (excluding the intercept) and Y the response variable. You are testing that the coefficients of $p_1 \geq 1$ predictors in a set X_1 are 0, controlling for all of the other predictors X_{-1}, which is comprised of $p - p_1 \geq 0$ variables.

The hypotheses can be expressed in two different ways. The first is in terms of $\rho_{YX_1|X_{-1}}$, the multiple partial correlation between the predictors in X_1 and the response Y adjusting for the predictors in X_{-1}:

$$\begin{aligned} H_0 &: \rho^2_{YX_1|X_{-1}} = 0 \\ H_1 &: \rho^2_{YX_1|X_{-1}} > 0 \end{aligned}$$

The second is in terms of the multiple correlations in full ($\rho_{Y|(X_1,X_{-1})}$) and reduced ($\rho_{Y|X_{-1}}$) nested models:

$$\begin{aligned} H_0 &: \rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}} = 0 \\ H_1 &: \rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}} > 0 \end{aligned}$$

Note that the squared values of $\rho_{Y|(X_1,X_{-1})}$ and $\rho_{Y|X_{-1}}$ are the population R^2 values for full and reduced models.

The test statistic can be written in terms of the sample multiple partial correlation $R_{YX_1|X_{-1}}$,

$$F = \begin{cases} (N-1-p)\frac{R^2_{YX_1|X_{-1}}}{1-R^2_{YX_1|X_{-1}}}, & \text{intercept} \\ (N-p)\frac{R^2_{YX_1|X_{-1}}}{1-R^2_{YX_1|X_{-1}}}, & \text{no intercept} \end{cases}$$

or the sample multiple correlations in full ($R_{Y|(X_1,X_{-1})}$) and reduced ($R_{Y|X_{-1}}$) models,

$$F = \begin{cases} (N-1-p)\frac{R^2_{Y|(X_1,X_{-1})}-R^2_{Y|X_{-1}}}{1-R^2_{Y|(X_1,X_{-1})}}, & \text{intercept} \\ (N-p)\frac{R^2_{Y|(X_1,X_{-1})}-R^2_{Y|X_{-1}}}{1-R^2_{Y|(X_1,X_{-1})}}, & \text{no intercept} \end{cases}$$

The test is the usual Type III F test in multiple regression:

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} F \geq F_{1-\alpha}(p_1, N-1-p), & \text{intercept} \\ F \geq F_{1-\alpha}(p_1, N-p), & \text{no intercept} \end{cases}$$

Although the test is invariant to whether the predictors are assumed to be random or fixed, the power is affected by this assumption. If the response and predictors are assumed to have a joint multivariate normal distribution, then the exact power is given by the following formula:

$$\begin{aligned} \text{power} &= \begin{cases} P\left[\left(\frac{N-1-p}{p_1}\right)\left(\frac{R^2_{Y|(X_1,X_{-1})}}{1-R^2_{Y|(X_1,X_{-1})}}\right) \geq F_{1-\alpha}(p_1, N-1-p)\right], & \text{intercept} \\ P\left[\left(\frac{N-p}{p_1}\right)\left(\frac{R^2_{Y|(X_1,X_{-1})}}{1-R^2_{Y|(X_1,X_{-1})}}\right) \geq F_{1-\alpha}(p_1, N-p)\right], & \text{no intercept} \end{cases} \\ &= \begin{cases} P\left[R^2_{Y|(X_1,X_{-1})} \geq \frac{F_{1-\alpha}(p_1,N-1-p)}{F_{1-\alpha}(p_1,N-1-p)+\frac{N-1-p}{p_1}}\right], & \text{intercept} \\ P\left[R^2_{Y|(X_1,X_{-1})} \geq \frac{F_{1-\alpha}(p_1,N-p)}{F_{1-\alpha}(p_1,N-p)+\frac{N-p}{p_1}}\right], & \text{no intercept} \end{cases} \end{aligned}$$

The distribution of $R^2_{Y|(X_1,X_{-1})}$ (for any $\rho^2_{Y|(X_1,X_{-1})}$) is given in Chapter 32 of Johnson, Kotz, and Balakrishnan (1995). Sample size tables are presented in Gatsonis and Sampson (1989).

If the predictors are assumed to have fixed values, then the exact power is given by the noncentral F distribution. The noncentrality parameter is

$$\lambda = N\frac{\rho^2_{YX_1|X_{-1}}}{1-\rho^2_{YX_1|X_{-1}}}$$

or equivalently,

$$\lambda = N\frac{\rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}}}{1-\rho^2_{Y|(X_1,X_{-1})}}$$

The power is

$$\text{power} = \begin{cases} P\left(F(p_1, N-1-p, \lambda) \geq F_{1-\alpha}(p_1, N-1-p)\right), & \text{intercept} \\ P\left(F(p_1, N-p, \lambda) \geq F_{1-\alpha}(p_1, N-p)\right), & \text{no intercept} \end{cases}$$

The minimum acceptable input value of N depends on several factors, as shown in Table 57.26.

Table 57.26. Minimum Acceptable Sample Size Values in the MULTREG Statement

Predictor Type	Intercept in Model?	$p_1 = 1$?	Minimum N
Random	Yes	Yes	$p+3$
Random	Yes	No	$p+2$
Random	No	Yes	$p+2$
Random	No	No	$p+1$
Fixed	Yes	Yes or No	$p+2$
Fixed	No	Yes or No	$p+1$

Analyses in the ONECORR Statement

Fisher's z Test for Pearson Correlation (TEST=PEARSON DIST=FISHERZ)

Fisher's z transformation (Fisher 1921) of the sample correlation $R_{Y|(X_1,X_{-1})}$ is defined as

$$z = \frac{1}{2}\log\left(\frac{1+R_{Y|(X_1,X_{-1})}}{1-R_{Y|(X_1,X_{-1})}}\right)$$

Fisher's z test assumes the approximate normal distribution $N(\mu,\sigma^2)$ for z, where

$$\mu = \frac{1}{2}\log\left(\frac{1+\rho_{Y|(X_1,X_{-1})}}{1-\rho_{Y|(X_1,X_{-1})}}\right) + \frac{\rho_{Y|(X_1,X_{-1})}}{2(N-1-p^\star)}$$

and

$$\sigma^2 = \frac{1}{N-3-p^\star}$$

where $p^\star$ is the number of variables partialled out (Anderson 1984, pp. 132–133) and $\rho_{Y|(X_1,X_{-1})}$ is the partial correlation between Y and X_1 adjusting for the set of zero or more variables X_{-1}.

The test statistic

$$z^\star = (N-3-p^\star)^{\frac{1}{2}}\left[z - \frac{1}{2}\log\left(\frac{1+\rho_0}{1-\rho_0}\right) - \frac{\rho_0}{2(N-1-p^\star)}\right]$$

is assumed to have a normal distribution $N(\delta,\nu)$ where ρ_0 is the null partial correlation and δ and ν are derived from section 16.33 of Stuart and Ord (1994):

$$\begin{aligned}\delta \quad = \quad & (N-3-p^\star)^{\frac{1}{2}}\left[\frac{1}{2}\log\left(\frac{1+\rho_{Y|(X_1,X_{-1})}}{1-\rho_{Y|(X_1,X_{-1})}}\right) + \right. \\ & \frac{\rho_{Y|(X_1,X_{-1})}}{2(N-1-p^\star)}\left(1 + \frac{5+\rho^2_{Y|(X_1,X_{-1})}}{4(N-1-p^\star)} + \right. \\ & \left.\frac{11+2\rho^2_{Y|(X_1,X_{-1})}+3\rho^4_{Y|(X_1,X_{-1})}}{8(N-1-p^\star)^2}\right)\end{aligned}$$

$$
\begin{aligned}
& -\frac{1}{2}\log\left(\frac{1+\rho_0}{1-\rho_0}\right) - \frac{\rho_0}{2(N-1-p^\star)}\Bigg] \\
\nu \quad = \quad & \frac{N-3-p^\star}{N-1-p^\star}\left[1 + \frac{4-\rho^2_{Y|(X_1,X_{-1})}}{2(N-1-p^\star)} + \right. \\
& \left. \frac{22-6\rho^2_{Y|(X_1,X_{-1})} - 3\rho^4_{Y|(X_1,X_{-1})}}{6(N-1-p^\star)^2}\right]
\end{aligned}
$$

The approximate power is computed as

$$
\text{power} \quad = \quad \begin{cases} \Phi\left(\frac{\delta - z_{1-\alpha}}{\nu^{\frac{1}{2}}}\right), & \text{upper 1-sided} \\ \Phi\left(\frac{-\delta - z_{1-\alpha}}{\nu^{\frac{1}{2}}}\right), & \text{lower 1-sided} \\ \Phi\left(\frac{\delta - z_{1-\frac{\alpha}{2}}}{\nu^{\frac{1}{2}}}\right) + \Phi\left(\frac{-\delta - z_{1-\frac{\alpha}{2}}}{\nu^{\frac{1}{2}}}\right), & \text{2-sided} \end{cases}
$$

Because the test is biased, the achieved significance level may differ from the nominal significance level. The actual alpha is computed in the same way as the power except with the correlation $\rho_{Y|(X_1,X_{-1})}$ replaced by the null correlation ρ_0.

t Test for Pearson Correlation (TEST=PEARSON DIST=T)

The 2-sided case is identical to multiple regression with an intercept and $p_1 = 1$, which is discussed in the "Analyses in the MULTREG Statement" section on page 3500.

Let $p^\star$ denote the number of variables partialled out. For the 1-sided cases, the test statistic is

$$
t = (N-2-p^\star)^{\frac{1}{2}} \frac{R_{YX_1|X_{-1}}}{\left(1 - R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}}
$$

which is assumed to have a null distribution of $t(N-2-p^\star)$.

If the X and Y variables are assumed to have a joint multivariate normal distribution, then the exact power is given by the following formula:

$$
\text{power} \quad = \quad \begin{cases} P\left[(N-2-p^\star)^{\frac{1}{2}} \frac{R_{YX_1|X_{-1}}}{\left(1-R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}} \ge t_{1-\alpha}(N-2-p^\star)\right], & \text{upper 1-sided} \\ P\left[(N-2-p^\star)^{\frac{1}{2}} \frac{R_{YX_1|X_{-1}}}{\left(1-R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}} \le t_{\alpha}(N-2-p^\star)\right], & \text{lower 1-sided} \end{cases}
$$

$$= \begin{cases} P\left[R_{Y|(X_1,X_{-1})} \geq \frac{t_{1-\alpha}(N-2-p^\star)}{\left(t^2_{1-\alpha}(N-2-p^\star)+\frac{1}{N-2-p^\star}\right)^{\frac{1}{2}}}\right], & \text{upper 1-sided} \\ P\left[R_{Y|(X_1,X_{-1})} \leq \frac{t_{\alpha}(N-2-p^\star)}{\left(t^2_{\alpha}(N-2-p^\star)+\frac{1}{N-2-p^\star}\right)^{\frac{1}{2}}}\right], & \text{lower 1-sided} \end{cases}$$

The distribution of $R_{Y|(X_1,X_{-1})}$ (given the underlying true correlation $\rho_{Y|(X_1,X_{-1})}$) is given in Chapter 32 of Johnson, Kotz, and Balakrishnan (1995).

If the X variables are assumed to have fixed values, then the exact power is given by the noncentral t distribution $t(N-2-p^\star,\delta)$, where the noncentrality is

$$\delta = N^{\frac{1}{2}} \frac{\rho_{YX_1|X_{-1}}}{\left(1-\rho^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}}$$

The power is

$$\text{power} = \begin{cases} P\left(t(N-2-p^\star,\delta) \geq t_{1-\alpha}(N-2-p^\star)\right), & \text{upper 1-sided} \\ P\left(t(N-2-p^\star,\delta) \leq t_{\alpha}(N-2-p^\star)\right), & \text{lower 1-sided} \end{cases}$$

Analyses in the ONESAMPLEFREQ Statement

Exact Test of a Binomial Proportion (TEST=EXACT)

Let X be distributed as $\text{Bin}(N,p)$. The hypotheses for the test of the proportion p are as follows:

$$\begin{aligned} H_0 &: \ p = p_0 \\ H_1 &: \ \begin{cases} p \neq p_0, & \text{2-sided} \\ p > p_0, & \text{upper 1-sided} \\ p < p_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

The exact test assumes binomially distributed data and requires $N \geq 1$ and $0 < p_0 < 1$. The test statistic is

$$X = \text{number of successes} \sim \text{Bin}(N,p)$$

The significance probability α is split symmetrically for 2-sided tests, in the sense that each tail is filled with as much as possible up to $\alpha/2$.

Exact power computations are based on the binomial distribution and computing formulas such as the following from Johnson and Kotz (1970, equation 3.20):

$$\begin{aligned} P(X \geq C|N,p) &= P\left(F_{\nu_1,\nu_2} \leq \frac{\nu_2 p}{\nu_1(1-p)}\right) \\ &\quad \text{where } \nu_1 = 2C \text{ and } \nu_2 = 2(N-C+1) \end{aligned}$$

Let C_L and C_U denote lower and upper critical values, respectively. Let α_a denote the achieved (actual) significance level, which for 2-sided tests is the sum of the favorable major tail (α_M) and the opposite minor tail (α_m).

For the upper 1-sided case,

$$\begin{aligned} C_U &= \min\{C : P(X \geq C|p_0) \leq \alpha\} \\ \text{Reject } H_0 &\quad \text{if} \quad X \geq C_U \\ \alpha_a &= P(X \geq C_U|p_0) \\ \text{power} &= P(X \geq C_U|p) \end{aligned}$$

For the lower 1-sided case,

$$\begin{aligned} C_L &= \max\{C : P(X \leq C|p_0) \leq \alpha\} \\ \text{Reject } H_0 &\quad \text{if} \quad X \leq C_L \\ \alpha_a &= P(X \leq C_L|p_0) \\ \text{power} &= P(X \leq C_L|p) \end{aligned}$$

For the 2-sided case,

$$\begin{aligned} C_L &= \max\{C : P(X \leq C|p_0) \leq \frac{\alpha}{2}\} \\ C_U &= \min\{C : P(X \geq C|p_0) \leq \frac{\alpha}{2}\} \\ \text{Reject } H_0 &\quad \text{if} \quad X \leq C_L \text{ or } X \geq C_U \\ \alpha_a &= P(X \leq C_L \text{ or } X \geq C_U|p_0) \\ \text{power} &= P(X \leq C_L \text{ or } X \geq C_U|p) \end{aligned}$$

z Test for Binomial Proportion (TEST=Z)

For the normal approximation test, the test statistic is

$$Z(X) = \frac{X - Np_0}{[Np_0(1-p_0)]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, the computations are the same as described in the "Exact Test of a Binomial Proportion (TEST=EXACT)" section on page 3504 except for the definitions of the critical values.

For the upper 1-sided case,

$$C_U = \min\{C : Z(C) \geq z_{1-\alpha}\}$$

For the lower 1-sided case,

$$C_L = \max\{C : Z(C) \leq z_{\alpha}\}$$

For the 2-sided case,

$$\begin{aligned} C_L &= \max\{C : Z(C) \le z_{\frac{\alpha}{2}}\} \\ C_U &= \min\{C : Z(C) \ge z_{1-\frac{\alpha}{2}}\} \end{aligned}$$

For the METHOD=NORMAL option, the test statistic $Z(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-p_0)}{[p_0(1-p_0)]^{\frac{1}{2}}}, \frac{p(1-p)}{p_0(1-p_0)} \right)$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\frac{z_\alpha + \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}} \right), & \text{upper 1-sided} \\ \Phi\left(\frac{z_\alpha - \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}} \right), & \text{lower 1-sided} \\ \Phi\left(\frac{z_{\frac{\alpha}{2}} + \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}} \right) + \Phi\left(\frac{z_{\frac{\alpha}{2}} - \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}} \right), & \text{2-sided} \end{cases}$$

The approximate sample size is computed in closed form for the 1-sided cases by inverting the power equation,

$$N = \left(\frac{z_{\text{power}}\sqrt{p(1-p)} + z_{1-\alpha}\sqrt{p_0(1-p_0}}{p-p_0} \right)^2$$

and by numerical inversion for the 2-sided case.

z Test for Binomial Proportion with Continuity Adjustment (TEST=ADJZ)

For the normal approximation test with continuity adjustment, the test statistic is (Pagano and Gauvreau 1993 p. 295):

$$Z_c(X) = \frac{X - Np_0 + 0.5(1_{\{X<Np_0\}}) - 0.5(1_{\{X>Np_0\}})}{[Np_0(1-p_0)]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, the computations are the same as described in the "Exact Test of a Binomial Proportion (TEST=EXACT)" section on page 3504 except for the definitions of the critical values.

For the upper 1-sided case,

$$C_U = \min\{C : Z_c(C) \ge z_{1-\alpha}\}$$

For the lower 1-sided case,

$$C_L = \max\{C : Z_c(C) \le z_\alpha\}$$

For the 2-sided case,

$$\begin{aligned} C_L &= \max\{C : Z_c(C) \le z_{\frac{\alpha}{2}}\} \\ C_U &= \min\{C : Z_c(C) \ge z_{1-\frac{\alpha}{2}}\} \end{aligned}$$

For the METHOD=NORMAL option, the test statistic $Z_c(X)$ is assumed to have the normal distribution $N(\mu, \sigma^2)$ where μ and σ^2 are derived as follows.

For convenience of notation, define

$$k = \frac{1}{2\sqrt{Np_0(1-p_0)}}$$

Then

$$E\left[Z_c(X)\right] = 2kNp - 2kNp_0 + kP(X < Np_0) - kP(X > Np_0)$$

and

$$\begin{aligned} \mathrm{Var}\left[Z_c(X)\right] &= \frac{p(1-p)}{p_0(1-p_0)} + k^2\left[1 - P(X = Np_0)\right] \\ &\quad -k^2\left[P(X < Np_0) - P(X > Np_0)\right]^2 \\ &\quad +4k^2\left[E\left(X1_{\{X<Np_0\}}\right) - E\left(X1_{\{X>Np_0\}}\right)\right] \\ &\quad -4k^2Np\left[P(X < Np_0) - P(X > Np_0)\right] \end{aligned}$$

The probabilities $P(X = Np_0)$, $P(X < Np_0)$, and $P(X > Np_0)$ and the truncated expectations $E\left(X1_{\{X<Np_0\}}\right)$ and $E\left(X1_{\{X>Np_0\}}\right)$ are approximated by assuming the normal-approximate distribution of X, $N(Np, Np(1-p))$. Letting $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively, and defining d as

$$d = \frac{Np_0 - Np}{\left[Np(1-p)\right]^{\frac{1}{2}}}$$

the terms are computed as follows:

$$\begin{aligned} P(X = Np_0) &= 0 \\ P(X < Np_0) &= \Phi(d) \end{aligned}$$

$$\begin{aligned} P(X > Np_0) &= 1 - \Phi(d) \\ E\left(X1_{\{X<Np_0\}}\right) &= Np\Phi(d) - [Np(1-p)]^{\frac{1}{2}}\,\phi(d) \\ E\left(X1_{\{X>Np_0\}}\right) &= Np\,[1-\Phi(d)] + [Np(1-p)]^{\frac{1}{2}}\,\phi(d) \end{aligned}$$

The mean and variance of $Z_c(X)$ are thus approximated by

$$\mu = k\,[2Np - 2Np_0 + 2\Phi(d) - 1]$$

and

$$\sigma^2 = 4k^2\left[Np(1-p) + \Phi(d)\,(1-\Phi(d)) - 2\,(Np(1-p))^{\frac{1}{2}}\,\phi(d)\right]$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\frac{z_\alpha+\mu}{\sigma}\right), & \text{upper 1-sided} \\ \Phi\left(\frac{z_\alpha-\mu}{\sigma}\right), & \text{lower 1-sided} \\ \Phi\left(\frac{z_{\frac{\alpha}{2}}+\mu}{\sigma}\right) + \Phi\left(\frac{z_{\frac{\alpha}{2}}-\mu}{\sigma}\right), & \text{2-sided} \end{cases}$$

Analyses in the ONESAMPLEMEANS Statement

One-sample t Test (TEST=T)

The hypotheses for the one-sample t test are

$$\begin{aligned} H_0 &: \mu = \mu_0 \\ H_1 &: \begin{cases} \mu \neq \mu_0, & \text{2-sided} \\ \mu > \mu_0, & \text{upper 1-sided} \\ \mu < \mu_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

The test assumes normally distributed data and requires $N \geq 2$. The test statistics are

$$\begin{aligned} t &= N^{\frac{1}{2}}\left(\frac{\bar{x}-\mu_0}{s}\right) \quad \sim t(N-1,\delta) \\ t^2 &\sim F(1, N-1, \delta^2) \end{aligned}$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation, and

$$\delta = N^{\frac{1}{2}}\left(\frac{\mu-\mu_0}{\sigma}\right)$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-1), & \text{2-sided} \\ t \geq t_{1-\alpha}(N-1), & \text{upper 1-sided} \\ t \leq t_{\alpha}(N-1), & \text{lower 1-sided} \end{cases}$$

Exact power computations for t tests are discussed in O'Brien and Muller (1993, section 8.2), although not specifically for the one-sample case. The power is based on the noncentral t and F distributions:

$$\text{power} \quad = \quad \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{2-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper 1-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower 1-sided} \end{cases}$$

Solutions for N, α, and δ are obtained by numerically inverting the power equation. Closed-form solutions for other parameters, in terms of δ, are as follows:

$$\begin{aligned} \mu &= \delta \sigma N^{-\frac{1}{2}} + \mu_0 \\ \sigma &= \begin{cases} \delta^{-1} N^{\frac{1}{2}} (\mu - \mu_0), & |\delta| > 0 \\ \text{undefined}, & \text{otherwise} \end{cases} \end{aligned}$$

One-sample t Test with Lognormal Data (TEST=T DIST=LOGNORMAL)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "One-sample t Test (TEST=T)" section on page 3508 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. This is because the transformation of a null arithmetic mean of lognormal data to the normal scale depends on the unknown coefficient of variation, resulting in an ill-defined hypothesis on the log-transformed data. Geometric means transform cleanly and are more natural for lognormal data.

The hypotheses for the one-sample t test with lognormal data are

$$\begin{aligned} H_0 &: \frac{\gamma}{\gamma_0} = 1 \\ H_1 &: \begin{cases} \frac{\gamma}{\gamma_0} \neq 1, & \text{2-sided} \\ \frac{\gamma}{\gamma_0} > 1, & \text{upper 1-sided} \\ \frac{\gamma}{\gamma_0} < 1, & \text{lower 1-sided} \end{cases} \end{aligned}$$

Let $\mu^\star$ and $\sigma^\star$ be the (arithmetic) mean and standard deviation of the normal distribution of the log-transformed data. The hypotheses can be rewritten as follows:

$$\begin{aligned} H_0 &: \mu^\star = \log(\gamma_0) \\ H_1 &: \begin{cases} \mu^\star \neq \log(\gamma_0), & \text{2-sided} \\ \mu^\star > \log(\gamma_0), & \text{upper 1-sided} \\ \mu^\star < \log(\gamma_0), & \text{lower 1-sided} \end{cases} \end{aligned}$$

where $\mu^\star = \log(\gamma)$.

The test assumes lognormally distributed data and requires $N \geq 2$.

The power is

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{2-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper 1-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower 1-sided} \end{cases}$$

where

$$\begin{aligned} \delta &= N^{\frac{1}{2}} \left(\frac{\mu^\star - \log(\gamma_0)}{\sigma^\star} \right) \\ \sigma^\star &= \left[\log(\text{CV}^2 + 1) \right]^{\frac{1}{2}} \end{aligned}$$

Equivalence Test for Mean of Normal Data (TEST=EQUIV DIST=NORMAL)

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \mu < \theta_L \quad \text{or} \quad \mu > \theta_U \\ H_1 &: \theta_L \leq \mu \leq \theta_U \end{aligned}$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \geq 2$. Phillips (1990) derives an expression for the exact power assuming a two-sample balanced design; the results are easily adapted to a one-sample design:

$$\begin{aligned} \text{power} = \; & Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\mu - \theta_U}{\sigma N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))} \right) - \\ & Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\mu - \theta_L}{\sigma N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))} \right) \end{aligned}$$

where $Q_{\cdot}(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

Equivalence Test for Mean of Lognormal Data (TEST=EQUIV DIST=LOGNORMAL)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "Equivalence Test for Mean of Normal Data (TEST=EQUIV DIST=NORMAL)" section on page 3510 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. This is because the transformation of an arithmetic mean of lognormal data to the normal scale depends on the unknown coefficient of variation, resulting in an ill-defined hypothesis on the log-transformed data. Geometric means transform cleanly and are more natural for lognormal data.

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \gamma \le \theta_L \quad \text{or} \quad \gamma \ge \theta_U \\ H_1 &: \theta_L < \gamma < \theta_U \end{aligned}$$

$$\text{where} \quad 0 < \theta_L < \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \ge 2$. Diletti, Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to a one-sample design:

$$\begin{aligned} \text{power} = \; & Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\log(\gamma) - \log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) - \\ & Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\log(\gamma) - \log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) \end{aligned}$$

where

$$\sigma^\star = \left[\log(\mathrm{CV}^2 + 1)\right]^{\frac{1}{2}}$$

is the standard deviation of the log-transformed data, and $Q_\cdot(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

Confidence Interval for Mean (CI=T)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{array}{ll}
\left[\bar{x} - t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}, \quad \bar{x} + t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}\right], & \text{2-sided} \\
\left[\bar{x} - t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}, \quad \infty\right), & \text{upper 1-sided} \\
\left(-\infty, \quad \bar{x} + t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}\right], & \text{lower 1-sided}
\end{array}$$

where $\bar{x}$ is the sample mean and s is the sample standard deviation. The "half-width" is defined as the distance from the point estimate $\bar{x}$ to a finite endpoint,

$$\text{half-width} = \left\{ \begin{array}{ll} t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}, & \text{2-sided} \\ t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}, & \text{1-sided} \end{array} \right.$$

A "valid" conference interval captures the true mean. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given by Beal (1989):

$$\begin{array}{rcl}
\Pr(\text{half-width} \le h) & = & \left\{ \begin{array}{ll} P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2 (t^2_{1-\frac{\alpha}{2}}(N-1))}\right), & \text{2-sided} \\ P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2 (t^2_{1-\alpha}(N-1))}\right), & \text{1-sided} \end{array} \right. \\
\begin{array}{r}\Pr(\text{half-width} \le h \mid \\ \text{validity})\end{array} & = & \left\{ \begin{array}{ll} \left(\frac{1}{1-\alpha}\right) 2 \left[Q_{N-1}\left((t_{1-\frac{\alpha}{2}}(N-1)), 0;\right.\right. & \\ \quad \left. 0, b_1) - Q_{N-1}(0,0;0,b_1)\right], & \text{2-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-1}\left((t_{1-\alpha}(N-1)), 0; 0, b_1\right), & \text{1-sided} \end{array} \right.
\end{array}$$

where

$$\begin{array}{rcl}
b_1 & = & \frac{h(N-1)^{\frac{1}{2}}}{\sigma(t_{1-\frac{\alpha}{c}}(N-1))N^{-\frac{1}{2}}} \\
c & = & \text{number of sides}
\end{array}$$

and $Q_{\cdot}(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

A "quality" confidence interval is both sufficiently narrow (half-width $\le h$) and valid:

$$\begin{array}{rcl}
\Pr(\text{quality}) & = & \Pr(\text{half-width} \le h \text{ and validity}) \\
& = & \Pr(\text{half-width} \le h \mid \text{validity})(1-\alpha)
\end{array}$$

Analyses in the ONEWAYANOVA Statement

One-Degree-of-Freedom Contrast (TEST=CONTRAST)

The hypotheses are

$$\begin{aligned} H_0 &: c_1\mu_1 + \cdots + c_G\mu_G = c_0 \\ H_1 &: \begin{cases} c_1\mu_1 + \cdots + c_G\mu_G \neq c_0, & \text{2-sided} \\ c_1\mu_1 + \cdots + c_G\mu_G > c_0, & \text{upper 1-sided} \\ c_1\mu_1 + \cdots + c_G\mu_G < c_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

where G is the number of groups, $\{c_1, \ldots, c_G\}$ are the contrast coefficients, and c_0 is the null contrast value.

The test is the usual F test for a contrast in one-way ANOVA. It assumes normal data with common group variances and requires $N \geq G + 1$ and $n_i \geq 1$.

O'Brien and Muller (1993, section 8.2.3.2) give the exact power as

$$\text{power} = \begin{cases} P\left(F(1, N-G, \delta^2) \geq F_{1-\alpha}(1, N-G)\right), & \text{2-sided} \\ P\left(t(N-G, \delta) \geq t_{1-\alpha}(N-G)\right), & \text{upper 1-sided} \\ P\left(t(N-G, \delta) \leq t_{\alpha}(N-G)\right), & \text{lower 1-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}} \left(\frac{\sum_{i=1}^{G} c_i\mu_i - c_0}{\sigma \left(\sum_{i=1}^{G} \frac{c_i^2}{w_i} \right)^{\frac{1}{2}}} \right)$$

Overall F Test (TEST=OVERALL)

The hypotheses are

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 = \cdots = \mu_G \\ H_1 &: \mu_i \neq \mu_j \text{ for some } i,j \end{aligned}$$

where G is the number of groups.

The test is the usual overall F test for equality of means in one-way ANOVA. It assumes normal data with common group variances and requires $N \geq G + 1$ and $n_i \geq 1$.

O'Brien and Muller (1993, section 8.2.3.1) give the exact power as

$$\text{power} = P\left(F(G-1, N-G, \lambda) \geq F_{1-\alpha}(G-1, N-G)\right)$$

where the noncentrality is

$$\lambda = N \left(\frac{\sum_{i=1}^{G} w_i(\mu_i - \bar{\mu})^2}{\sigma^2} \right)$$

and

$$\bar{\mu} = \sum_{i=1}^{G} w_i \mu_i$$

Analyses in the PAIREDFREQ Statement

Overview of Conditional McNemar tests

Notation:

		Case		
		Failure	Success	
Control	Failure	n_{00}	n_{01}	$n_{0\cdot}$
	Success	n_{10}	n_{11}	$n_{1\cdot}$
		$n_{\cdot 0}$	$n_{\cdot 1}$	N

$$\begin{aligned}
n_{00} &= \#\{\text{control=failure, case=failure }\} \\
n_{01} &= \#\{\text{control=failure, case=success }\} \\
n_{10} &= \#\{\text{control=success, case=failure }\} \\
n_{11} &= \#\{\text{control=success, case=success }\} \\
N &= n_{00} + n_{01} + n_{10} + n_{11} \\
n_D &= n_{01} + n_{10} \equiv \text{ \# discordant pairs} \\
\hat{\pi}_{ij} &= \frac{n_{ij}}{N} \\
\pi_{ij} &= \text{theoretical population value of } \hat{\pi}_{ij} \\
\pi_{1\cdot} &= \pi_{10} + \pi_{11} \\
\pi_{\cdot 1} &= \pi_{01} + \pi_{11} \\
OR &= \text{“odds ratio”} = \frac{\pi_{01}}{\pi_{10}} \\
OR_0 &= \text{null odds ratio}
\end{aligned}$$

All McNemar tests covered in PROC POWER are *conditional*, meaning that n_D is assumed fixed at its observed value.

For the usual $OR_0 = 0$, the hypotheses are

$$\begin{aligned} H_0 &: \pi_{\cdot 1} = \pi_{1 \cdot} \\ H_1 &: \begin{cases} \pi_{\cdot 1} \neq \pi_{1 \cdot}, & \text{2-sided} \\ \pi_{\cdot 1} > \pi_{1 \cdot}, & \text{upper 1-sided} \\ \pi_{\cdot 1} < \pi_{1 \cdot}, & \text{lower 1-sided} \end{cases} \end{aligned}$$

The test statistic for both tests covered in PROC POWER (DIST=EXACT_COND and DIST=NORMAL) is the McNemar statistic Q_M, which has the following form when $OR_0 = 0$:

$$Q_{M_0} = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}}$$

For the conditional McNemar tests, this is equivalent to the square of the $Z(X)$ statistic for the test of a single proportion (normal approximation to binomial), where the proportion is $\frac{\pi_{01}}{\pi_{01}+\pi_{10}}$, the null is 0.5, and "N" is n_D (see, e.g., Schork and Williams 1980):

$$\begin{aligned} Z(X) &= \frac{n_{01} - n_D(0.5)}{[n_D 0.5(1-0.5)]^{\frac{1}{2}}} \quad \dot{\sim} \mathrm{N} \left(\frac{n_D^{\frac{1}{2}}(\frac{\pi_{01}}{\pi_{01}+\pi_{10}} - 0.5)}{[0.5(1-0.5)]^{\frac{1}{2}}}, \frac{\frac{\pi_{01}}{\pi_{01}+\pi_{10}}\left(1 - \frac{\pi_{01}}{\pi_{01}+\pi_{10}}\right)}{0.5(1-0.5)} \right) \\ &= \frac{n_{01} - (n_{01}+n_{10})(0.5)}{[(n_{01}+n_{10})0.5(1-0.5)]^{\frac{1}{2}}} \\ &= \frac{n_{01} - n_{10}}{[n_{01}+n_{10}]^{\frac{1}{2}}} \\ &= \sqrt{Q_{M_0}} \end{aligned}$$

This can be generalized to a custom null for $\frac{\pi_{01}}{\pi_{01}+\pi_{10}}$, which is equivalent to specifying a custom odds ratio:

$$\left[\frac{\pi_{01}}{\pi_{01}+\pi_{10}} \right]_0 \equiv \left[\frac{1}{1 + \frac{1}{\frac{\pi_{01}}{\pi_{10}}}} \right]_0 \equiv \frac{1}{1 + \frac{1}{OR_0}}$$

So, a conditional McNemar test (asymptotic or exact) with a custom null is equivalent to the test of a single proportion $p_1 \equiv \frac{\pi_{01}}{\pi_{01}+\pi_{10}}$ with a null value $p_0 \equiv \frac{1}{1+\frac{1}{OR_0}}$, with a sample size of n_D:

$$\begin{aligned} H_0 &: p_1 = p_0 \\ H_1 &: \begin{cases} p_1 \neq p_0, & \text{2-sided} \\ p_1 > p_0, & \text{1-sided U} \\ p_1 < p_0, & \text{1-sided L} \end{cases} \end{aligned}$$

which is equivalent to

$$\begin{aligned} H_0 &: \quad OR = OR_0 \\ H_1 &: \quad \begin{cases} OR \neq OR_0, & \text{2-sided} \\ OR > OR_0, & \text{1-sided U} \\ OR < OR_0, & \text{1-sided L} \end{cases} \end{aligned}$$

The general form of the test statistic is thus

$$Q_M = \frac{(n_{01} - n_D p_0)^2}{n_D p_0 (1 - p_0)}$$

The two most common conditional McNemar tests assume either the exact conditional distribution of Q_M (covered by the DIST=EXACT_COND analysis) or a standard normal distribution for Q_M (covered by the DIST=NORMAL analysis).

McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)

For DIST=EXACT_COND, the power is calculated assuming that the test is conducted using the exact conditional distribution of Q_M (conditional on n_D). The power is calculated by first computing the conditional power for each possible n_D. The unconditional power is computed as a weighted average over all possible outcomes of n_D:

$$\text{power} = \sum_{n_D=0}^{N} P(n_D) P(\text{Reject } p_1 = p_0 | n_D)$$

where $n_D \sim \text{Bin}(\pi_{01} + \pi_{10}, N)$, and $P(\text{Reject } p_1 = p_0 | n_D)$ is calculated using the exact method in the "Exact Test of a Binomial Proportion (TEST=EXACT)" section on page 3504.

The achieved significance level, reported as Actual Alpha in the analysis, is computed in the same way except using the actual alpha of the one-sample test in place of its power:

$$\text{actual alpha} = \sum_{n_D=0}^{N} P(n_D) \alpha^{\star}(p_1, p_0 | n_D)$$

where $\alpha^{\star}(p_1, p_0 | n_D)$ is the actual alpha calculated using the exact method in the "Exact Test of a Binomial Proportion (TEST=EXACT)" section on page 3504 with proportion p_1, null p_0, and sample size n_D.

McNemar Normal Approximation Test (TEST=MCNEMAR DIST=NORMAL)

For DIST=NORMAL, power is calculated assuming the test is conducted using the normal-approximate distribution of Q_M (conditional on n_D).

For the METHOD=EXACT option, the power is calculated in the same way as described in the "McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)" section on page 3516, except that $P(\text{Reject } p_1 = p_0 | n_D)$ is calculated using the exact method in the "z Test for Binomial Proportion (TEST=Z)" section on page 3505. The achieved significance level is calculated in the same way as described at the end of the "McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)" section on page 3516.

For the METHOD=MIETTINEN option, approximate sample size for the 1-sided cases is computed according to equation (5.6) in Miettinen (1968):

$$N = \frac{\left\{ z_{1-\alpha}(p_{10} + p_{01}) + z_{power} \left[(p_{10} + p_{01})^2 - \frac{1}{4}(p_{01} - p_{10})^2(3 + p_{10} + p_{01}) \right]^{\frac{1}{2}} \right\}^2}{(p_{10} + p_{01})(p_{01} - p_{10})^2}$$

Approximate power for the 1-sided cases is computed by solving the sample size equation for power, and approximate power for the 2-sided case follows easily by summing the 1-sided powers each at $\alpha/2$:

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}} \right), & \text{upper 1-sided} \\ \Phi\left(\frac{-(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}} \right), & \text{lower 1-sided} \\ \Phi\left(\frac{(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}} \right) + & \\ \quad \Phi\left(\frac{-(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}} \right), & \text{2-sided} \end{cases}$$

The 2-sided solution for N is obtained by numerically inverting the power equation.

In general, compared to METHOD=CONNOR, the METHOD=MIETTINEN approximation tends to be slightly more accurate but may be slightly anticonservative in the sense of underestimating sample size and overestimating power (Lachin 1992, p. 1250).

For the METHOD=CONNOR option, approximate sample size for the 1-sided cases is computed according to equation (3) in Connor (1987):

$$N = \frac{\left\{ z_{1-\alpha}(p_{10} + p_{01})^{\frac{1}{2}} + z_{power} \left[p_{10} + p_{01} - (p_{01} - p_{10})^2 \right]^{\frac{1}{2}} \right\}^2}{(p_{01} - p_{10})^2}$$

Approximate power for the 1-sided cases is computed by solving the sample size equation for power, and approximate power for the 2-sided case follows easily by summing the 1-sided powers each at $\alpha/2$:

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})^{\frac{1}{2}}}{[p_{10}+p_{01}-(p_{01}-p_{10})^2]^{\frac{1}{2}}} \right), & \text{upper 1-sided} \\ \Phi\left(\frac{-(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})^{\frac{1}{2}}}{[p_{10}+p_{01}-(p_{01}-p_{10})^2]^{\frac{1}{2}}} \right), & \text{lower 1-sided} \\ \Phi\left(\frac{(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})^{\frac{1}{2}}}{[p_{10}+p_{01}-(p_{01}-p_{10})^2]^{\frac{1}{2}}} \right) + & \\ \quad \Phi\left(\frac{-(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})^{\frac{1}{2}}}{[p_{10}+p_{01}-(p_{01}-p_{10})^2]^{\frac{1}{2}}} \right), & \text{2-sided} \end{cases}$$

The 2-sided solution for N is obtained by numerically inverting the power equation.

In general, compared to METHOD=MIETTINEN, the METHOD=CONNOR approximation tends to be slightly less accurate but slightly conservative in the sense of overestimating sample size and underestimating power (Lachin 1992, p. 1250).

Analyses in the PAIREDMEANS Statement

Paired t Test (TEST=DIFF)

The hypotheses for the paired t test are

$$\begin{aligned} H_0 &: \mu_{\text{diff}} = \mu_0 \\ H_1 &: \begin{cases} \mu_{\text{diff}} \neq \mu_0, & \text{2-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper 1-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

The test assumes normally distributed data and requires $N \geq 2$. The test statistics are

$$\begin{aligned} t &= N^{\frac{1}{2}} \left(\frac{\bar{d} - \mu_0}{s_d} \right) \quad \sim t(N-1, \delta) \\ t^2 &\sim F(1, N-1, \delta^2) \end{aligned}$$

where $\bar{d}$ and s_d are the sample mean and standard deviation of the differences and

$$\delta = N^{\frac{1}{2}} \left(\frac{\mu_{\text{diff}} - \mu_0}{\sigma_{\text{diff}}} \right)$$

and

$$\sigma_{\text{diff}} = \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2 \right)^{\frac{1}{2}}$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-1), & \text{2-sided} \\ t \geq t_{1-\alpha}(N-1), & \text{upper 1-sided} \\ t \leq t_{\alpha}(N-1), & \text{lower 1-sided} \end{cases}$$

Exact power computations for t tests are given in O'Brien and Muller (1993, section 8.2.2):

$$\text{power} \quad = \quad \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{2-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper 1-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower 1-sided} \end{cases}$$

Paired t Test for Mean Ratio with Lognormal Data (TEST=RATIO)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "Paired t Test (TEST=DIFF)" section on page 3518 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the paired t test with lognormal pairs $\{Y_1, Y_2\}$ are

$$\begin{aligned} H_0 \quad &: \quad \frac{\gamma_2}{\gamma_1} = \gamma_0 \\ H_1 \quad &: \quad \begin{cases} \frac{\gamma_2}{\gamma_1} \neq \gamma_0, & \text{2-sided} \\ \frac{\gamma_2}{\gamma_1} > \gamma_0, & \text{upper 1-sided} \\ \frac{\gamma_2}{\gamma_1} < \gamma_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

Let $\mu_1^\star$, $\mu_2^\star$, $\sigma_1^\star$, $\sigma_2^\star$, and $\rho^\star$ be the (arithmetic) means, standard deviations, and correlation of the bivariate normal distribution of the log-transformed data $\{\log Y_1, \log Y_2\}$. The hypotheses can be rewritten as follows:

$$\begin{aligned} H_0 \quad &: \quad \mu_2^\star - \mu_1^\star = \log(\gamma_0) \\ H_1 \quad &: \quad \begin{cases} \mu_2^\star - \mu_1^\star \neq \log(\gamma_0), & \text{2-sided} \\ \mu_2^\star - \mu_1^\star > \log(\gamma_0), & \text{upper 1-sided} \\ \mu_2^\star - \mu_1^\star < \log(\gamma_0), & \text{lower 1-sided} \end{cases} \end{aligned}$$

where

$$\begin{aligned} \mu_1^\star &= \log \gamma_1 \\ \mu_2^\star &= \log \gamma_2 \\ \sigma_1^\star &= \left[\log(\text{CV}_1^2 + 1)\right]^{\frac{1}{2}} \end{aligned}$$

$$\begin{aligned}\sigma_2^\star &= \left[\log(\mathrm{CV}_2^2+1)\right]^{\frac{1}{2}} \\ \rho^\star &= \frac{\log\left\{\rho\mathrm{CV}_1\mathrm{CV}_2+1\right\}}{\sigma_1^\star\sigma_2^\star}\end{aligned}$$

and CV_1, CV_2, and ρ are the coefficients of variation and the correlation of the original untransformed pairs $\{Y_1, Y_2\}$. The conversion from ρ to $\rho^\star$ is shown in Jones and Miller (1966).

The test assumes lognormally distributed data and requires $N \geq 2$. The power is

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{2-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper 1-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower 1-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}}\left(\frac{\mu_1^\star - \mu_2^\star - \log(\gamma_0)}{\sigma^\star}\right)$$

and

$$\sigma^\star = \left(\sigma_1^{\star 2} + \sigma_2^{\star 2} - 2\rho^\star\sigma_1^\star\sigma_2^\star\right)^{\frac{1}{2}}$$

Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \mu_{\text{diff}} < \theta_L \quad \text{or} \quad \mu_{\text{diff}} > \theta_U \\ H_1 &: \theta_L \leq \mu_{\text{diff}} \leq \theta_U \end{aligned}$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \geq 2$. Phillips (1990) derives an expression for the exact power assuming a two-sample balanced design; the results are easily adapted to a paired design:

$$\begin{aligned}\text{power} = \; & Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\mu_{\text{diff}}-\theta_U}{\sigma_{\text{diff}}N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U-\theta_L)}{2\sigma_{\text{diff}}N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) - \\ & Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\mu_{\text{diff}}-\theta_L}{\sigma_{\text{diff}}N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U-\theta_L)}{2\sigma_{\text{diff}}N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right)\end{aligned}$$

where

$$\sigma_{\text{diff}} = \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2\right)^{\frac{1}{2}}$$

and $Q_{\cdot}(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data (TEST=EQUIV_RATIO)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)" section on page 3520 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \frac{\gamma_T}{\gamma_R} \le \theta_L \quad \text{or} \quad \frac{\gamma_T}{\gamma_R} \ge \theta_U \\ H_1 &: \theta_L < \frac{\gamma_T}{\gamma_R} < \theta_U \\ \text{where} & \quad 0 < \theta_L < \theta_U \end{aligned}$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \ge 2$. Diletti, Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to a paired design:

$$\begin{aligned} \text{power} = \; & Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))} \right) - \\ & Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))} \right) \end{aligned}$$

where $\sigma^\star$ is the standard deviation of the differences between the log-transformed pairs (in other words, the standard deviation of $\log(Y_T) - \log(Y_R)$, where Y_T and Y_R are observations from the treatment and reference, respectively), computed as

$$\begin{aligned} \sigma^\star &= \left(\sigma_R^{\star 2} + \sigma_T^{\star 2} - 2\rho^\star \sigma_R^\star \sigma_T^\star \right)^{\frac{1}{2}} \\ \sigma_R^\star &= \left[\log(\text{CV}_R^2 + 1) \right]^{\frac{1}{2}} \\ \sigma_T^\star &= \left[\log(\text{CV}_T^2 + 1) \right]^{\frac{1}{2}} \\ \rho^\star &= \frac{\log\left\{ \rho \text{CV}_R \text{CV}_T + 1 \right\}}{\sigma_R^\star \sigma_T^\star} \end{aligned}$$

where CV_R, CV_T, and ρ are the coefficients of variation and the correlation of the original untransformed pairs $\{Y_T, Y_R\}$, and $Q_\cdot(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function. The conversion from ρ to $\rho^\star$ is shown in Jones and Miller (1966), and Owen's Q function is defined in the "Common Notation" section on page 3498.

Confidence Interval for Mean Difference (CI=DIFF)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{array}{ll}
\left[\bar{d} - t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}, \quad \bar{d} + t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}\right], & \text{2-sided} \\
\left[\bar{d} - t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}, \quad \infty\right), & \text{upper 1-sided} \\
\left(-\infty, \quad \bar{d} + t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}\right], & \text{lower 1-sided}
\end{array}$$

where $\bar{d}$ and s_d are the sample mean and standard deviation of the differences. The "half-width" is defined as the distance from the point estimate $\bar{d}$ to a finite endpoint,

$$\text{half-width} = \begin{cases} t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}, & \text{2-sided} \\ t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}, & \text{1-sided} \end{cases}$$

A "valid" conference interval captures the true mean difference. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given by Beal (1989):

$$\begin{aligned}
\text{Pr(half-width} \le h) &= \begin{cases} P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2_{\text{diff}}(t^2_{1-\frac{\alpha}{2}}(N-1))}\right), & \text{2-sided} \\ P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2_{\text{diff}}(t^2_{1-\alpha}(N-1))}\right), & \text{1-sided} \end{cases} \\
\begin{array}{r}\text{Pr(half-width} \le h \mid \\ \text{validity)}\end{array} &= \begin{cases} \left(\frac{1}{1-\alpha}\right) 2 \left[Q_{N-1}\left((t_{1-\frac{\alpha}{2}}(N-1)), 0; \right.\right. & \\ \quad \left.0, b_1) - Q_{N-1}(0,0;0,b_1)\right], & \text{2-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-1}\left((t_{1-\alpha}(N-1)), 0; 0, b_1\right), & \text{1-sided} \end{cases}
\end{aligned}$$

where

$$\begin{aligned}
\sigma_{\text{diff}} &= \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2\right)^{\frac{1}{2}} \\
b_1 &= \frac{h(N-1)^{\frac{1}{2}}}{\sigma_{\text{diff}}(t_{1-\frac{\alpha}{c}}(N-1))N^{-\frac{1}{2}}} \\
c &= \text{number of sides}
\end{aligned}$$

and $Q_\cdot(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

A "quality" confidence interval is both sufficiently narrow (half-width $\le h$) and valid:

$$\begin{aligned}
\text{Pr(quality)} &= \text{Pr(half-width} \le h \text{ and validity)} \\
&= \text{Pr(half-width} \le h \mid \text{validity)}(1-\alpha)
\end{aligned}$$

Analyses in the TWOSAMPLEFREQ Statement

Overview of the 2×2 Table

Notation:

		Group 2		
		Failure	Success	
Group 1	Failure	x_1	x_2	m
	Success	$n_1 - x_1$	$n_2 - x_2$	$N - m$
		n_1	n_2	N

$$\begin{aligned}
x_1 &= \text{\# successes in group 1} \\
x_2 &= \text{\# successes in group 2} \\
m &= x_1 + x_2 = \text{ total \# successes} \\
\hat{p}_1 &= \frac{x_1}{n_1} \\
\hat{p}_2 &= \frac{x_2}{n_2} \\
\hat{p} &= \frac{m}{N} = w_1\hat{p}_1 + w_2\hat{p}_2
\end{aligned}$$

The hypotheses are

$$\begin{aligned}
H_0 &: \; p_2 - p_1 = p_0 \\
H_1 &: \; \begin{cases} p_2 - p_1 \neq p_0, & \text{2-sided} \\ p_2 - p_1 > p_0, & \text{upper 1-sided} \\ p_2 - p_1 < p_0, & \text{lower 1-sided} \end{cases}
\end{aligned}$$

where p_0 is constrained to be 0 for all but the unconditional Pearson chi-square test.

Internal calculations are performed in terms of p_1, p_2, and p_0. An input set consisting of OR, p_1, and OR_0 is transformed as follows:

$$\begin{aligned}
p_2 &= \frac{(OR)p_1}{1 - p_1 + (OR)p_1} \\
p_{10} &= p_1 \\
p_{20} &= \frac{OR_0 p_{10}}{1 - p_{10} + (OR_0)p_{10}} \\
p_0 &= p_{20} - p_{10}
\end{aligned}$$

An input set consisting of RR, p_1, and RR_0 is transformed as follows:

$$\begin{aligned} p_2 &= (RR)p_1 \\ p_{10} &= p_1 \\ p_{20} &= (RR_0)p_{10} \\ p_0 &= p_{20} - p_{10} \end{aligned}$$

Note that the transformation of either OR_0 or RR_0 to p_0 is not unique. The chosen parameterization fixes the null value p_{10} at the input value of p_1.

Pearson Chi-square Test for Two Proportions (TEST=PCHI)

The usual Pearson chi-square test is unconditional. The test statistic

$$z_P = \frac{\hat{p_2} - \hat{p_1} - p_0}{\left[\hat{p}(1-\hat{p})\left(\frac{1}{n_1}+\frac{1}{n_2}\right)\right]^{\frac{1}{2}}} = [Nw_1w_2]^{\frac{1}{2}} \frac{\hat{p_2} - \hat{p_1} - p_0}{\hat{p}(1-\hat{p})}$$

is assumed to have a null distribution of $N(0,1)$.

Sample size for the 1-sided cases is given by equation (4) in Fleiss, Tytun, and Ury (1980). One-sided power is computed as suggested by Diegert and Diegert (1981) by inverting the sample size formula. Power for the 2-sided case is computed by adding the lower-sided and upper-sided powers each with $\alpha/2$, and sample size for the 2-sided case is obtained by numerically inverting the power formula. A custom null value p_0 for the proportion difference $p_2 - p_1$ is also supported.

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\alpha}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}} \right), & \text{upper 1-sided} \\ \Phi\left(\frac{-(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\alpha}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}} \right), & \text{lower 1-sided} \\ \Phi\left(\frac{(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}} \right) + & \\ \quad \Phi\left(\frac{-(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}} \right), & \text{2-sided} \end{cases}$$

For the 1-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \frac{\left[z_{1-\alpha} \left\{(w_1p_1 + w_2p_2)(1-w_1p_1-w_2p_2)\right\}^{\frac{1}{2}} + z_{\text{power}} \left\{w_2p_1(1-p_1) + w_1p_2(1-p_2)\right\}^{\frac{1}{2}} \right]^2}{w_1w_2(p_2-p_1-p_0)^2}$$

For the 2-sided case, the solution for N is obtained by numerically inverting the power equation.

Likelihood Ratio chi-square Test for Two Proportions (TEST=LRCHI)

The usual likelihood ratio chi-square test is unconditional. The test statistic

$$z_{LR} = (-1_{\{p_2<p_1\}})\sqrt{2N\sum_{i=1}^{2}\left[w_i\hat{p}_i\log\left(\frac{\hat{p}_i}{\hat{p}}\right)+w_i(1-\hat{p}_i)\log\left(\frac{1-\hat{p}_i}{1-\hat{p}}\right)\right]}$$

is assumed to have a null distribution of $N(0,1)$ and an alternative distribution of $N(\delta,1)$ where

$$\delta = N^{\frac{1}{2}}(-1_{\{p_2<p_1\}})\sqrt{2\sum_{i=1}^{2}\left[w_ip_i\log\left(\frac{p_i}{w_1p_1+w_2p_2}\right)+w_i(1-p_i)\log\left(\frac{1-p_i}{1-(w_1p_1+w_2p_2)}\right)\right]}$$

The approximate power is

$$\text{power} = \begin{cases} \Phi\left(\delta - z_{1-\alpha}\right), & \text{upper 1-sided} \\ \Phi\left(-\delta - z_{1-\alpha}\right), & \text{lower 1-sided} \\ \Phi\left(\delta - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(-\delta - z_{1-\frac{\alpha}{2}}\right), & \text{2-sided} \end{cases}$$

For the 1-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \left(\frac{z_{\text{power}} + z_{1-\alpha}}{\delta}\right)^2$$

For the 2-sided case, the solution for N is obtained by numerically inverting the power equation.

Fisher's Exact Conditional Test for Two Proportions (Test=FISHER)

Fisher's exact test is conditional on the observed total number of successes m. Power and sample size computations for the METHOD=WALTERS option are based on a test with similar power properties, the continuity-adjusted arcsine test. The test statistic

$$\begin{aligned} z_A \quad = \quad & (4Nw_1w_2)^{\frac{1}{2}}\left[\arcsin\left(\left[\hat{p_2}+\frac{1}{2Nw_2}(1_{\{\hat{p_2}<\hat{p_1}\}}-1_{\{\hat{p_2}>\hat{p_1}\}})\right]^{\frac{1}{2}}\right)\right. \\ & \left. -\arcsin\left(\left[\hat{p_1}+\frac{1}{2Nw_1}(1_{\{\hat{p_1}<\hat{p_2}\}}-1_{\{\hat{p_1}>\hat{p_2}\}})\right]^{\frac{1}{2}}\right)\right] \end{aligned}$$

is assumed to have a null distribution of $N(0,1)$ and an alternative distribution of $N(\delta,1)$ where

$$\begin{aligned} \delta \quad = \quad & (4Nw_1w_2)^{\frac{1}{2}}\left[\arcsin\left(\left[p_2+\frac{1}{2Nw_2}(1_{\{p_2<p_1\}}-1_{\{p_2>p_1\}})\right]^{\frac{1}{2}}\right)\right. \\ & \left. -\arcsin\left(\left[p_1+\frac{1}{2Nw_1}(1_{\{p_1<p_2\}}-1_{\{p_1>p_2\}})\right]^{\frac{1}{2}}\right)\right] \end{aligned}$$

The approximate power for the 1-sided balanced case is given by Walters (1979) and is easily extended to the unbalanced and 2-sided cases:

$$\text{power} = \begin{cases} \Phi\left(\delta - z_{1-\alpha}\right), & \text{upper 1-sided} \\ \Phi\left(-\delta - z_{1-\alpha}\right), & \text{lower 1-sided} \\ \Phi\left(\delta - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(-\delta - z_{1-\frac{\alpha}{2}}\right), & \text{2-sided} \end{cases}$$

Analyses in the TWOSAMPLEMEANS Statement

Two-sample t Test Assuming Equal Variances (TEST=DIFF)

The hypotheses for the two-sample t test are

$$\begin{aligned} H_0 &: \mu_{\text{diff}} = \mu_0 \\ H_1 &: \begin{cases} \mu_{\text{diff}} \neq \mu_0, & \text{2-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper 1-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

The test assumes normally distributed data and common standard deviation per group, and it requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. The test statistics are

$$\begin{aligned} t &= N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}} \left(\frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{s_p}\right) \quad \sim t(N-2, \delta) \\ t^2 &\sim F(1, N-2, \delta^2) \end{aligned}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_p is the pooled standard deviation, and

$$\delta = N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}} \left(\frac{\mu_{\text{diff}} - \mu_0}{\sigma}\right)$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-2), & \text{2-sided} \\ t \geq t_{1-\alpha}(N-2), & \text{upper 1-sided} \\ t \leq t_{\alpha}(N-2), & \text{lower 1-sided} \end{cases}$$

Exact power computations for t tests are given in O'Brien and Muller (1993, section 8.2.1):

$$\text{power} = \begin{cases} P\left(F(1, N-2, \delta^2) \geq F_{1-\alpha}(1, N-2)\right), & \text{2-sided} \\ P\left(t(N-2, \delta) \geq t_{1-\alpha}(N-2)\right), & \text{upper 1-sided} \\ P\left(t(N-2, \delta) \leq t_{\alpha}(N-2)\right), & \text{lower 1-sided} \end{cases}$$

Solutions for N, n_1, n_2, α, and δ are obtained by numerically inverting the power equation. Closed-form solutions for other parameters, in terms of δ, are as follows:

$$\begin{aligned}
\mu_{\text{diff}} &= \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0 \\
\mu_1 &= \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0 - \mu_2 \\
\mu_2 &= \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0 - \mu_1 \\
\sigma &= \begin{cases} \delta^{-1}(Nw_1w_2)^{\frac{1}{2}}(\mu_{\text{diff}} - \mu_0), & |\delta| > 0 \\ \text{undefined}, & \text{otherwise} \end{cases} \\
w_1 &= \begin{cases} \frac{1}{2} \pm \frac{1}{2}\left[1 - \frac{4\delta^2\sigma^2}{N(\mu_{\text{diff}}-\mu_0)^2}\right]^{\frac{1}{2}}, & 0 < |\delta| \le \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}}-\mu_0|}{\sigma} \\ \text{undefined}, & \text{otherwise} \end{cases} \\
w_2 &= \begin{cases} \frac{1}{2} \pm \frac{1}{2}\left[1 - \frac{4\delta^2\sigma^2}{N(\mu_{\text{diff}}-\mu_0)^2}\right]^{\frac{1}{2}}, & 0 < |\delta| \le \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}}-\mu_0|}{\sigma} \\ \text{undefined}, & \text{otherwise} \end{cases}
\end{aligned}$$

Finally, here is a derivation of the solution for w_1:

Solve the δ equation for w_1 (which requires the quadratic formula). Then determine the range of δ given w_1:

$$\begin{aligned}
\min_{w_1}(\delta) &= \begin{cases} 0, & \text{when} \quad w_1 = 0 \quad \text{or} \quad 1, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) \ge 0 \\ \frac{1}{2}N^{\frac{1}{2}}\frac{(\mu_{\text{diff}}-\mu_0)}{\sigma}, & \text{when} \quad w_1 = \frac{1}{2}, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) < 0 \end{cases} \\
\max_{w_1}(\delta) &= \begin{cases} 0, & \text{when} \quad w_1 = 0 \quad \text{or} \quad 1, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) < 0 \\ \frac{1}{2}N^{\frac{1}{2}}\frac{(\mu_{\text{diff}}-\mu_0)}{\sigma}, & \text{when} \quad w_1 = \frac{1}{2}, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) \ge 0 \end{cases}
\end{aligned}$$

This implies

$$|\delta| \le \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}} - \mu_0|}{\sigma}$$

Two-sample Satterthwaite t Test Assuming Unequal Variances (TEST=DIFF_SATT)

The hypotheses for the two-sample Satterthwaite t test are

$$\begin{aligned}
H_0 &: \mu_{\text{diff}} = \mu_0 \\
H_1 &: \begin{cases} \mu_{\text{diff}} \ne \mu_0, & \text{2-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper 1-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower 1-sided} \end{cases}
\end{aligned}$$

The test assumes normally distributed data and requires $N \ge 3$, $n_1 \ge 1$, and $n_2 \ge 1$. The test statistics are

$$\begin{aligned}
t &= \frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^{\frac{1}{2}}} = N^{\frac{1}{2}}\frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{\left[\frac{s_1^2}{w_1} + \frac{s_2^2}{w_2}\right]^{\frac{1}{2}}} \\
F &= t^2
\end{aligned}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_1 and s_2 are the sample standard deviations.

As DiSantostefano and Muller (1995, p. 585) state, the test is based on assuming that under H_0, F is distributed as $F(1,\nu)$, where ν is given by Satterthwaite's approximation (Satterthwaite 1946),

$$\nu = \frac{\left[\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\right]^2}{\frac{\left[\frac{\sigma_1^2}{n_1}\right]^2}{n_1-1} + \frac{\left[\frac{\sigma_2^2}{n_2}\right]^2}{n_2-1}} = \frac{\left[\frac{\sigma_1^2}{w_1} + \frac{\sigma_2^2}{w_2}\right]^2}{\frac{\left[\frac{\sigma_1^2}{w_1}\right]^2}{Nw_1-1} + \frac{\left[\frac{\sigma_2^2}{w_2}\right]^2}{Nw_2-1}}$$

Since ν is unknown, in practice it must be replaced by an estimate

$$\hat{\nu} = \frac{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^2}{\frac{\left[\frac{s_1^2}{n_1}\right]^2}{n_1-1} + \frac{\left[\frac{s_2^2}{n_2}\right]^2}{n_2-1}} = \frac{\left[\frac{s_1^2}{w_1} + \frac{s_2^2}{w_2}\right]^2}{\frac{\left[\frac{s_1^2}{w_1}\right]^2}{Nw_1-1} + \frac{\left[\frac{s_2^2}{w_2}\right]^2}{Nw_2-1}}$$

So the test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} F \geq F_{1-\alpha}(1,\hat{\nu}), & \text{2-sided} \\ t \geq t_{1-\alpha}(\hat{\nu}), & \text{upper 1-sided} \\ t \leq t_{\alpha}(\hat{\nu}), & \text{lower 1-sided} \end{cases}$$

Exact solutions for power for the 2-sided and upper 1-sided cases are given in Moser, Stevens, and Watts (1989). The lower 1-sided case follows easily using symmetry. The equations are as follows:

$$\text{power} \quad = \begin{cases} \int_0^\infty P\left(F(1,N-2,\lambda) > \right. & \\ \qquad \left. h(u)F_{1-\alpha}(1,v(u))|u\right) f(u)\mathrm{d}u, & \text{2-sided} \\ \int_0^\infty P\left(t(N-2,\lambda^{\frac{1}{2}}) > \right. & \\ \qquad \left. [h(u)]^{\frac{1}{2}}\, t_{1-\alpha}(v(u))|u\right) f(u)\mathrm{d}u, & \text{upper 1-sided} \\ \int_0^\infty P\left(t(N-2,\lambda^{\frac{1}{2}}) < \right. & \\ \qquad \left. [h(u)]^{\frac{1}{2}}\, t_{\alpha}(v(u))|u\right) f(u)\mathrm{d}u, & \text{lower 1-sided} \end{cases}$$

where

$$h(u) \quad = \quad \frac{\left(\frac{1}{n_1} + \frac{u}{n_2}\right)(n_1+n_2-2)}{\left[(n_1-1) + (n_2-1)\frac{u\sigma_1^2}{\sigma_2^2}\right]\left(\frac{1}{n_1} + \frac{\sigma_2^2}{\sigma_1^2 n_2}\right)}$$

$$v(u) \quad = \quad \frac{\left(\frac{1}{n_1} + \frac{u}{n_2}\right)^2}{\frac{1}{n_1^2(n_1-1)} + \frac{u^2}{n_2^2(n_2-1)}}$$

$$\lambda \quad = \quad \frac{(\mu_{\text{diff}} - \mu_0)^2}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

$$f(u) = \frac{\Gamma\left(\frac{n_1+n_2-2}{2}\right)}{\Gamma\left(\frac{n_1-1}{2}\right)\Gamma\left(\frac{n_2-1}{2}\right)} \left[\frac{\sigma_1^2(n_2-1)}{\sigma_2^2(n_1-1)}\right]^{\frac{n_2-1}{2}} \cdot$$
$$u^{\frac{n_2-3}{2}} \left[1 + \left(\frac{n_2-1}{n_1-1}\right)\frac{u\sigma_1^2}{\sigma_2^2}\right]^{-\left(\frac{n_1+n_2-2}{2}\right)}$$

The density $f(u)$ is obtained from the fact that

$$\frac{u\sigma_1^2}{\sigma_2^2} \sim F(n_2-1, n_1-1)$$

Two-sample Pooled t Test of Mean Ratio with Lognormal Data (TEST=RATIO)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "Two-sample t Test Assuming Equal Variances (TEST=DIFF)" section on page 3526 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. The test assumes equal coefficients of variation in the two groups.

The hypotheses for the two-sample t test with lognormal data are

$$\begin{aligned} H_0 &: \frac{\gamma_2}{\gamma_1} = \gamma_0 \\ H_1 &: \begin{cases} \frac{\gamma_2}{\gamma_1} \neq \gamma_0, & \text{2-sided} \\ \frac{\gamma_2}{\gamma_1} > \gamma_0, & \text{upper 1-sided} \\ \frac{\gamma_2}{\gamma_1} < \gamma_0, & \text{lower 1-sided} \end{cases} \end{aligned}$$

Let $\mu_1^\star$, $\mu_2^\star$, and $\sigma^\star$ be the (arithmetic) means and common standard deviation of the corresponding normal distributions of the log-transformed data. The hypotheses can be rewritten as follows:

$$\begin{aligned} H_0 &: \mu_2^\star - \mu_1^\star = \log(\gamma_0) \\ H_1 &: \begin{cases} \mu_2^\star - \mu_1^\star \neq \log(\gamma_0), & \text{2-sided} \\ \mu_2^\star - \mu_1^\star > \log(\gamma_0), & \text{upper 1-sided} \\ \mu_2^\star - \mu_1^\star < \log(\gamma_0), & \text{lower 1-sided} \end{cases} \end{aligned}$$

where

$$\begin{aligned} \mu_1^\star &= \log \gamma_1 \\ \mu_2^\star &= \log \gamma_2 \end{aligned}$$

The test assumes lognormally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$.

The power is

$$\text{power} = \begin{cases} P\left(F(1, N-2, \delta^2) \geq F_{1-\alpha}(1, N-2)\right), & \text{2-sided} \\ P\left(t(N-2, \delta) \geq t_{1-\alpha}(N-2)\right), & \text{upper 1-sided} \\ P\left(t(N-2, \delta) \leq t_{\alpha}(N-2)\right), & \text{lower 1-sided} \end{cases}$$

where

$$\begin{aligned} \delta &= N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}} \left(\frac{\mu_2^\star - \mu_1^\star - \log(\gamma_0)}{\sigma^\star} \right) \\ \sigma^\star &= \left[\log(\mathrm{CV}^2 + 1)\right]^{\frac{1}{2}} \end{aligned}$$

Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \mu_{\text{diff}} < \theta_L \quad \text{or} \quad \mu_{\text{diff}} > \theta_U \\ H_1 &: \theta_L \leq \mu_{\text{diff}} \leq \theta_U \end{aligned}$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. Phillips (1990) derives an expression for the exact power assuming a balanced design; the results are easily adapted to an unbalanced design:

$$\begin{aligned} \text{power} = \; & Q_{N-2}\left((-t_{1-\alpha}(N-2)), \frac{\mu_{\text{diff}} - \theta_U}{\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-2)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))} \right) \quad - \\ & Q_{N-2}\left((t_{1-\alpha}(N-2)), \frac{\mu_{\text{diff}} - \theta_L}{\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-2)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))} \right) \end{aligned}$$

where $Q_{\cdot}(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data (TEST=EQUIV_RATIO)

The lognormal case is handled by re-expressing the analysis equivalently as a normality-based test on the log-transformed data, using properties of the lognormal distribution as discussed in Johnson and Kotz (1970, chapter 14). The approaches in the "Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)" section on page 3530 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the equivalence test are

$$\begin{aligned} H_0 &: \frac{\gamma_T}{\gamma_R} \le \theta_L \quad \text{or} \quad \frac{\gamma_T}{\gamma_R} \ge \theta_U \\ H_1 &: \theta_L < \frac{\gamma_T}{\gamma_R} < \theta_U \end{aligned}$$

$$\text{where} \quad 0 < \theta_L < \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \ge 3$, $n_1 \ge 1$, and $n_2 \ge 1$. Diletti, Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to an unbalanced two-sample design:

$$\begin{aligned} \text{power} = {} & Q_{N-2}\left((-t_{1-\alpha}(N-2)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-2)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))} \right) - \\ & Q_{N-2}\left((t_{1-\alpha}(N-2)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; \right. \\ & \left. 0, \frac{(N-2)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))} \right) \end{aligned}$$

where

$$\sigma^\star = \left[\log(\mathrm{CV}^2 + 1)\right]^{\frac{1}{2}}$$

is the (assumed common) standard deviation of the normal distribution of the log-transformed data, and $Q_{\cdot}(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

Confidence Interval for Mean Difference (CI=DIFF)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{array}{ll}
\Big[(\bar{x}_2 - \bar{x}_1) - t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, & \\
\quad (\bar{x}_2 - \bar{x}_1) + t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}} \Big], & \text{2-sided} \\
\Big[(\bar{x}_2 - \bar{x}_1) - t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, \quad \infty \Big), & \text{upper 1-sided} \\
\Big(-\infty, \quad (\bar{x}_2 - \bar{x}_1) + t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}} \Big], & \text{lower 1-sided}
\end{array}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_p is the pooled standard deviation. The "half-width" is defined as the distance from the point estimate $\bar{x}_2 - \bar{x}_1$ to a finite endpoint,

$$\text{half-width} = \begin{cases} t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, & \text{2-sided} \\ t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, & \text{1-sided} \end{cases}$$

A "valid" conference interval captures the true mean. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given by Beal (1989):

$$\begin{aligned}
\Pr(\text{half-width} \le h) &= \begin{cases} P\left(\chi^2(N-2) \le \frac{h^2 N(N-2)(w_1 w_2)}{\sigma^2 (t^2_{1-\frac{\alpha}{2}}(N-2))} \right), & \text{2-sided} \\ P\left(\chi^2(N-2) \le \frac{h^2 N(N-2)(w_1 w_2)}{\sigma^2 (t^2_{1-\alpha}(N-2))} \right), & \text{1-sided} \end{cases} \\
\begin{array}{r}\Pr(\text{half-width} \le h \mid \\ \text{validity})\end{array} &= \begin{cases} \left(\frac{1}{1-\alpha}\right) 2 \Big[Q_{N-2}\left((t_{1-\frac{\alpha}{2}}(N-2)), 0; \right. & \\ \quad \left. 0, b_2 \right) - Q_{N-2}(0,0;0,b_2) \Big], & \text{2-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-2}\left((t_{1-\alpha}(N-2)), 0; 0, b_2 \right), & \text{1-sided} \end{cases}
\end{aligned}$$

where

$$\begin{aligned}
b_2 &= \frac{h(N-2)^{\frac{1}{2}}}{\sigma (t_{1-\frac{\alpha}{c}}(N-2)) N^{-\frac{1}{2}} (w_1 w_2)^{-\frac{1}{2}}} \\
c &= \text{number of sides}
\end{aligned}$$

and $Q_{\cdot}(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the "Common Notation" section on page 3498.

A "quality" confidence interval is both sufficiently narrow (half-width $\le h$) and valid:

$$\begin{aligned}
\Pr(\text{quality}) &= \Pr(\text{half-width} \le h \text{ and validity}) \\
&= \Pr(\text{half-width} \le h \mid \text{validity})(1-\alpha)
\end{aligned}$$

Analyses in the TWOSAMPLESURVIVAL Statement

Rank Tests for Two Survival Curves (TEST=LOGRANK, TEST=GEHAN, TEST=TARONEWARE)

The method is from Lakatos (1988) and Cantor (1997, pp. 83–92).

Define the following notation:

$$\begin{aligned}
X_j(i) &= i\text{th input time point on survival curve for group } j \\
S_j(i) &= \text{input survivor function value corresponding to } X_j(i) \\
h_j(t) &= \text{hazard rate for group } j \text{ at time } t \\
\Psi_j(t) &= \text{loss hazard rate for group } j \text{ at time } t \\
\lambda_j &= \text{exponential hazard rate for group } j \\
R &= \text{hazard ratio of group 2 to group 1} \equiv \text{(assumed constant) value of } \frac{h_2(t)}{h_1(t)} \\
m_j &= \text{median survival time for group } j \\
b &= \text{number of subintervals per time unit} \\
T &= \text{accrual time} \\
\tau &= \text{post-accrual follow-up time} \\
L_j &= \text{exponential loss rate for group } j \\
XL_j &= \text{input time point on loss curve for group } j \\
SL_j &= \text{input survivor function value corresponding to } XL_j \\
mL_j &= \text{median survival time for group } j \\
r_i &= \text{rank for } i\text{th time point}
\end{aligned}$$

Each survival curve can be specified in one of several ways.

- For exponential curves:
  - a single point $(X_j(1), S_j(1))$ on the curve
  - median survival time
  - hazard rate
  - hazard ratio (for curve 2, with respect to curve 1)
- For piecewise linear curves with proportional hazards:
  - a set of points $\{(X_1(1), S_1(1)), (X_1(2), S_1(2)), \ldots\}$ (for curve 1)
  - hazard ratio (for curve 2, with respect to curve 1)
- For arbitrary piecewise linear curves:
  - a set of points $\{(X_j(1), S_j(1)), (X_j(2), S_j(2)), \ldots\}$

A total of M evenly spaced time points $\{t_0 = 0, t_1, t_2, \ldots, t_M = T + \tau\}$ are used in calculations, where

$$M = \text{floor}\left((T + \tau)b\right)$$

The hazard function is calculated for each survival curve at each time point. For an exponential curve, the (constant) hazard is given by one of the following, depending on the input parameterization:

$$h_j(t) = \begin{cases} \lambda_j \\ \lambda_1 R \\ \frac{-\log(\frac{1}{2})}{m_j} \\ \frac{-\log(S_j(1))}{X_j(1)} \\ \frac{-\log(S_1(1))}{X_1(1)} R \end{cases}$$

For a piecewise linear curve, define the following additional notation:

$$\begin{aligned} t_i^- &= \text{largest input time } X \text{ such that } X \le t_i \\ t_i^+ &= \text{smallest input time } X \text{ such that } X > t_i \end{aligned}$$

The hazard is computed using linear interpolation as follows:

$$h_j(t_i) = \frac{S_j(t_i^-) - S_j(t_i^+)}{\left[S_j(t_i^+) - S_j(t_i^-)\right]\left[t_i - t_i^-\right] + S_j(t_i^-)\left[t_i^+ - t_i^-\right]}$$

With proportional hazards, the hazard rate of group 2's curve in terms of the hazard rate of group 1's curve is

$$h_2(t) = h_1(t)R$$

Hazard function values $\{\Psi_j(t_i)\}$ for the loss curves are computed in an analogous way from $\{L_j, XL_j, SL_j, mL_j\}$.

The expected number at risk $N_j(i)$ at time i in group j is calculated for each group and time points 0 through $M-1$, as follows:

$$\begin{aligned} N_j(0) &= Nw_j \\ N_j(i+1) &= N_j(i)\left[1 - h_j(t_i)\left(\frac{1}{b}\right) - \Psi_j(t_i)\left(\frac{1}{b}\right) - \left(\frac{1}{b(T+\tau-t_i)}\right)1_{\{t_i>\tau\}}\right] \end{aligned}$$

Define θ_i as the ratio of hazards and ϕ_i as the ratio of expected numbers at risk for time t_i:

$$\begin{aligned} \theta_i &= \frac{h_2(t_i)}{h_1(t_i)} \\ \phi_i &= \frac{N_2(i)}{N_1(i)} \end{aligned}$$

The expected number of deaths in each subinterval is calculated as follows:

$$D_i = [h_1(t_i)N_1(i) + h_2(t_i)N_2(i)]\left(\frac{1}{b}\right)$$

The rank values are calculated as follows according to which test statistic is used:

$$r_i = \begin{cases} 1, & \text{log-rank} \\ N_1(i) + N_2(i), & \text{Gehan} \\ \sqrt{N_1(i) + N_2(i)}, & \text{Tarone-Ware} \end{cases}$$

The distribution of the test statistic is approximated by $N(E, 1)$ where

$$E = \frac{\sum_{i=0}^{M-1} D_i r_i \left[\frac{\phi_i\theta_i}{1+\phi_i\theta_i} - \frac{\phi_i}{1+\phi_i}\right]}{\sqrt{\sum_{i=0}^{M-1} D_i r_i^2 \frac{\phi_i}{(1+\phi_i)^2}}}$$

Note that $N^{\frac{1}{2}}$ can be factored out of the mean E, and so it can be expressed equivalently as

$$E = N^{\frac{1}{2}} E^\star = N^{\frac{1}{2}} \left[\frac{\sum_{i=0}^{M-1} D_i^\star r_i^\star \left[\frac{\phi_i\theta_i}{1+\phi_i\theta_i} - \frac{\phi_i}{1+\phi_i}\right]}{\sqrt{\sum_{i=0}^{M-1} D_i^\star r_i^{\star 2} \frac{\phi_i}{(1+\phi_i)^2}}}\right]$$

where $E^\star$ is free of N and

$$\begin{aligned} D_i^\star &= [h_1(t_i)N_1^\star(i) + h_2(t_i)N_2^\star(i)]\left(\frac{1}{b}\right) \\ r_i^\star &= \begin{cases} 1, & \text{log-rank} \\ N_1^\star(i) + N_2^\star(i), & \text{Gehan} \\ \sqrt{N_1^\star(i) + N_2^\star(i)}, & \text{Tarone-Ware} \end{cases} \\ N_j^\star(0) &= w_j \\ N_j^\star(i+1) &= N_j^\star(i)\left[1 - h_j(t_i)\left(\frac{1}{b}\right) - \Psi_j(t_i)\left(\frac{1}{b}\right) - \left(\frac{1}{b(T+\tau-t_i)}\right) 1_{\{t_i>\tau\}}\right] \end{aligned}$$

The approximate power is

$$\text{power} = \begin{cases} \Phi\left(-N^{\frac{1}{2}}E^\star - z_{1-\alpha}\right), & \text{upper 1-sided} \\ \Phi\left(N^{\frac{1}{2}}E^\star - z_{1-\alpha}\right), & \text{lower 1-sided} \\ \Phi\left(-N^{\frac{1}{2}}E^\star - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(N^{\frac{1}{2}}E^\star - z_{1-\frac{\alpha}{2}}\right), & \text{2-sided} \end{cases}$$

Note that the upper and lower 1-sided cases are expressed differently than in other analyses. This is because $E^\star > 0$ corresponds to a higher survival curve in group

1 and thus, by the convention used in PROC power for 2-group analyses, the lower side.

For the 1-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \left(\frac{z_{\text{power}} + z_{1-\alpha}}{E^\star} \right)^2$$

For the 2-sided case, the solution for N is obtained by numerically inverting the power equation.

Examples

Example 57.1. One-Way ANOVA

This example deals with the same situation as in Example 34.1 on page 1951 of Chapter 34, “The GLMPOWER Procedure.”

Hocking (1985, p. 109) describes a study of the effectiveness of electrolytes in reducing lactic acid buildup for long-distance runners. You are planning a similar study in which you will allocate five different fluids to runners on a 10-mile course and measure lactic acid buildup immediately after the race. The fluids consist of water and two commercial electrolyte drinks, EZDure and LactoZap, each prepared at two concentrations, low (EZD1 and LZ1) and high (EZD2 and LZ2).

You conjecture that the standard deviation of lactic acid measurements given any particular fluid is about 3.75, and that the expected lactic acid values will correspond roughly to those in Table 57.27. You are least familiar with the LZ1 drink and hence decide to consider a range of reasonable values for that mean.

Table 57.27. Mean Lactic Acid Buildup by Fluid

Water	EZD1	EZD2	LZ1	LZ2
35.6	33.7	30.2	29 or 28	25.9

You are interested in four different comparisons, shown in Table 57.28 with appropriate contrast coefficients.

Table 57.28. Planned Comparisons

	Contrast Coefficients				
Comparison	Water	EZD1	EZD2	LZ1	LZ2
Water versus electrolytes	4	-1	-1	-1	-1
EZD versus LZ	0	1	1	-1	-1
EZD1 versus EZD2	0	1	-1	0	0
LZ1 versus LZ2	0	0	0	1	-1

For each of these contrasts you want to determine the sample size required to achieve a power of 0.9 for detecting an effect with magnitude in accord with Table 57.27. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed for individual contrasts. You plan to test each contrast at $\alpha = 0.025$. In the interests of reducing costs, you will provide twice as many runners with water as with any of the electrolytes; in other words, you will use a sample size weighting scheme of 2:1:1:1:1. Use the ONEWAYANOVA statement in the POWER procedure to compute the sample sizes. The statements required to perform this analysis are as follows:

```
proc power;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = .
      power = 0.9
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
run;
```

The NTOTAL= option with a missing value (.) indicates total sample size as the result parameter. The GROUPMEANS= option with values from Table 57.27 specifies your conjectures for the means. With only one mean varying (the LZ1 mean), the "crossed" notation is simpler, showing scenarios for each group mean separated by a vertical bar (|). See the "Specifying Value Lists in Analysis Statements" section on page 3490 for more details on crossed and matched notations for grouped values. The contrasts in Table 57.28 are specified with the CONTRAST option, using the "matched" notation with each contrast enclosed in parentheses. The STDDEV=, ALPHA=, and POWER= options specify the error standard deviation, significance level, and power. The GROUPWEIGHTS= option specifies the weighting schemes. Default values for the NULL= and SIDES= options specify a 2-sided t test of the contrast equal to 0. See Output 57.1.1 for the output.

Output 57.1.1. Sample Sizes for One-Way ANOVA Contrasts

```
                            The POWER Procedure
                     Single DF Contrast in One-Way ANOVA

                           Fixed Scenario Elements

                     Method                         Exact
                     Alpha                          0.025
                     Standard Deviation              3.75
                     Group Weights              2 1 1 1 1
                     Nominal Power                    0.9
                     Number of Sides                    2
                     Null Contrast Value                0


                              Computed N Total

                                                                  Actual      N
   Index  -----Contrast-----  -------------Means-------------      Power  Total

       1   4  -1  -1  -1  -1   35.6    33.7    30.2   29   25.9    0.947     30
       2   4  -1  -1  -1  -1   35.6    33.7    30.2   28   25.9    0.901     24
       3   0   1   1  -1  -1   35.6    33.7    30.2   29   25.9    0.929     60
       4   0   1   1  -1  -1   35.6    33.7    30.2   28   25.9    0.922     48
       5   0   1  -1   0   0   35.6    33.7    30.2   29   25.9    0.901    174
       6   0   1  -1   0   0   35.6    33.7    30.2   28   25.9    0.901    174
       7   0   0   0   1  -1   35.6    33.7    30.2   29   25.9    0.902    222
       8   0   0   0   1  -1   35.6    33.7    30.2   28   25.9    0.902    480
```

The sample sizes in Output 57.1.1 range from 24 for the comparison of water versus electrolytes to 480 for the comparison of LZ1 versus LZ2, both assuming the smaller LZ1 mean. The sample size for the latter comparison is relatively large because the small mean difference of $28 - 25.9 = 2.1$ is hard to detect.

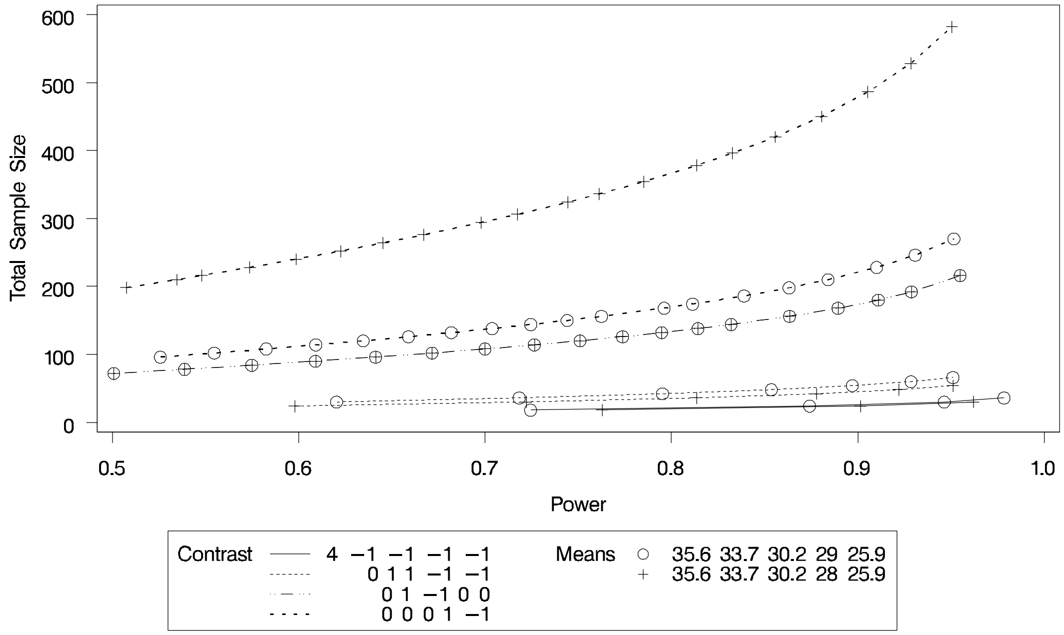
The Nominal Power of 0.9 in the Fixed Scenario Elements table in Output 57.1.1 represents the input target power, and the Actual Power column in the Computed N Total table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting. Note that all of the sample sizes are rounded up to multiples of 6 to preserve integer group sizes (since the group weights add up to 6). You can use the NFRACTIONAL option in the ONEWAYANOVA statement to compute raw fractional sample sizes.

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC POWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the x-axis. (The result parameter, here sample size, is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the power range.

```
proc power plotonly;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = .
      power = 0.9
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
   plot x=power min=.5 max=.95;
run;
```

See Output 57.1.2 for the resulting plot.

Output 57.1.2. Plot of Sample Size versus Power for One-Way ANOVA Contrasts



In Output 57.1.2, the line style identifies the contrast, and the plotting symbol identifies the group means scenario. The plot shows that the required sample size is highest for the (0 0 0 1 -1) contrast, corresponding to the test of LZ1 versus LZ2 that was previously found to require the most resources, in either cell means scenario.

Note that some of the plotted points in Output 57.1.2 are unevenly spaced. This is because the plotted points are the *rounded* sample size results at their corresponding *actual* power levels. The range specified with the MIN= and MAX= values in the PLOT statement correspond to *nominal* power levels. In some cases, actual power is substantially higher than nominal power. To obtain plots with evenly spaced points (but with *fractional* sample sizes at the computed points), you can use the NFRACTIONAL option in the analysis statement preceding the PLOT statement.

Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 24 to 480 that achieves 0.9 power for different comparisons). In the ONEWAYANOVA statement, identify power as the result

(POWER=.), and specify NTOTAL=24. The following statements produce the plot:

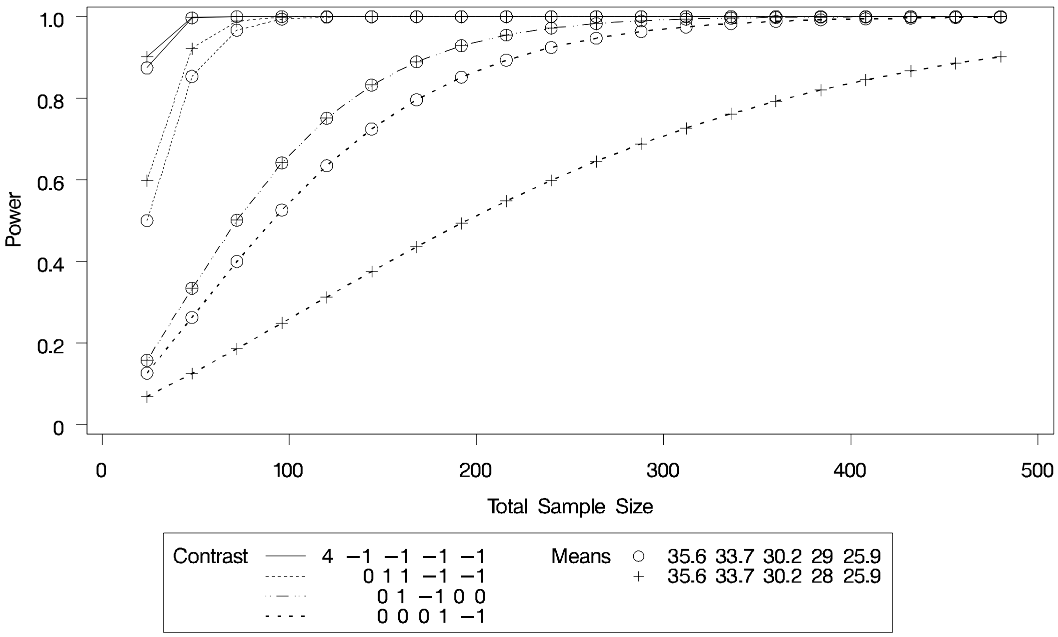
```
proc power plotonly;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = 24
      power = .
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
   plot x=n min=24 max=480;
run;
```

The X=N option in the PLOT statement requests a plot with sample size on the x-axis.

Note that the value specified with the NTOTAL=24 option is not used. It is overridden in the plot by the MIN= and MAX= options in the PLOT statement, and the PLOTONLY option in the PROC POWER statement disables nongraphical results. But the NTOTAL= option (along with a value) is still needed in the ONEWAYANOVA statement as a placeholder, to identify the desired parameterization for sample size.

See Output 57.1.3 for the plot.

Output 57.1.3. Plot of Power versus Sample Size for One-Way ANOVA Contrasts



Although Output 57.1.2 and Output 57.1.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment. Output 57.1.2 reveals the range of required sample sizes for powers of interest, and Output 57.1.3 reveals the range of achieved powers for sample sizes of interest.

Example 57.2. The Sawtooth Power Function in Proportion Analyses

For many common statistical analyses, the power curve is monotonically increasing: the more samples you take, the more power you achieve. However, in statistical analyses of discrete data, such as tests of proportions, the power curve is often non-monotonic. A small increase in sample size can result in a *decrease* in power, a decrease that is sometimes substantial. The explanation is that the actual significance level (in other words, the achieved Type 1 error rate) for discrete tests strays below the target level and varies with sample size. The power loss from a decrease in the Type 1 error rate may outweigh the power gain from an increase in sample size. The example discussed in this section demonstrates this “sawtooth” phenomenon. For additional discussion on the topic, refer to Chernick and Liu (2002).

Suppose you have a new scheduling system for an airline, and you want to determine how many flights you must observe to have at least an 80% chance of establishing an improvement in the proportion of late arrivals on a specific travel route. You will use a 1-sided exact binomial proportion test with a null proportion of 30%, the frequency of late arrivals under the previous scheduling system, and a nominal significance level of α = 0.05. Well-supported predictions estimate the new late arrival rate to be about 20%, and you will base your sample size determination on this assumption.

The POWER procedure does not currently compute exact sample size directly for the exact binomial test. But you can get an initial estimate by computing the approximate sample size required for a z test. Use the ONESAMPLEFREQ statement in the POWER procedure with TEST=Z and METHOD=NORMAL to compute the approximate sample size to achieve a power of 0.8 using the z test. The following statements perform the analysis:

```
proc power;
   onesamplefreq test=z method=normal
      sides           = 1
      alpha           = 0.05
      nullproportion = 0.3
      proportion      = 0.2
      ntotal          = .
      power           = 0.8;
run;
```

The NTOTAL= option with a missing value (.) indicates sample size as the result parameter. The SIDES=1 option specifies a 1-sided test. The ALPHA=, NULLPROPORTION=, and POWER= options specify the significance level of 0.05, null value of 0.3, and target power of 0.8. The PROPORTION= option specifies your conjecture of 0.3 for the true proportion.

Output 57.2.1. Approximate Sample Size for z Test of a Proportion

```
                         The POWER Procedure
                    Z Test for Binomial Proportion

                       Fixed Scenario Elements

              Method                 Normal approximation
              Number of Sides                           1
              Null Proportion                         0.3
              Alpha                                  0.05
              Binomial Proportion                     0.2
              Nominal Power                           0.8


                          Computed N Total

                          Actual         N
                           Power     Total

                           0.800       119
```

The results, shown in Output 57.2.1, indicate that you need to observe about N=119 flights to have an 80% chance of rejecting the hypothesis of a late arrival proportion of 30% or higher, if the true proportion is 20%, using the z test. A similar analysis (Output 57.2.2) reveals an approximate sample size of N=129 for the z test with continuity correction, which performed using TEST=ADJZ:

```
proc power;
   onesamplefreq test=adjz method=normal
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.3
      proportion     = 0.2
      ntotal         = .
      power          = 0.8;
run;
```

Output 57.2.2. Approximate Sample Size for z Test with Continuity Correction

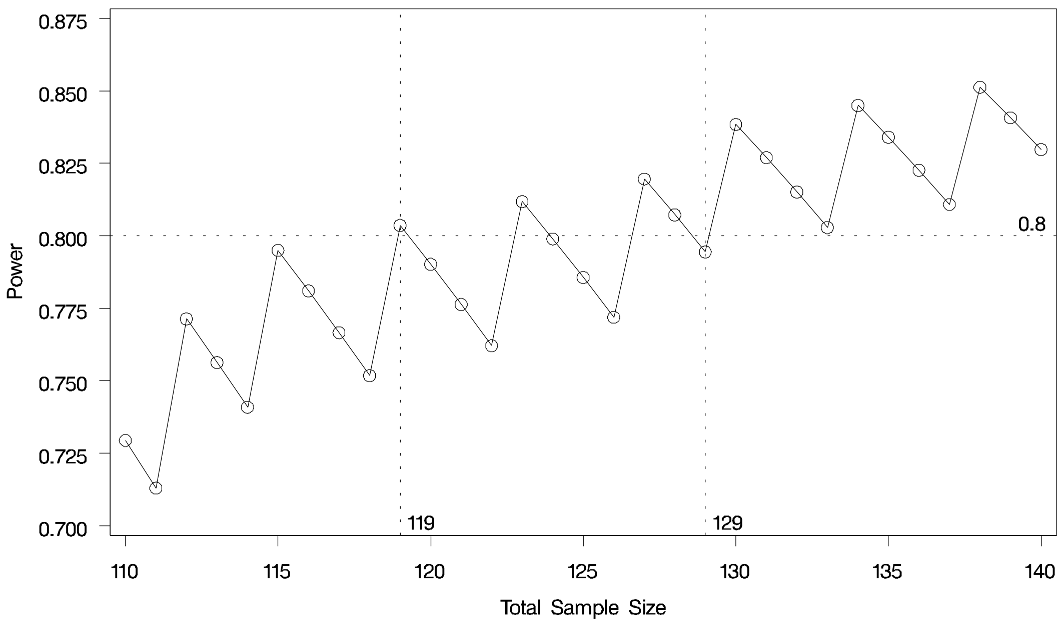
```
                          The POWER Procedure
           Z Test for Binomial Proportion with Continuity Adjustment

                        Fixed Scenario Elements

                  Method                  Normal approximation
                  Number of Sides                            1
                  Null Proportion                          0.3
                  Alpha                                   0.05
                  Binomial Proportion                      0.2
                  Nominal Power                            0.8


                          Computed N Total

                          Actual        N
                           Power    Total

                           0.801      129
```

Based on the approximate sample size results, you decide to explore the power of the exact binomial test for sample sizes between 110 and 140. The following statements produce the plot:

```
proc power plotonly;
   onesamplefreq test=exact
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.3
      proportion     = 0.2
      ntotal         = 119
      power          = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

The TEST=EXACT option in the ONESAMPLEFREQ statement specifies the exact binomial test, and the missing value (.) for the POWER= option indicates power as the result parameter. The PLOTONLY option in the PROC POWER statement disables nongraphical output. The PLOT statement with X=N requests a plot with sample size on the x-axis. The MIN= and MAX= options in the PLOT statement specify the sample size range. The YOPTS=(REF=) and XOPTS=(REF=) options add reference lines to highlight the approximate sample size results. The STEP=1 option produces a point at each integer sample size. The sample size value specified with the NTOTAL= option in the ONESAMPLEFREQ statement is overridden by the MIN= and MAX= options in the PLOT statement. Output 57.2.3 shows the plot.

Output 57.2.3. Plot of Power versus Sample Size for Exact Binomial Test



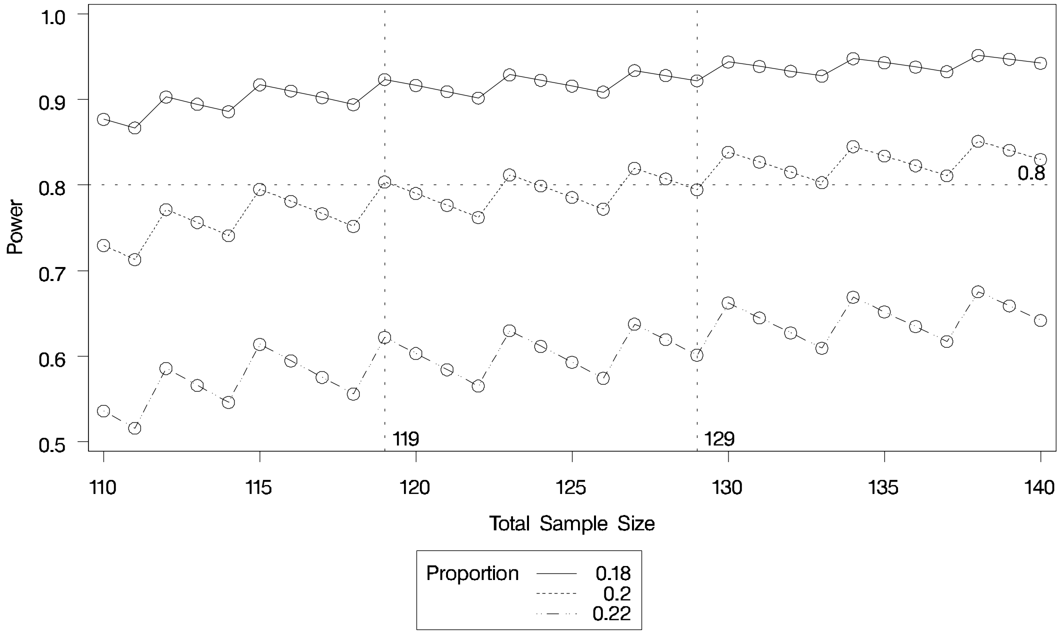
Note the sawtooth pattern in Output 57.2.3. Although the power surpasses the target level of 0.8 at N=119, it decreases to 0.79 with N=120 and further to 0.76 with N=122 before rising again to 0.81 with N=123. Not until N=130 does the power stay above the 0.8 target. Thus, a more conservative sample size recommendation of 130 might be appropriate, depending on the precise goals of the sample size determination.

In addition to considering alternative sample sizes, you may also want to assess the sensitivity of the power to inaccuracies in assumptions about the true proportion. The following statements produce a plot including true proportion values of 0.18 and 0.22. They are identical to the previous statements except for the additional true proportion values specified with the PROPORTION= option in the ONESAMPLEFREQ statement.

```
proc power plotonly;
   onesamplefreq test=exact
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.3
      proportion     = 0.18 0.2 0.22
      ntotal         = 119
      power          = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

Output 57.2.4 shows the plot.

Output 57.2.4. Plot for Assessing Sensitivity to True Proportion Value



The plot reveals a dramatic sensitivity to the true proportion value. For N=119, the power is about 0.92 if the true proportion is 0.22, and as low as 0.62 if the proportion is 0.18. Note also that the power jumps occur at the same sample sizes in all three curves; the curves are only shifted and stretched vertically. This is because spikes and valleys in power curves are invariant to the true proportion value; they are due to changes in the critical value of the test.

A closer look at some ancillary output from the analysis sheds light on this property of the sawtooth pattern. You can add an ODS OUTPUT statement to save the plot content corresponding to Output 57.2.3 to a data set:

```
proc power plotonly;
   ods output plotcontent=PlotData;
   onesamplefreq test=exact
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.3
      proportion     = 0.2
      ntotal         = 119
      power          = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

The PlotData data set contains parameter values for each point in the plot. The parameters including underlying characteristics of the putative test. The following statements print the critical value and actual significance level along with sample size and power.

```
proc print data=PlotData;
   var NTotal LowerCritVal Alpha Power;
run;
```

Output 57.2.5 shows the plot data.

Output 57.2.5. Numerical Content of Plot

```
                         Lower
       Obs    NTotal    CritVal      Alpha     Power

         1      110        24       0.0356     0.729
         2      111        24       0.0313     0.713
         3      112        25       0.0446     0.771
         4      113        25       0.0395     0.756
         5      114        25       0.0349     0.741
         6      115        26       0.0490     0.795
         7      116        26       0.0435     0.781
         8      117        26       0.0386     0.767
         9      118        26       0.0341     0.752
        10      119        27       0.0478     0.804
        11      120        27       0.0425     0.790
        12      121        27       0.0377     0.776
        13      122        27       0.0334     0.762
        14      123        28       0.0465     0.812
        15      124        28       0.0414     0.799
        16      125        28       0.0368     0.786
        17      126        28       0.0327     0.772
        18      127        29       0.0453     0.820
        19      128        29       0.0404     0.807
        20      129        29       0.0359     0.794
        21      130        30       0.0493     0.838
        22      131        30       0.0441     0.827
        23      132        30       0.0394     0.815
        24      133        30       0.0351     0.803
        25      134        31       0.0480     0.845
        26      135        31       0.0429     0.834
        27      136        31       0.0384     0.823
        28      137        31       0.0342     0.811
        29      138        32       0.0466     0.851
        30      139        32       0.0418     0.841
        31      140        32       0.0374     0.830
```

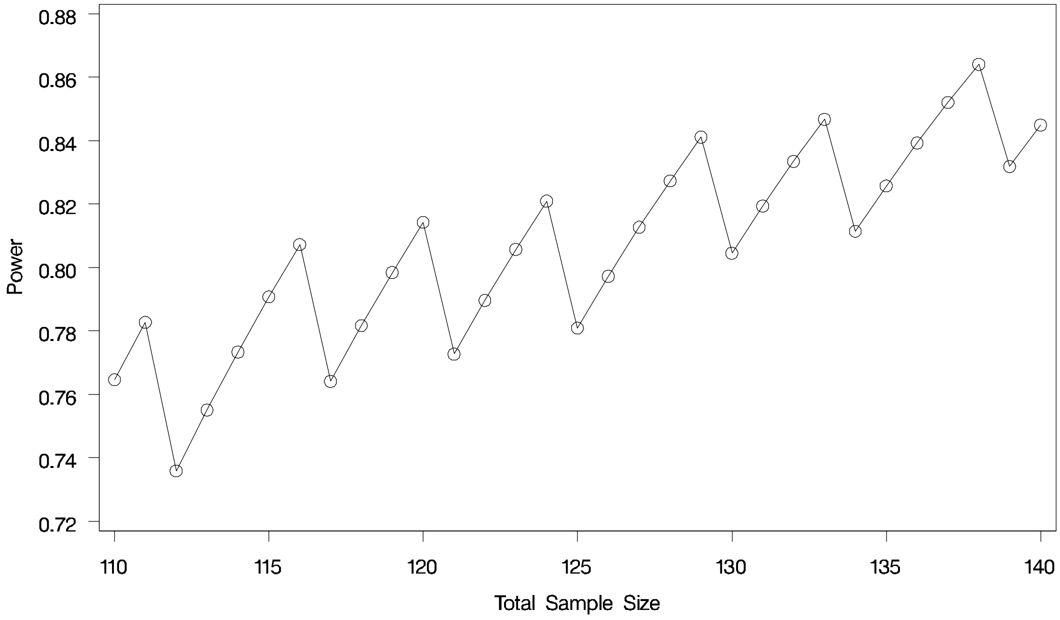
Note that whenever the critical value changes, the actual α jumps up to a value close to the nominal α=0.05, and the power also jumps up. Then while the critical value stays constant, the actual α and power slowly decrease. The critical value is independent of the true proportion value. So, you can achieve a locally maximal power by choosing a sample size corresponding to a spike on the sawtooth curve, and this choice is locally optimal *regardless* of the unknown value of the true proportion. Locally optimal sample sizes in this case include 115, 119, 123, 127, 130, and 134.

As a point of interest, the power does not always jump sharply and decrease gradually. The shape of the sawtooth depends on the direction of the test and the location of the null proportion relative to 0.5. For example, if the direction of the hypothesis in this example is reversed (by switching true and null proportion values) so that the rejection region is in the upper tail, then the power curve exhibits sharp decreases and gradual increases. The following statements are similar to those producing the plot in Output 57.2.3 but with values of the PROPORTION= and NULLPROPORTION= options switched.

```
proc power plotonly;
   onesamplefreq test=exact
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.2
      proportion     = 0.3
      ntotal         = 119
      power          = .;
   plot x=n min=110 max=140 step=1;
run;
```

The resulting plot is shown in Output 57.2.6.

Output 57.2.6. Plot of Power versus Sample Size for Another 1-Sided Test

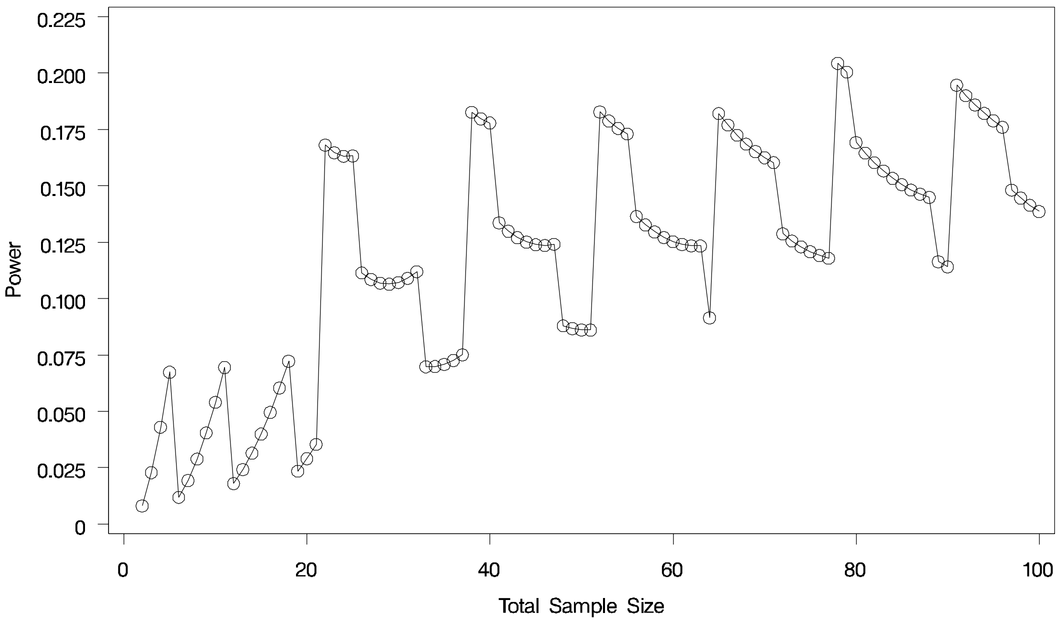


Finally, 2-sided tests can lead to even more irregular power curve shapes, since changes in lower and upper critical values affect the power in different ways. The following statements produce a plot of power versus sample size for the scenario of a 2-sided test with high alpha and a true proportion close to the null value.

```
proc power plotonly;
   onesamplefreq test=exact
      sides          = 2
      alpha          = 0.2
      nullproportion = 0.1
      proportion     = 0.09
      ntotal         = 10
      power          = .;
   plot x=n min=2 max=100 step=1;
run;
```

The resulting plot is shown in Output 57.2.7.

Output 57.2.7. Plot of Power versus Sample Size for a 2-Sided Test



Due to the irregular shapes of power curves for proportion tests, the question "Which sample size should I use?" is often insufficient. A sample size solution produced directly in PROC POWER reveals the smallest possible sample size to achieve your target power. But as the Examples in this section demonstrate, it is helpful to consult graphs for answers to questions such as the following:

- Which sample size will guarantee that all higher sample sizes also achieve my target power?
- Given a candidate sample size, can I increase it slightly to achieve locally maximal power, or perhaps even decrease it and get higher power?

Example 57.3. Simple AB/BA Crossover Designs

Crossover trials are experiments in which each subject is given a sequence of different treatments. They are especially common in clinical trials for medical studies. The reduction in variability from taking multiple measurements on a subject allows for more precise treatment comparisons. The simplest such design is the AB/BA crossover, in which each subject receives each of two treatments in a randomized order.

Under certain simplifying assumptions, you can test the treatment difference in an AB/BA crossover trial using either a paired or two-sample t test (or equivalence test, depending on the hypothesis). This example will demonstrate when and how you can use the PAIREDMEANS statement in PROC POWER to perform power analyses for AB/BA crossover designs.

Senn (1993, Chapter 3) discusses a study comparing the effects of two bronchodilator medications in treatment of asthma, using an AB/BA crossover design. Suppose you want to plan a similar study comparing two new medications, “Xilodol” and “Brantium.” Half of the patients would be assigned to sequence AB, getting a dose of Xilodol in the first treatment period, a wash-out period of one week, and then a dose of Brantium in the second treatment period. The other half would be assigned to sequence BA, following the same schedule but with the drugs reversed. In each treatment period you would administer the drugs in the morning and then measure peak expiratory flow (PEF) at the end of the day, with higher PEF representing better lung function.

You conjecture that the mean and standard deviation of PEF are about μ_A = 310 and σ_A = 40 for Xilodol and μ_B = 330 and σ_B = 55 for Brantium, and that each pair of measurements on the same subject will have a correlation of about 0.3. You want to compute the power of both 1-sided and 2-sided tests of mean difference, with a significance level of $\alpha = 0.01$, for a sample size of 100 patients and also plot the power for a range of 50 to 200 patients. Note that the allocation ratio of patients to the two sequences is irrelevant in this analysis.

The choice of statistical test depends on which assumptions are reasonable. One possibility is a t test. A paired or two-sample t test is valid when there is no carryover effect and no interactions between patients, treatments, and periods. See Senn (1993, Chapter 3) for more details. The choice between a paired or a two-sample test depends on what you assume about the period effect. If you assume no period effect, then a paired t test is the appropriate analysis for the design, with the first member of each pair being the Xilodol measurement (regardless of which sequence the patient belongs to). Otherwise the two-sample t test approach is called for, since this analysis adjusts for the period effect using an extra degree of freedom.

Suppose you assume no period effect. Then you can use the PAIREDMEANS statement in PROC POWER with the TEST=DIFF option to perform a sample size analysis for the paired t test. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify the conjectured means and standard deviations for each drug using the PAIREDMEANS= and PAIREDSTDDEVS= options and the correlation using the CORR= option. Specify both 1- and 2-sided tests using the SIDES= option, the significance level using the ALPHA= option, and the sample size (in terms of number of pairs) using the NPAIRS= option. Generate a plot of power versus sample size by specifying the PLOT statement with X=N to request a plot with sample size on the x-axis. (The result parameter, here power, is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the sample size range (as numbers of pairs).

The following statements perform the sample size analysis.

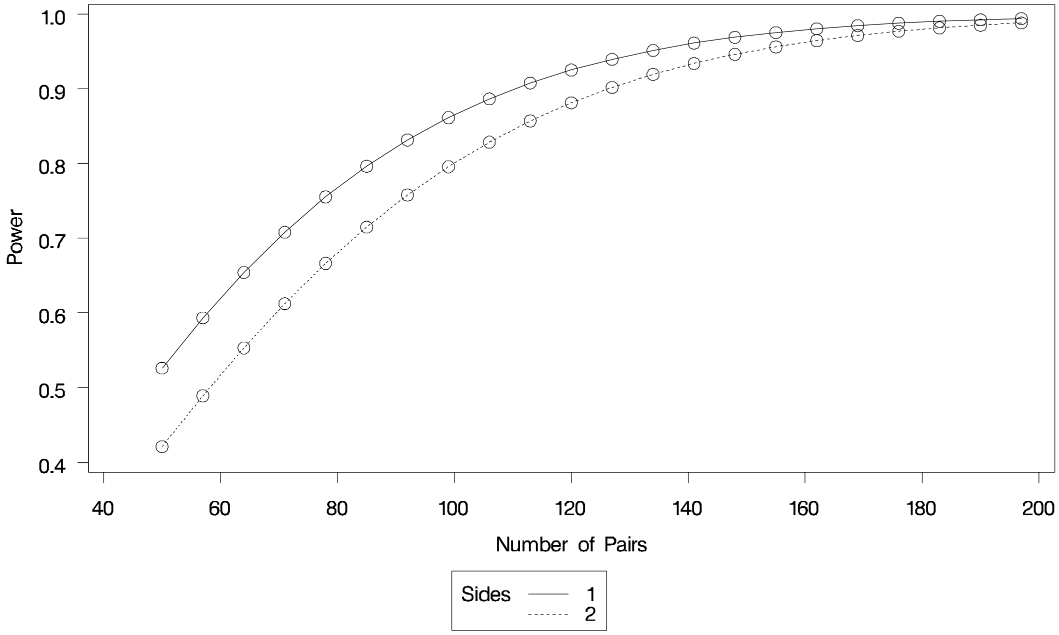
```
proc power;
   pairedmeans test=diff
      pairedmeans   = (330 310)
      pairedstddevs = (40 55)
      corr          = 0.3
      sides         = 1 2
      alpha         = 0.01
      npairs        = 100
      power         = .;
   plot x=n min=50 max=200;
run;
```

Default values for the NULLDIFF= and DIST= options specify a null mean difference of 0 and the assumption of normally distributed data. The output is shown in Output 57.3.1 and Output 57.3.2.

Output 57.3.1. Power for Paired t Analysis of Crossover Design

```
                      The POWER Procedure
               Paired t Test for Mean Difference

                    Fixed Scenario Elements

             Distribution                  Normal
             Method                         Exact
             Alpha                           0.01
             Mean 1                           330
             Mean 2                           310
             Standard Deviation 1              40
             Standard Deviation 2              55
             Correlation                      0.3
             Number of Pairs                  100
             Null Difference                    0


                         Computed Power

                     Index    Sides    Power

                         1        1    0.865
                         2        2    0.801
```

Output 57.3.2. Plot of Power versus Sample Size for Paired t Analysis of Crossover Design



The Computed Power table in Output 57.3.1 shows that the power with 100 patients is about 0.8 for the 2-sided test and 0.87 for the 1-sided test with the alternative of larger Brantium mean. In Output 57.3.2, the line style identifies the number of sides of the test. The plotting symbols identify locations of actual computed powers; the curves are linear interpolations of these points. The plot demonstrates how much higher the power is for the 1-sided test than the 2-sided test for the range of sample sizes.

Suppose now that instead of detecting a difference between Xilodol and Brantium, you want to establish that they are similar, in particular, that the absolute mean PEF difference is at most 35. You might consider this goal if, for example, one of the drugs has fewer side effects and if a difference of no more than 35 is considered clinically small. Instead of a standard t test, you would conduct an *equivalence test* of the treatment mean difference for the two drugs. You would test the hypothesis that the true difference is less than -35 or more than 35 against the alternative that the mean difference is between -35 and 35, using an additive model and a two one-sided tests (“TOST”) analysis.

Assuming no period effect, you can use the PAIREDMEANS statement with the TEST=EQUIV_DIFF option to perform a sample size analysis for the paired equivalence test. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Use the LOWER= and UPPER= options to specify the equivalence bounds of -35 and 35. Use the PAIREDMEANS=, PAIREDSTDDEVS=, CORR=, and ALPHA= options in the same way as in the t test at the beginning of this example to specify the remaining parameters.

The following statements perform the sample size analysis.

```
proc power;
   pairedmeans test=equiv_add
      lower         = -35
      upper         = 35
      pairedmeans   = (330 310)
      pairedstddevs = (40 55)
      corr          = 0.3
      alpha         = 0.01
      npairs        = 100
      power         = .;
run;
```

The default option DIST=NORMAL specifies an assumption of normally distributed data. The output is shown in Output 57.3.3.

Output 57.3.3. Power for Paired Equivalence Test for Crossover Design

```
            The POWER Procedure
Equivalence Test for Paired Mean Difference

          Fixed Scenario Elements

  Distribution                       Normal
  Method                              Exact
  Lower Equivalence Bound               -35
  Upper Equivalence Bound                35
  Alpha                                0.01
  Reference Mean                        330
  Treatment Mean                        310
  Standard Deviation 1                   40
  Standard Deviation 2                   55
  Correlation                           0.3
  Number of Pairs                       100


               Computed Power

                   Power

                   0.598
```

The power for the paired equivalence test with 100 patients is about 0.6.

Example 57.4. Noninferiority Test with Lognormal Data

The typical goal in noninferiority testing is to conclude that a new treatment or process or product is not appreciably worse than some standard. This is accomplished by convincingly rejecting a 1-sided null hypothesis that the new treatment is appreciably worse than the standard. When designing such studies, investigators must define precisely what constitutes “appreciably worse.”

You can use the POWER procedure for sample size analyses for a variety of noninferiority tests, by specifying custom, 1-sided null hypotheses for common tests. This example illustrates the strategy (often called Blackwelder's scheme, Blackwelder 1982) by comparing the means of two independent lognormal samples. The logic applies to one-sample, two-sample, and paired-sample problems involving normally distributed measures and proportions.

Suppose you are designing a study hoping to show that a new (less expensive) manufacturing process does not produce appreciably more pollution than the current process. Quantifying "appreciably worse" as 10%, you seek to show that the mean pollutant level from the new process is less than 110% of that from the current process. In standard hypothesis testing notation, you seek to reject

$$H_0\colon \frac{\mu_{\mathrm{new}}}{\mu_{\mathrm{current}}} \geq 1.10$$

in favor of

$$H_A\colon \frac{\mu_{\mathrm{new}}}{\mu_{\mathrm{current}}} < 1.10$$

This is described graphically in Figure 57.8. Mean ratios below 100% are better levels for the new process; a ratio of 100% indicates absolute equivalence; ratios of 100–110% are "tolerably" worse; and ratios exceeding 110% are appreciably worse.

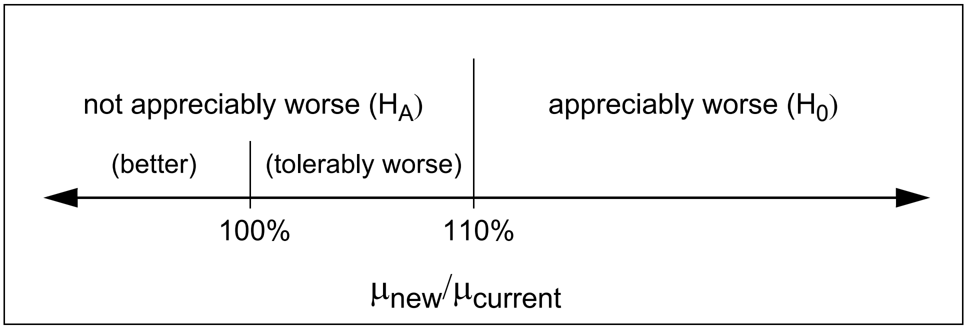

not appreciably worse (H_A)
appreciably worse (H_0)
(better)
(tolerably worse)
100%
110%
$\mu_{new}/\mu_{current}$

Figure 57.8. Hypotheses for the Pollutant Study

An appropriate test for this situation is the common two-group t test on log-transformed data. The hypotheses become

$$\begin{array}{lcl} H_0 & : & \log(\mu_{\mathrm{new}}) - \log(\mu_{\mathrm{current}}) \geq \log(1.10) \\ H_A & : & \log(\mu_{\mathrm{new}}) - \log(\mu_{\mathrm{current}}) < \log(1.10) \end{array}$$

Measurements of the pollutant level will be taken using laboratory models of the two processes and will be treated as independent lognormal observations with a coefficient of variation (σ/μ) between 0.5 and 0.6 for both processes. You will end up with 300 measurements for the current process and 180 for the new one. It is important to avoid a Type 1 error here, so you set the Type 1 error rate to 0.01. Your theoretical work suggests that the new process will actually reduce the pollutant by about 10% (to 90% of current), but you need to compute and graph the power of the study if the new levels are actually between 70% and 120% of current levels.

Implement the sample size analysis using the TWOSAMPLEMEANS statement in PROC POWER with the TEST=RATIO option, Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify a series of scenarios for the mean ratio between 0.7 and 1.2 using the MEANRATIO= option. Use the NULLRATIO= option to specify the null mean ratio of 1.10. Specify SIDES=L to indicate a 1-sided test with the alternative hypothesis stating that the mean ratio is *lower* than the null value. Specify the significance level, scenarios for the coefficient of variation, and the group sample sizes using the ALPHA=, CV=, and GROUPNS= options. Generate a plot of power versus mean ratio by specifying the PLOT statement with X=EFFECT to request a plot with mean ratio on the x-axis. (The result parameter, here power, is always plotted on the other axis.) Use the STEP= option in the PLOT statement to specify an interval of 0.05 between computed points in the plot.

The following statements perform the desired analysis.

```
proc power;
   twosamplemeans test=ratio
      meanratio = 0.7 to 1.2 by 0.1
      nullratio = 1.10
      sides     = L
      alpha     = 0.01
      cv        = 0.5 0.6
      groupns   = (300 180)
      power     = .;
   plot x=effect step=0.05;
run;
```

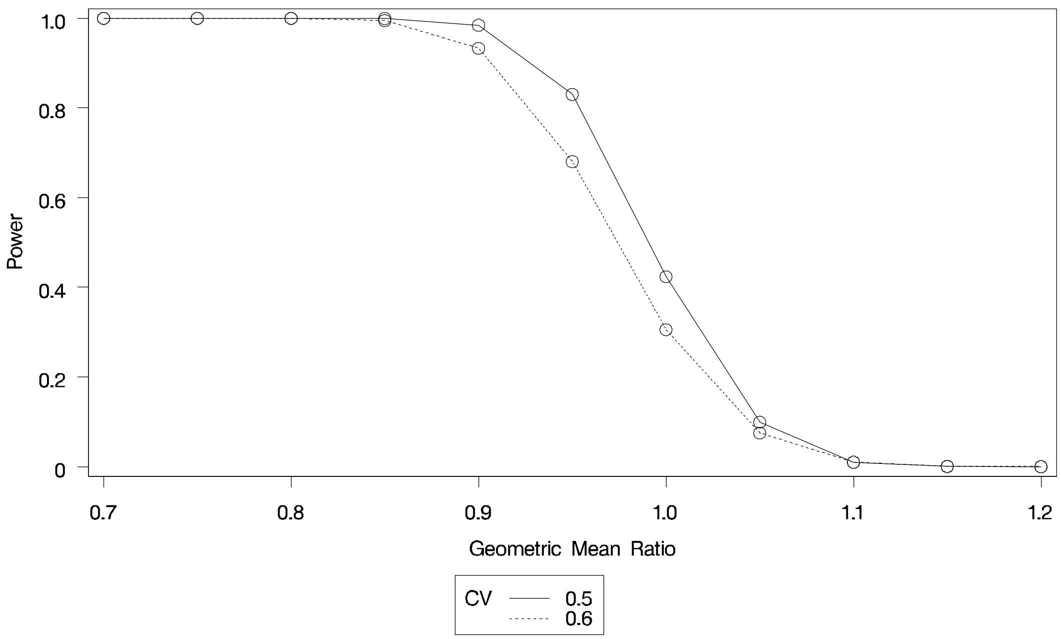
Note the use of SIDES=L, which forces computations for cases that need a rejection region that is opposite to the one providing the most one-tailed power; in this case, it is the lower tail. Such cases will show power that is less than the prescribed Type 1 error rate. The default option DIST=LOGNORMAL specifies the assumption of lognormally distributed data. The default MIN= and MAX= options in the plot statement specify an x-axis range identical to the effect size range in the TWOSAMPLEMEANS statement (mean ratios between 0.7 and 1.2).

See the output in Output 57.4.1 and Output 57.4.2.

Output 57.4.1. Power for Noninferiority Test of Ratio

```
                              The POWER Procedure
                        Two-sample t Test for Mean Ratio

                            Fixed Scenario Elements

                   Distribution                    Lognormal
                   Method                              Exact
                   Number of Sides                         L
                   Null Geometric Mean Ratio             1.1
                   Alpha                                0.01
                   Group 1 Sample Size                   300
                   Group 2 Sample Size                   180


                                 Computed Power

                                     Geo
                                    Mean
                          Index    Ratio      CV    Power

                              1      0.7     0.5    >.999
                              2      0.7     0.6    >.999
                              3      0.8     0.5    >.999
                              4      0.8     0.6    >.999
                              5      0.9     0.5    0.985
                              6      0.9     0.6    0.933
                              7      1.0     0.5    0.424
                              8      1.0     0.6    0.306
                              9      1.1     0.5    0.010
                             10      1.1     0.6    0.010
                             11      1.2     0.5    <.001
                             12      1.2     0.6    <.001
```

Output 57.4.2. Plot of Power versus Mean Ratio for Noninferiority Test



The Computed Power table in Output 57.4.1 shows that power exceeds 0.90 if the true mean ratio is 90% or less, as surmised. But power is unacceptably low (0.31–0.42) if the processes happen to be truly equivalent. Note that the power is identical to the alpha level (0.01) if the true mean ratio is 1.10 and below 0.01 if the true mean ratio is appreciably worse (> 110%). In Output 57.4.2, the line style identifies the coefficient of variation. The plotting symbols identify locations of actual computed powers; the curves are linear interpolations of these points.

Example 57.5. Multiple Regression and Correlation

You are working with a team of preventive cardiologists investigating whether elevated serum homocysteine levels are linked to atherosclerosis (plaque buildup in coronary arteries). The planned analysis is an ordinary least squares regression to assess the relationship between total homocysteine level (tHcy) and a plaque burden index (PBI), adjusting for six other variables: age, gender, plasma levels of folate, vitamins B_6 and B_{12}, and a serum cholesterol index. You will regress PBI on tHcy and the six other predictors (plus the intercept) and use a Type III F test to assess whether tHcy is a significant predictor after adjusting for the others. You wonder whether 100 subjects will provide adequate statistical power.

This is a correlational study at a single time. Subjects will be screened so that about half will have had a heart problem. All eight variables will be measured during one visit. Most clinicians are familiar with simple correlations between two variables, so you decide to pose the statistical problem in terms of estimating and testing the partial correlation between X_1 = tHcy and Y = PBI, controlling for the six other predictor variables ($R_{YX_1|X_{-1}}$). This greatly simplifies matters, especially the elicitation of the conjectured effect.

You use partial regression plots like that shown in Figure 57.9 to teach the team that the partial correlation between PBI and tHcy is the correlation of two sets of residuals obtained from ordinary regression models, one from regressing PBI on the six covariates and the other from regressing tHcy on the same covariates. Thus each subject has "expected" tHcy and PBI values based on the six covariates. The cardiologists believe that subjects who are relatively higher than expected on tHcy will also be relatively higher than expected on PBI. The partial correlation quantifies that adjusted association just like a standard simple correlation does with the unadjusted linear association between two variables.

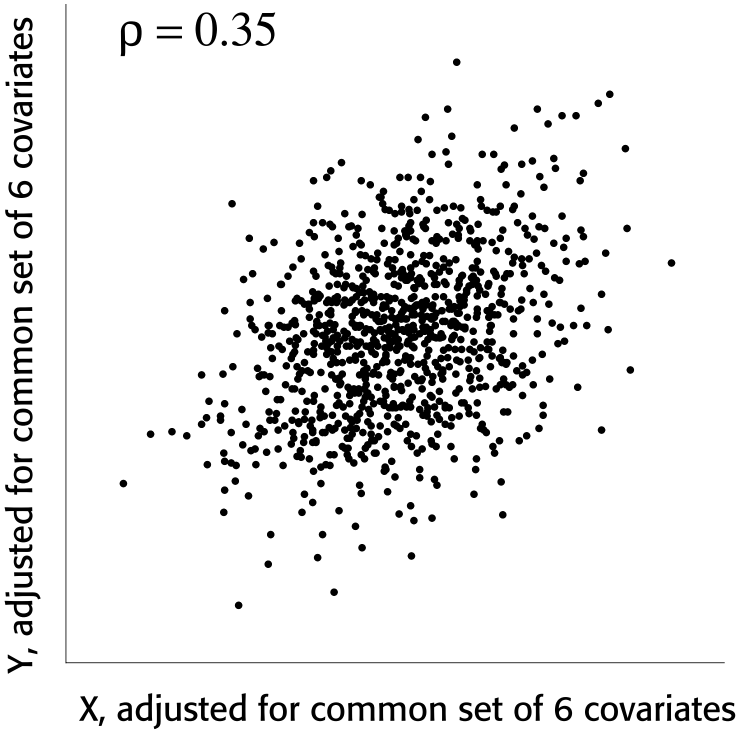


Figure 57.9. Partial Regression Plot

Based on previously published studies of various coronary risk factors and after viewing a set of scatterplots showing various correlations, the team surmises that the true partial correlation is likely to be at least 0.35.

You want to compute the statistical power for a sample size of N = 100, using α = 0.05. You also want to plot power for sample sizes between 50 and 150. Use the MULTREG statement to compute the power and the PLOT statement to produce the graph. Since the predictors are observed rather than fixed in advanced, and a joint multivariate normal assumption seems tenable, use MODEL=RANDOM. The following statements perform the power analysis:

```
proc power;
   multreg
      model = random
      nfullpredictors = 7
      ntestpredictors = 1
      partialcorr = 0.35
      ntotal = 100
      power = .;
   plot x=n min=50 max=150;
run;
```

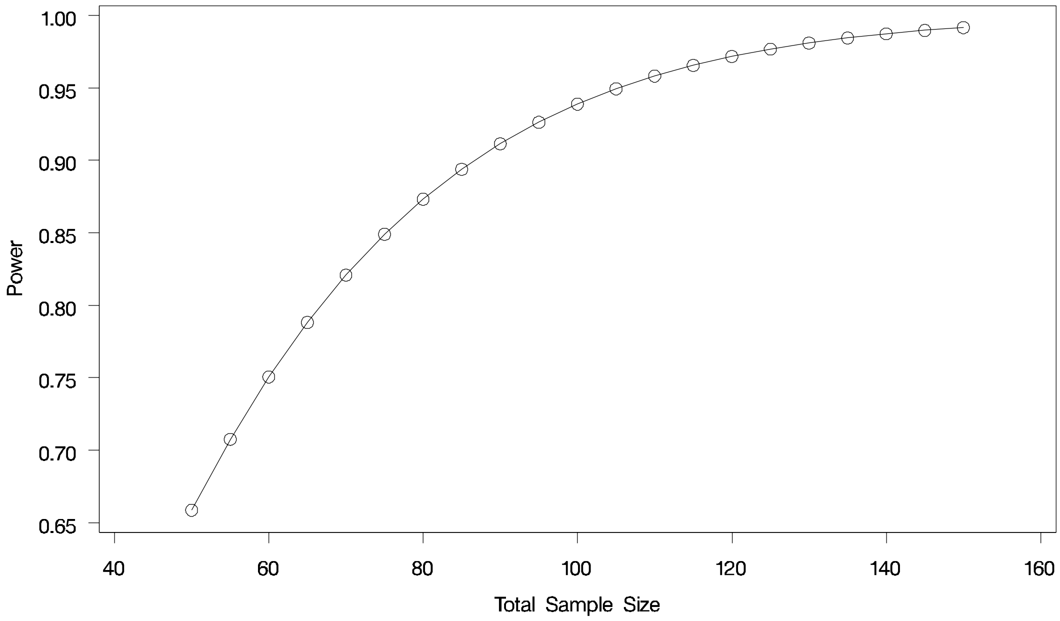
The POWER=. option identifies power as the parameter to compute. The NFULLPREDICTORS= option specifies 7 total predictors (not including the intercept), and the NTESTPREDICTORS= option indicates that 1 of those predictors is being tested. The PARTIALCORR= and NTOTAL= options specify the partial correlation and sample size, respectively. The default value for the ALPHA= option sets the significance level to 0.05. The X=N option in the plot statement requests a plot of sample size on the x-axis, and the MIN= and MAX= options specify the sample size range.

Output 57.5.1 shows the output, and Output 57.5.2 shows the plot.

Output 57.5.1. Power Analysis for Multiple Regression

```
                          The POWER Procedure
                 Type III F Test in Multiple Regression

                        Fixed Scenario Elements

        Method                                          Exact
        Model                                        Random X
        Number of Predictors in Full Model                  7
        Number of Test Predictors                           1
        Partial Correlation                              0.35
        Total Sample Size                                 100
        Alpha                                            0.05


                             Computed Power

                                 Power

                                 0.939
```

Output 57.5.2. Plot of Power versus Sample Size for Multiple Regression



For the sample size N = 100, the study is almost balanced with respect to Type 1 and Type 2 error rates, with α = 0.05 and β = 1 - 0.937 = 0.063. The study thus seems well designed at this sample size.

Now suppose that in a follow-up meeting with the cardiologists, you discover that their specific intent is to demonstrate that the (partial) correlation between PBI and tHcy is greater than 0.2. You suggest changing the planned data analysis to a 1-sided Fisher's z test with a null correlation of 0.2. The following statements perform a power analysis for this test:

```
proc power;
   onecorr dist=fisherz
      npvars = 6
      corr = 0.35
      nullcorr = 0.2
      sides = 1
      ntotal = 100
      power = .;
run;
```

The DIST=FISHERZ option in the ONECORR statement specifies Fisher's z test. The NPVARS= option specifies that 6 additional variables are adjusted for in the partial correlation. The CORR= option specifies the conjectured correlation of 0.35, and the NULLCORR= option indicates the null value of 0.2. The SIDES= option specifies a 1-sided test.

Output 57.5.3 shows the output.

Output 57.5.3. Power Analysis for Fisher’s z Test

```
                             The POWER Procedure
                    Fisher’s z Test for Pearson Correlation

                           Fixed Scenario Elements

      Distribution                          Fisher’s z transformation of r
      Method                                          Normal approximation
      Number of Sides                                                    1
      Null Correlation                                                 0.2
      Number of Variables Partialled Out                                 6
      Correlation                                                     0.35
      Total Sample Size                                                100
      Nominal Alpha                                                   0.05


                                Computed Power

                              Actual
                               Alpha    Power

                                0.05    0.466
```

The power for Fisher’s z test is less than 50%, the decrease being mostly due to the smaller effect size (relative to the null value). When asked for a recommendation for a new sample size goal, you compute the required sample size to achieve a power of 0.95 (to balance Type 1 and Type 2 errors) and 0.85 (a threshold deemed to be minimally acceptable to the team). The following statements perform the sample size determination:

```
proc power;
   onecorr dist=fisherz
      npvars = 6
      corr = 0.35
      nullcorr = 0.2
      sides = 1
      ntotal = .
      power = 0.85 0.95;
run;
```

The NTOTAL=. option identifies sample size as the parameter to compute, and the POWER= option specifies the target powers.

Output 57.5.4. Sample Size Determination for Fisher's z Test

```
                         The POWER Procedure
               Fisher's z Test for Pearson Correlation

                       Fixed Scenario Elements

      Distribution                        Fisher's z transformation of r
      Method                                        Normal approximation
      Number of Sides                                                  1
      Null Correlation                                               0.2
      Number of Variables Partialled Out                               6
      Correlation                                                   0.35
      Nominal Alpha                                                 0.05


                           Computed N Total

                        Nominal     Actual     Actual         N
                Index     Power      Alpha      Power     Total

                    1      0.85       0.05      0.850       280
                    2      0.95       0.05      0.950       417
```

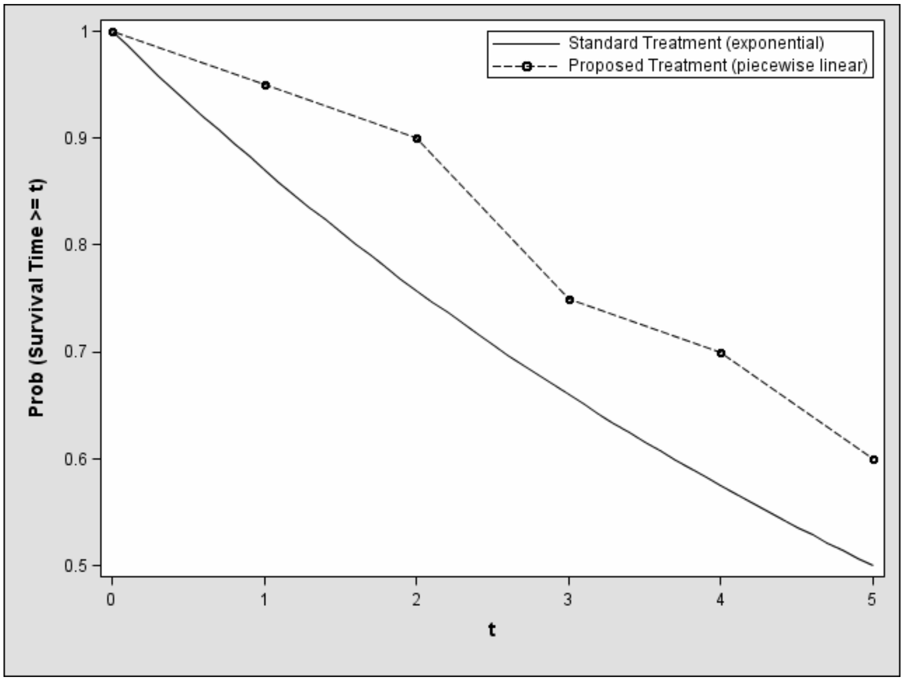
The results in Output 57.5.4 reveal a required sample size of 417 to achieve a power of 0.95 and 280 to achieve a power of 0.85.

Example 57.6. Comparing Two Survival Curves

You are consulting for a clinical research group planning a trial to compare survival rates for proposed and standard cancer treatments. The planned data analysis is a log-rank test to nonparametrically compare the overall survival curves for the two treatments. Your goal is to determine an appropriate sample size to achieve a power of 0.8 for a 2-sided test with α = 0.05 using a balanced design.

The survival curve for patients on the standard treatment is well-known to be approximately exponential with a median survival time of five years. The research group conjectures that the new proposed treatment will yield a (nonexponential) survival curve similar to the dashed line in Output 57.6.1. Patients will be accrued uniformly over two years and then followed for an additional three years past the accrual period. Some loss to follow-up is expected, with roughly exponential rates that would result in about 50% loss with the standard treatment within 10 years. The loss to follow-up with the proposed treatment is more difficult to predict, but 50% loss would expected to occur sometime between years 5 and 20.

Output 57.6.1. Survival Curves



Use the TWOSAMPLESURVIVAL statement with the TEST=LOGRANK option to compute the required sample size for the log-rank test. The following statements perform the analysis:

```
proc power;
   twosamplesurvival test=logrank
      curve("Standard") = 5 : 0.5
      curve("Proposed") = (1 to 5 by 1):(0.95 0.9 0.75 0.7 0.6)
      groupsurvival = "Standard" | "Proposed"
      accrualtime = 2
      followuptime = 3
      groupmedlosstimes = 10 | 20 5
      power = 0.8
      npergroup = .;
run;
```

The CURVE= option defines the two survival curves. The "Standard" curve has only one point, specifying an exponential form with a survival probability of 0.5 at year 5. The "Proposed" curve is a piecewise linear curve defined by the five points shown in Output 57.6.1. The GROUPSURVIVAL= option assigns the survival curves to the two groups, and the ACCRUALTIME= and FOLLOWUPTIME= options specify the accrual and follow-up times. The GROUPMEDLOSSTIMES= option specifies the years at which 50% loss is expected to occur. The POWER= option specifies the target power, and the NPERGROUP=. option identifies sample size per group as the parameter to compute. Default values for the SIDES= and ALPHA= options specify a 2-sided test with α = 0.05.

Output 57.6.2 shows the results.

Output 57.6.2. Sample Size Determination for Log-Rank Test

```
                           The POWER Procedure
                   Log-Rank Test for Two Survival Curves

                         Fixed Scenario Elements

           Method                           Lakatos normal approximation
           Accrual Time                                                2
           Follow-up Time                                              3
           Group 1 Survival Curve                               Standard
           Form of Survival Curve 1                          Exponential
           Group 2 Survival Curve                               Proposed
           Form of Survival Curve 2                     Piecewise Linear
           Group 1 Median Loss Time                                   10
           Nominal Power                                             0.8
           Number of Sides                                             2
           Number of Time Sub-Intervals                               12
           Alpha                                                    0.05


                           Computed N Per Group

                              Median
                                Loss     Actual     N Per
                     Index    Time 2      Power     Group

                         1        20      0.800       228
                         2         5      0.801       234
```

The required sample size per group to achieve a power of 0.8 is 228 if the median loss time is 20 years for the proposed treatment. Only six more patients are required in each group if the median loss time is as short as five years.

Example 57.7. Confidence Interval Precision

An investment firm has hired you to help plan a study to estimate the success of a new investment strategy called “IntuiVest.” The study involves complex simulations of market conditions over time, and it tracks the balance of a hypothetical brokerage account starting with $50,000. Each simulation is very expensive in terms of computing time. You are asked to determine an appropriate number of simulations to estimate the average change in the account balance at the end of three years. The goal is to have a 95% chance of obtaining a 90% confidence interval whose half-width is at most $1,000. That is, the firm wants to have a 95% chance of being able to correctly claim at the end of the study that “Our research shows with 90% confidence that IntuiVest yields a profit of $X +/- $1,000 at the end of three years on an initial investment of $50,000 (under simulated market conditions).”

The probability of achieving the desired precision (that is, a small interval width) can be calculated either unconditionally or conditionally given that the true mean is captured by the interval. You decide to use the conditional form, considering two of its advantages:

- The conditional probability is usually lower than the unconditional probability for the same sample size, meaning that the conditional form is generally conservative.
- The overall probability of achieving the desired precision *and* capturing the true mean is easily computed as the product of the half-width probability and the confidence level. In this case, the overall probability is $0.95 \times 0.9 = 0.855$.

Based on some initial simulations, you expect a standard deviation between $25,000 and $45,000 for the ending account balance. You will consider both of these values in the sample size analysis.

As mentioned in the "Overview of Power Concepts" section on page 3488, an analysis of confidence interval precision is analogous to a traditional power analysis, with "CI Half-Width" taking the place of effect size and "Prob(Width)" taking the place of power. In this example, the target CI Half-Width is 1000, and the desired Prob(Width) is 0.95.

In addition to computing sample sizes for a half-width of $1,000, you are asked to plot the required number of simulations for a range of half-widths between $500 and $2,000. Use the ONESAMPLEMEANS statement with the CI=T option to implement the sample size determination. The following statements perform the analysis:

```
proc power;
   onesamplemeans ci=t
      alpha = 0.1
      halfwidth = 1000
      stddev = 25000 45000
      probwidth = 0.95
      ntotal = .;
   plot x=effect min=500 max=2000;
run;
```

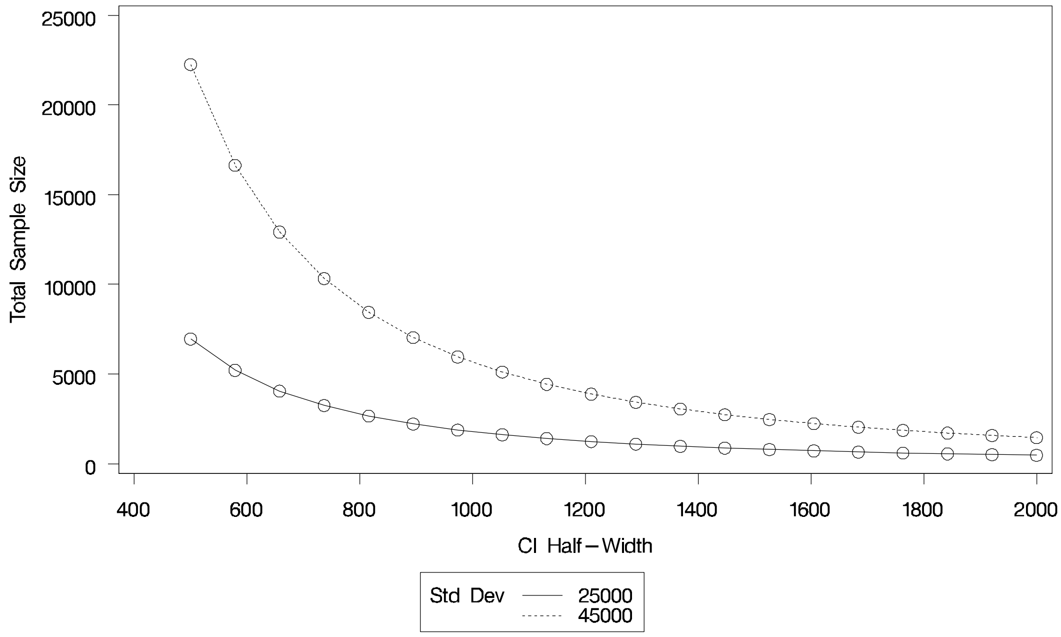
The NTOTAL=. option identifies sample size as the parameter to compute. The ALPHA=0.1 option specifies a confidence level of $1 - \alpha = 0.9$. The HALFWIDTH= option specifies the target half-width, and the STDDEV= option specifies the conjectured standard deviation values. The PROBWIDTH= option specifies the desired probability of achieving the target precision. The default value PROBTYPE=CONDITIONAL specifies that this probability is conditional on the true mean being captured by the interval. The default of SIDES=2 indicates a 2-sided interval.

Output 57.7.1 shows the output, and Output 57.7.2 shows the plot.

Output 57.7.1. Sample Size Determination for Confidence Interval Precision

```
                    The POWER Procedure
                Confidence Interval for Mean

                  Fixed Scenario Elements

            Distribution                  Normal
            Method                         Exact
            Alpha                            0.1
            CI Half-Width                   1000
            Nominal Prob(Width)             0.95
            Number of Sides                    2
            Prob Type                Conditional


                     Computed N Total

                                 Actual
                        Std        Prob          N
              Index     Dev     (Width)      Total

                  1   25000       0.951       1788
                  2   45000       0.950       5652
```

Output 57.7.2. Plot of Sample Size vs. Confidence Interval Half-Width



The number of simulations required in order to have a 95% chance of obtaining a half-width of at most 1000 is between 1788 and 5652, depending on the standard deviation. The plot reveals that over 20,000 simulations would be required for a half-width of 500 assuming the higher standard deviation.

Example 57.8. Customizing Plots

The example in this section demonstrates various ways you can modify and enhance plots:

- assigning analysis parameters to axes
- fine-tuning a sample size axis
- adding reference lines
- linking plot features to analysis parameters
- choosing key (legend) styles
- modifying symbol locations

The example plots are all based on a sample size analysis for a two-sample t test of group mean difference. You start by computing the sample size required to achieve a power of 0.9 using a 2-sided test with $\alpha = 0.05$, assuming the first mean is 12, the second mean is either 15 or 18, and the standard deviation is either 7 or 9.

Use the TWOSAMPLEMEANS statement with the TEST=DIFF option to compute the required sample sizes. Indicate total sample size as the result parameter by supplying a missing value (.) with the NTOTAL= option. Use the GROUPMEANS=, STDDEV=, and POWER= option to specify values of the other parameters. The following statements perform the sample size computations.

```
proc power;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
run;
```

Default values for the NULLDIFF=, SIDES=, GROUPWEIGHTS=, and DIST= options specify a null mean difference of 0, 2-sided test, balanced design, and assumption of normally distributed data.

Output 57.8.1 shows that the required sample size ranges from 60 to 382 depending on the unknown standard deviation and second mean.

Output 57.8.1. Computed Sample Sizes

```
                        The POWER Procedure
              Two-sample t Test for Mean Difference

                      Fixed Scenario Elements

                 Distribution                 Normal
                 Method                        Exact
                 Group 1 Mean                     12
                 Nominal Power                   0.9
                 Number of Sides                   2
                 Null Difference                   0
                 Alpha                          0.05
                 Group 1 Weight                    1
                 Group 2 Weight                    1


                         Computed N Total

                                Std     Actual         N
              Index     Mean2     Dev      Power     Total

                  1        15       7      0.902       232
                  2        15       9      0.901       382
                  3        18       7      0.904        60
                  4        18       9      0.904        98
```

Assigning Analysis Parameters to Axes

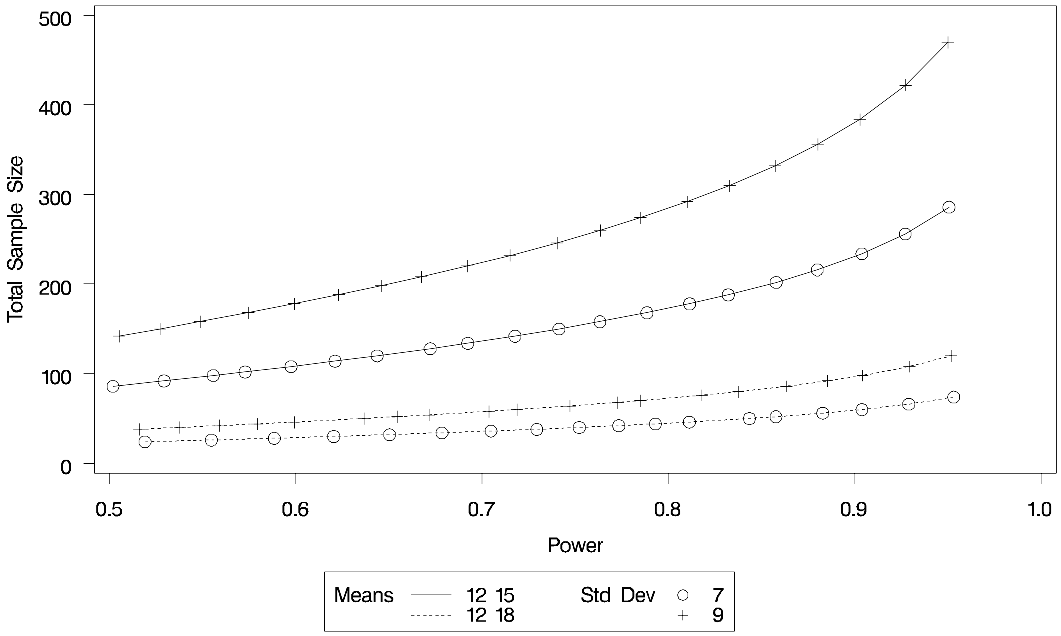
Use the PLOT statement to produce plots for all power and sample size analyses in PROC POWER. For the sample size analysis described at the beginning of this example, suppose you want to plot the required sample size on the y-axis against a range of powers between 0.5 and 0.95 on the x-axis. The X= and Y= options specify which parameter to plot against the result, and which axis to assign to this parameter. You can use either the X= or Y= option, but not both. Use the X=POWER option in PLOT statement to request a plot with power on the x-axis. The result parameter, here total sample size, is always plotted on the other axis. Use the MIN= and MAX= options to specify the range of the axis indicated with either the X= or the Y= option. Here, specify MIN=0.5 and MAX=0.95 to specify the power range. The following statements produce the plot.

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
   plot x=power min=0.5 max=0.95;
run;
```

Note that the value (0.9) of the POWER= option in the TWOSAMPLEMEANS statement is only a placeholder when the PLOTONLY option is used and both the MIN= and MAX= options are used, because the values of the MIN= and MAX= options override the value of 0.9. But the POWER= option itself is still required in the TWOSAMPLEMEANS statement, to provide a complete specification of the sample size analysis.

The resulting plot is shown in Output 57.8.2.

Output 57.8.2. Plot of Sample Size versus Power

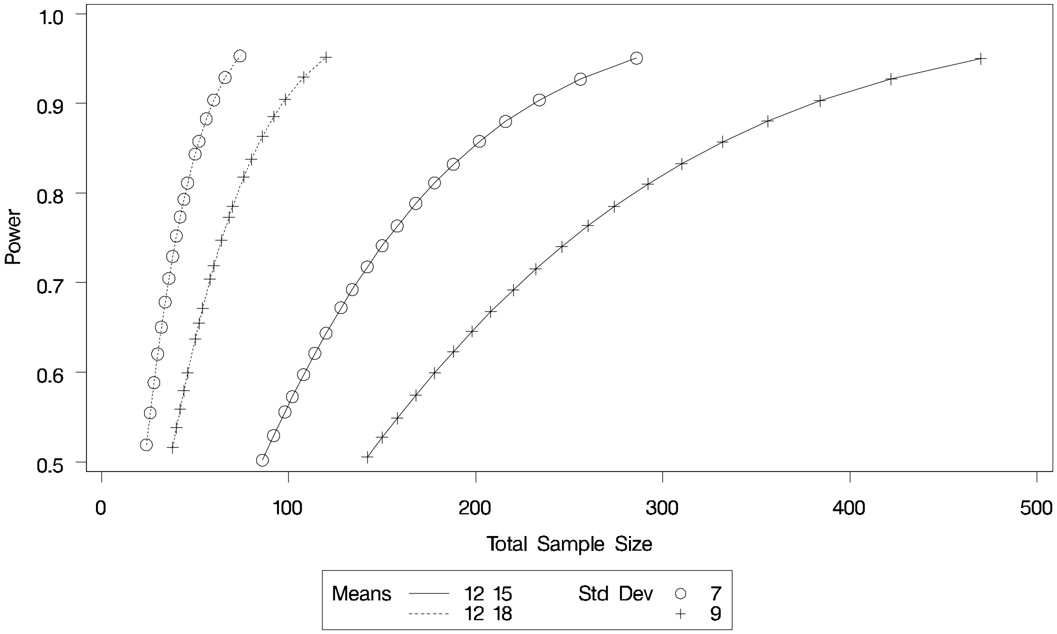


The line style identifies the group means scenario, and the plotting symbol identifies the standard deviation scenario. The locations of plotting symbols indicate computed sample sizes; the curves are linear interpolations of these points. By default, each curve consists of approximately 20 computed points (sometimes slightly more or less, depending on the analysis).

If you would rather plot power on the y-axis versus sample size on the x-axis, you have two general strategies to choose from. One strategy is to use the Y= option instead of the X= option in the PLOT statement:

```
plot y=power min=0.5 max=0.95;
```

Output 57.8.3. Plot of Power versus Sample Size using First Strategy



Note that the resulting plot (Output 57.8.3) is essentially a mirror image of Output 57.8.2. The axis ranges are set such that each curve in Output 57.8.3 contains similar values of Y instead of X. Each plotted point represents the computed value of the x-axis at the input value of the y-axis.

A second strategy for plotting power versus sample size (when originally solving for sample size) is to invert the analysis and base the plot on computed power for a given range of sample sizes. This strategy works well for monotonic power curves (as is the case for the t test and most other continuous analyses). It is advantageous in the sense of preserving the traditional role of the y-axis as the computed parameter. A common way to implement this strategy is

- Determine the range of sample sizes sufficient to cover at the desired power range for all curves (where each "curve" represents a scenario for standard deviation and second group mean).
- Use this range for the x-axis of a plot.

To determine the required sample sizes for target powers of 0.5 and 0.95, change the values in the POWER= option to reflect this range:

```
proc power;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.5 0.95
      ntotal       = .;
run;
```

Output 57.8.4 reveals that a sample size range of 24 to 470 is approximately sufficient to cover the desired power range of 0.5 to 0.95 for all curves (“approximately” because the actual power at the rounded sample size of 24 is slightly higher than the nominal power of 0.5).

Output 57.8.4. Computed Sample Sizes

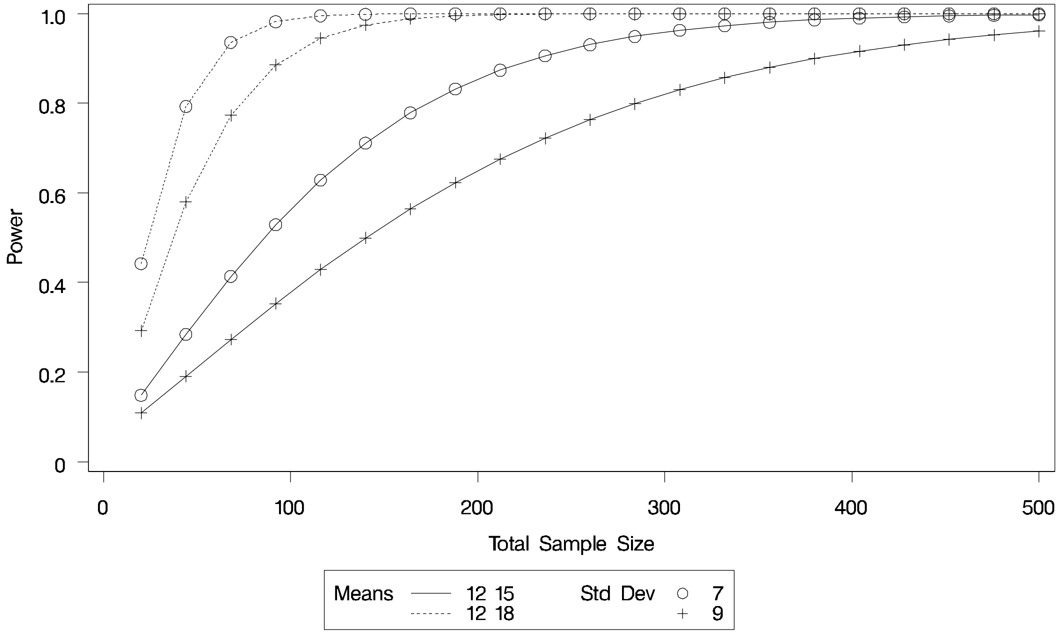
```
                         The POWER Procedure
                Two-sample t Test for Mean Difference

                      Fixed Scenario Elements

                   Distribution               Normal
                   Method                      Exact
                   Group 1 Mean                   12
                   Number of Sides                 2
                   Null Difference                 0
                   Alpha                        0.05
                   Group 1 Weight                  1
                   Group 2 Weight                  1


                          Computed N Total

                             Std     Nominal     Actual         N
           Index     Mean2   Dev       Power      Power     Total

               1        15     7        0.50      0.502        86
               2        15     7        0.95      0.951       286
               3        15     9        0.50      0.505       142
               4        15     9        0.95      0.950       470
               5        18     7        0.50      0.519        24
               6        18     7        0.95      0.953        74
               7        18     9        0.50      0.516        38
               8        18     9        0.95      0.952       120
```

To plot power on the y-axis for sample sizes between 20 and 500, use the X=N option in the PLOT statement with MIN=20 and MAX=500:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 200;
   plot x=n min=20 max=500;
run;
```

Each curve in the resulting plot in Output 57.8.5 covers at least a power range of 0.5 to 0.95.

Output 57.8.5. Plot of Power versus Sample Size Using Second Strategy

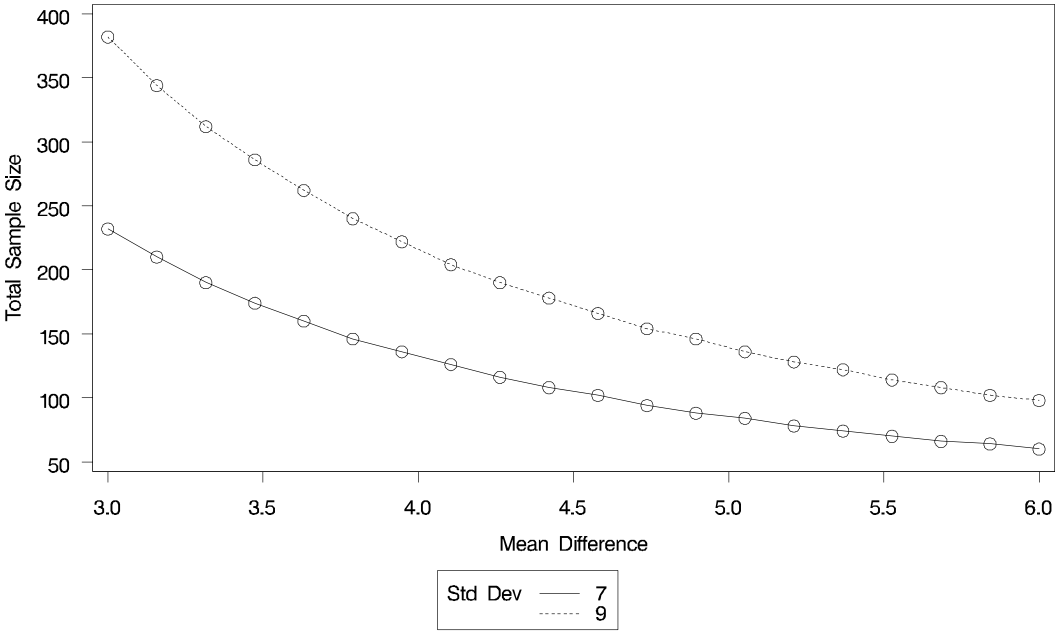


Finally, suppose you want to produce a plot of sample size versus effect size for a power of 0.9. In this case, the “effect size” is defined to be the mean difference. You need to reparameterize the analysis by using the MEANDIFF= option instead of the GROUPMEANS= option to produce a plot, since each plot axis must be represented by a scalar parameter. Use the X=EFFECT option in the PLOT statement to assign the mean difference to the x-axis. The following statements produce a plot of required sample size to detect mean differences between 3 and 6.

```
proc power plotonly;
   twosamplemeans test=diff
      meandiff     = 3 6
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
   plot x=effect min=3 max=6;
run;
```

The resulting plot Output 57.8.6 shows how the required sample size decreases with increasing mean difference.

Output 57.8.6. Plot of Sample Size versus Mean Difference



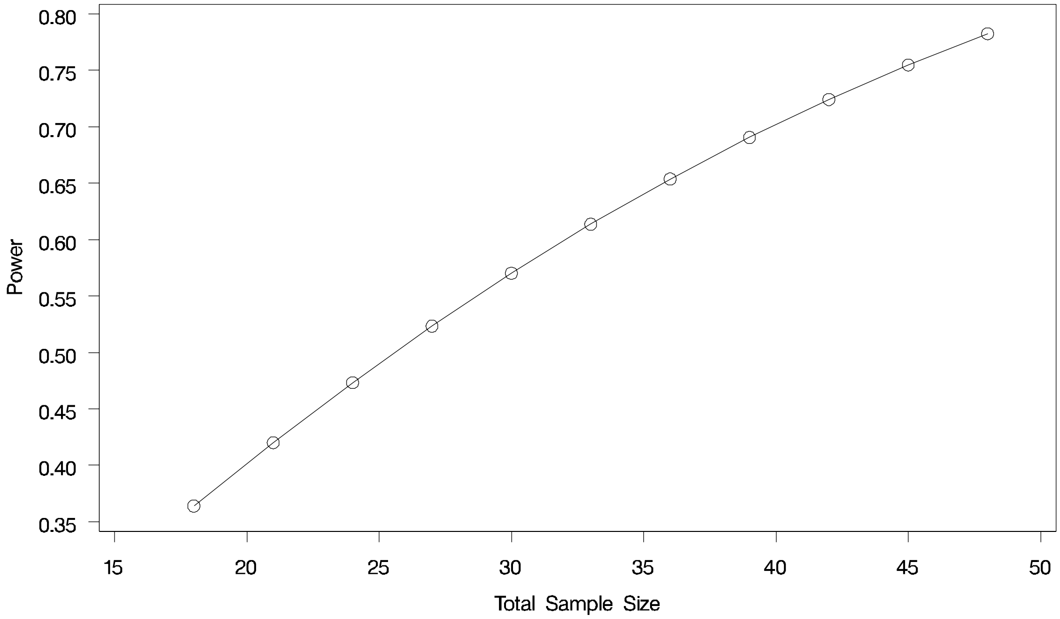
Fine-Tuning a Sample Size Axis

Consider the following plot request for a sample size analysis similar to the one in Output 57.8.1 but with only a single scenario, and with unbalanced sample size allocation of 2:1.

```
proc power plotonly;
   ods output plotcontent=PlotData;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=20;
run;
```

The MIN=, MAX=, and NPOINTS= options in the PLOT statement request a plot with 20 points between 20 and 50. But the resulting plot (Output 57.8.7) appears to have only 11 points, and they range from 18 to 48.

Output 57.8.7. Plot with Overlapping Points



The reason that this plot has fewer points than usual is due to the rounding of sample sizes. If you do not use the NFRACTIONAL option in the analysis statement (here, the TWOSAMPLEMEANS statement), then the set of sample size points determined by the MIN=, MAX=, NPOINTS=, and STEP= options in the PLOT statement may be rounded to satisfy the allocation weights. In this case, they are rounded down to the nearest multiples of 3 (the sum of the weights), and many of the points overlap. To see the overlap, you can print the NominalNTotal (unadjusted) and NTotal (rounded) variables in the PlotContent ODS object (here saved to a data set called PlotData):

```
proc print data=PlotData;
   var NominalNTotal NTotal;
run;
```

The output is shown in Output 57.8.8.

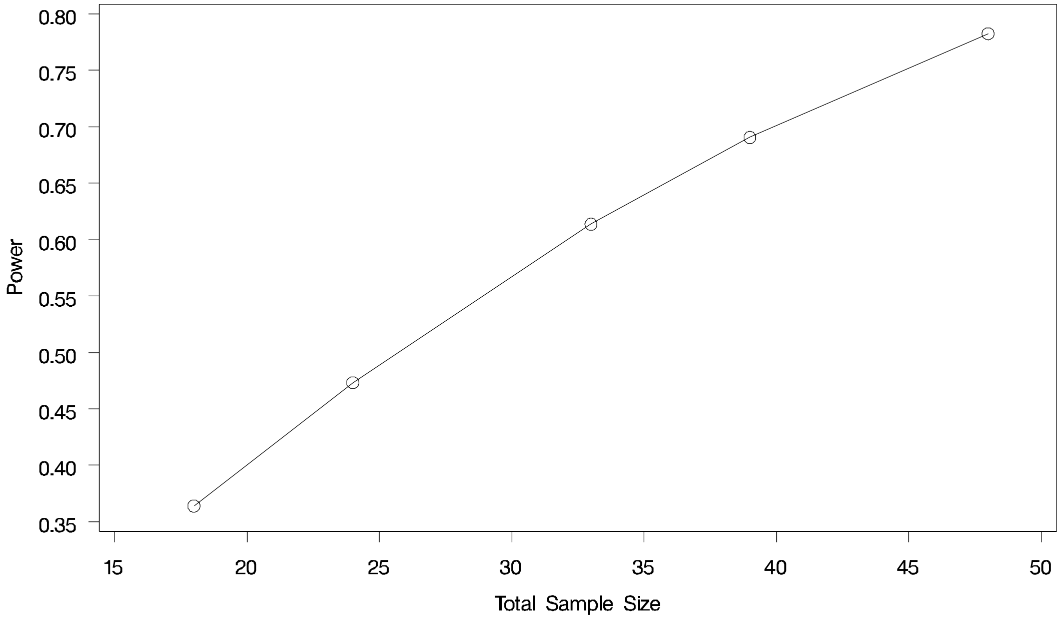
Output 57.8.8. Sample Sizes

```
                   Nominal
          Obs      NTotal      NTotal

            1       18.0         18
            2       19.6         18
            3       21.2         21
            4       22.7         21
            5       24.3         24
            6       25.9         24
            7       27.5         27
            8       29.1         27
            9       30.6         30
           10       32.2         30
           11       33.8         33
           12       35.4         33
           13       36.9         36
           14       38.5         36
           15       40.1         39
           16       41.7         39
           17       43.3         42
           18       44.8         42
           19       46.4         45
           20       48.0         48
```

Besides overlapping of sample size points, another peculiarity that might occur without the NFRACTIONAL option is unequal spacing; for example, in the plot in Output 57.8.9, created with the following statements.

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=5;
run;
```

Output 57.8.9. Plot with Unequally Spaced Points

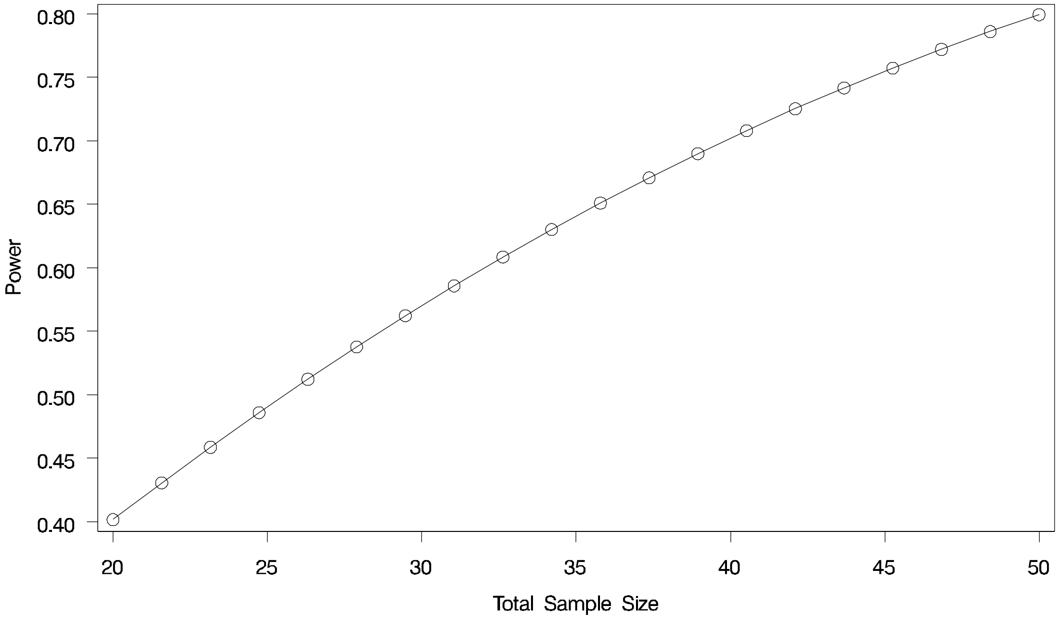


If you want to guarantee evenly spaced, nonoverlapping sample size points in your plots, you can either (1) use the NFRACTIONAL option in the analysis statement preceding the PLOT statement, or (2) use the STEP= option and provide values for the MIN=, MAX=, and STEP= options in the PLOT statement that are multiples of the sum of the allocation weights. Note that this sum is simply 1 for one-sample and paired designs and 2 for balanced two-sample designs. So, any integer step value works well for one-sample and paired designs, and any even step value works well for balanced two-sample designs. Both of these strategies will avoid rounding adjustments.

The following statements implement the first strategy to create the plot in Output 57.8.10, using the NFRACTIONAL option in the TWOSAMPLEMEANS statement.

```
proc power plotonly;
   twosamplemeans test=diff
      nfractional
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=20;
run;
```

Output 57.8.10. Plot with Fractional Sample Sizes



To implement the second strategy, use multiples of 3 for the STEP=, MIN=, and MAX= options in the PLOT statement (because the sum of the allocation weights is $2 + 1 = 3$). The following statements use STEP=3, MIN=18, and MAX=48 to create a plot that looks identical to Output 57.8.7 but suffers no overlapping of points.

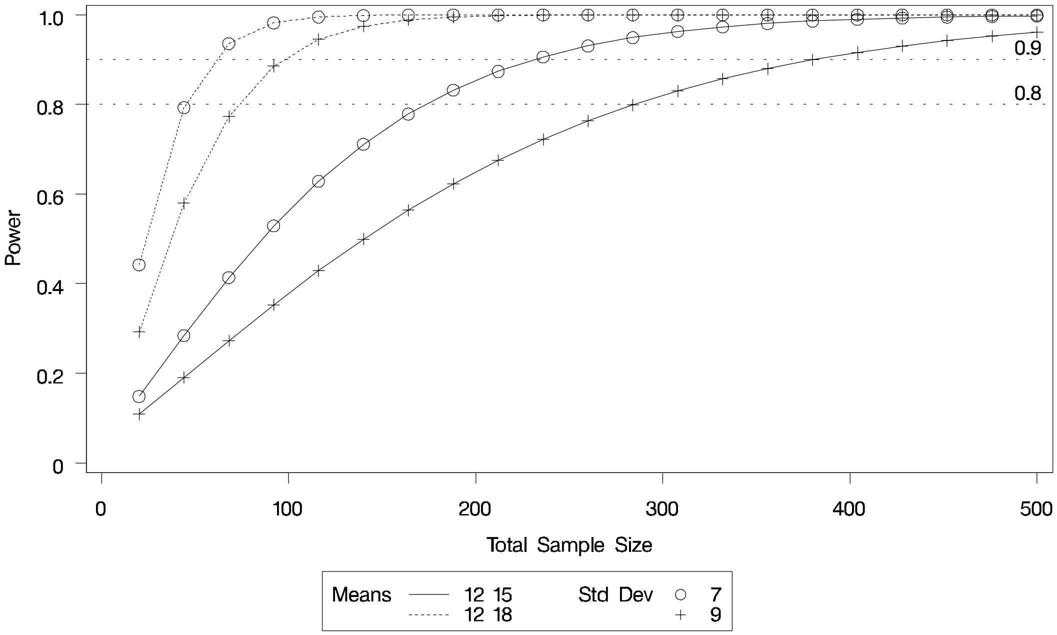
```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=18 max=48 step=3;
run;
```

Adding Reference Lines

Suppose you want to add reference lines to highlight power=0.8 and power=0.9 on the plot in Output 57.8.5. You can add simple reference lines using the YOPTS= option and REF= sub-option in the PLOT statement to produce Output 57.8.11, using the following statements.

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 100;
   plot x=n min=20 max=500
      yopts=(ref=0.8 0.9);
run;
```

Output 57.8.11. Plot with Simple Reference Lines on Y-Axis

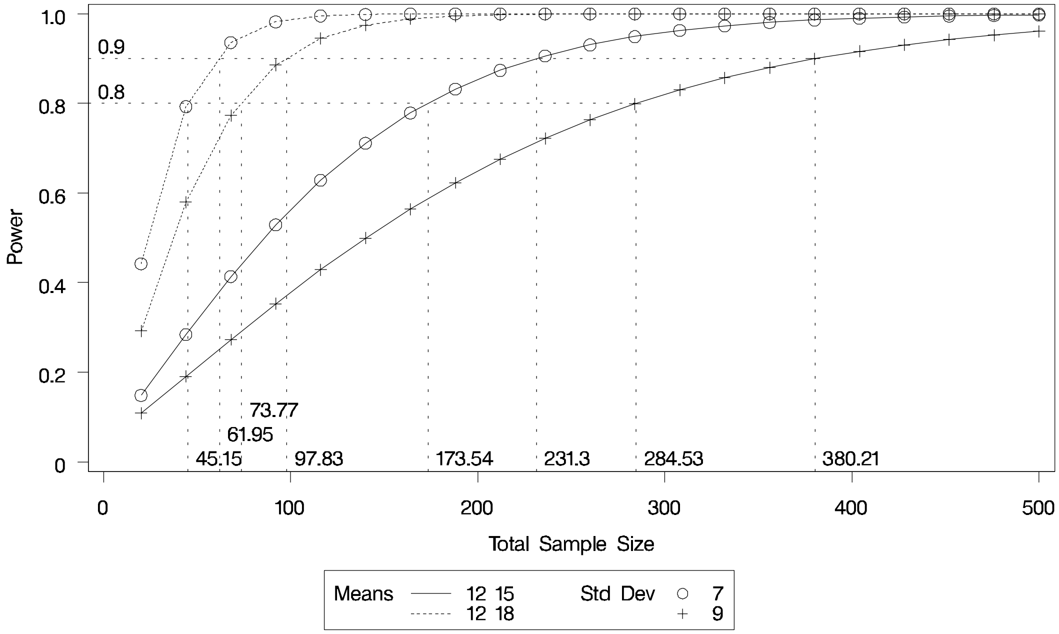


Or, you can specify CROSSREF=YES to add reference lines that intersect each curve and cross over to the other axis:

```
   plot x=n min=20 max=500
      yopts=(ref=0.8 0.9 crossref=yes);
```

The resulting plot is shown in Output 57.8.12.

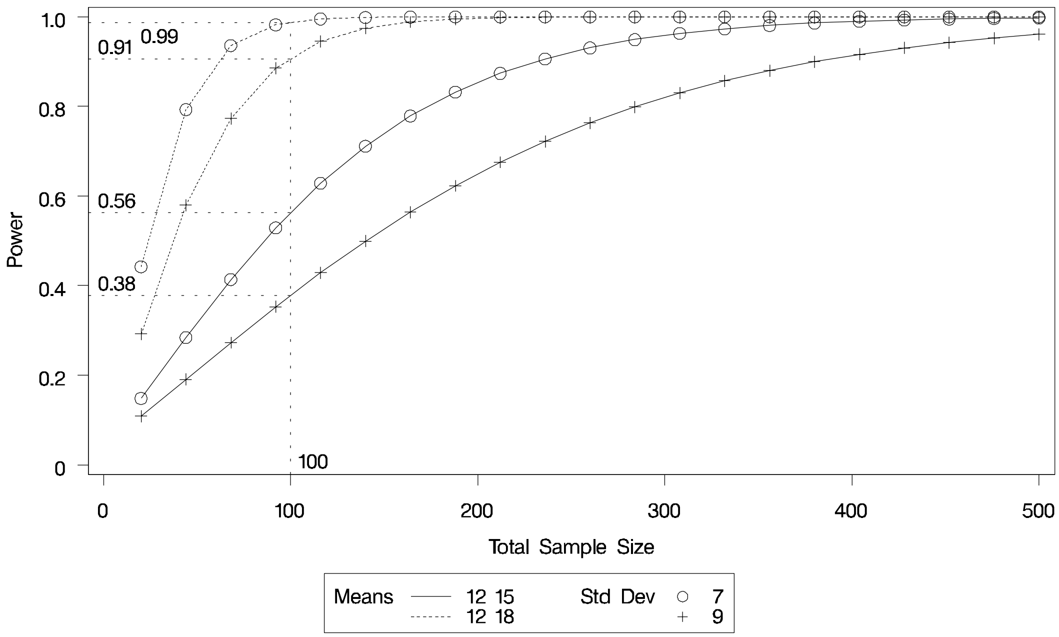
Output 57.8.12. Plot with CROSSREF=YES Style Reference Lines from Y-Axis



You can also add reference lines for the x-axis by using the XOPTS= option instead of the YOPTS= option. For example, the following plot statement produces Output 57.8.13, which has crossing reference lines highlighting the sample size of 100.

```
plot x=n min=20 max=500
   xopts=(ref=100 crossref=yes);
```

Output 57.8.13. Plot with CROSSREF=YES Style Reference Lines from X-Axis



Linking Plot Features to Analysis Parameters

You can use the VARY option in the PLOT statement to specify which of the following features you wish to associate with analysis parameters.

- line style
- plotting symbol
- color
- panel

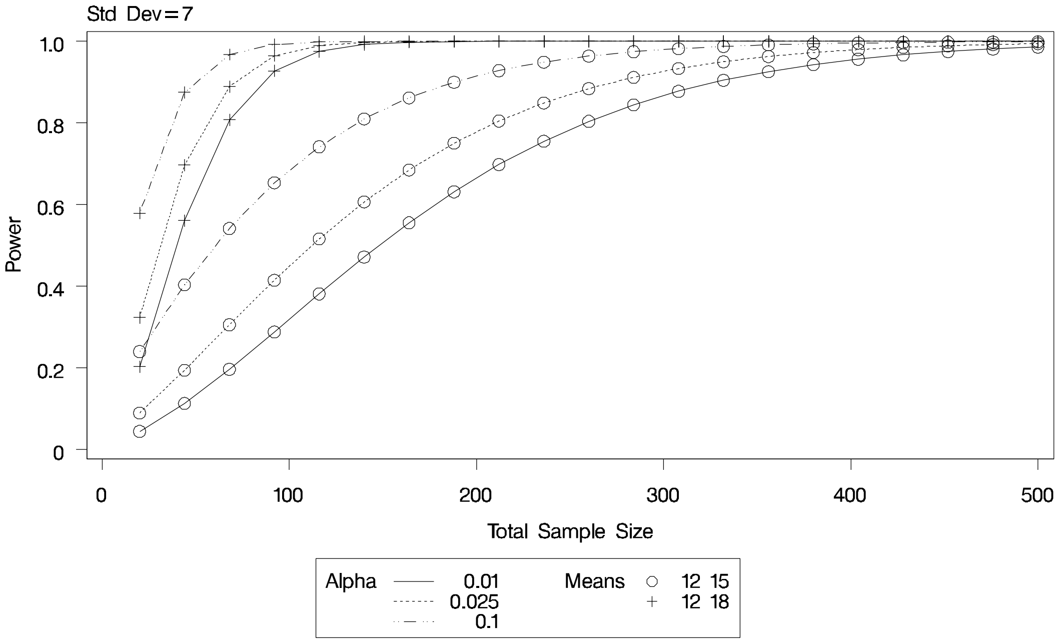
You can specify mappings between each of these features and one or more analysis parameters, or you can simply choose a subset of these features to use (and rely on default settings to associate these features with multiple-valued analysis parameters).

Suppose you supplement the sample size analysis in Output 57.8.5 to include three values of alpha, using the following statements.

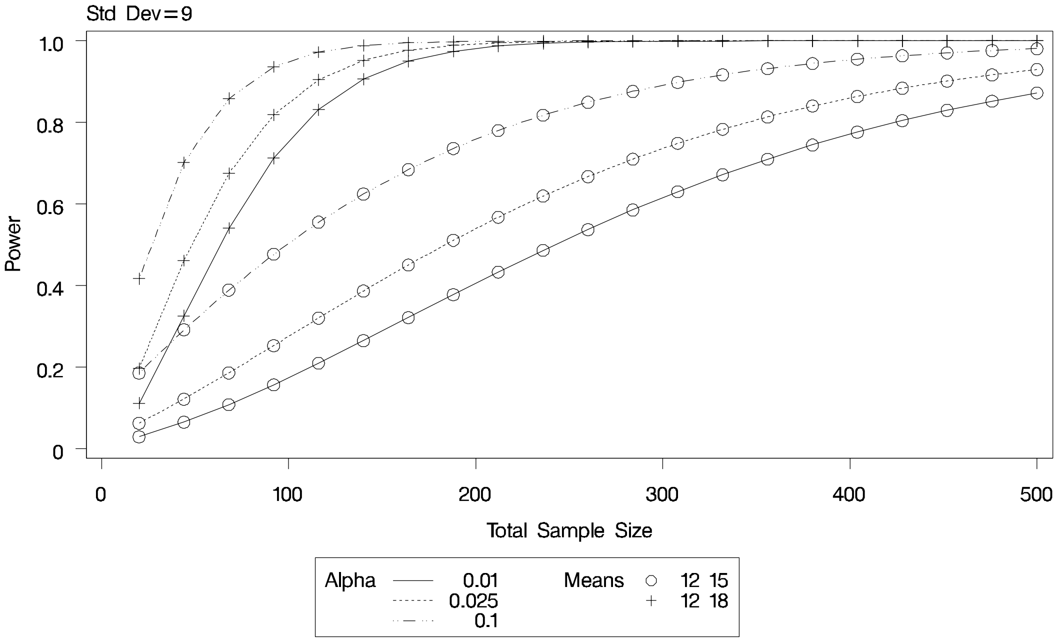
```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      alpha        = 0.01 0.025 0.1
      power        = .
      ntotal       = 100;
   plot x=n min=20 max=500;
run;
```

The defaults for the VARY option in the PLOT statement specify line style varying by the ALPHA= parameter, plotting symbol varying by the GROUPMEANS= parameter, panel varying by the STDDEV= parameter, and color remaining constant. The resulting plot, consisting of two panels, is shown in Output 57.8.14 and Output 57.8.15.

Output 57.8.14. Plot with Default VARY Settings: Panel 1 of 2



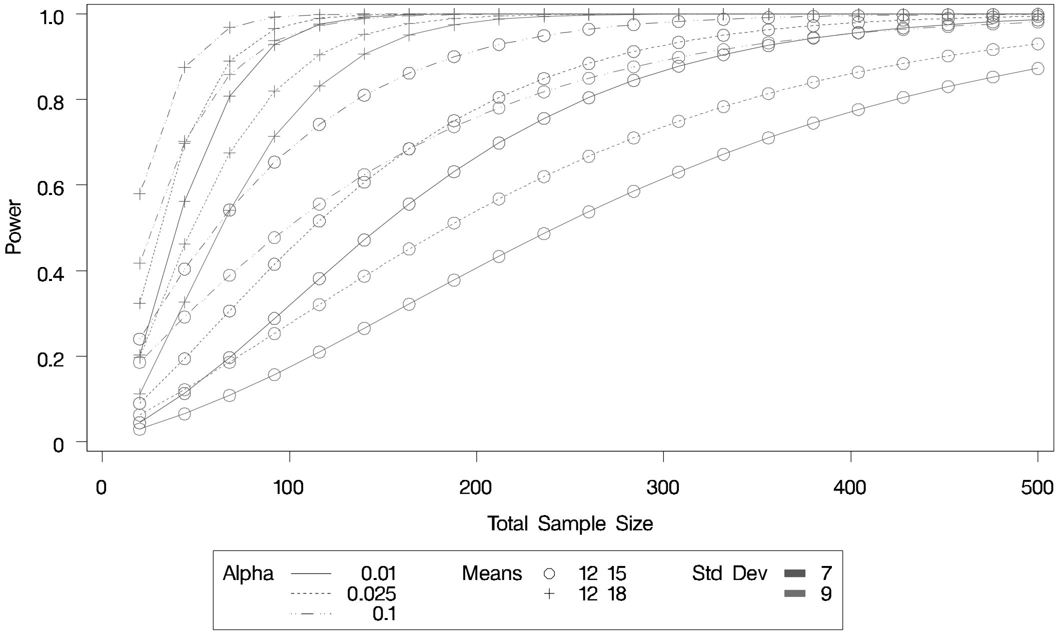
Output 57.8.15. Plot with Default VARY Settings: Panel 2 of 2



Suppose you want to produce a plot with only one panel that varies color in addition to line style and plotting symbol. Include the LINESTYLE, SYMBOL, and COLOR keywords in the VARY option in the PLOT statement, as follows, to produce the plot in Output 57.8.16.

```
plot x=n min=20 max=500
   vary (linestyle, symbol, color);
```

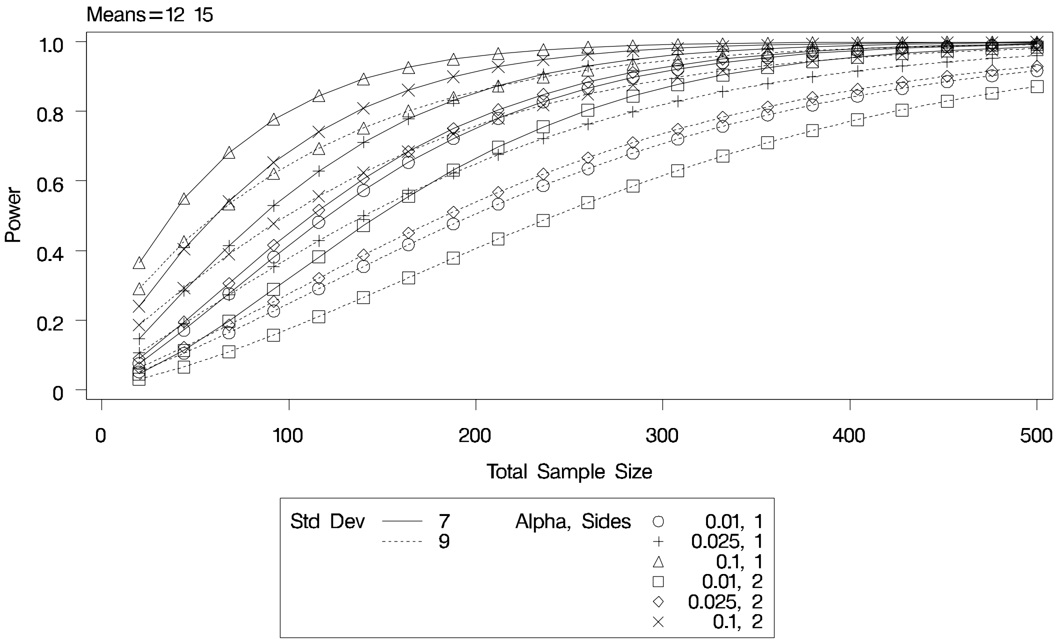
Output 57.8.16. Plot with Varying Color Instead of Panel



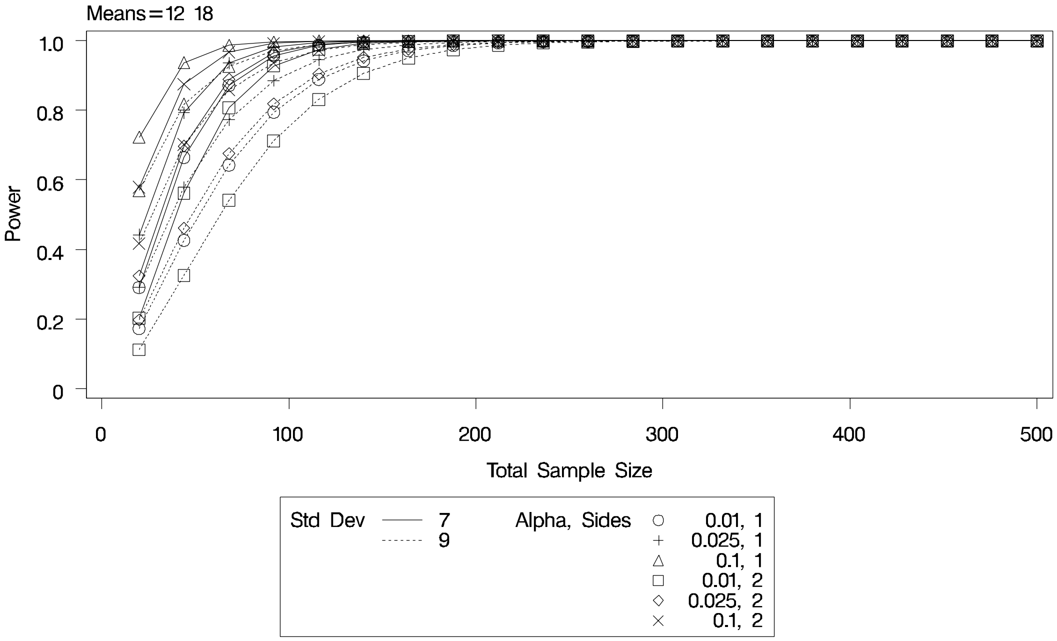
Finally, suppose you want to specify which features are used *and* which analysis parameters they are linked to. The following PLOT statement produces a two-panel plot (shown in Output 57.8.17 and Output 57.8.18) in which line style varies by standard deviation, plotting symbol varies by both alpha and sides, and panel varies by means.

```
plot x=n min=20 max=500
   vary (linestyle by stddev,
         symbol by alpha sides,
         panel by groupmeans);
```

Output 57.8.17. Plot with Features Explicitly Linked to Parameters: Panel 1 of 2



Output 57.8.18. Plot with Features Explicitly Linked to Parameters: Panel 2 of 2



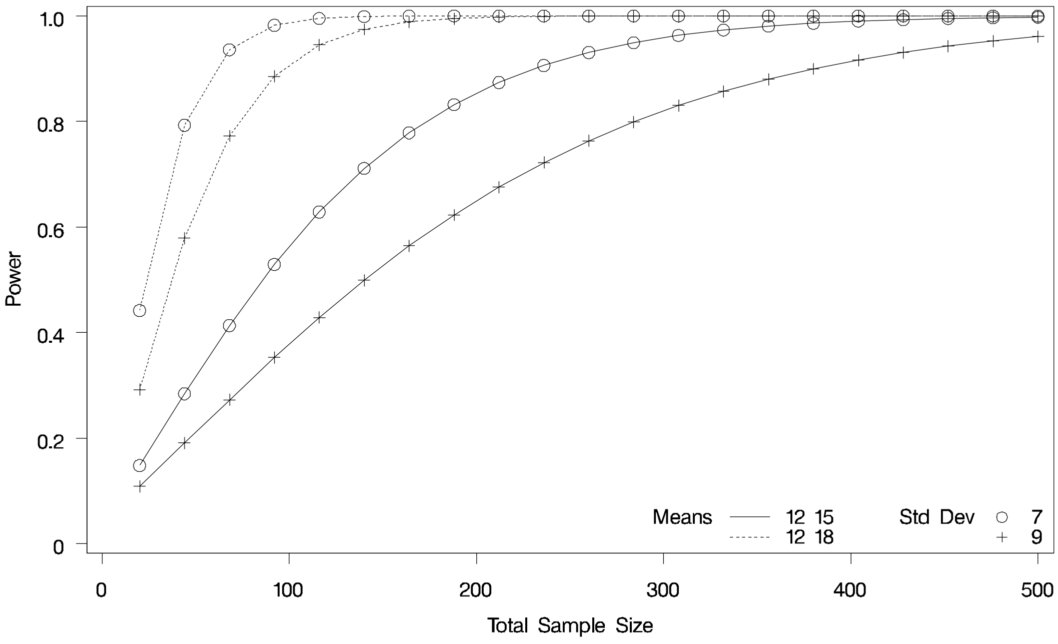
Choosing Key (Legend) Styles

The default style for the key (or "legend") is one that displays the association between levels of features and levels of analysis parameters, located below the x-axis. For example, Output 57.8.5 demonstrates this style of key.

You can reproduce Output 57.8.5 with the same key but a different location, inside the plotting region, using the POS=INSET option within the KEY=BYFEATURE option in the PLOT statement. The following statements product the plot in Output 57.8.19.

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 200;
   plot x=n min=20 max=500
      key = byfeature(pos=inset);
run;
```

Output 57.8.19. Plot with a By-Feature Key Inside the Plotting Region

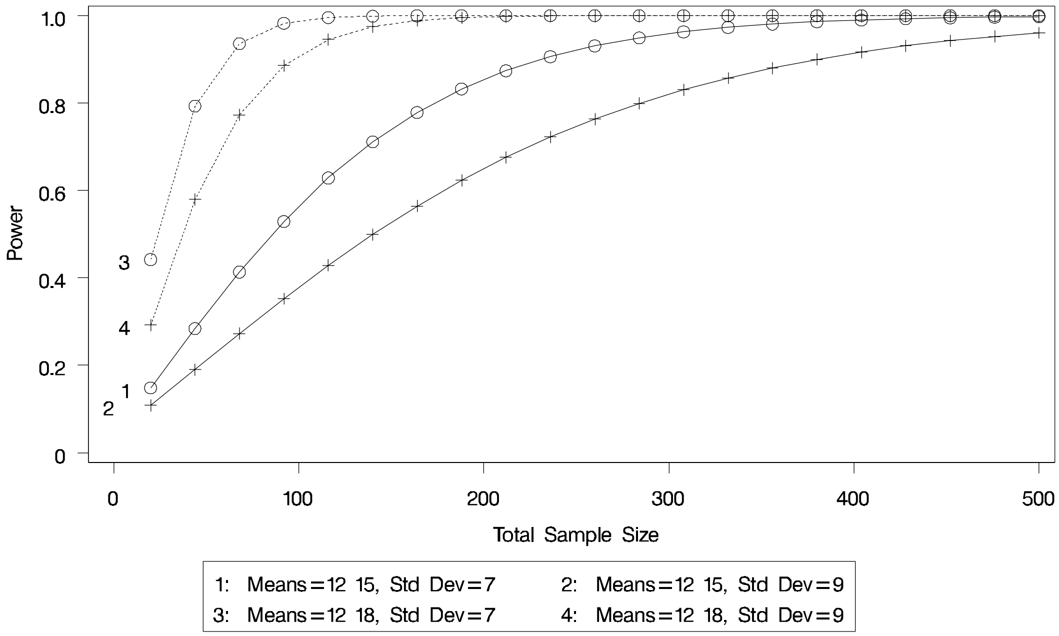


Alternatively, you can specify a key that identifies each individual curve separately by number using the KEY=BYCURVE option in the PLOT statement:

```
   plot x=n min=20 max=500
      key = bycurve;
```

The resulting plot is shown in Output 57.8.20.

Output 57.8.20. Plot with a Numbered By-Curve Key

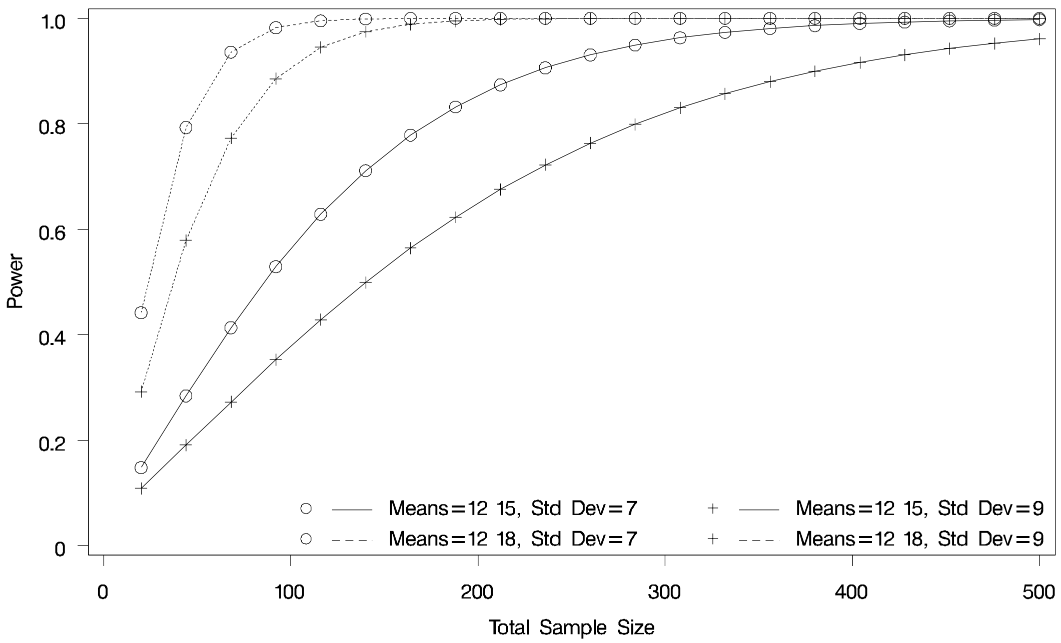


Use the NUMBERS=OFF option within the KEY=BYCURVE option to specify a nonnumbered key that identifies curves with samples of line styles, symbols, and colors:

```
plot x=n min=20 max=500
   key = bycurve(numbers=off pos=inset);
```

The POS=INSET suboption places the key within the plotting region. The resulting plot is shown in Output 57.8.21.

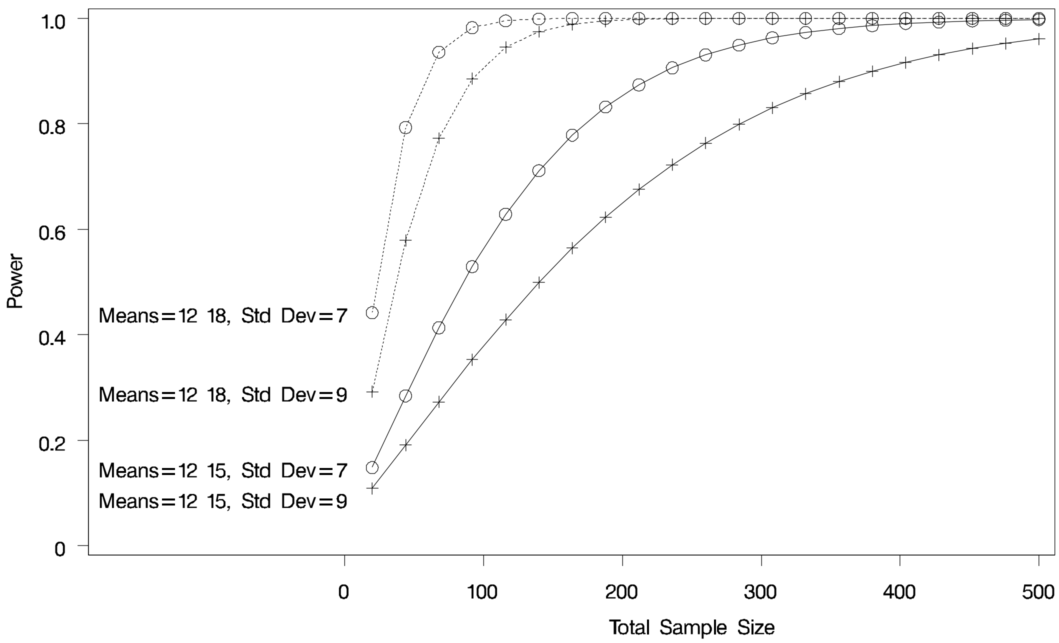
Output 57.8.21. Plot with a Nonnumbered By-Curve Key



Finally, you can attach labels directly to curves with the KEY=ONCURVES option. The following plot statement produces Output 57.8.22.

```
plot x=n min=20 max=500
   key = oncurves;
```

Output 57.8.22. Plot with Directly Labeled Curves



Modifying Symbol Locations

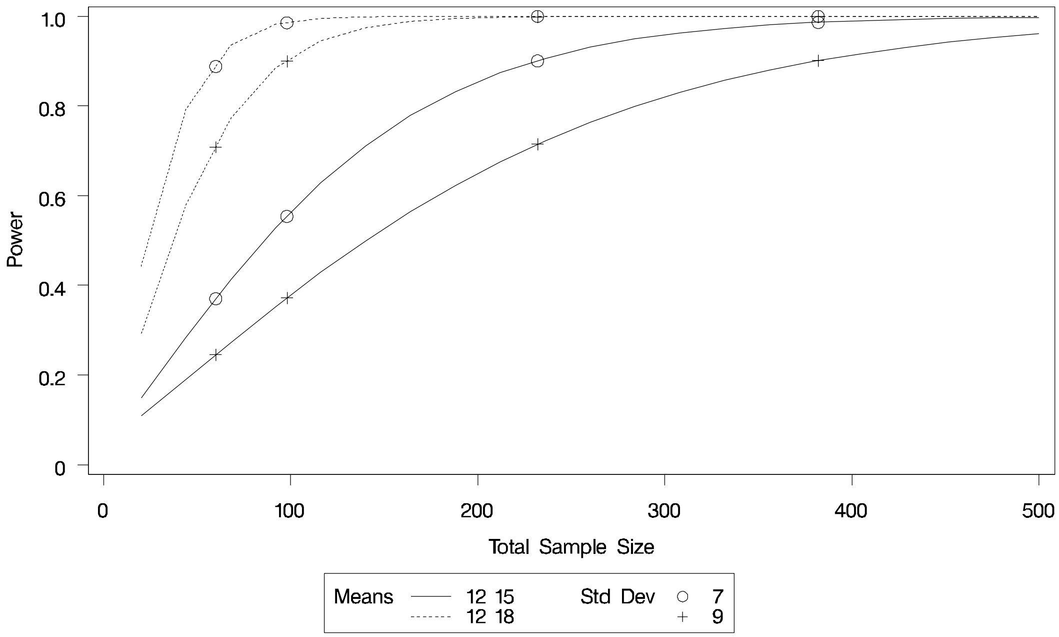
The default locations for plotting symbols are the points computed directly from the power and sample size algorithms. For example, Output 57.8.5 shows plotting symbols corresponding to computed points. The curves connecting these points are interpolated (as indicated by the INTERPOL= option in the PLOT statement).

You can modify the locations of plotting symbols using the MARKERS= option in the plot statement. The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the input specified in the analysis statement preceding the PLOT statement. You may prefer this as an alternative to using reference lines to highlight specific points. For example, you can reproduce Output 57.8.5, but with the plotting symbols located at the sample sizes shown in Output 57.8.1, using the following statements.

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 232 382 60 98;
   plot x=n min=20 max=500
      markers=analysis;
run;
```

The analysis statement here is the TWOSAMPLEMEANS statement. The MARKERS=ANALYSIS option in the PLOT statement causes the plotting symbols to occur at sample sizes specified by the NTOTAL= option in the TWOSAMPLEMEANS statement: 232, 382, 60, and 98. The resulting plot is shown in Output 57.8.23.

Output 57.8.23. Plot with MARKERS=ANALYSIS

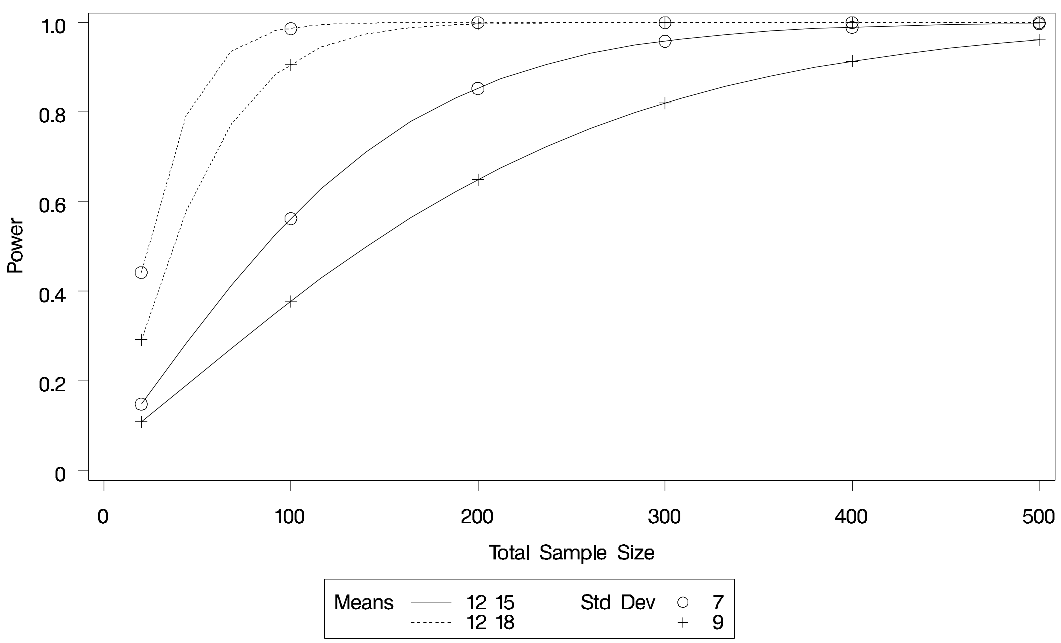


You can also use the MARKERS=NICE option to align symbols with the tick marks on one of the axes (the x-axis when the X= option is used, or the y-axis when the Y= is used):

```
plot x=n min=20 max=500
   markers=nice;
```

The plot created by this PLOT statement is shown in Output 57.8.24.

Output 57.8.24. Plot with MARKERS=NICE



Note that the plotting symbols are aligned with the tick marks on the x-axis because the X= option is specified.

References

Anderson, T.W. (1984), *An Introduction to Multivariate Statistical Analysis*, Second Edition, New York: John Wiley & Sons.

Beal, S.L. (1989), “Sample Size Determination for Confidence Intervals on the Population Means and on the Difference between Two Population Means,” *Biometrics*, 45, 969–977.

Blackwelder, W.C. (1982), “‘Proving the Null Hypothesis’ in Clinical Trials,” *Controlled Clinical Trials*, 3, 345–353.

Cantor, A. B. (1997), *Extending SAS® Survival Analysis Techniques for Medical Research*, Cary, NC: SAS Institute Inc.

Castelloe, J.M. (2000), “Sample Size Computations and Power Analysis with the SAS ® System,” *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference*, Paper 265-25, Cary, NC: SAS Institute Inc.

Castelloe, J.M. and O’Brien, R.G. (2001), “Power and Sample Size Determination for Linear Models,” *Proceedings of the Twenty-sixth Annual SAS Users Group International Conference*, Paper 240-26. Cary, NC: SAS Institute Inc.

Chernick, M.R. and Liu, C.Y. (2002), “The Saw-Toothed Behavior of Power Versus Sample Size and Software Solutions: Single Binomial Proportion Using Exact Methods,” *The American Statistician*, 56, 149–155.

Connor, R.J. (1987), “Sample Size for Testing Differences in Proportions for the Paired-Sample Design,” *Biometrics*, 43, 207–211.

Diegert, C. and Diegert, K.V. (1981), “Note on Inversion of Casagrande-Pike-Smith Approximate Sample-Size Formula for Fisher-Irwin Test on 2 X 2 Tables,” *Biometrics*, 37, 595.

Diletti, D., Hauschke, D., and Steinijans, V.W. (1991), “Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals,” *International Journal of Clinical Pharmacology, Therapy and Toxicology*, 29, 1–8.

DiSantostefano, R.L. and Muller, K.E. (1995), “A Comparison of Power Approximations for Satterthwaite’s Test,” *Communications in Statistics — Simulation and Computation*, 24 (3), 583–593.

Fisher, R.A. (1921), “On the ‘Probable Error’ of a Coefficient of Correlation Deduced from a Small Sample,” *Metron*, 1, 3–32.

Fleiss, J.L., Tytun, A. and Ury, H.K. (1980), “A Simple Approximation for Calculating Sample Sizes for Comparing Independent Proportions,” *Biometrics*, 36, 343–346.

Gatsonis, C. and Sampson, A.R. (1989), “Multiple Correlation: Exact Power and Sample Size Calculations,” *Psychological Bulletin*, 106, 516–524.

Hocking, R.R. (1985), *The Analysis of Linear Models*, Monterey, CA: Brooks/Cole Publishing Company.

Johnson, N.L. and Kotz, S. (1970), *Distributions in Statistics: Continuous Univariate Distributions — 1*, New York: John Wiley & Sons.

Johnson, N.L. Kotz, S. and Balakrishnan, N. (1995), *Continuous Univariate Distributions, Volume 2*, Second Edition, New York: John Wiley & Sons.

Jones R.M. and Miller, K.S. (1966), “On the Multivariate Lognormal Distribution,” *Journal of Industrial Mathematics*, 16, 63–76.

Lachin, J.M. (1992), “Power and Sample Size Evaluation for the McNemar Test with Application to Matched Case-Control Studies,” *Statistics in Medicine*, 11, 1239–1251.

Lakatos, E. (1988), “Sample Sizes Based on the Log-Rank Statistic in Complex Clinical Trials,” *Biometrics*, 44, 229–241.

Lenth, R.V. (2001), “Some Practical Guidelines for Effective Sample Size Determination,” *The American Statistician*, 55, 187–193.

Maxwell, S.E. (2000), “Sample Size and Multiple Regression Analysis,” *Psychological Methods*, 5, 434–458.

Miettinen, O.S. (1968), “The Matched Pairs Design in the Case of All-or-None Responses,” *Biometrics*, 339–352.

Moser, B.K., Stevens, G.R., and Watts, C.L. (1989), "The Two-Sample T Test Versus Satterthwaite's Approximate F Test," *Communications in Statistics A — Theory and Methods*, 18, 3963–3975.

Muller, K.E. and Benignus, V.A. (1992), "Increasing Scientific Power with Statistical Power," *Neurotoxicology and Teratology*, 14, 211–219.

O'Brien, R.G. and Muller, K.E. (1993), "Unified Power Analysis for *t*-Tests Through Multivariate Hypotheses," in *Applied Analysis of Variance in Behavioral Science*, ed. L.K. Edwards, New York: Marcel Dekker, Chapter 8, 297–344.

Owen, D.B. (1965), "A Special Case of a Bivariate Non-Central *t*-Distribution," *Biometrika*, 52, 437–446.

Pagano, M. and Gauvreau, K. (1993), *Principles of Biostatistics*, Belmont, CA: Wadsworth, Inc.

Phillips, K.F. (1990), "Power of the Two One-Sided Tests Procedure in Bioequivalence," *Journal of Pharmacokinetics and Biopharmaceutics*, 18, 137–144.

Satterthwaite, F.W. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.

Schork, M. and Williams, G. (1980), "Number of Observations Required for the Comparison of Two Correlated Proportions," *Communications in Statistics–Simulation and Computation 9,* 349–357.

Schuirmann, D.J. (1987), "A Comparison of the Two One-Sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability," *Journal of Pharmacokinetics and Biopharmaceutics*, 15, 657–680.

Senn, S. (1993), *Cross-over Trials in Clinical Research*, New York: John Wiley & Sons, Inc.

Stuart, A. and Ord, J.K. (1994), *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory*, Sixth Edition, Baltimore: Edward Arnold Publishers Ltd.

Walters, D.E. (1979). "In Defence of the Arc Sine Approximation," *The Statistician*, 28, 219–232.

Chapter 58
The PRINCOMP Procedure

Chapter Contents

Chapter 58
The PRINCOMP Procedure

Overview

The PRINCOMP procedure performs principal component analysis. As input you can use raw data, a correlation matrix, a covariance matrix, or a sums of squares and crossproducts (SSCP) matrix. You can create output data sets containing eigenvalues, eigenvectors, and standardized or unstandardized principal component scores.

Principal component analysis is a multivariate technique for examining relationships among several quantitative variables. The choice between using factor analysis and principal component analysis depends in part upon your research objectives. You should use the PRINCOMP procedure if you are interested in summarizing data and detecting linear relationships. Plots of principal components are especially valuable tools in exploratory data analysis. You can use principal components to reduce the number of variables in regression, clustering, and so on. See Chapter 5, "Introduction to Multivariate Procedures," for a detailed comparison of the PRINCOMP and FACTOR procedures.

Experimental graphics are now available with the PRINCOMP procedure. For more information, see the "ODS Graphics" section on page 3613.

Principal component analysis was originated by Pearson (1901) and later developed by Hotelling (1933). The application of principal components is discussed by Rao (1964), Cooley and Lohnes (1971), and Gnanadesikan (1977). Excellent statistical treatments of principal components are found in Kshirsagar (1972), Morrison (1976), and Mardia, Kent, and Bibby (1979).

Given a data set with p numeric variables, you can compute p principal components. Each principal component is a linear combination of the original variables, with coefficients equal to the eigenvectors of the correlation or covariance matrix. The eigenvectors are customarily taken with unit length. The principal components are sorted by descending order of the eigenvalues, which are equal to the variances of the components.

Principal components have a variety of useful properties (Rao 1964; Kshirsagar 1972):

- The eigenvectors are orthogonal, so the principal components represent jointly perpendicular directions through the space of the original variables.
- The principal component scores are jointly uncorrelated. Note that this property is quite distinct from the previous one.
- The first principal component has the largest variance of any unit-length linear combination of the observed variables. The jth principal component has the largest variance of any unit-length linear combination orthogonal to the first

$j-1$ principal components. The last principal component has the smallest variance of any linear combination of the original variables.

- The scores on the first j principal components have the highest possible generalized variance of any set of unit-length linear combinations of the original variables.
- The first j principal components provide a least-squares solution to the model

  $$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

  where $\mathbf{Y}$ is an $n \times p$ matrix of the centered observed variables; $\mathbf{X}$ is the $n \times j$ matrix of scores on the first j principal components; $\mathbf{B}$ is the $j \times p$ matrix of eigenvectors; $\mathbf{E}$ is an $n \times p$ matrix of residuals; and you want to minimize trace$(\mathbf{E}'\mathbf{E})$, the sum of all the squared elements in $\mathbf{E}$. In other words, the first j principal components are the best linear predictors of the original variables among all possible sets of j variables, although any nonsingular linear transformation of the first j principal components would provide equally good prediction. The same result is obtained if you want to minimize the determinant or the Euclidean (Schur, Frobenious) norm of $\mathbf{E}'\mathbf{E}$ rather than the trace.
- In geometric terms, the j-dimensional linear subspace spanned by the first j principal components provides the best possible fit to the data points as measured by the sum of squared perpendicular distances from each data point to the subspace. This is in contrast to the geometric interpretation of least squares regression, which minimizes the sum of squared vertical distances. For example, suppose you have two variables. Then, the first principal component minimizes the sum of squared perpendicular distances from the points to the first principal axis. This is in contrast to least squares, which would minimize the sum of squared vertical distances from the points to the fitted line.

Principal component analysis can also be used for exploring polynomial relationships and for multivariate outlier detection (Gnanadesikan 1977), and it is related to factor analysis, correspondence analysis, allometry, and biased regression techniques (Mardia, Kent, and Bibby 1979).

Getting Started

The following example uses the PRINCOMP procedure to analyze job performance. Police officers were rated by their supervisors in 14 categories as part of standard police departmental administrative procedure.

The following statements create the Jobratings data set:

```
options validvarname=any;

data Jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
datalines;
26838853879867
74758876857667
56757863775875
67869777988997

  ...

99899899899899
76656399567486
;
```

The data set Jobratings contains 14 variables. Each variable contains the job ratings using a scale measurement from 1 to 10 (1=fail to comply, 10=exceptional). The last variable Overall Rating contains a score as an overall index on how each officer performs.

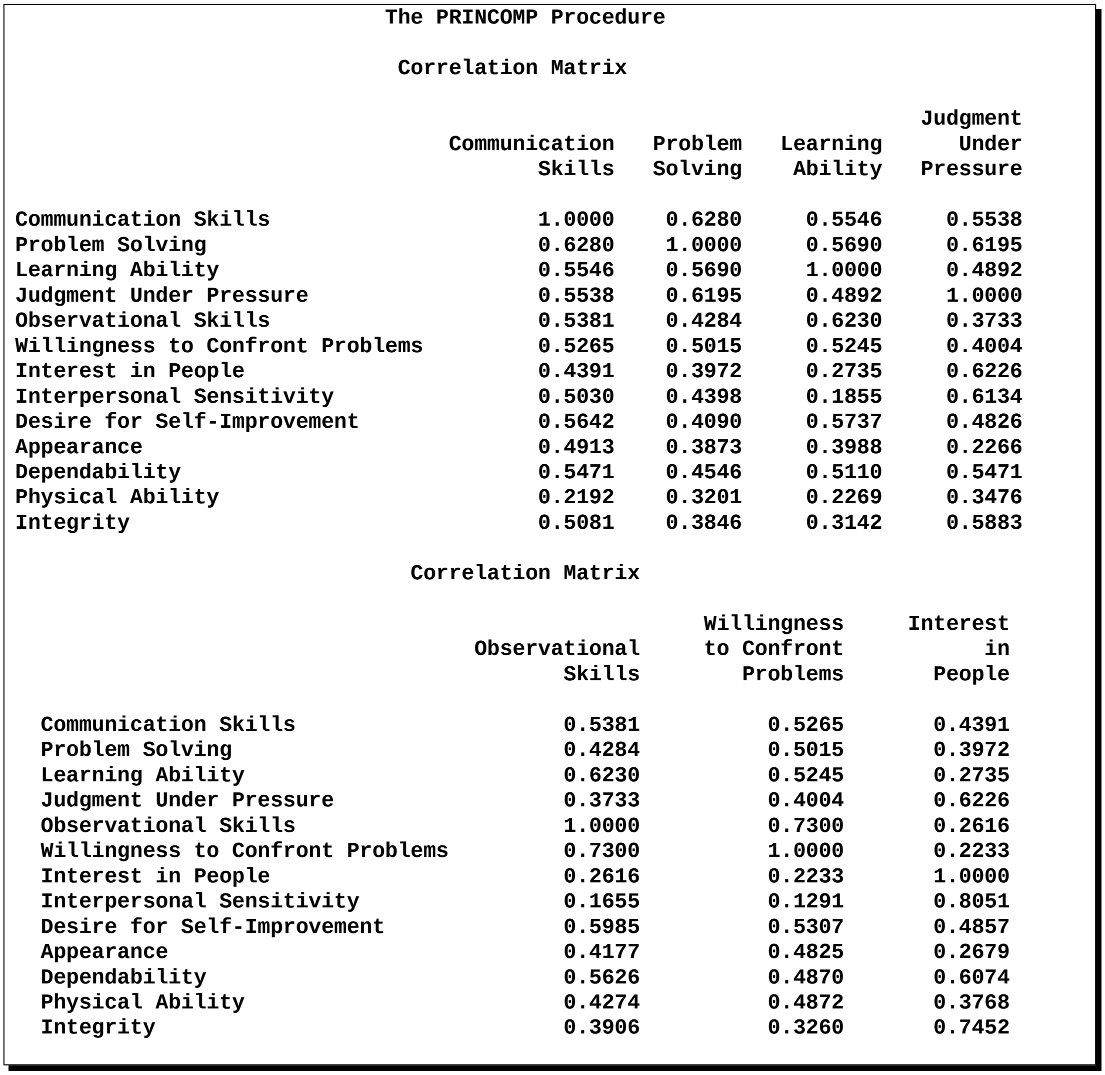
The following statement requests a principal component analysis on the Jobratings data set and outputs the scores to the Scores data set (OUT= Scores). Note that variable Overall Rating is excluded from the analysis.

```
proc princomp data=Jobratings(drop='Overall Rating'n) out=scores;
   run;
```

Figure 58.1 to Figure 58.3 display the PROC PRINCOMP output, beginning with simple statistics followed by the correlation matrix. The PROC PRINCOMP statement requests by default principal components computed from the correlation matrix, so the total variance is equal to the number of variables, 13. In this example, it would also be reasonable to use the COV option, which would cause variables with a high variance (such as Dependability) to have more influence on the results than variables with a low variance (such as Learning Ability). If you used the COV option, scores would be computed from centered rather than standardized variables.

```
                              The PRINCOMP Procedure

                             Observations         103
                             Variables             13


                                Simple Statistics

                                                              Judgment
         Communication        Problem       Learning         Under  Observational
                Skills        Solving        Ability      Pressure         Skills

  Mean     6.650485437    6.631067961    6.990291262   6.737864078    6.932038835
  StD      1.764068036    1.590352602    1.339411238   1.731830976    1.761584269

                                Simple Statistics

        Willingness
        to Confront      Interest  Interpersonal        Desire for
           Problems     in People    Sensitivity  Self-Improvement     Appearance

Mean    7.291262136   6.708737864    6.621359223       6.572815534    7.000000000
StD     1.525155524   1.892353385    1.760773587       1.729796212    1.798692335

                                Simple Statistics

                                              Physical
                  Dependability                Ability           Integrity

            Mean    6.825242718            7.203883495         7.213592233
            StD     1.917040123            1.555251845         1.845240223
```

Figure 58.1. Number of Observations and Simple Statistics from the PRINCOMP Procedure



```
                         The PRINCOMP Procedure

                           Correlation Matrix

                                                                     Judgment
                                 Communication   Problem   Learning     Under
                                        Skills   Solving    Ability  Pressure

Communication Skills                    1.0000    0.6280     0.5546    0.5538
Problem Solving                         0.6280    1.0000     0.5690    0.6195
Learning Ability                        0.5546    0.5690     1.0000    0.4892
Judgment Under Pressure                 0.5538    0.6195     0.4892    1.0000
Observational Skills                    0.5381    0.4284     0.6230    0.3733
Willingness to Confront Problems        0.5265    0.5015     0.5245    0.4004
Interest in People                      0.4391    0.3972     0.2735    0.6226
Interpersonal Sensitivity               0.5030    0.4398     0.1855    0.6134
Desire for Self-Improvement             0.5642    0.4090     0.5737    0.4826
Appearance                              0.4913    0.3873     0.3988    0.2266
Dependability                           0.5471    0.4546     0.5110    0.5471
Physical Ability                        0.2192    0.3201     0.2269    0.3476
Integrity                               0.5081    0.3846     0.3142    0.5883

                            Correlation Matrix

                                                   Willingness     Interest
                                   Observational   to Confront           in
                                          Skills      Problems       People

  Communication Skills                    0.5381        0.5265       0.4391
  Problem Solving                         0.4284        0.5015       0.3972
  Learning Ability                        0.6230        0.5245       0.2735
  Judgment Under Pressure                 0.3733        0.4004       0.6226
  Observational Skills                    1.0000        0.7300       0.2616
  Willingness to Confront Problems        0.7300        1.0000       0.2233
  Interest in People                      0.2616        0.2233       1.0000
  Interpersonal Sensitivity               0.1655        0.1291       0.8051
  Desire for Self-Improvement             0.5985        0.5307       0.4857
  Appearance                              0.4177        0.4825       0.2679
  Dependability                           0.5626        0.4870       0.6074
  Physical Ability                        0.4274        0.4872       0.3768
  Integrity                               0.3906        0.3260       0.7452
```

Figure 58.2. Correlation Matrix from the PRINCOMP Procedure

```
                               Correlation Matrix

                                    Interpersonal        Desire for
                                      Sensitivity  Self-Improvement   Appearance

Communication Skills                       0.5030            0.5642       0.4913
Problem Solving                            0.4398            0.4090       0.3873
Learning Ability                           0.1855            0.5737       0.3988
Judgment Under Pressure                    0.6134            0.4826       0.2266
Observational Skills                       0.1655            0.5985       0.4177
Willingness to Confront Problems           0.1291            0.5307       0.4825
Interest in People                         0.8051            0.4857       0.2679
Interpersonal Sensitivity                  1.0000            0.3713       0.2600
Desire for Self-Improvement                0.3713            1.0000       0.4474
Appearance                                 0.2600            0.4474       1.0000
Dependability                              0.5408            0.5981       0.5089
Physical Ability                           0.2182            0.3752       0.3820
Integrity                                  0.6920            0.5664       0.4135

                                Correlation Matrix

                                                      Physical
                                      Dependability    Ability      Integrity

  Communication Skills                       0.5471     0.2192         0.5081
  Problem Solving                            0.4546     0.3201         0.3846
  Learning Ability                           0.5110     0.2269         0.3142
  Judgment Under Pressure                    0.5471     0.3476         0.5883
  Observational Skills                       0.5626     0.4274         0.3906
  Willingness to Confront Problems           0.4870     0.4872         0.3260
  Interest in People                         0.6074     0.3768         0.7452
  Interpersonal Sensitivity                  0.5408     0.2182         0.6920
  Desire for Self-Improvement                0.5981     0.3752         0.5664
  Appearance                                 0.5089     0.3820         0.4135
  Dependability                              1.0000     0.4461         0.6536
  Physical Ability                           0.4461     1.0000         0.3810
  Integrity                                  0.6536     0.3810         1.0000
```

Figure 58.3. (Continued) Correlation Matrix from the PRINCOMP Procedure

Figure 58.4 displays the eigenvalues. The first principal component explains about 50% of the total variance, the second principal component explains about 13.6%, and the third principal component explains about 7.7%. Note that the eigenvalues sum to the total variance. The eigenvalues indicate that three to five components provide a good summary of the data, with three components accounting for about 71.7% of the total variance and five components explaining about 82.7%. Subsequent components contribute less than 5% each.

```
                       The PRINCOMP Procedure

                Eigenvalues of the Correlation Matrix

            Eigenvalue    Difference    Proportion    Cumulative

       1    6.54740242    4.77468744        0.5036        0.5036
       2    1.77271499    0.76747933        0.1364        0.6400
       3    1.00523565    0.26209665        0.0773        0.7173
       4    0.74313901    0.06479499        0.0572        0.7745
       5    0.67834402    0.22696368        0.0522        0.8267
       6    0.45138034    0.06922167        0.0347        0.8614
       7    0.38215866    0.08432613        0.0294        0.8908
       8    0.29783254    0.02340663        0.0229        0.9137
       9    0.27442591    0.01208809        0.0211        0.9348
      10    0.26233782    0.01778332        0.0202        0.9550
      11    0.24455450    0.04677622        0.0188        0.9738
      12    0.19777828    0.05508241        0.0152        0.9890
      13    0.14269586                      0.0110        1.0000
```

Figure 58.4. Eigenvalues from the PRINCOMP Procedure

Figure 58.5 and Figure 58.6 display the eigenvectors. From the eigenvectors matrix, you can represent the first principal component Prin1 as a linear combination of the original variables

$$\begin{aligned}
\textsf{Prin1} \quad = \quad & 0.303548 \times (\textsf{Communication Skills}) \\
& +0.278034 \times (\textsf{Problem Solving}) \\
& +0.266521 \times (\textsf{Learning Ability}) \\
& . \\
& . \\
& . \\
& +0.298246 \times (\textsf{Integrity})
\end{aligned}$$

and, similarly, the second principal component Prin2 is

$$\begin{aligned}
\textsf{Prin2} \quad = \quad & 0.052039 \times (\textsf{Communication Skills}) \\
& +0.057046 \times (\textsf{Problem Solving}) \\
& +0.288152 \times (\textsf{Learning Ability}) \\
& . \\
& . \\
& . \\
& -0.301812 \times (\textsf{Integrity})
\end{aligned}$$

where the variables are standardized.

```
                              The PRINCOMP Procedure

                                   Eigenvectors

                                       Prin1        Prin2        Prin3        Prin4

Communication Skills                0.303548     0.052039     -.329181     -.227039
Problem Solving                     0.278034     0.057046     -.400112     0.300476
Learning Ability                    0.266521     0.288152     -.354591     -.020735
Judgment Under Pressure             0.294376     -.199458     -.255164     0.397306
Observational Skills                0.276641     0.366979     0.065959     0.035711
Willingness to Confront Problems    0.267580     0.392989     0.098723     0.184409
Interest in People                  0.278060     -.432916     0.118113     0.046047
Interpersonal Sensitivity           0.253814     -.495662     -.064547     -.060000
Desire for Self-Improvement         0.299833     0.099077     0.061097     -.211279
Appearance                          0.237358     0.190065     0.248353     -.544587
Dependability                       0.319480     -.049742     0.169476     -.156070
Physical Ability                    0.213868     0.097499     0.614959     0.514519
Integrity                           0.298246     -.301812     0.190222     -.169062

                                   Eigenvectors

                                       Prin5        Prin6        Prin7        Prin8

Communication Skills                0.181087     -.416563     0.143543     0.333846
Problem Solving                     0.453604     0.096750     0.048904     0.199259
Learning Ability                    -.219329     0.578388     -.114808     0.064088
Judgment Under Pressure             -.030188     0.102087     0.068204     -.591822
Observational Skills                -.325257     -.301254     -.297894     0.163484
Willingness to Confront Problems    0.038278     -.458585     -.044796     -.365684
Interest in People                  -.111279     0.030870     -.011105     0.154829
Interpersonal Sensitivity           0.107807     -.170305     -.088194     0.192725
Desire for Self-Improvement         -.427477     0.105369     0.689011     0.087453
Appearance                          0.568044     0.221643     0.049267     -.257497
Dependability                       -.130575     0.202301     -.594850     0.081242
Physical Ability                    0.203995     0.173168     0.169247     0.302536
Integrity                           -.130757     -.100039     0.029456     -.317545
```

Figure 58.5. Eigenvectors from the PRINCOMP Procedure

```
                                  Eigenvectors

                                       Prin9      Prin10      Prin11      Prin12

Communication Skills                -.430955    0.375983    0.028370    -.252778
Problem Solving                     0.256098    -.372914    -.434417    0.069863
Learning Ability                    0.224706    0.287031    0.210540    -.284355
Judgment Under Pressure             -.358618    0.178270    0.118318    0.306490
Observational Skills                0.258377    0.223793    -.079692    0.565290
Willingness to Confront Problems    0.129976    -.330710    0.275249    -.386151
Interest in People                  0.321200    -.081470    0.393841    -.210915
Interpersonal Sensitivity           0.137468    -.074821    0.285447    0.276824
Desire for Self-Improvement         -.121474    -.363854    -.052085    0.151436
Appearance                          0.087395    0.061890    0.168369    0.236655
Dependability                       -.495598    -.377561    -.164909    -.090904
Physical Ability                    -.149625    0.258321    -.006202    -.055828
Integrity                           0.271060    0.297010    -.612497    -.276273

                                  Eigenvectors

                                                    Prin13

                  Communication Skills            -.122809
                  Problem Solving                 -.116642
                  Learning Ability                0.248555
                  Judgment Under Pressure         -.126636
                  Observational Skills            -.168555
                  Willingness to Confront Problems 0.177688
                  Interest in People              -.610215
                  Interpersonal Sensitivity       0.643410
                  Desire for Self-Improvement     0.053834
                  Appearance                      -.113705
                  Dependability                   -.018094
                  Physical Ability                0.133430
                  Integrity                       0.114965
```

Figure 58.6. (Continued) Eigenvectors from the PRINCOMP Procedure

The first component reflects overall performance since the first eigenvector shows approximately equal loadings on all variables. The second eigenvector has high positive loadings on the variables Observational Skills and Willingness to Confront Problems but even higher negative loadings on the variables Interest in People and Interpersonal Sensitivity. This component seems to reflect the ability to take action, but it also reflects a lack of interpersonal skills. The third eigenvector has a very high positive loading on the variable Physical Ability and high negative loadings on the variables Problem Solving and Learning Ability. This component seems to reflect physical strength, but also shows poor learning and problem-solving skills.

In short, the three components represent:

First Component: overall performance

Second Component: smart, tough, and introverted

Third Component: superior strength and average intellect

Syntax

The following statements are available in PROC PRINCOMP.

PROC PRINCOMP < *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
PARTIAL *variables* **;**
VAR *variables* **;**
WEIGHT *variable* **;**

Usually only the VAR statement is used in addition to the PROC PRINCOMP statement. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC PRINCOMP statement. The remaining statements are described in alphabetical order.

PROC PRINCOMP Statement

PROC PRINCOMP < *options* > **;**

The PROC PRINCOMP statement starts the PRINCOMP procedure and, optionally, identifies input and output data sets, specifies details of the analysis, or suppresses the display of output. You can specify the following options in the PROC PRINCOMP statement.

Task	**Options**
Specify data sets	DATA=
	OUT=
	OUTSTAT=
Specify details of analysis	COV
	N=
	NOINT
	PREFIX=
	SINGULAR=
	STD
	VARDEF=
Suppress the display of output	NOPRINT

The following list provides details on these options.

COVARIANCE
COV

computes the principal components from the covariance matrix. If you omit the COV option, the correlation matrix is analyzed. Use of the COV option causes variables with large variances to be more strongly associated with components with large eigenvalues and causes variables with small variances to be more strongly associated with components with small eigenvalues. You should not specify the COV option unless the units in which the variables are measured are comparable or the variables are standardized in some way.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set can be an ordinary SAS data set or a TYPE=ACE, TYPE=CORR, TYPE=COV, TYPE=FACTOR, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV data set (see Appendix A, "Special SAS Data Sets,"). Also, the PRINCOMP procedure can read the _TYPE_='COVB' matrix from a TYPE=EST data set. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

N=*number*

specifies the number of principal components to be computed. The default is the number of variables. The value of the N= option must be an integer greater than or equal to zero.

NOINT

omits the intercept from the model. In other words, the NOINT option requests that the covariance or correlation matrix not be corrected for the mean. When you use the PRINCOMP procedure with the NOINT option, the covariance matrix and, hence, the standard deviations are not corrected for the mean. If you are interested in the standard deviations corrected for the mean, you can get them by using a procedure such as the MEANS procedure.

If you use a TYPE=SSCP data set as input to the PRINCOMP procedure and list the variable Intercept in the VAR statement, the procedure acts as if you had also specified the NOINT option. If you use NOINT and also create an OUTSTAT= data set, the data set is TYPE=UCORR or TYPE=UCOV rather than TYPE=CORR or TYPE=COV.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set that contains all the original data as well as the principal component scores. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for information on permanent SAS data sets).

OUTSTAT=*SAS-data-set*

creates an output SAS data set that contains means, standard deviations, number of observations, correlations or covariances, eigenvalues, and eigenvectors. If you specify the COV option, the data set is TYPE=COV or TYPE=UCOV, depending on the NOINT option, and it contains covariances; otherwise, the data set is TYPE=CORR or TYPE=UCORR, depending on the NOINT option, and it contains correlations. If you specify the PARTIAL statement, the OUTSTAT= data set contains R-squares as well. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for information on permanent SAS data sets).

PREFIX=*name*

specifies a prefix for naming the principal components. By default, the names are Prin1, Prin2, . . . , Prin$n$. If you specify PREFIX=ABC, the components are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the current name length defined by the VALIDVARNAME= system option.

SINGULAR=*p*
SING=*p*

specifies the singularity criterion, where $0 < p < 1$. If a variable in a PARTIAL statement has an R-square as large as $1 - p$ when predicted from the variables listed before it in the statement, the variable is assigned a standardized coefficient of 0. By default, SINGULAR=1E-8.

STANDARD
STD

standardizes the principal component scores in the OUT= data set to unit variance. If you omit the STANDARD option, the scores have variance equal to the corresponding eigenvalue. Note that STANDARD has no effect on the eigenvalues themselves.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor used in calculating variances and standard deviations. By default, VARDEF=DF. The following table displays the values and associated divisors.

Value	**Divisor**	**Formula**	
DF	error degrees of freedom	$n - i$	(before partialling)
		$n - p - i$	(after partialling)
N	number of observations	n	
WEIGHT | WGT	sum of weights	$\sum_{j=1}^{n} w_j$	
WDF	sum of weights minus one	$\left(\sum_{j=1}^{n} w_j\right) - i$	(before partialling)
		$\left(\sum_{j=1}^{n} w_j\right) - p - i$	(after partialling)

In the formulas for VARDEF=DF and VARDEF=WDF, p is the number of degrees of freedom of the variables in the PARTIAL statement, and i is 0 if the NOINT option is specified and 1 otherwise.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PRINCOMP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PRINCOMP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

The analysis produced using a FREQ statement reflects the expanded number of observations. The total number of observations is considered equal to the sum of the FREQ variable. You could produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set would be identical. Each observation in the old data set would be replicated n_j times in the new data set, where n_j is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

PARTIAL Statement

PARTIAL *variables* **;**

If you want to analyze a partial correlation or covariance matrix, specify the names of the numeric variables to be partialled out in the PARTIAL statement. The PRINCOMP procedure computes the principal components of the residuals from the prediction of the VAR variables by the PARTIAL variables. If you request an OUT= or OUTSTAT= data set, the residual variables are named by prefixing the characters R_ to the VAR variables. Thus, the number of characters required to distinguish the VAR variables should be, at most, two characters fewer than the current name length defined by the VALIDVARNAME= system option.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not specified in other statements are analyzed. If, however, the DATA= data set is TYPE=SSCP, the default set of variables used as VAR variables does not include Intercept so that the correlation or covariance matrix is constructed correctly. If you want to analyze Intercept as a separate variable, you should specify it in the VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances.

The observation is used in the analysis only if the value of the WEIGHT statement variable is nonmissing and is greater than zero.

Details

Missing Values

Observations with missing values for any variable in the VAR, PARTIAL, FREQ, or WEIGHT statement are omitted from the analysis and are given missing values for principal component scores in the OUT= data set. If a correlation, covariance, or SSCP matrix is read, it can contain missing values as long as every pair of variables has at least one nonmissing entry.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the principal component scores. The N= option determines the number of new variables. The names of the new variables are formed by concatenating the value given by the PREFIX= option (or Prin if PREFIX= is omitted) and the numbers 1, 2, 3, and so on. The new variables have mean 0 and variance equal to the corresponding eigenvalue, unless you specify the STANDARD option to standardize the scores to unit variance. Also, if you specify the COV option, the procedure computes the principal component scores from the corrected or the uncorrected (if the NOINT option is specified) variables rather than the standardized variables.

If you use a PARTIAL statement, the OUT= data set also contains the residuals from predicting the VAR variables from the PARTIAL variables. The names of the residual variables are formed by prefixing R_ to the names of the VAR variables.

An OUT= data set cannot be created if the DATA= data set is TYPE=ACE, TYPE=CORR, TYPE=COV, TYPE=EST, TYPE=FACTOR, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure. The following table relates the TYPE= value for the OUTSTAT= data set to the options specified in the PROC PRINCOMP statement.

Options	**TYPE=**
(default)	CORR
COV	COV
NOINT	UCORR
COV NOINT	UCOV

Notice that the default (neither the COV nor NOINT option) produces a TYPE=CORR data set.

The new data set contains the following variables:

- the BY variables, if any
- two new variables, _TYPE_ and _NAME_, both character variables
- the variables analyzed, that is, those in the VAR statement; or, if there is no VAR statement, all numeric variables not listed in any other statement; or, if there is a PARTIAL statement, the residual variables as described under the OUT= data set

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

TYPE

MEAN mean of each variable. If you specify the PARTIAL statement, this observation is omitted.

STD standard deviations. If you specify the COV option, this observation is omitted, so the SCORE procedure does not standardize the variables before computing scores. If you use the PARTIAL statement, the standard deviation of a variable is computed as its root mean squared error as predicted from the PARTIAL variables.

USTD uncorrected standard deviations. When you specify the NOINT option in the PROC PRINCOMP statement, the OUTSTAT= data set contains standard deviations not corrected for the mean. However, if you also specify the COV option in the PROC PRINCOMP statement, this observation is omitted.

N number of observations on which the analysis is based. This value is the same for each variable. If you specify the PARTIAL statement and the value of the VARDEF= option is DF or unspecified, then the number of observations is decremented by the degrees of freedom for the PARTIAL variables.

SUMWGT the sum of the weights of the observations. This value is the same for each variable. If you specify the PARTIAL statement and VARDEF=WDF, then the sum of the weights is decremented by the degrees of freedom for the PARTIAL variables. This observation is output only if the value is different from that in the observation with _TYPE_='N'.

CORR correlations between each variable and the variable specified by the _NAME_ variable. The number of observations with _TYPE_='CORR' is equal to the number of variables being analyzed. If you specify the COV option, no _TYPE_='CORR' observations are produced. If you use the PARTIAL statement, the partial correlations, not the raw correlations, are output.

UCORR uncorrected correlation matrix. When you specify the NOINT option without the COV option in the PROC PRINCOMP statement, the OUTSTAT= data set contains a matrix of correlations not corrected for the means. However, if you also specify the COV option in the PROC PRINCOMP statement, this observation is omitted.

COV covariances between each variable and the variable specified by the _NAME_ variable. _TYPE_='COV' observations are produced only if you specify the COV option. If you use the PARTIAL statement, the partial covariances, not the raw covariances, are output.

UCOV uncorrected covariance matrix. When you specify the NOINT and COV options in the PROC PRINCOMP statement, the OUTSTAT= data set contains a matrix of covariances not corrected for the means.

EIGENVAL eigenvalues. If the N= option requested fewer than the maximum number of principal components, only the specified number of eigenvalues are produced, with missing values filling out the observation.

SCORE eigenvectors. The _NAME_ variable contains the name of the corresponding principal component as constructed from the PREFIX= option. The number of observations with _TYPE_='SCORE' equals the number of principal components computed. The eigenvectors have unit length unless you specify the STD option, in which case the unit-length eigenvectors are divided by the square roots of the eigenvalues to produce scores with unit standard deviations.

To obtain the principal component scores, if the COV option is not specified, these coefficients should be multiplied by the standardized data. With the COV option, these coefficients should be multiplied by the centered data. Means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD' should be used for centering and standardizing the data.

USCORE scoring coefficients to be applied without subtracting the mean from the raw variables. _TYPE_='USCORE' observations are produced when you specify the NOINT option in the PROC PRINCOMP statement.

To obtain the principal component scores, these coefficients should be multiplied by the data that are standardized by the uncorrected standard deviations obtained from the observation with _TYPE_='USTD'.

RSQUARED R-squares for each VAR variable as predicted by the PARTIAL variables

B regression coefficients for each VAR variable as predicted by the PARTIAL variables. This observation is produced only if you specify the COV option.

STB standardized regression coefficients for each VAR variable as predicted by the PARTIAL variables. If you specify the COV option, this observation is omitted.

The data set can be used with the SCORE procedure to compute principal component scores, or it can be used as input to the FACTOR procedure specifying METHOD=SCORE to rotate the components. If you use the PARTIAL statement, the scoring coefficients should be applied to the residuals, not the original variables.

Computational Resources

Let

n = number of observations
v = number of VAR variables
p = number of PARTIAL variables
c = number of components

- The minimum allocated memory required is

  $$232v + 120p + 48c + \max(8cv, 8vp + 4(v+p)(v+p+1))$$

  bytes
- The time required to compute the correlation matrix is roughly proportional to

  $$n(v+p)^2 + \frac{p}{2}(v+p)(v+p+1)$$

- The time required to compute eigenvalues is roughly proportional to v^3.
- The time required to compute eigenvectors is roughly proportional to cv^2.

Displayed Output

The PRINCOMP procedure displays the following items if the DATA= data set is not TYPE=CORR, TYPE=COV, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV:

- Simple Statistics, including the Mean and Std (standard deviation) for each variable. If you specify the NOINT option, the uncorrected standard deviation (UStD) is displayed.
- the Correlation or, if you specify the COV option, the Covariance Matrix

The PRINCOMP procedure displays the following items if you use the PARTIAL statement.

- Regression Statistics, giving the R-square and RMSE (root mean square error) for each VAR variable as predicted by the PARTIAL variables (not shown)
- Standardized Regression Coefficients or, if you specify the COV option, Regression Coefficients for predicting the VAR variables from the PARTIAL variables (not shown)
- the Partial Correlation Matrix or, if you specify the COV option, the Partial Covariance Matrix (not shown)

The PRINCOMP procedure displays the following item if you specify the COV option:

- the Total Variance

The PRINCOMP procedure displays the following items unless you specify the NOPRINT option:

- Eigenvalues of the correlation or covariance matrix, as well as the Difference between successive eigenvalues, the Proportion of variance explained by each eigenvalue, and the Cumulative proportion of variance explained
- the Eigenvectors

ODS Table Names

PROC PRINCOMP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 58.1. ODS Tables Produced in PROC PRINCOMP

ODS Table Name	Description	Statement / Option
Corr	Correlation Matrix	default unless COV is specified
Cov	Covariance Matrix	default if COV is specified
Eigenvalues	Eigenvalues	default
Eigenvectors	Eigenvectors	default
NObsNVar	Number of Observations, Variables and (Partial) Variables	default
ParCorr	Partial Correlation Matrix	PARTIAL statement
ParCov	Uncorrected Partial Covariance Matrix	PARTIAL statement COV
RegCoef	Regression Coefficients	PARTIAL statement COV
RSquareRMSE	Regression Statistics: R-Squares and RMSEs	PARTIAL statement
SimpleStatistics	Simple Statistics	default
StdRegCoef	Standardized Regression Coefficients	PARTIAL statement
TotalVariance	Total Variance	PROC PRINCOMP COV

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the PRINCOMP procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs, you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

You can specify the N= option in the PRINCOMP statement to control the number of principal components to be displayed.

ODS Graph Names

PROC PRINCOMP assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 58.2.

To request these graphs, you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 58.2. ODS Graphics Produced by PROC PRINCOMP

ODS Graph Name	Plot Description	Statement
EigenvaluePlot	Eigenvalues and Proportion Plot	default
PaintedPrinCompScoresPlot	Painted Component Scores Plot: 2nd versus 3rd, painted by 1st	default and $nvar^{*} >= 3$
PrinCompMatrixPlot	Component Scores Matrix Plot	default and $nvar >= 2$
PrinCompPatternPlot	Component Pattern Plot	default
PrinCompScoresPlot12	Component Scores Plot: 1st versus 2nd	default and $nvar >= 2$
PrinCompScoresPlot13	Component Scores Plot: 1st versus 3rd	default and $nvar >= 3$

Examples

Example 58.1. Temperatures

This example analyzes mean daily temperatures in selected cities in January and July. Both the raw data and the principal components are plotted to illustrate how principal components are orthogonal rotations of the original variables.

The following statements create the Temperature data set:

```
data Temperature;
   title 'Mean Temperature in January and July for Selected Cities ';
   input City $1-15 January July;
   cards;
Mobile          51.2 81.6
Phoenix         51.2 91.2
Little Rock     39.5 81.4
Sacramento      45.1 75.2
Denver          29.9 73.0
Hartford        24.8 72.7
Wilmington      32.0 75.8
Washington DC   35.6 78.7
Jacksonville    54.6 81.0
Miami           67.2 82.3
Atlanta         42.4 78.0
Boise           29.0 74.5
Chicago         22.9 71.9
Peoria          23.8 75.1
Indianapolis    27.9 75.0
Des Moines      19.4 75.1
Wichita         31.3 80.7
Louisville      33.3 76.9
```

[*] number of variables to be analyzed

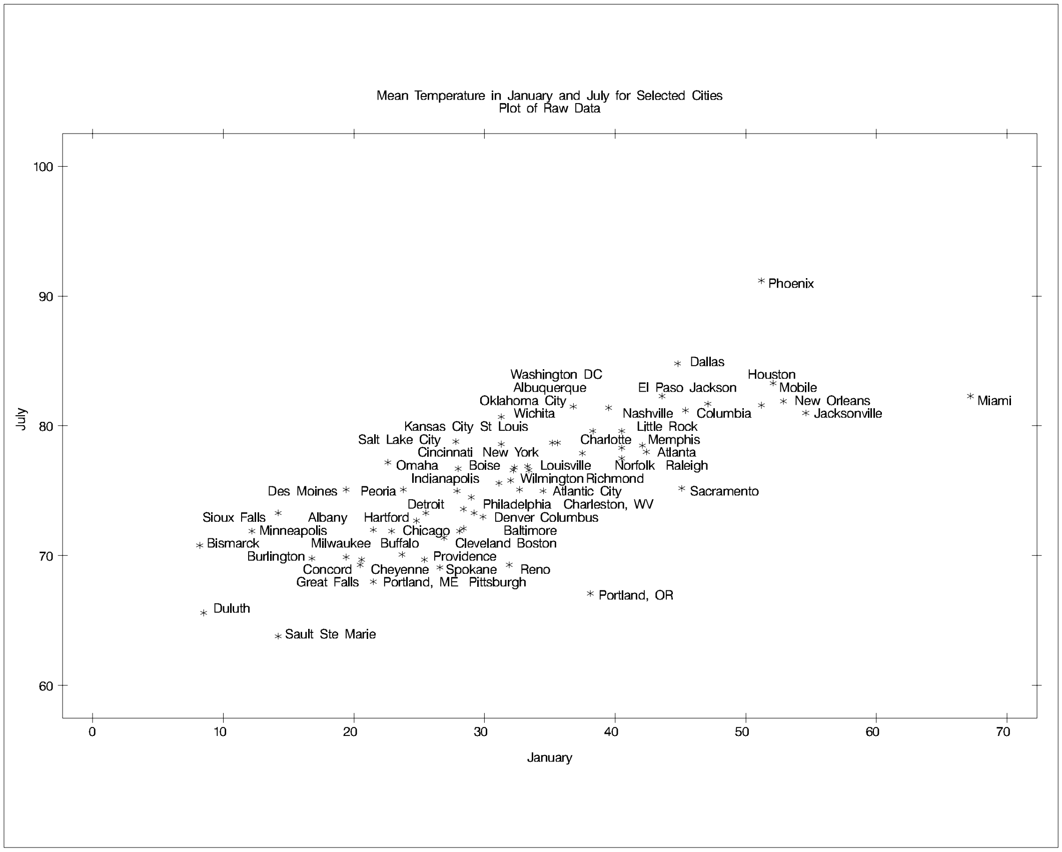
```
New Orleans     52.9 81.9
Portland, ME    21.5 68.0
Baltimore       33.4 76.6
Boston          29.2 73.3
Detroit         25.5 73.3
Sault Ste Marie 14.2 63.8
Duluth           8.5 65.6
Minneapolis     12.2 71.9
Jackson         47.1 81.7
Kansas City     27.8 78.8
St Louis        31.3 78.6
Great Falls     20.5 69.3
Omaha           22.6 77.2
Reno            31.9 69.3
Concord         20.6 69.7
Atlantic City   32.7 75.1
Albuquerque     35.2 78.7
Albany          21.5 72.0
Buffalo         23.7 70.1
New York        32.2 76.6
Charlotte       42.1 78.5
Raleigh         40.5 77.5
Bismarck         8.2 70.8
Cincinnati      31.1 75.6
Cleveland       26.9 71.4
Columbus        28.4 73.6
Oklahoma City   36.8 81.5
Portland, OR    38.1 67.1
Philadelphia    32.3 76.8
Pittsburgh      28.1 71.9
Providence      28.4 72.1
Columbia        45.4 81.2
Sioux Falls     14.2 73.3
Memphis         40.5 79.6
Nashville       38.3 79.6
Dallas          44.8 84.8
El Paso         43.6 82.3
Houston         52.1 83.3
Salt Lake City  28.0 76.7
Burlington      16.8 69.8
Norfolk         40.5 78.3
Richmond        37.5 77.9
Spokane         25.4 69.7
Charleston, WV  34.5 75.0
Milwaukee       19.4 69.9
Cheyenne        26.6 69.1
;
```

The following statements plot the temperature data set. For information on the %PLOTIT macro, see Appendix B, “Using the %PLOTIT Macro.”

```
title2 'Plot of Raw Data';
%plotit(data=Temperature,labelvar=City,
        plotvars=July January, color=black, colors=black);
run;
```

The results are displayed in Output 58.1.1, which shows a scatter diagram of the 64 pairs of data points with July temperatures plotted against January temperatures.

Output 58.1.1. Plot of Raw Data



The following statement requests a principal component analysis on the Temperature data set and outputs the scores to the Prin data set (OUT= Prin):

```
proc princomp data=Temperature cov out=Prin;
var July January;
run;
```

Output 58.1.2 displays the PROC PRINCOMP output. The standard deviation of January (11.712) is higher than the standard deviation of July (5.128). The COV option in the PROC PRINCOMP statement requests the principal components to be computed from the covariance matrix. The total variance is 163.474. The first principal component explains about 94% of the total variance, and the second principal component explains only about 6%. The eigenvalues sum to the total variance.

Note that January receives a higher loading on Prin1 because it has a higher standard deviation than July, and the PRINCOMP procedure calculates the scores using the centered variables rather than the standardized variables.

Output 58.1.2. Results of Principal Component Analysis

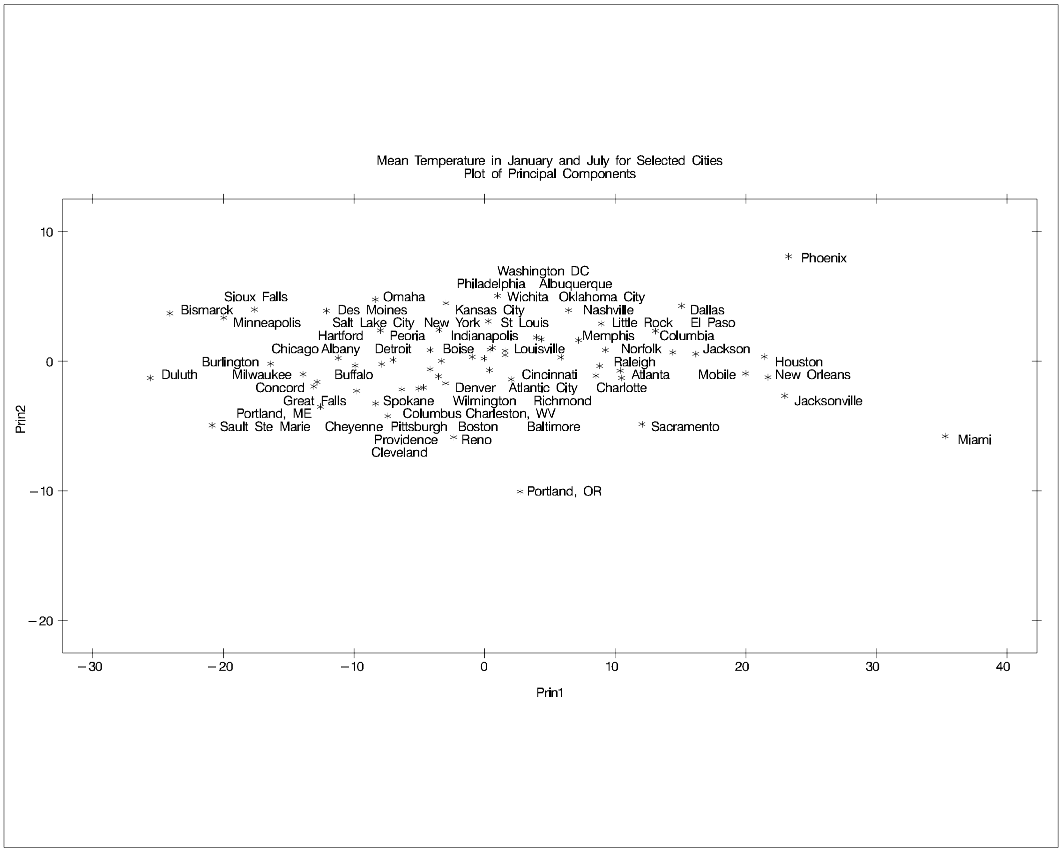
```
            Mean Temperature in January and July for Selected Cities

                              The PRINCOMP Procedure

                             Observations         64
                             Variables             2


                                Simple Statistics

                                       July           January

                     Mean       75.60781250       32.09531250
                     StD         5.12761910       11.71243309



                                Covariance Matrix

                                        July           January

                   July           26.2924777        46.8282912
                   January        46.8282912       137.1810888



                         Total Variance    163.47356647


                          Eigenvalues of the Covariance Matrix

                   Eigenvalue    Difference    Proportion    Cumulative

              1    154.310607    145.147647        0.9439        0.9439
              2      9.162960                      0.0561        1.0000


                                   Eigenvectors

                                       Prin1         Prin2

                       July         0.343532      0.939141
                       January      0.939141      -.343532
```

The following statement plots the Prin data set created from the previous PROC PRINCOMP statement:

```
title2 'Plot of Principal Components';
%plotit(data=Prin,labelvar=City,
        plotvars=Prin2 Prin1, color=black, colors=black);
run;
```

Output 58.1.3 displays a plot of the second principal component Prin2 against the first principal component Prin1. It is clear from this plot that the principal components are orthogonal rotations of the original variables and that the first principal component has a larger variance than the second principal component. In fact, Prin1 has a larger variance than either of the original variables July and January.

Output 58.1.3. Plot of Principal Components



Example 58.2. Crime Rates

The following data provide crime rates per 100,000 people in seven categories for each of the fifty states in 1977. Since there are seven numeric variables, it is impossible to plot all the variables simultaneously. Principal components can be used to summarize the data in two or three dimensions, and they help to visualize the data. The following statements produce Output 58.2.1:

```
data Crime;
   title 'Crime Rates per 100,000 Population by State';
   input State $1-15 Murder Rape Robbery Assault
         Burglary Larceny Auto_Theft;
   cards;
Alabama        14.2 25.2  96.8 278.3 1135.5 1881.9 280.7
Alaska         10.8 51.6  96.8 284.0 1331.7 3369.8 753.3
Arizona         9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas        8.8 27.6  83.2 203.4  972.6 1862.1 183.4
California     11.5 49.4 287.0 358.0 2139.4 3499.8 663.5
Colorado        6.3 42.0 170.7 292.9 1935.2 3903.2 477.1
Connecticut     4.2 16.8 129.5 131.8 1346.0 2620.7 593.2
Delaware        6.0 24.9 157.0 194.2 1682.6 3678.4 467.0
Florida        10.2 39.6 187.9 449.1 1859.9 3840.5 351.4
Georgia        11.7 31.1 140.5 256.5 1351.1 2170.2 297.9
Hawaii          7.2 25.5 128.0  64.1 1911.5 3920.4 489.4
Idaho           5.5 19.4  39.6 172.5 1050.8 2599.6 237.6
Illinois        9.9 21.8 211.3 209.0 1085.0 2828.5 528.6
Indiana         7.4 26.5 123.2 153.5 1086.2 2498.7 377.4
Iowa            2.3 10.6  41.2  89.8  812.5 2685.1 219.9
Kansas          6.6 22.0 100.7 180.5 1270.4 2739.3 244.3
Kentucky       10.1 19.1  81.1 123.3  872.2 1662.1 245.4
Louisiana      15.5 30.9 142.9 335.5 1165.5 2469.9 337.7
Maine           2.4 13.5  38.7 170.0 1253.1 2350.7 246.9
Maryland        8.0 34.8 292.1 358.9 1400.0 3177.7 428.5
Massachusetts   3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
Michigan        9.3 38.9 261.9 274.6 1522.7 3159.0 545.5
Minnesota       2.7 19.5  85.9  85.8 1134.7 2559.3 343.1
Mississippi    14.3 19.6  65.7 189.1  915.6 1239.9 144.4
Missouri        9.6 28.3 189.0 233.5 1318.3 2424.2 378.4
Montana         5.4 16.7  39.2 156.8  804.9 2773.2 309.2
Nebraska        3.9 18.1  64.7 112.7  760.0 2316.1 249.1
Nevada         15.8 49.1 323.1 355.0 2453.1 4212.6 559.2
New Hampshire   3.2 10.7  23.2  76.0 1041.7 2343.9 293.4
New Jersey      5.6 21.0 180.4 185.1 1435.8 2774.5 511.5
New Mexico      8.8 39.1 109.6 343.4 1418.7 3008.6 259.5
New York       10.7 29.4 472.6 319.1 1728.0 2782.0 745.8
North Carolina 10.6 17.0  61.3 318.3 1154.1 2037.8 192.1
North Dakota    0.9  9.0  13.3  43.8  446.1 1843.0 144.7
Ohio            7.8 27.3 190.5 181.1 1216.0 2696.8 400.4
Oklahoma        8.6 29.2  73.8 205.0 1288.2 2228.1 326.8
Oregon          4.9 39.9 124.1 286.9 1636.4 3506.1 388.9
Pennsylvania    5.6 19.0 130.3 128.0  877.5 1624.1 333.2
Rhode Island    3.6 10.5  86.5 201.0 1489.5 2844.1 791.4
South Carolina 11.9 33.0 105.9 485.3 1613.6 2342.4 245.1
South Dakota    2.0 13.5  17.9 155.7  570.5 1704.4 147.5
Tennessee      10.1 29.7 145.8 203.9 1259.7 1776.5 314.0
Texas          13.3 33.8 152.4 208.2 1603.1 2988.7 397.6
Utah            3.5 20.3  68.8 147.3 1171.6 3004.6 334.5
Vermont         1.4 15.9  30.8 101.2 1348.2 2201.0 265.2
Virginia        9.0 23.3  92.1 165.7  986.2 2521.2 226.7
Washington      4.3 39.6 106.2 224.8 1605.6 3386.9 360.3
West Virginia   6.0 13.2  42.2  90.9  597.4 1341.7 163.3
Wisconsin       2.8 12.9  52.2  63.7  846.9 2614.2 220.7
Wyoming         5.4 21.9  39.7 173.9  811.6 2772.2 282.0
;


proc princomp out=Crime_Components;
run;
```

Output 58.2.1. Results of Principal Component Analysis: PROC PRINCOMP

```
                  Crime Rates per 100,000 Population by State

                             The PRINCOMP Procedure

                            Observations          50
                            Variables              7


                               Simple Statistics

                  Murder              Rape           Robbery           Assault

 Mean        7.444000000       25.73400000       124.0920000       211.3000000
 StD         3.866768941       10.75962995        88.3485672       100.2530492

                               Simple Statistics

                         Burglary           Larceny         Auto_Theft

          Mean        1291.904000       2671.288000       377.5260000
          StD          432.455711        725.908707       193.3944175



                               Correlation Matrix

                                                                         Auto_
              Murder      Rape    Robbery    Assault    Burglary    Larceny    Theft

 Murder       1.0000    0.6012     0.4837     0.6486      0.3858     0.1019   0.0688
 Rape         0.6012    1.0000     0.5919     0.7403      0.7121     0.6140   0.3489
 Robbery      0.4837    0.5919     1.0000     0.5571      0.6372     0.4467   0.5907
 Assault      0.6486    0.7403     0.5571     1.0000      0.6229     0.4044   0.2758
 Burglary     0.3858    0.7121     0.6372     0.6229      1.0000     0.7921   0.5580
 Larceny      0.1019    0.6140     0.4467     0.4044      0.7921     1.0000   0.4442
 Auto_Theft   0.0688    0.3489     0.5907     0.2758      0.5580     0.4442   1.0000



                      Eigenvalues of the Correlation Matrix

                  Eigenvalue    Difference    Proportion    Cumulative

              1   4.11495951    2.87623768        0.5879        0.5879
              2   1.23872183    0.51290521        0.1770        0.7648
              3   0.72581663    0.40938458        0.1037        0.8685
              4   0.31643205    0.05845759        0.0452        0.9137
              5   0.25797446    0.03593499        0.0369        0.9506
              6   0.22203947    0.09798342        0.0317        0.9823
              7   0.12405606                      0.0177        1.0000


                                  Eigenvectors

               Prin1      Prin2      Prin3      Prin4      Prin5      Prin6      Prin7

Murder      0.300279   -.629174   0.178245   -.232114   0.538123   0.259117   0.267593
Rape        0.431759   -.169435   -.244198   0.062216   0.188471   -.773271   -.296485
Robbery     0.396875   0.042247   0.495861   -.557989   -.519977   -.114385   -.003903
Assault     0.396652   -.343528   -.069510   0.629804   -.506651   0.172363   0.191745
Burglary    0.440157   0.203341   -.209895   -.057555   0.101033   0.535987   -.648117
Larceny     0.357360   0.402319   -.539231   -.234890   0.030099   0.039406   0.601690
Auto_Theft  0.295177   0.502421   0.568384   0.419238   0.369753   -.057298   0.147046
```

The eigenvalues indicate that two or three components provide a good summary of the data, two components accounting for 76% of the total variance and three components explaining 87%. Subsequent components contribute less than 5% each.

The first component is a measure of overall crime rate since the first eigenvector shows approximately equal loadings on all variables. The second eigenvector has high positive loadings on variables Auto_Theft and Larceny and high negative loadings on variables Murder and Assault. There is also a small positive loading on Burglary and a small negative loading on Rape. This component seems to measure the preponderance of property crime over violent crime. The interpretation of the third component is not obvious.

A simple way to examine the principal components in more detail is to display the output data set sorted by each of the large components. The following statements produce Output 58.2.2 through Output 58.2.3:

```
proc sort data=Crime_Components;
   by Prin1;
run;

proc print;
   id State;
   var Prin1 Prin2 Murder Rape Robbery
       Assault Burglary Larceny Auto_Theft;
   title2 'States Listed in Order of Overall Crime Rate';
   title3 'As Determined by the First Principal Component';
run;



proc sort data=Crime_Components;
   by Prin2;
run;

proc print;
   id State;
   var Prin1 Prin2 Murder Rape Robbery
       Assault Burglary Larceny Auto_Theft;
   title2 'States Listed in Order of Property Vs. Violent Crime';
   title3 'As Determined by the Second Principal Component';
run;
```

Output 58.2.2. OUT= Data Set Sorted by First Principal Component

```
                Crime Rates per 100,000 Population by State
                States Listed in Order of Overall Crime Rate
               As Determined by the First Principal Component

                                                                        A
                                                                        u
                                                          B             t
                                             R     A      u      L      o
                                    M        o     s      r      a      _
S                  P        P       u        b     s      g      r      T
t                  r        r       r   R    b     a      l      c      h
a                  i        i       d   a    e     u      a      e      e
t                  n        n       e   p    r     l      r      n      f
e                  1        2       r   e    y     t      y      y      t

North Dakota   -3.96408  0.38767  0.9  9.0  13.3  43.8  446.1 1843.0  144.7
South Dakota   -3.17203 -0.25446  2.0 13.5  17.9 155.7  570.5 1704.4  147.5
West Virginia  -3.14772 -0.81425  6.0 13.2  42.2  90.9  597.4 1341.7  163.3
Iowa           -2.58156  0.82475  2.3 10.6  41.2  89.8  812.5 2685.1  219.9
Wisconsin      -2.50296  0.78083  2.8 12.9  52.2  63.7  846.9 2614.2  220.7
New Hampshire  -2.46562  0.82503  3.2 10.7  23.2  76.0 1041.7 2343.9  293.4
Nebraska       -2.15071  0.22574  3.9 18.1  64.7 112.7  760.0 2316.1  249.1
Vermont        -2.06433  0.94497  1.4 15.9  30.8 101.2 1348.2 2201.0  265.2
Maine          -1.82631  0.57878  2.4 13.5  38.7 170.0 1253.1 2350.7  246.9
Kentucky       -1.72691 -1.14663 10.1 19.1  81.1 123.3  872.2 1662.1  245.4
Pennsylvania   -1.72007 -0.19590  5.6 19.0 130.3 128.0  877.5 1624.1  333.2
Montana        -1.66801  0.27099  5.4 16.7  39.2 156.8  804.9 2773.2  309.2
Minnesota      -1.55434  1.05644  2.7 19.5  85.9  85.8 1134.7 2559.3  343.1
Mississippi    -1.50736 -2.54671 14.3 19.6  65.7 189.1  915.6 1239.9  144.4
Idaho          -1.43245 -0.00801  5.5 19.4  39.6 172.5 1050.8 2599.6  237.6
Wyoming        -1.42463  0.06268  5.4 21.9  39.7 173.9  811.6 2772.2  282.0
Arkansas       -1.05441 -1.34544  8.8 27.6  83.2 203.4  972.6 1862.1  183.4
Utah           -1.04996  0.93656  3.5 20.3  68.8 147.3 1171.6 3004.6  334.5
Virginia       -0.91621 -0.69265  9.0 23.3  92.1 165.7  986.2 2521.2  226.7
North Carolina -0.69925 -1.67027 10.6 17.0  61.3 318.3 1154.1 2037.8  192.1
Kansas         -0.63407 -0.02804  6.6 22.0 100.7 180.5 1270.4 2739.3  244.3
Connecticut    -0.54133  1.50123  4.2 16.8 129.5 131.8 1346.0 2620.7  593.2
Indiana        -0.49990  0.00003  7.4 26.5 123.2 153.5 1086.2 2498.7  377.4
Oklahoma       -0.32136 -0.62429  8.6 29.2  73.8 205.0 1288.2 2228.1  326.8
Rhode Island   -0.20156  2.14658  3.6 10.5  86.5 201.0 1489.5 2844.1  791.4
Tennessee      -0.13660 -1.13498 10.1 29.7 145.8 203.9 1259.7 1776.5  314.0
Alabama        -0.04988 -2.09610 14.2 25.2  96.8 278.3 1135.5 1881.9  280.7
New Jersey      0.21787  0.96421  5.6 21.0 180.4 185.1 1435.8 2774.5  511.5
Ohio            0.23953  0.09053  7.8 27.3 190.5 181.1 1216.0 2696.8  400.4
Georgia         0.49041 -1.38079 11.7 31.1 140.5 256.5 1351.1 2170.2  297.9
Illinois        0.51290  0.09423  9.9 21.8 211.3 209.0 1085.0 2828.5  528.6
Missouri        0.55637 -0.55851  9.6 28.3 189.0 233.5 1318.3 2424.2  378.4
Hawaii          0.82313  1.82392  7.2 25.5 128.0  64.1 1911.5 3920.4  489.4
Washington      0.93058  0.73776  4.3 39.6 106.2 224.8 1605.6 3386.9  360.3
Delaware        0.96458  1.29674  6.0 24.9 157.0 194.2 1682.6 3678.4  467.0
Massachusetts   0.97844  2.63105  3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
Louisiana       1.12020 -2.08327 15.5 30.9 142.9 335.5 1165.5 2469.9  337.7
New Mexico      1.21417 -0.95076  8.8 39.1 109.6 343.4 1418.7 3008.6  259.5
Texas           1.39696 -0.68131 13.3 33.8 152.4 208.2 1603.1 2988.7  397.6
Oregon          1.44900  0.58603  4.9 39.9 124.1 286.9 1636.4 3506.1  388.9
South Carolina  1.60336 -2.16211 11.9 33.0 105.9 485.3 1613.6 2342.4  245.1
Maryland        2.18280 -0.19474  8.0 34.8 292.1 358.9 1400.0 3177.7  428.5
Michigan        2.27333  0.15487  9.3 38.9 261.9 274.6 1522.7 3159.0  545.5
Alaska          2.42151  0.16652 10.8 51.6  96.8 284.0 1331.7 3369.8  753.3
Colorado        2.50929  0.91660  6.3 42.0 170.7 292.9 1935.2 3903.2  477.1
Arizona         3.01414  0.84495  9.5 34.2 138.2 312.3 2346.1 4467.4  439.5
Florida         3.11175 -0.60392 10.2 39.6 187.9 449.1 1859.9 3840.5  351.4
New York        3.45248  0.43289 10.7 29.4 472.6 319.1 1728.0 2782.0  745.8
California      4.28380  0.14319 11.5 49.4 287.0 358.0 2139.4 3499.8  663.5
Nevada          5.26699 -0.25262 15.8 49.1 323.1 355.0 2453.1 4212.6  559.2
```

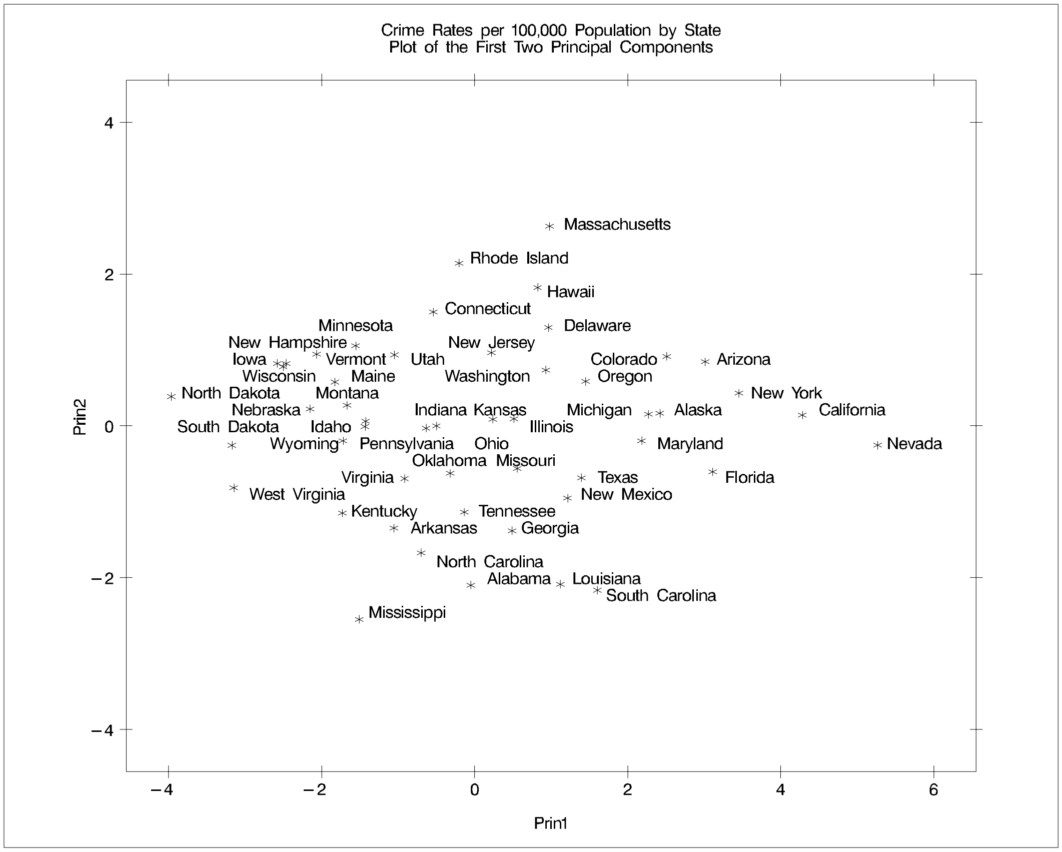
Output 58.2.3. OUT= Data Set Sorted by Second Principal Component

```
                Crime Rates per 100,000 Population by State
            States Listed in Order of Property Vs. Violent Crime
              As Determined by the Second Principal Component

                                                                        A
                                                                        u
                                                          B             t
                                             R     A      u      L      o
                                    M        o     s      r      a      _
S                  P        P       u        b     s      g      r      T
t                  r        r       r   R    b     a      l      c      h
a                  i        i       d   a    e     u      a      e      e
t                  n        n       e   p    r     l      r      n      f
e                  1        2       r   e    y     t      y      y      t

Mississippi    -1.50736 -2.54671 14.3 19.6  65.7 189.1  915.6 1239.9  144.4
South Carolina  1.60336 -2.16211 11.9 33.0 105.9 485.3 1613.6 2342.4  245.1
Alabama        -0.04988 -2.09610 14.2 25.2  96.8 278.3 1135.5 1881.9  280.7
Louisiana       1.12020 -2.08327 15.5 30.9 142.9 335.5 1165.5 2469.9  337.7
North Carolina -0.69925 -1.67027 10.6 17.0  61.3 318.3 1154.1 2037.8  192.1
Georgia         0.49041 -1.38079 11.7 31.1 140.5 256.5 1351.1 2170.2  297.9
Arkansas       -1.05441 -1.34544  8.8 27.6  83.2 203.4  972.6 1862.1  183.4
Kentucky       -1.72691 -1.14663 10.1 19.1  81.1 123.3  872.2 1662.1  245.4
Tennessee      -0.13660 -1.13498 10.1 29.7 145.8 203.9 1259.7 1776.5  314.0
New Mexico      1.21417 -0.95076  8.8 39.1 109.6 343.4 1418.7 3008.6  259.5
West Virginia  -3.14772 -0.81425  6.0 13.2  42.2  90.9  597.4 1341.7  163.3
Virginia       -0.91621 -0.69265  9.0 23.3  92.1 165.7  986.2 2521.2  226.7
Texas           1.39696 -0.68131 13.3 33.8 152.4 208.2 1603.1 2988.7  397.6
Oklahoma       -0.32136 -0.62429  8.6 29.2  73.8 205.0 1288.2 2228.1  326.8
Florida         3.11175 -0.60392 10.2 39.6 187.9 449.1 1859.9 3840.5  351.4
Missouri        0.55637 -0.55851  9.6 28.3 189.0 233.5 1318.3 2424.2  378.4
South Dakota   -3.17203 -0.25446  2.0 13.5  17.9 155.7  570.5 1704.4  147.5
Nevada          5.26699 -0.25262 15.8 49.1 323.1 355.0 2453.1 4212.6  559.2
Pennsylvania   -1.72007 -0.19590  5.6 19.0 130.3 128.0  877.5 1624.1  333.2
Maryland        2.18280 -0.19474  8.0 34.8 292.1 358.9 1400.0 3177.7  428.5
Kansas         -0.63407 -0.02804  6.6 22.0 100.7 180.5 1270.4 2739.3  244.3
Idaho          -1.43245 -0.00801  5.5 19.4  39.6 172.5 1050.8 2599.6  237.6
Indiana        -0.49990  0.00003  7.4 26.5 123.2 153.5 1086.2 2498.7  377.4
Wyoming        -1.42463  0.06268  5.4 21.9  39.7 173.9  811.6 2772.2  282.0
Ohio            0.23953  0.09053  7.8 27.3 190.5 181.1 1216.0 2696.8  400.4
Illinois        0.51290  0.09423  9.9 21.8 211.3 209.0 1085.0 2828.5  528.6
California      4.28380  0.14319 11.5 49.4 287.0 358.0 2139.4 3499.8  663.5
Michigan        2.27333  0.15487  9.3 38.9 261.9 274.6 1522.7 3159.0  545.5
Alaska          2.42151  0.16652 10.8 51.6  96.8 284.0 1331.7 3369.8  753.3
Nebraska       -2.15071  0.22574  3.9 18.1  64.7 112.7  760.0 2316.1  249.1
Montana        -1.66801  0.27099  5.4 16.7  39.2 156.8  804.9 2773.2  309.2
North Dakota   -3.96408  0.38767  0.9  9.0  13.3  43.8  446.1 1843.0  144.7
New York        3.45248  0.43289 10.7 29.4 472.6 319.1 1728.0 2782.0  745.8
Maine          -1.82631  0.57878  2.4 13.5  38.7 170.0 1253.1 2350.7  246.9
Oregon          1.44900  0.58603  4.9 39.9 124.1 286.9 1636.4 3506.1  388.9
Washington      0.93058  0.73776  4.3 39.6 106.2 224.8 1605.6 3386.9  360.3
Wisconsin      -2.50296  0.78083  2.8 12.9  52.2  63.7  846.9 2614.2  220.7
Iowa           -2.58156  0.82475  2.3 10.6  41.2  89.8  812.5 2685.1  219.9
New Hampshire  -2.46562  0.82503  3.2 10.7  23.2  76.0 1041.7 2343.9  293.4
Arizona         3.01414  0.84495  9.5 34.2 138.2 312.3 2346.1 4467.4  439.5
Colorado        2.50929  0.91660  6.3 42.0 170.7 292.9 1935.2 3903.2  477.1
Utah           -1.04996  0.93656  3.5 20.3  68.8 147.3 1171.6 3004.6  334.5
Vermont        -2.06433  0.94497  1.4 15.9  30.8 101.2 1348.2 2201.0  265.2
New Jersey      0.21787  0.96421  5.6 21.0 180.4 185.1 1435.8 2774.5  511.5
Minnesota      -1.55434  1.05644  2.7 19.5  85.9  85.8 1134.7 2559.3  343.1
Delaware        0.96458  1.29674  6.0 24.9 157.0 194.2 1682.6 3678.4  467.0
Connecticut    -0.54133  1.50123  4.2 16.8 129.5 131.8 1346.0 2620.7  593.2
Hawaii          0.82313  1.82392  7.2 25.5 128.0  64.1 1911.5 3920.4  489.4
Rhode Island   -0.20156  2.14658  3.6 10.5  86.5 201.0 1489.5 2844.1  791.4
Massachusetts   0.97844  2.63105  3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
```

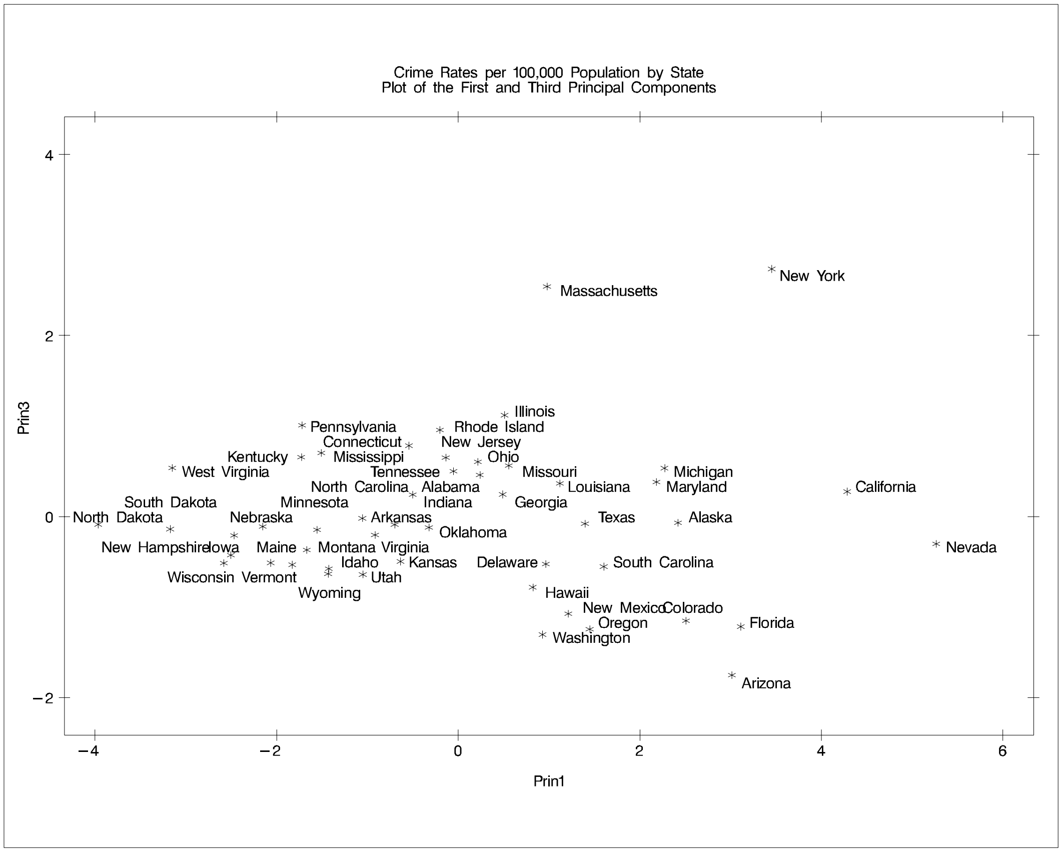
Another recommended procedure is to make scatter plots of the first few components. The sorted listings help to identify observations on the plots. The following statements produce Output 58.2.4 through Output 58.2.5:

```
title2 'Plot of the First Two Principal Components';

%plotit(data=Crime_Components,labelvar=State,
        plotvars=Prin2 Prin1, color=black, colors=black);

run;


title2 'Plot of the First and Third Principal Components';

%plotit(data=Crime_Components,labelvar=State,
        plotvars=Prin3 Prin1, color=black, colors=black);
run;
```

Output 58.2.4. Plot of the First Two Principal Components



Output 58.2.5. Plot of the First and Third Principal Components



It is possible to identify regional trends on the plot of the first two components. Nevada and California are at the extreme right, with high overall crime rates but an average ratio of property crime to violent crime. North and South Dakota are on the extreme left with low overall crime rates. Southeastern states tend to be in the bottom of the plot, with a higher-than-average ratio of violent crime to property crime. New England states tend to be in the upper part of the plot, with a greater-than-average ratio of property crime to violent crime.

The most striking feature of the plot of the first and third principal components is that Massachusetts and New York are outliers on the third component.

Example 58.3. Basketball Data

The data in this example are rankings of 35 college basketball teams. The rankings were made before the start of the 1985–86 season by 10 news services.

The purpose of the principal component analysis is to compute a single variable that best summarizes all 10 of the preseason rankings.

Note that the various news services rank different numbers of teams, varying from 20 through 30 (there is a missing rank in one of the variables, WashPost). And, of course, each service does not rank the same teams, so there are missing values in these data. Each of the 35 teams is ranked by at least one news service.

The PRINCOMP procedure omits observations with missing values. To obtain principal component scores for all of the teams, it is necessary to replace the missing values. Since it is the best teams that are ranked, it is not appropriate to replace missing values with the mean of the nonmissing values. Instead, an ad hoc method is used that replaces missing values by the mean of the unassigned ranks. For example, if 20 teams are ranked by a news service, then ranks 21 through 35 are unassigned. The mean of ranks 21 through 35 is 28, so missing values for that variable are replaced by the value 28. To prevent the method of missing-value replacement from having an undue effect on the analysis, each observation is weighted according to the number of nonmissing values it has. See Example 59.3 in Chapter 59, "The PRINQUAL Procedure," for an alternative analysis of these data.

Since the first principal component accounts for 78% of the variance, there is substantial agreement among the rankings. The eigenvector shows that all the news services are about equally weighted, so a simple average would work almost as well as the first principal component. The following statements produce Output 58.3.1 through Output 58.3.3:

```
/*-----------------------------------------------------------*/
/*                                                           */
/* Preseason 1985 College Basketball Rankings                */
/* (rankings of 35 teams by 10 news services)                */
/*                                                           */
/* Note: (a) news services rank varying numbers of teams;    */
/*       (b) not all teams are ranked by all news services;  */
/*       (c) each team is ranked by at least one service;    */
/*       (d) rank 20 is missing for UPI.                     */
/*                                                           */
/*-----------------------------------------------------------*/

data HoopsRanks;
   input School $13. CSN DurSun DurHer WashPost USAToday
         Sport InSports UPI AP SI;
   label CSN      = 'Community Sports News (Chapel Hill, NC)'
         DurSun   = 'Durham Sun'
         DurHer   = 'Durham Morning Herald'
         WashPost = 'Washington Post'
         USAToday = 'USA Today'
         Sport    = 'Sport Magazine'
         InSports = 'Inside Sports'
         UPI      = 'United Press International'
         AP       = 'Associated Press'
         SI       = 'Sports Illustrated'
         ;
   format CSN--SI 5.1;
   cards;
Louisville     1  8  1  9  8  9  6 10  9  9
Georgia Tech   2  2  4  3  1  1  1  2  1  1
Kansas         3  4  5  1  5 11  8  4  5  7
Michigan       4  5  9  4  2  5  3  1  3  2
Duke           5  6  7  5  4 10  4  5  6  5
UNC            6  1  2  2  3  4  2  3  2  3
```

```
Syracuse       7 10  6 11  6  6  5  6  4 10
Notre Dame     8 14 15 13 11 20 18 13 12  .
Kentucky       9 15 16 14 14 19 11 12 11 13
LSU           10  9 13  . 13 15 16  9 14  8
DePaul        11  . 21 15 20  . 19  .  . 19
Georgetown    12  7  8  6  9  2  9  8  8  4
Navy          13 20 23 10 18 13 15  . 20  .
Illinois      14  3  3  7  7  3 10  7  7  6
Iowa          15 16  .  . 23  .  . 14  . 20
Arkansas      16  .  .  . 25  .  .  .  . 16
Memphis State 17  . 11  . 16  8 20  . 15 12
Washington    18  .  .  .  .  .  . 17  .  .
UAB           19 13 10  . 12 17  . 16 16 15
UNLV          20 18 18 19 22  . 14 18 18  .
NC State      21 17 14 16 15  . 12 15 17 18
Maryland      22  .  .  . 19  .  .  . 19 14
Pittsburgh    23  .  .  .  .  .  .  .  .  .
Oklahoma      24 19 17 17 17 12 17  . 13 17
Indiana       25 12 20 18 21  .  .  .  .  .
Virginia      26  . 22  .  . 18  .  .  .  .
Old Dominion  27  .  .  .  .  .  .  .  .  .
Auburn        28 11 12  8 10  7  7 11 10 11
St. Johns     29  .  .  .  . 14  .  .  .  .
UCLA          30  .  .  .  .  .  . 19  .  .
St. Joseph’s   .  . 19  .  .  .  .  .  .  .
Tennessee      .  . 24  .  . 16  .  .  .  .
Montana        .  .  . 20  .  .  .  .  .  .
Houston        .  .  .  . 24  .  .  .  .  .
Virginia Tech  .  .  .  .  .  . 13  .  .  .
;

   /* PROC MEANS is used to output a data set containing the      */
   /* maximum value of each of the newspaper and magazine         */
   /* rankings.  The output data set, maxrank, is then used       */
   /* to set the missing values to the next highest rank plus     */
   /* thirty-six, divided by two (that is, the mean of the        */
   /* missing ranks).  This ad hoc method of replacing missing    */
   /* values is based more on intuition than on rigorous          */
   /* statistical theory.  Observations are weighted by the       */
   /* number of nonmissing values.                                */
   /*                                                             */
```

```
title ’Pre-Season 1985 College Basketball Rankings’;
proc means data=HoopsRanks;
   output out=MaxRank
          max=CSNMax DurSunMax DurHerMax
              WashPostMax USATodayMax SportMax
              InSportsMax UPIMax APMax SIMax;
run;
```

Output 58.3.1. Summary Statistics for Basketball Rankings Using PROC MEANS

```
                  Pre-Season 1985 College Basketball Rankings

                              The MEANS Procedure

    Variable   Label                                      N            Mean
    -----------------------------------------------------------------------
    CSN        Community Sports News (Chapel Hill, NC)   30      15.5000000
    DurSun     Durham Sun                                20      10.5000000
    DurHer     Durham Morning Herald                     24      12.5000000
    WashPost   Washington Post                           19      10.4210526
    USAToday   USA Today                                 25      13.0000000
    Sport      Sport Magazine                            20      10.5000000
    InSports   Inside Sports                             20      10.5000000
    UPI        United Press International                19      10.0000000
    AP         Associated Press                          20      10.5000000
    SI         Sports Illustrated                        20      10.5000000
    -----------------------------------------------------------------------

Variable   Label                                            Std Dev         Minimum
--------------------------------------------------------------------------------
CSN        Community Sports News (Chapel Hill, NC)      8.8034084       1.0000000
DurSun     Durham Sun                                   5.9160798       1.0000000
DurHer     Durham Morning Herald                        7.0710678       1.0000000
WashPost   Washington Post                              6.0673607       1.0000000
USAToday   USA Today                                    7.3598007       1.0000000
Sport      Sport Magazine                               5.9160798       1.0000000
InSports   Inside Sports                                5.9160798       1.0000000
UPI        United Press International                   5.6273143       1.0000000
AP         Associated Press                             5.9160798       1.0000000
SI         Sports Illustrated                           5.9160798       1.0000000
--------------------------------------------------------------------------------

       Variable   Label                                          Maximum
       -----------------------------------------------------------------
       CSN        Community Sports News (Chapel Hill, NC)     30.0000000
       DurSun     Durham Sun                                  20.0000000
       DurHer     Durham Morning Herald                       24.0000000
       WashPost   Washington Post                             20.0000000
       USAToday   USA Today                                   25.0000000
       Sport      Sport Magazine                              20.0000000
       InSports   Inside Sports                               20.0000000
       UPI        United Press International                  19.0000000
       AP         Associated Press                            20.0000000
       SI         Sports Illustrated                          20.0000000
       -----------------------------------------------------------------
```

```
   data Basketball;
      set HoopsRanks;
      if _n_=1 then set MaxRank;
      array Services{10} CSN--SI;
      array MaxRanks{10} CSNMax--SIMax;
      keep School CSN--SI Weight;
      Weight=0;
      do i=1 to 10;
         if Services{i}=. then Services{i}=(MaxRanks{i}+36)/2;
         else Weight=Weight+1;
      end;
   run;
```

```
proc princomp data=Basketball n=1 out=PCBasketball standard;
   var CSN--SI;
   weight Weight;
run;
```

Output 58.3.2. Principal Components Analysis of Basketball Rankings Using PROC PRINCOMP

```
                              The PRINCOMP Procedure

                             Observations          35
                             Variables             10


                                Simple Statistics

                   CSN           DurSun           DurHer         WashPost         USAToday

Mean       13.33640553      13.06451613      12.88018433      13.83410138      12.55760369
StD        22.08036285      21.66394183      21.38091837      23.47841791      20.48207965

                                Simple Statistics

                 Sport         InSports              UPI               AP               SI

Mean       13.83870968      13.24423963      13.59216590      12.83410138      13.52534562
StD        23.37756267      22.20231526      23.25602811      21.40782406      22.93219584
```

Output 58.3.2. (continued)

```
                              Correlation Matrix

                                                           CSN    DurSun    DurHer

 CSN         Community Sports News (Chapel Hill, NC)    1.0000    0.6505    0.6415
 DurSun      Durham Sun                                 0.6505    1.0000    0.8341
 DurHer      Durham Morning Herald                      0.6415    0.8341    1.0000
 WashPost    Washington Post                            0.6121    0.7667    0.7035
 USAToday    USA Today                                  0.7456    0.8860    0.8877
 Sport       Sport Magazine                             0.4806    0.6940    0.7788
 InSports    Inside Sports                              0.6558    0.7702    0.7900
 UPI         United Press International                 0.7007    0.9015    0.7676
 AP          Associated Press                           0.6779    0.8437    0.8788
 SI          Sports Illustrated                         0.6135    0.7518    0.7761

                              Correlation Matrix

             Wash                               In
             Post    USAToday     Sport     Sports       UPI        AP        SI

CSN        0.6121      0.7456    0.4806     0.6558    0.7007    0.6779    0.6135
DurSun     0.7667      0.8860    0.6940     0.7702    0.9015    0.8437    0.7518
DurHer     0.7035      0.8877    0.7788     0.7900    0.7676    0.8788    0.7761
WashPost   1.0000      0.7984    0.6598     0.8717    0.6953    0.7809    0.5952
USAToday   0.7984      1.0000    0.7716     0.8475    0.8539    0.9479    0.8426
Sport      0.6598      0.7716    1.0000     0.7176    0.6220    0.8217    0.7701
InSports   0.8717      0.8475    0.7176     1.0000    0.7920    0.8830    0.7332
UPI        0.6953      0.8539    0.6220     0.7920    1.0000    0.8436    0.7738
AP         0.7809      0.9479    0.8217     0.8830    0.8436    1.0000    0.8212
SI         0.5952      0.8426    0.7701     0.7332    0.7738    0.8212    1.0000



                      Eigenvalues of the Correlation Matrix

                  Eigenvalue    Difference    Proportion    Cumulative

             1    7.88601647                      0.7886        0.7886


                                 Eigenvectors

                                                                       Prin1

      CSN           Community Sports News (Chapel Hill, NC)         0.270205
      DurSun        Durham Sun                                      0.326048
      DurHer        Durham Morning Herald                           0.324392
      WashPost      Washington Post                                 0.300449
      USAToday      USA Today                                       0.345200
      Sport         Sport Magazine                                  0.293881
      InSports      Inside Sports                                   0.324088
      UPI           United Press International                      0.319902
      AP            Associated Press                                0.342151
      SI            Sports Illustrated                              0.308570
```

```
proc sort data=PCBasketball;
   by Prin1;
run;

proc print;
   var School Prin1;
   title ’Pre-Season 1985 College Basketball Rankings’;
   title2 ’College Teams as Ordered by PROC PRINCOMP’;
run;
```

Output 58.3.3. Basketball Rankings Using PROC PRINCOMP

```
                   Pre-Season 1985 College Basketball Rankings
                    College Teams as Ordered by PROC PRINCOMP

                         OBS    School              Prin1

                           1    Georgia Tech     -0.58068
                           2    UNC              -0.53317
                           3    Michigan         -0.47874
                           4    Kansas           -0.40285
                           5    Duke             -0.38464
                           6    Illinois         -0.33586
                           7    Syracuse         -0.31578
                           8    Louisville       -0.31489
                           9    Georgetown       -0.29735
                          10    Auburn           -0.09785
                          11    Kentucky          0.00843
                          12    LSU               0.00872
                          13    Notre Dame        0.09407
                          14    NC State          0.19404
                          15    UAB               0.19771
                          16    Oklahoma          0.23864
                          17    Memphis State     0.25319
                          18    Navy              0.28921
                          19    UNLV              0.35103
                          20    DePaul            0.43770
                          21    Iowa              0.50213
                          22    Indiana           0.51713
                          23    Maryland          0.55910
                          24    Arkansas          0.62977
                          25    Virginia          0.67586
                          26    Washington        0.67756
                          27    Tennessee         0.70822
                          28    St. Johns         0.71425
                          29    Virginia Tech     0.71638
                          30    St. Joseph’s      0.73492
                          31    UCLA              0.73965
                          32    Pittsburgh        0.75078
                          33    Houston           0.75534
                          34    Montana           0.75790
                          35    Old Dominion      0.76821
```

Example 58.4. PRINCOMP Graphics (Experimental)

This example illustrates the experimental ODS graphics in PROC PRINCOMP, using the example in the the “Getting Started” section on page 3596.

The following statements request plots in PROC PRINCOMP.

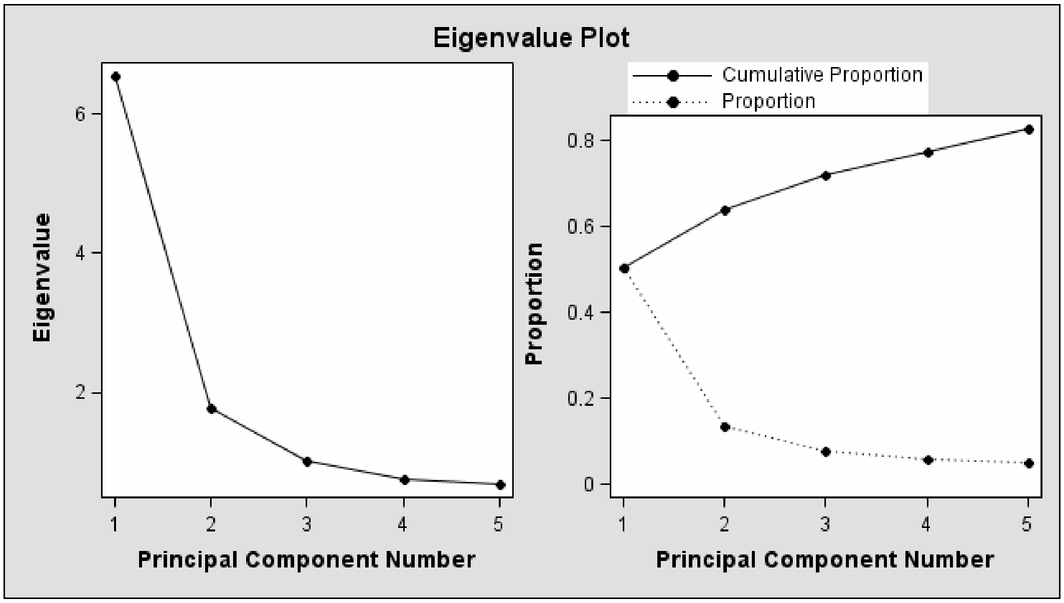
```
ods html;
ods graphics on;

proc princomp data=Jobratings(drop='Overall Rating'n) n=5;
   run;

ods graphics off;
ods html close;
```

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the PRINCOMP procedure, see the “ODS Graphics” section on page 3613.

The N= 5 option in the PROC PRINCOMP statement sets the number of principal components to 5.

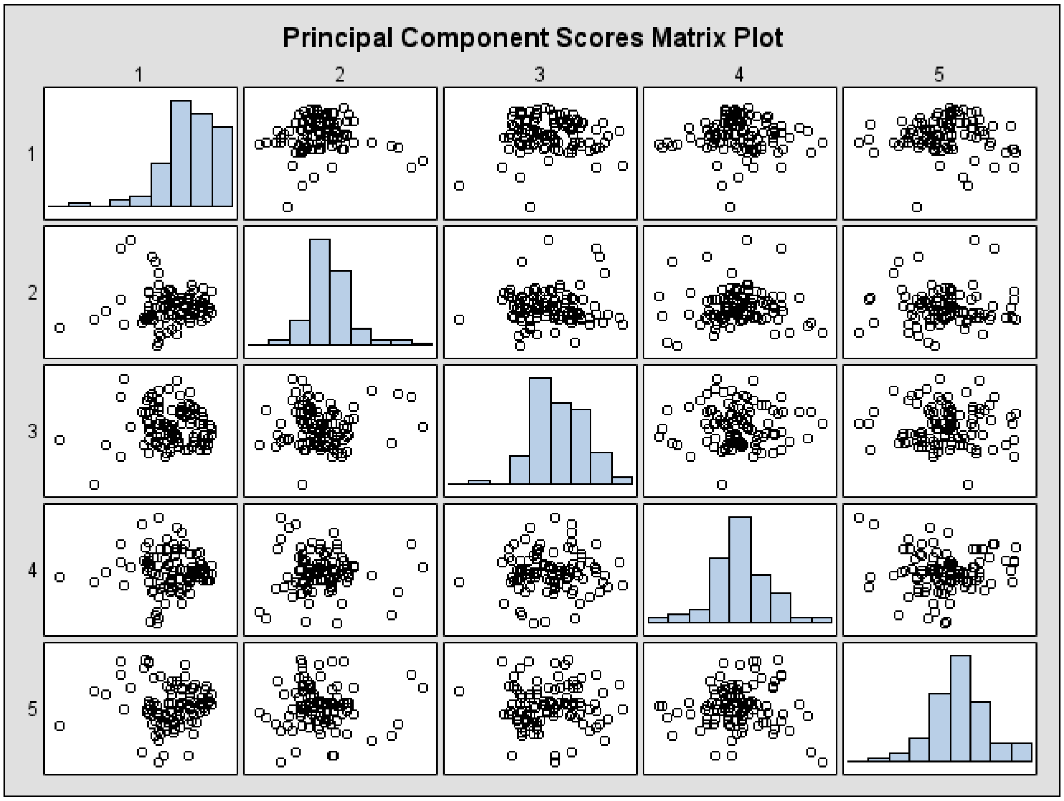
Output 58.4.1 shows the eigenvalue plots. Each point in the plot on the left shows an eigenvalue; each point in the plot on the right shows the (cumulative) proportion of variance explained by each component.

Output 58.4.1. Eigenvalue Scatter Plot (Experimental)



Output 58.4.2 shows a scatter matrix plot between the first five components. The histogram of each component is displayed in the diagonal element of the matrix.

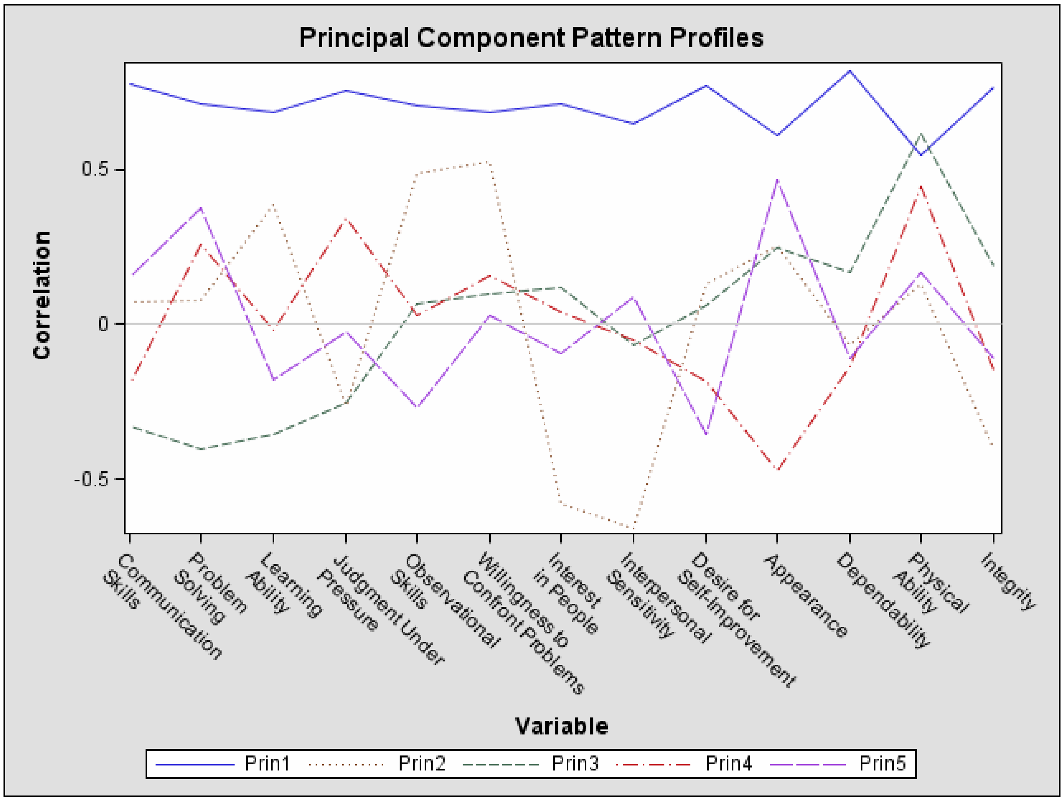
Output 58.4.2. Component Scores Matrix Plot (Experimental)



Output 58.4.3 shows a component pattern profile. The Y-axis shows the correlation between a component and a variable. There is one profile for each component. Line patterns are used to differentiate correlations between components.

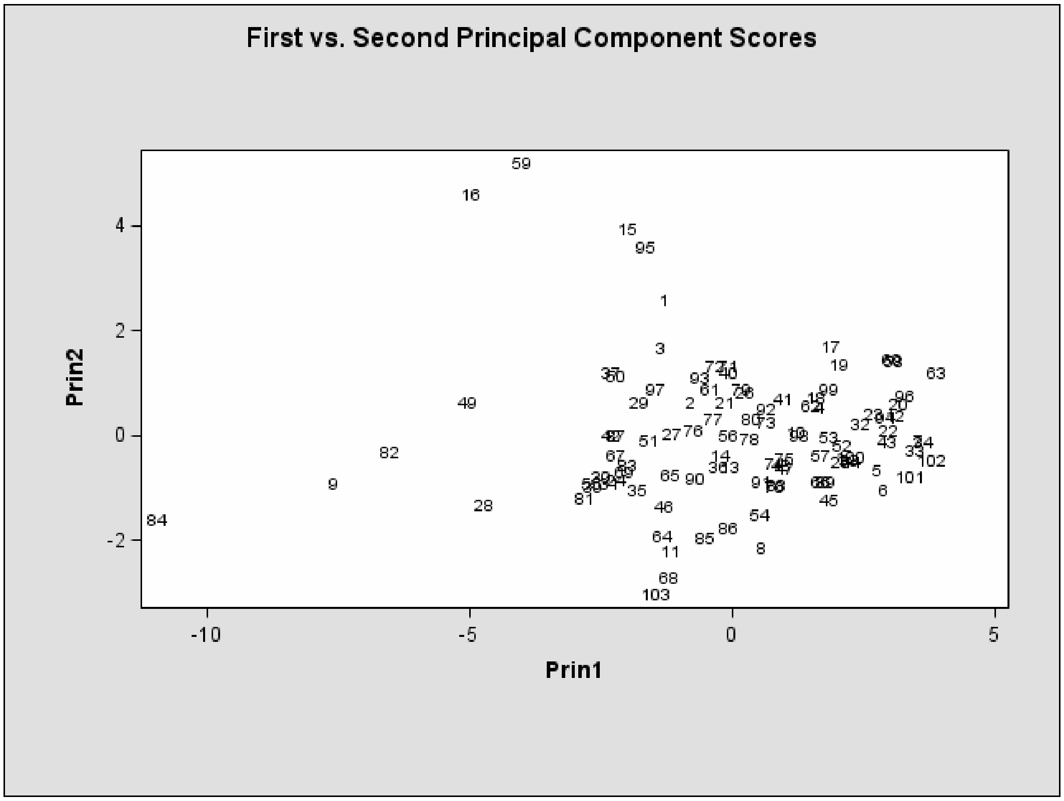
The nearly horizontal profile from the first component indicates that the first component is mostly correlated evenly across all variables. The second component is positively correlated with the variables Observational Skills and Willingness to Confront Problems and is negatively correlated with the variables Interest in People and Interpersonal Sensitivity.

Output 58.4.3. Component Pattern Plot (Experimental)



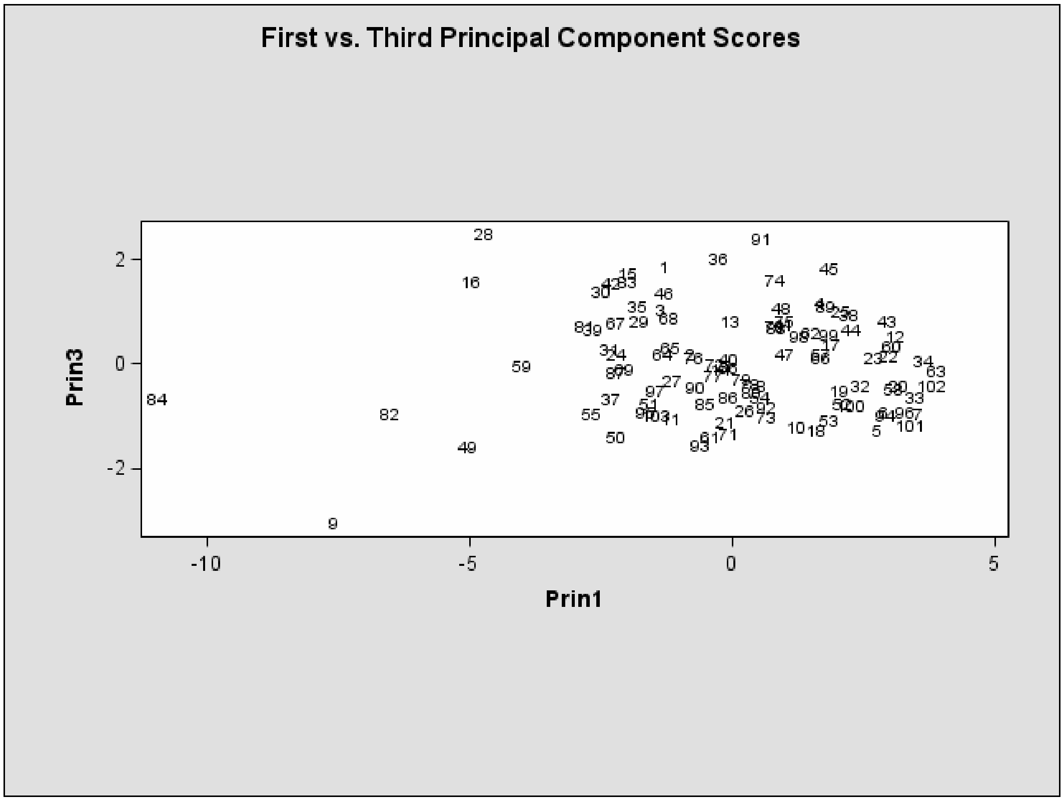
Output 58.4.4 shows a scatter plot of the first and second components. Observation numbers are used as the plotting symbol.

Output 58.4.4. Component Scores Plot: 1st versus 2nd (Experimental)



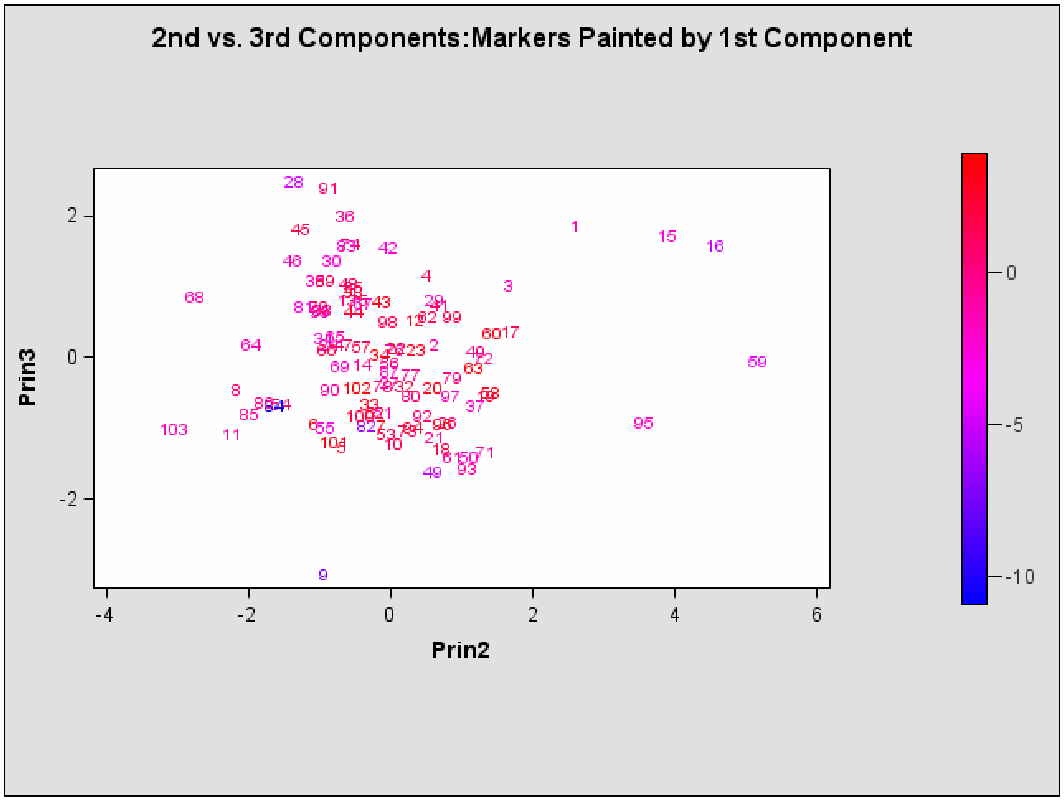
[Output 58.4.5](#) shows a scatter plot of the first and third components.

Output 58.4.5. Component Scores Plot: 1st versus 3rd (Experimental)



Output 58.4.6 shows a scatter plot of the second and third components, displaying density with color. Color interpolation is based on the first component, going from blue (or light gray) (minimum density), magenta (or dark gray) (median density), and to red (or black) (maximum density).

Output 58.4.6. Painted Components Scores Plot: 2nd versus 3rd, Painted by 1st (Experimental)



References

Cooley, W.W. and Lohnes, P.R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons, Inc.

Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, New York: John Wiley & Sons, Inc.

Hotelling, H. (1933), “Analysis of a Complex of Statistical Variables into Principal Components,” *Journal of Educational Psychology*, 24, 417–441, 498–520.

Kshirsagar, A.M. (1972), *Multivariate Analysis*, New York: Marcel Dekker, Inc.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis*, London: Academic Press.

Morrison, D.F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill Book Co.

Pearson, K. (1901), “On Lines and Planes of Closest Fit to Systems of Points in Space,” *Philosophical Magazine*, 6(2), 559–572.

Rao, C.R. (1964), “The Use and Interpretation of Principal Component Analysis in Applied Research,” *Sankhya A*, 26, 329–358.

Chapter 59
The PRINQUAL Procedure

Chapter Contents

Chapter 59
The PRINQUAL Procedure

Overview

The PRINQUAL procedure obtains linear and nonlinear transformations of variables by using the method of alternating least squares to optimize properties of the transformed variables' covariance or correlation matrix. Nonoptimal transformations for logarithm, rank, exponentiation, inverse sine, and logit are also available with PROC PRINQUAL. Experimental graphics are now available with the PRINQUAL procedure. For more information, see the "ODS Graphics" section on page 3677.

The PRINQUAL (principal components of qualitative data) procedure is a data transformation procedure that is based on the work of Kruskal and Shepard (1974); Young, Takane, and de Leeuw (1978); Young (1981); and Winsberg and Ramsay (1983). You can use PROC PRINQUAL to

- generalize ordinary principal component analysis to a method capable of analyzing data that are not quantitative
- perform metric and nonmetric multidimensional preference (MDPREF) analyses (Carroll 1972)
- preprocess data, transforming variables prior to their use in other data analyses
- summarize mixed quantitative and qualitative data and detect nonlinear relationships
- reduce the number of variables for subsequent use in regression analyses, cluster analyses, and other analyses

The PRINQUAL procedure provides three methods of transforming a set of qualitative and quantitative variables to optimize the transformed variables' covariance or correlation matrix. These methods are

- maximum total variance (MTV)
- minimum generalized variance (MGV)
- maximum average correlation (MAC)

All three methods attempt to find transformations that decrease the rank of the covariance matrix computed from the transformed variables. Transforming the variables to maximize the variance accounted for by a few linear combinations (using the MTV method) locates the observations in a space with dimensionality that approximates the stated number of linear combinations as much as possible, given the transformation constraints. Transforming the variables to minimize their generalized variance or maximize the sum of correlations also reduces the dimensionality. The transformed qualitative (nominal and ordinal) variables can be thought of as quantified by the analysis, with the quantification done in the context set by the algorithm. The data are quantified so that the proportion of variance accounted for by a stated number of principal components is locally maximal, the generalized variance of the variables is locally minimal, or the average of the correlations is locally maximal.

The data can contain variables with nominal, ordinal, interval, and ratio scales of measurement (Siegel 1956). Any mix is allowed with all methods. PROC PRINQUAL can

- transform nominal variables by scoring the categories to optimize the covariance matrix (Fisher 1938)
- transform ordinal variables monotonically by scoring the ordered categories so that order is weakly preserved (adjacent categories can be merged) and the covariance matrix is optimized. You can untie ties optimally or leave them tied (Kruskal 1964). You can also transform ordinal variables to ranks.
- transform interval and ratio scale of measurement variables linearly, or transform them nonlinearly with spline transformations (de Boor 1978; van Rijckevorsel 1982) or monotone spline transformations (Winsberg and Ramsay 1983). In addition, nonoptimal transformations for logarithm, exponential, power, logit, and inverse trigonometric sine are available.
- for all transformations, estimate missing data without constraint, with category constraints (missing values within the same group get the same value), and with order constraints (missing value estimates in adjacent groups can be tied to preserve a specified ordering). Refer to Gifi (1990) and Young (1981).

The PROC PRINQUAL iterations produce a set of transformed variables. Each variable's new scoring satisfies a set of constraints based on the original scoring of the variable and the specified transformation type. First, all variables are required to satisfy transformation standardization constraints; that is, all variables have a fixed mean and variance. The other constraints include linear constraints, weak order constraints, category constraints, and smoothness constraints. The new set of scores is selected from the sets of possible scorings that do not violate the constraints so that the method criterion is locally optimized.

The displayed output from PROC PRINQUAL is a listing of the iteration history. However, the primary output from PROC PRINQUAL is an output data set. By default, the procedure creates an output data set that contains variables with _TYPE_='SCORE'. These observations contain original variables, transformed variables, components, or data approximations. If you specify the CORRELATIONS option in the PROC PRINQUAL statement, the data set also contains observations with _TYPE_='CORR'; these observations contain correlations or component structure information.

The Three Methods of Variable Transformation

The three methods of variable transformation provided by PROC PRINQUAL are discussed in the following sections.

The Maximum Total Variance (MTV) Method

The MTV method (Young, Takane, and de Leeuw 1978) is based on the principal component model, and it attempts to maximize the sum of the first r eigenvalues of the covariance matrix. This method transforms variables to be (in a least-squares sense) as similar to linear combinations of r principal component score variables as possible, where r can be much smaller than the number of variables. This maximizes the total variance of the first r components (the trace of the covariance matrix of the first r principal components). Refer to Kuhfeld, Sarle, and Young (1985).

On each iteration, the MTV algorithm alternates classical principal component analysis (Hotelling 1933) with optimal scaling (Young 1981). When all variables are ordinal preference ratings, this corresponds to Carroll's (1972) MDPREF analysis. You can request the dummy variable initialization method suggested by Tenenhaus and Vachette (1977), who independently proposed the same iterative algorithm for nominal and interval scale-of-measurement variables.

The Minimum Generalized Variance (MGV) Method

The MGV method (Sarle 1984) uses an iterated multiple regression algorithm in an attempt to minimize the determinant of the covariance matrix of the transformed variables. This method transforms each variable to be (in a least-squares sense) as similar to linear combinations of the remaining variables as possible. This locally minimizes the generalized variance of the transformed variables, the determinant of the covariance matrix, the volume of the parallelepiped defined by the transformed variables, and the sphericity (the extent to which a quadratic form in the optimized covariance matrix defines a sphere). Refer to Kuhfeld, Sarle, and Young (1985).

On each iteration for each variable, the MGV algorithm alternates multiple regression with optimal scaling. The multiple regression involves predicting the selected variable from all other variables. You can request a dummy variable initialization using a modification of the Tenenhaus and Vachette (1977) method that is appropriate with a regression algorithm. This method can be viewed as a way of investigating the nature of the linear and nonlinear dependencies in, and the rank of, a data matrix containing variables that can be nonlinearly transformed. This method tries to create a less-than-full-rank data matrix. The matrix contains the transformation of each variable that is most similar to what the other transformed variables predict.

The Maximum Average Correlation (MAC) Method

The MAC method (de Leeuw 1985) uses an iterated constrained multiple regression algorithm in an attempt to maximize the average of the elements of the correlation matrix. This method transforms each variable to be (in a least-squares sense) as similar to the average of the remaining variables as possible.

On each iteration for each variable, the MAC algorithm alternates computing an equally weighted average of the other variables with optimal scaling. The MAC method is similar to the MGV method in that each variable is scaled to be as similar to a linear combination of the other variables as possible, given the constraints on the transformation. However, optimal weights are not computed. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. Do not use this method with negatively correlated variables when some optimal transformations are constrained to be increasing because the signs of the correlations are not taken into account. The MAC method is useful as an initialization method for the MTV and MGV methods.

Getting Started

In the following example, PROC PRINQUAL uses the MTV method. Suppose that the problem is to linearize a curve through three-dimensional space. Let

$$
\begin{aligned}
\mathrm{X}_1 &= \mathrm{X}^3 \\
\mathrm{X}_2 &= \mathrm{X}_1 - \mathrm{X}^5 \\
\mathrm{X}_3 &= \mathrm{X}_2 - \mathrm{X}^6
\end{aligned}
$$

where $\mathrm{X} = -1.00, -0.98, -0.96, \ldots, 1.00$.

These three variables define a curve in three-dimensional space. The GPLOT procedure is used to display two-dimensional views of this curve. These data are completely described by three linear components, but they define a single curve, which could be described as a single nonlinear component.

PROC PRINQUAL is used to attempt to straighten the curve into a one-dimensional line with a continuous transformation of each variable. The N=1 option in the PROC PRINQUAL statement requests one principal component. The TRANSFORM statement requests a cubic spline transformation with nine knots. *Splines* are curves, which are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n - 1$ derivatives that agree at the points where they join. The abscissa values of the join points are called *knots*. The term “spline” is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Knots give the curve freedom to bend to more closely follow the data. Refer to Smith (1979) for an excellent introduction to splines. For another example of using splines, see Example 75.1 in Chapter 75, “The TRANSREG Procedure.”

One component accounts for 71 percent of the variance of the untransformed data, and after 45 iterations, over 98 percent of the variance of the transformed data is accounted for by one component (see Figure 59.2). Note that the algorithm would not have converged with 50 iterations and the default convergence criterion, so more iterations may be needed for this problem.

PROC PRINQUAL creates an output data set (which is not displayed) that contains both the original and transformed variables. The original variables have the names X1, X2, and X3. Transformed variables are named TX1, TX2, and TX3. All observations in the output data set have _TYPE_='SCORE', since the CORRELATIONS option is not specified in the PROC PRINQUAL statement. The GPLOT procedure uses this output data set and displays the nonlinear transformations of all three variables and the nearly one-dimensional scatter plot (see Figure 59.3 and Figure 59.4).

PROC PRINQUAL tries to project each variable on the first principal component. Notice that the curve in this example is closer to a circle than to a function from some views (see the plot of X3 vs. X2 in Figure 59.1) and that the first component does not run approximately from one end point of the curve to the other (see Figure 59.4). Since the curve has these characteristics, PROC PRINQUAL linearizes the scatter plot by collapsing the scatter around the principal axis, not by straightening the curve into a single line. PROC PRINQUAL would straighten simpler curves.

The following statements produce Figure 59.1 through Figure 59.4:

```
* Generate a Three-Dimensional Curve;
data X;
   do X = -1 to 1 by 0.02;
      X1 =      X ** 3;
      X2 = X1 - X ** 5;
      X3 = X2 - X ** 6;
      output;
   end;
   drop X;
run;

goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot data=X;
   title;
   axis1 minor=none label=(angle=90 rotate=0)
         order=(-1 to 1);
   axis2 minor=none order=(-1 to 1);
   plot X1*X2 / &opts name='prqin1';
   plot X3*X2 / &opts name='prqin2' vreverse;
   plot X1*X3 / &opts name='prqin3';
   symbol1 color=blue;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:prqin1 2:prqin2 3:prqin3;
run; quit;
```

```
* Try to Straighten the Curve;
proc prinqual data=X n=1 maxiter=50 covariance converge=0.007;
   title ’Iteratively Derive Variable Transformations’;
   transform spline(X1-X3 / nknots=9);
run;

* Plot the Transformations;
goptions nodisplay;
proc gplot;
   title;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   plot TX1*X1 / &opts name=’prqin4’;
   plot TX2*X2 / &opts name=’prqin5’;
   plot TX3*X3 / &opts name=’prqin6’;
   symbol1 color=blue;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:prqin4 2:prqin6 3:prqin5;
run; quit;

* Plot the Straightened Scatter Plot;
goptions nodisplay;
proc gplot;
   axis1 minor=none label=(angle=90 rotate=0)
         order=(-1 to 1);
   axis2 minor=none order=(-1 to 1);
   plot TX1*TX2 / &opts name=’prqin7’;
   plot TX3*TX2 / &opts name=’prqin8’ vreverse;
   plot TX1*TX3 / &opts name=’prqin9’;
   symbol1 color=blue;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:prqin7 2:prqin8 3:prqin9;
run; quit;
```

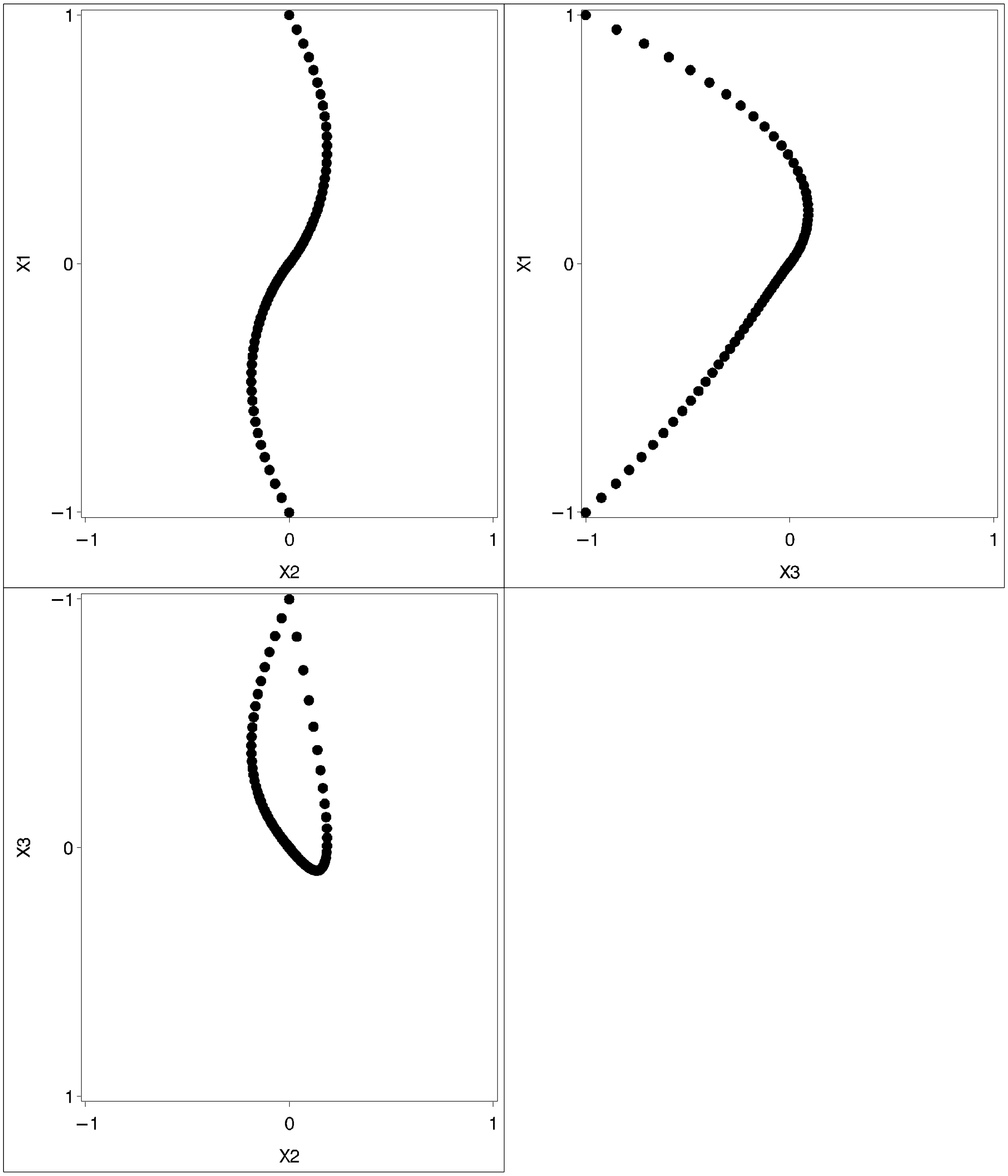


Figure 59.1. Three-Dimensional Curve Example Output

```
                  Iteratively Derive Variable Transformations

                           The PRINQUAL Procedure

                   PRINQUAL MTV Algorithm Iteration History

Iteration    Average    Maximum    Proportion    Criterion
   Number     Change     Change   of Variance       Change    Note
----------------------------------------------------------------------------
        1    0.16253    1.33045       0.71369
        2    0.07871    0.94549       0.79035      0.07667
        3    0.06518    0.80219       0.86334      0.07299
        4    0.05322    0.57928       0.91379      0.05045
        5    0.04154    0.38404       0.94204      0.02825
        6    0.03181    0.24391       0.95640      0.01436
        7    0.02461    0.15397       0.96349      0.00709
        8    0.01982    0.10205       0.96704      0.00355
        9    0.01662    0.07393       0.96894      0.00189
       10    0.01439    0.06232       0.97005      0.00112
       11    0.01288    0.05436       0.97081      0.00075
       12    0.01189    0.04911       0.97139      0.00058
       13    0.01119    0.04531       0.97188      0.00049
       14    0.01068    0.04276       0.97232      0.00044
       15    0.01027    0.04115       0.97273      0.00041
       16    0.00993    0.04039       0.97313      0.00040
       17    0.00965    0.04249       0.97351      0.00038
       18    0.00940    0.04400       0.97388      0.00037
       19    0.00919    0.04509       0.97423      0.00036
       20    0.00900    0.04587       0.97458      0.00034
       21    0.00883    0.04643       0.97491      0.00033
       22    0.00867    0.04681       0.97523      0.00032
       23    0.00852    0.04705       0.97555      0.00031
       24    0.00839    0.04719       0.97585      0.00031
       25    0.00827    0.04724       0.97615      0.00030
       26    0.00816    0.04722       0.97644      0.00029
       27    0.00805    0.04713       0.97672      0.00028
       28    0.00795    0.04699       0.97700      0.00027
       29    0.00785    0.04680       0.97726      0.00027
       30    0.00776    0.04656       0.97752      0.00026
       31    0.00768    0.04629       0.97777      0.00025
       32    0.00760    0.04598       0.97802      0.00025
       33    0.00752    0.04564       0.97826      0.00024
       34    0.00745    0.04528       0.97849      0.00023
       35    0.00739    0.04489       0.97872      0.00023
       36    0.00733    0.04448       0.97894      0.00022
       37    0.00729    0.04405       0.97915      0.00022
       38    0.00724    0.04361       0.97936      0.00021
       39    0.00720    0.04315       0.97957      0.00021
       40    0.00716    0.04268       0.97977      0.00020
       41    0.00713    0.04219       0.97997      0.00020
       42    0.00709    0.04170       0.98016      0.00019
       43    0.00706    0.04120       0.98035      0.00019
       44    0.00703    0.04070       0.98054      0.00019
       45    0.00699    0.04019       0.98072      0.00018    Converged

 Algorithm converged.
```

Figure 59.2. PROC PRINQUAL MTV Iteration History

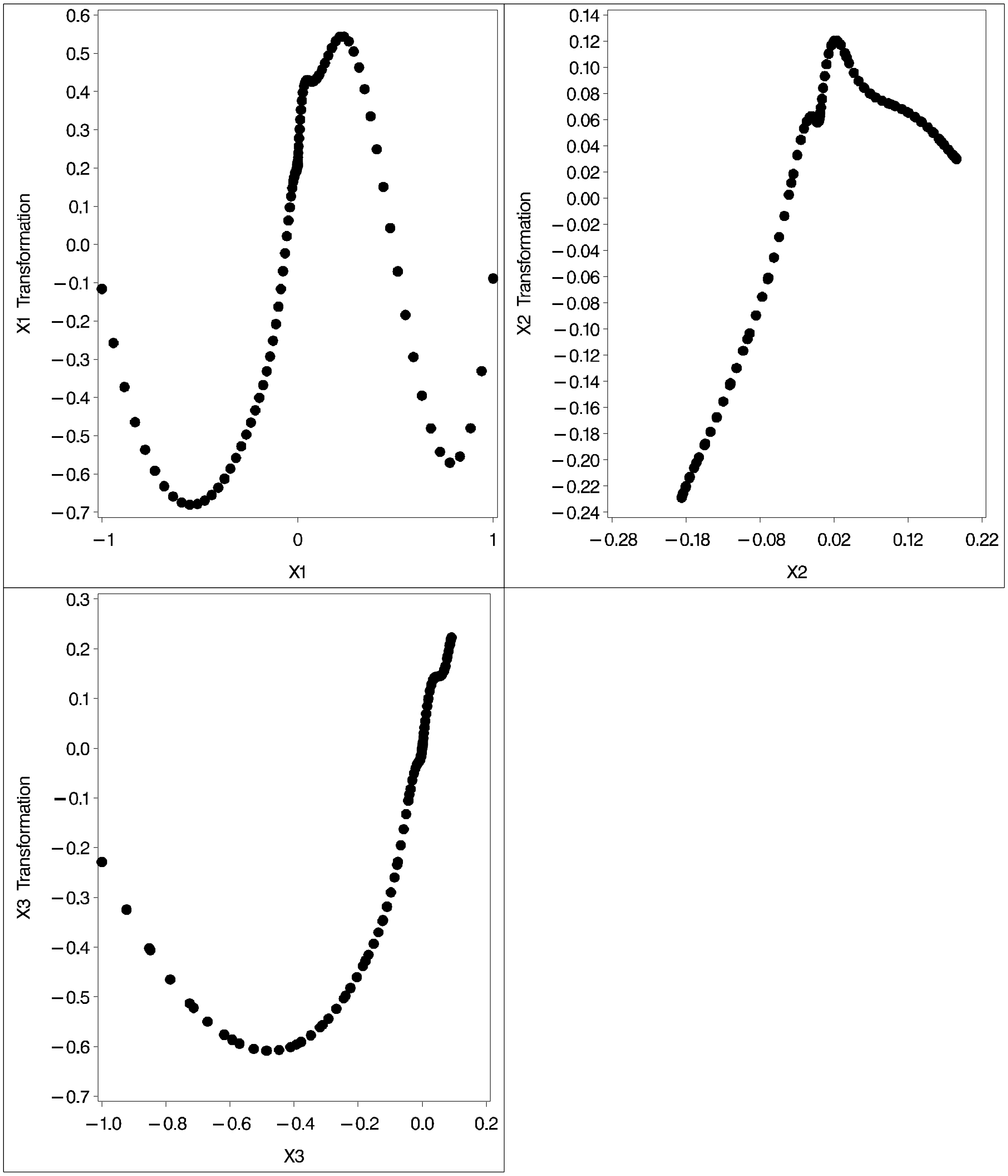


Figure 59.3. Variable Transformation Plots

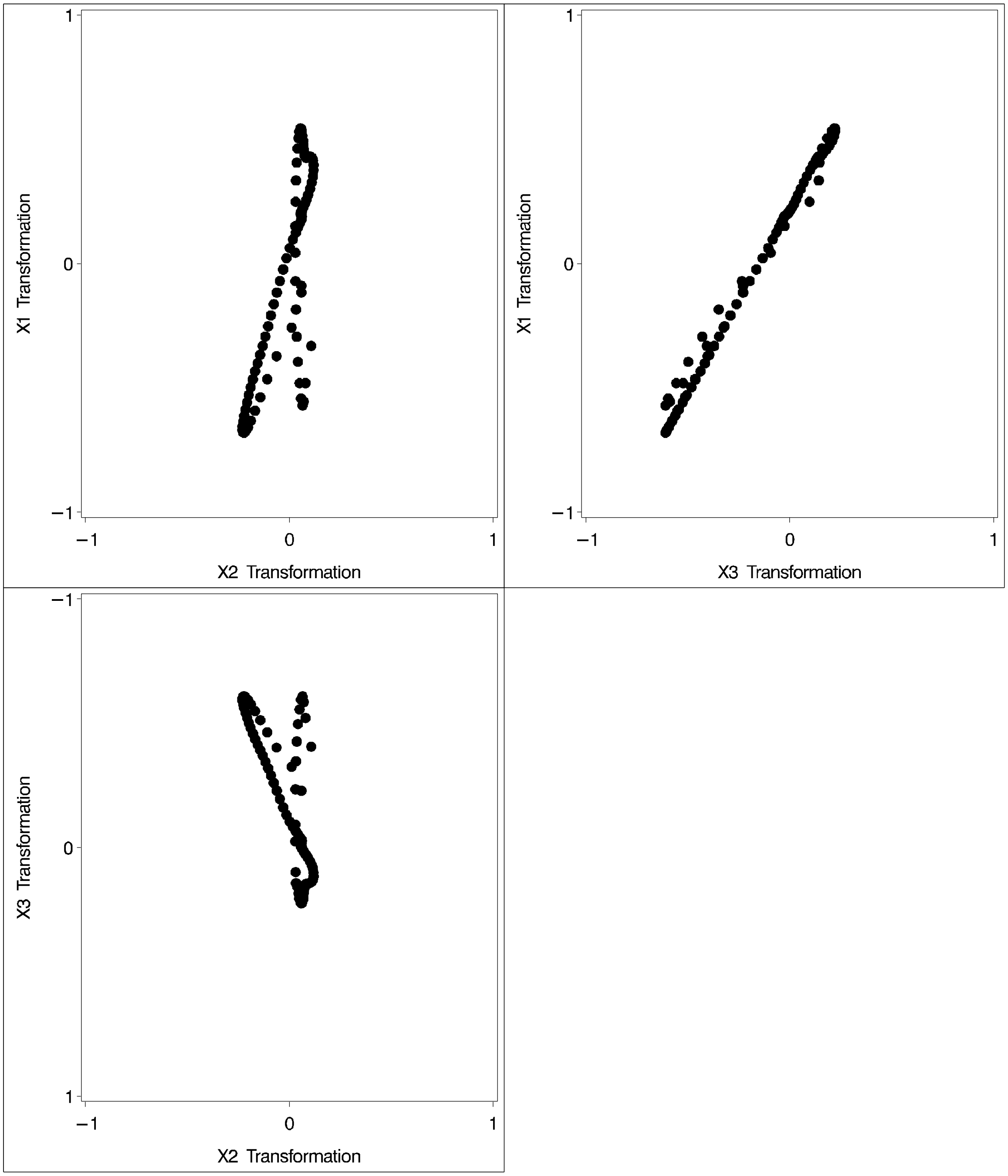


Figure 59.4. Plots of the Nearly One-Dimensional Curve

Syntax

The following statements are available in PROC PRINQUAL.

> **PROC PRINQUAL** < *options* > **;**
> **TRANSFORM** *transform(variables* < */ t-options* >*)*
> < . . . *transform(variables* < */ t-options* >*)* > **;**
> **BY** *variables* **;**
> **FREQ** *variable* **;**
> **ID** *variables* **;**
> **WEIGHT** *variable* **;**

To use PROC PRINQUAL, you need the PROC PRINQUAL and TRANSFORM statements. You can abbreviate all *options* and *t-options* to their first three letters. This is a special feature of PROC PRINQUAL and is not generally true of other SAS/STAT procedures.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC PRINQUAL statement. The remaining statements are described in alphabetical order.

PROC PRINQUAL Statement

> **PROC PRINQUAL** < *options* > **;**

The PROC PRINQUAL statement starts the PRINQUAL procedure. Optionally, this statement identifies an input data set, creates an output data set, specifies the algorithm and other computational details, and controls displayed output. The following table summarizes options available in the PROC PRINQUAL statement.

Task	Option
Identify input data set	
specifies input SAS data set	DATA=
Specify details for output data set	
outputs approximations to transformed variables	APPROXIMATIONS
specifies prefix for approximation variables	APREFIX=
outputs correlations and component structure matrix	CORRELATIONS
specifies a multidimensional preference analysis	MDPREF
specifies output data set	OUT=
specifies prefix for principal component scores variables	PREFIX=
replaces raw data with transformed data	REPLACE
outputs principal component scores	SCORES
standardizes principal component scores	STANDARD
specifies transformation standardization	TSTANDARD=
specifies prefix for transformed variables	TPREFIX=
Control iterative algorithm	
analyzes covariances	COVARIANCE
initializes using dummy variables	DUMMY
specifies iterative algorithm	METHOD=
specifies number of principal components	N=
suppresses numerical error checking	NOCHECK
specifies number of MGV models before refreshing	REFRESH=
restarts iterations	REITERATE
specifies singularity criterion	SINGULAR=
specifies input observation type	TYPE=
Control the number of iterations	
specifies minimum criterion change	CCONVERGE=
specifies number of first iteration to be displayed	CHANGE=
specifies minimum data change	CONVERGE=
specifies number of MAC initialization iterations	INITITER=
specifies maximum number of iterations	MAXITER=
Specify details for handling missing values	
includes monotone special missing values	MONOTONE=
excludes observations with missing values	NOMISS
unties special missing values	UNTIE=
Suppress displayed output	
suppresses displayed output	NOPRINT

The following list describes these options in alphabetical order.

APREFIX=*name*
APR=*name*

specifies a prefix for naming the approximation variables. By default, APREFIX=A. Specifying the APREFIX= option also implies the APPROXIMATIONS option.

APPROXIMATIONS
APPROX
APP

includes principal component approximations to the transformed variables (Eckart and Young 1936) in the output data set. Variable names are constructed from the value of the APREFIX= option and the input variable names. If you specify the APREFIX= option, then approximations are automatically included. If you specify the APPROXIMATIONS option and not the APREFIX= option, then the APPROXIMATIONS option uses the default, APREFIX=A, to construct the variable names.

CCONVERGE=*n*
CCO=*n*

specifies the minimum change in the criterion being optimized that is required to continue iterating. By default, CCONVERGE=0.0. The CCONVERGE= option is ignored for METHOD=MAC. For the MGV method, specify CCONVERGE=-2 to ensure data convergence.

CHANGE=*n*
CHA=*n*

specifies the number of the first iteration to be displayed in the iteration history table. The default is CHANGE=1. When you specify a larger value for n, the first $n - 1$ iterations are not displayed, thus speeding up the analysis. The CHANGE= option is most useful with the MGV method, which is much slower than the other methods.

CONVERGE=*n*
CON=*n*

specifies the minimum average absolute change in standardized variable scores that is required to continue iterating. By default, CONVERGE=0.00001. Average change is computed over only those variables that can be transformed by the iterations, that is, all LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE variables and nonoptimal transformation variables with missing values. For more information, see the section "Optimal Transformations" on page 3662.

COVARIANCE
COV

computes the principal components from the covariance matrix. The variables are always centered to mean zero. If you do not specify the COVARIANCE option, the variables are also standardized to variance one, which means the analysis is based on the correlation matrix.

CORRELATIONS
COR

includes correlations and the component structure matrix in the output data set. By default, this information is not included.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set must be an ordinary SAS data set; it cannot be a TYPE=CORR or TYPE=COV data set. If you omit the DATA= option, the PRINQUAL procedure uses the most recently created SAS data set.

DUMMY
DUM

expands variables specified for OPSCORE optimal transformations to dummy variables for the initialization (Tenenhaus and Vachette 1977). By default, the initial values of OPSCORE variables are the actual data values. The dummy variable nominal initialization requires considerable time and memory, so it might not be possible to use the DUMMY option with large data sets. No separate report of the initialization is produced. Initialization results are incorporated into the first iteration displayed in the iteration history table. For details, see the section "Optimal Transformations" on page 3662.

INITITER=*n*
INI=*n*

specifies the number of MAC iterations required to initialize the data before starting MTV or MGV iterations. By default, INITITER=0. The INITITER= option is ignored if METHOD=MAC.

MAXITER=*n*
MAX=*n*

specifies the maximum number of iterations. By default, MAXITER=30.

MDPREF
MDP

specifies a multidimensional preference analysis by implying the STANDARD, SCORES, and CORRELATIONS options. This option also suppresses warnings when there are more variables than observations.

METHOD=MAC | MGV | MTV
MET=MAC | MGV | MTV

specifies the optimization method. By default, METHOD=MTV. Values of the METHOD= option are MTV for maximum total variance, MGV for minimum generalized variance, or MAC for maximum average correlation. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. See the section "The Three Methods of Variable Transformation" on page 3643.

MONOTONE=*two-letters*
MON=*two-letters*

specifies the first and last special missing value in the list of those special missing values to be estimated using within-variable order and category constraints. By default, there are no order constraints on missing value estimates. The *two-letters* value must consist of two letters in alphabetical order. For example, MONOTONE=DF means that the estimate of .D must be less than or equal to the estimate of .E, which must be less than or equal to the estimate of .F; no order constraints are placed on estimates of ._, .A through .C, and .G through .Z. For details, see the "Missing Values" section on page 3667, and "Optimal Scaling" in Chapter 75, "The TRANSREG Procedure."

N=*n*

specifies the number of principal components to be computed. By default, N=2.

NOCHECK
NOC

turns off computationally intensive numerical error checking for the MGV method. If you do not specify the NOCHECK option, the procedure computes R^2 from the squared length of the predicted values vector and compares this value to the R^2 computed from the error sum of squares that is a by-product of the sweep algorithm (Goodnight 1978). If the two values of R^2 differ by more than the square root of the value of the SINGULAR= option, a warning is displayed, the value of the REFRESH= option is halved, and the model is refit after refreshing. Specifying the NOCHECK option slightly speeds up the algorithm. Note that other less computationally intensive error checking is always performed.

NOMISS
NOM

excludes all observations with missing values from the analysis, but does not exclude them from the OUT= data set. If you omit the NOMISS option, PROC PRINQUAL simultaneously computes the optimal transformations of the nonmissing values and estimates the missing values that minimize squared error.

Casewise deletion of observations with missing values occurs when you specify the NOMISS option, when there are missing values in IDENTITY variables, when there are weights less than or equal to 0, or when there are frequencies less than 1. Excluded observations are output with a blank value for the _TYPE_ variable, and they have a weight of 0. They do not contribute to the analysis but are scored and transformed as *supplementary* or passive observations. See the "Passive Observations" section on page 3674 and the "Missing Values" section on page 3667 for more information on excluded observations and missing data.

NOPRINT
NOP

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUT=*SAS-data-set*

specifies an output SAS data set that contains results of the analysis. If you omit the OUT= option, PROC PRINQUAL still creates an output data set and names it using the DATAn convention. If you want to create a permanent SAS data set, you must specify a two-level name. (Refer to the discussion in *SAS Language Reference: Concepts*.) You can use the REPLACE, APPROXIMATIONS, SCORES, and CORRELATIONS options to control what information is included in the output data set. For details, see the "Output Data Set" section on page 3669.

PREFIX=*name*
PRE=*name*

specifies a prefix for naming the principal components. By default, PREFIX=Prin. As a result, the principal component default names are Prin1, Prin2,. . ., Prin$n$.

REFRESH=*n*
REF=*n*

specifies the number of variables to scale in the MGV method before computing a new inverse. By default, REFRESH=5. PROC PRINQUAL uses the REFRESH= option in the sweep algorithm of the MGV method. Large values for the REFRESH= option make the method run faster but with increased error. Small values make the method run more slowly and with more numerical accuracy.

REITERATE
REI

enables the PRINQUAL procedure to use previous transformations as starting points. The REITERATE option affects only variables that are iteratively transformed (specified as LINEAR, SPLINE, MSPLINE, SSPLINE, UNTIE, OPSCORE, and MONOTONE). For iterative transformations, the REITERATE option requests a search in the input data set for a variable that consists of the value of the TPREFIX= option followed by the original variable name. If such a variable is found, it is used to provide the initial values for the first iteration. The final transformation is a member of the transformation family defined by the original variable, not the transformation family defined by the initialization variable. See the "REITERATE Option Usage" section on page 3673.

REPLACE
REP

replaces the original data with the transformed data in the output data set. The names of the transformed variables in the output data set correspond to the names of the original variables in the input data set. If you do not specify the REPLACE option, both original variables and transformed variables (with names constructed from the TPREFIX= option and the original variable names) are included in the output data set.

SCORES
SCO

includes principal component scores in the output data set. By default, scores are not included.

SINGULAR=*n*
SIN=*n*

specifies the largest value within rounding error of zero. By default, SINGULAR=1E$-$8. The PRINQUAL procedure uses the value of the SINGULAR= option for checking $(1 - \mathrm{R}^2)$ when constructing full-rank matrices of predictor variables, checking denominators before dividing, and so on.

STANDARD
STD

standardizes the principal component scores in the output data set to mean zero and variance one instead of the default mean zero and variance equal to the corresponding eigenvalue. See the SCORES option.

TPREFIX=*name*
TPR=*name*

specifies a prefix for naming the transformed variables. By default, TPREFIX=T. The TPREFIX= option is ignored if you specify the REPLACE option.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When the TSTANDARD= option is specified in the PROC statement, it specifies the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization just for selected variables.

CENTER centers the output variables to mean zero, but the variances are the same as the variances of the input variables.

NOMISS sets the means and variances of the transformed variables in the OUT= data set, computed over all output values that correspond to nonmissing values in the input data set, to the means and variances computed from the nonmissing observations of the original variables. The TSTANDARD=NOMISS specification is useful with missing data. When a variable is linearly transformed, the final variable contains the original nonmissing values and the missing value estimates. In other words, the nonmissing values are unchanged. If your data have no missing values, TSTANDARD=NOMISS and TSTANDARD=ORIGINAL produce the same results.

ORIGINAL sets the means and variances of the transformed variables to the means and variances of the original variables. This is the default.

Z standardizes the variables to mean zero, variance one.

For nonoptimal variable transformations, the means and variances of the original variables are actually the means and variances of the nonlinearly transformed variables, unless you specify the ORIGINAL nonoptimal *t-option* in the TRANSFORM statement. For example, if a variable X with no missing values is specified as LOG, then, by default, the final transformation of X is simply LOG(X), not LOG(X) standardized to the mean of X and variance of X.

TYPE=’*text* **’|***name*
TYP=’*text* **’|***name*

specifies the valid value for the _TYPE_ variable in the input data set. If PROC PRINQUAL finds an input _TYPE_ variable, it uses only observations with a _TYPE_ value that matches the TYPE= value. This enables a PROC PRINQUAL OUT= data set containing correlations to be used as input to PROC PRINQUAL without requiring a WHERE statement to exclude the correlations. If a _TYPE_ variable is not in the data set, all observations are used. The default is TYPE=’SCORE’, so if you do not specify the TYPE= option, only observations with _TYPE_ = ’SCORE’ are used.

PROC PRINQUAL displays a note when it reads observations with blank values of _TYPE_, but it does not automatically exclude those observations. Data sets created by the TRANSREG and PRINQUAL procedures have blank _TYPE_ values for those observations that were excluded from the analysis due to nonpositive weights, nonpositive frequencies, or missing data. When these observations are read again, they are excluded for the same reason that they were excluded from their original analysis, not because their _TYPE_ value is blank.

UNTIE=*two-letters*
UNT=*two-letters*

specifies the first and last special missing value in the list of those special missing values that are to be estimated with within-variable order constraints but no category constraints. The *two-letters* value must consist of two letters in alphabetical order. By default, there are category constraints but no order constraints on special missing value estimates. For details, see the "Missing Values" section on page 3667. Also, see "Optimal Scaling" in Chapter 75, "The TRANSREG Procedure."

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PRINQUAL to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the PRINQUAL procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, list the variable's name in a FREQ statement. PROC PRINQUAL then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Noninteger values of the

FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ statement variable is greater than or equal to 1.

ID Statement

ID *variables* **;**

The ID statement includes additional character or numeric variables in the output data set. The variables must be contained in the input data set.

TRANSFORM Statement

TRANSFORM *transform(variables < / t-options >)*

< . . . transform(variables < / t-options >) > **;**

The TRANSFORM statement lists the variables to be analyzed (*variables*) and specifies the transformation (*transform*) to apply to each variable listed. You must specify a transformation for each variable list in the TRANSFORM statement. The variables are variables in the data set. The *t-options* are transformation options that provide details for the transformation; these depend on the *transform* chosen. The *t-options* are listed after a slash in the parentheses that enclose the variables.

For example, the following statements find a quadratic polynomial transformation of all variables in the data set:

```
proc prinqual;
   transform spline(_all_ / degree=2);
run;
```

Or, if N1 through N10 are nominal variables and M1 through M10 are ordinal variables, you can use the following statements.

```
proc prinqual;
   transform opscore(N1-N10) monotone(M1-M10);
run;
```

The following sections describe the transformations available (specified with *transform*) and the options available for some of the transformations (specified with *t-options*).

Families of Transformations

There are three types of transformation families: nonoptimal, optimal, and other. Each family is summarized as follows.

Nonoptimal transformations preprocess the specified variables, replacing each one with a single new nonoptimal, nonlinear transformation.

Optimal transformations	replace the specified variables with new, iteratively derived optimal transformation variables that fit the specified model better than the original variable (except for contrived cases where the transformation fits the model exactly as well as the original variable).
Other transformations	are the IDENTITY and SSPLINE transformations. These do not fit into either of the preceding categories.

The following table summarizes the transformations in each family.

Family	**Members of Family**
Nonoptimal transformations	
inverse trigonometric sine	ARSIN
exponential	EXP
logarithm	LOG
logit	LOGIT
raises variables to specified power	POWER
transforms to ranks	RANK
Optimal transformations	
linear	LINEAR
monotonic, ties preserved	MONOTONE
monotonic B-spline	MSPLINE
optimal scoring	OPSCORE
B-spline	SPLINE
monotonic, ties not preserved	UNTIE
Other transformations	
identity, no transformation	IDENTITY
iterative smoothing spline	SSPLINE

The *transform* is followed by a variable (or list of variables) enclosed in parentheses. Optionally, depending on the *transform*, the parentheses can also contain *t-options*, which follow the variables and a slash. For example,

```
transform log(X Y);
```

computes the LOG transformation of X and Y. A more complex example is

```
transform spline(Y / nknots=2) log(X1 X2 X3);
```

The preceding statement uses the SPLINE transformation of the variable Y and the LOG transformation of the variables X1, X2, and X3. In addition, it uses the NKNOTS= option with the SPLINE transformation and specifies two knots.

The rest of this section provides syntax details for members of the three families of transformations. The *t-options* are discussed in the section "Transformation Options (t-options)" on page 3663.

Nonoptimal Transformations

Nonoptimal transformations are computed before the iterative algorithm begins. Nonoptimal transformations create a single new transformed variable that replaces the original variable. The new variable is not transformed by the subsequent iterative algorithms (except for a possible linear transformation and missing value estimation).

The following list provides syntax and details for nonoptimal variable transformations.

ARSIN
ARS

finds an inverse trigonometric sine transformation. Variables following ARSIN must be numeric, in the interval $(-1.0 \leq \text{X} \leq 1.0)$, and they are typically continuous.

EXP

exponentiates variables (the variable X is transformed to a^X). To specify the value of a, use the PARAMETER= *t-option*. By default, a is the mathematical constant $e = 2.718\ldots$. Variables following EXP must be numeric, and they are typically continuous.

LOG

transforms variables to logarithms (the variable X is transformed to $\log_a(X)$). To specify the base of the logarithm, use the PARAMETER= *t-option*. The default is a natural logarithm with base $e = 2.718\ldots$. Variables following LOG must be numeric and positive, and they are typically continuous.

LOGIT

finds a logit transformation on the variables. The logit of X is $\log(X/(1-X))$. Unlike other transformations, LOGIT does not have a three-letter abbreviation. Variables following LOGIT must be numeric, in the interval $(0.0 < \text{X} < 1.0)$, and they are typically continuous.

POWER
POW

raises variables to a specified power (the variable X is transformed to X^a). You must specify the power parameter a by specifying the PARAMETER= *t-option* following the variables:

```
power(variable / parameter=number)
```

You can use POWER for squaring variables (PARAMETER=2), reciprocal transformations (PARAMETER=-1), square roots (PARAMETER=0.5), and so on. Variables following POWER must be numeric, and they are typically continuous.

RANK
RAN

transforms variables to ranks. Ranks are averaged within ties. The smallest input value is assigned the smallest rank. Variables following RANK must be numeric.

Optimal Transformations

Optimal transformations are iteratively derived. Missing values for these types of variables can be optimally estimated (see the "Missing Values" section on page 3667).

The following list provides syntax and details for optimal transformations.

LINEAR
LIN

finds an optimal linear transformation of each variable. For variables with no missing values, the transformed variable is the same as the original variable. For variables with missing values, the transformed nonmissing values have a different scale and origin than the original values. Variables following LINEAR must be numeric.

MONOTONE
MON

finds a monotonic transformation of each variable, with the restriction that ties are preserved. The Kruskal (1964) secondary least-squares monotonic transformation is used. This transformation weakly preserves order and category membership (ties). Variables following MONOTONE must be numeric, and they are typically discrete.

MSPLINE
MSP

finds a monotonically increasing B-spline transformation with monotonic coefficients (de Boor 1978; de Leeuw 1986) of each variable. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with MSPLINE. By default, PROC PRINQUAL uses a quadratic spline. Variables following MSPLINE must be numeric, and they are typically continuous.

OPSCORE
OPS

finds an optimal scoring of each variable. The OPSCORE transformation assigns scores to each class (level) of the variable. Fisher's (1938) optimal scoring method is used. Variables following OPSCORE can be either character or numeric; numeric variables should be discrete.

SPLINE
SPL

finds a B-spline transformation (de Boor 1978) of each variable. By default, PROC PRINQUAL uses a cubic polynomial transformation. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with SPLINE. Variables following SPLINE must be numeric, and they are typically continuous.

UNTIE
UNT

finds a monotonic transformation of each variable without the restriction that ties are preserved. The PRINQUAL procedure uses the Kruskal (1964) primary least-squares monotonic transformation method. This transformation weakly preserves order but not category membership (it may untie some previously tied values). Variables following UNTIE must be numeric, and they are typically discrete.

Other Transformations

IDENTITY
IDE

specifies variables that are not changed by the iterations. The IDENTITY transformation is used for variables when no transformation and no missing data estimation are desired. However, the REFLECT, ADDITIVE, TSTANDARD=Z, and TSTANDARD=CENTER options can linearly transform all variables, including IDENTITY variables, after the iterations. Observations with missing values in IDENTITY variables are excluded from the analysis, and no optimal scores are computed for missing values in IDENTITY variables. Variables following IDENTITY must be numeric.

SSPLINE
SSP

finds an iterative smoothing spline transformation of each variable. The SSPLINE transformation does not generally minimize squared error. You can specify the smoothing parameter with either the SM= *t-option* or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables following SSPLINE must be numeric, and they are typically continuous.

Transformation Options (t-options)

If you use a nonoptimal, optimal or other transformation, you can use *t-options*, which specify additional details of the transformation. The *t-options* are specified within the parentheses that enclose variables and are listed after a slash. For example,

```
proc prinqual;
   transform spline(X Y / nknots=3);
run;
```

The preceding statements find an optimal variable transformation (SPLINE) of the variables X and Y and use a *t-option* to specify the number of knots (NKNOTS=). The following is a more complex example.

```
proc prinqual;
   transform spline(Y / nknots=3) spline(X1 X2 / nknots=6);
run;
```

These statements use the SPLINE transformation for all three variables and use *t-options* as well; the NKNOTS= option specifies the number of knots for the spline.

The following sections discuss the *t-options* available for nonoptimal, optimal, and other transformations.

The following table summarizes the *t-options*.

Table 59.1. t-options Available in the TRANSFORM Statement

Task	Option
Nonoptimal transformation t-options	
uses original mean and variance	ORIGINAL
Parameter t-options	
specifies miscellaneous parameters	PARAMETER=
specifies smoothing parameter	SM=
Spline t-options	
specifies the degree of the spline	DEGREE=
spaces the knots evenly	EVENLY
specifies the interior knots or break points	KNOTS=
creates n knots	NKNOTS=
Other t-options	
renames variables	NAME=
reflects the variable around the mean	REFLECT
specifies transformation standardization	TSTANDARD=

Nonoptimal Transformation t-options

ORIGINAL
ORI

matches the variable's final mean and variance to the mean and variance of the original variable. By default, the mean and variance are based on the transformed values. The ORIGINAL *t-option* is available for all of the nonoptimal transformations.

Parameter t-options

PARAMETER=*number*
PAR=*number*

specifies the transformation parameter. The PARAMETER= *t-option* is available for the EXP, LOG, POWER, SMOOTH, and SSPLINE transformations. For EXP, the parameter is the value to be exponentiated; for LOG, the parameter is the base value; and for POWER, the parameter is the power. For SMOOTH and SSPLINE, the parameter is the raw smoothing parameter. (You can specify a SAS/GRAPH-style smoothing parameter with the SM= *t-option*.) The default for the PARAMETER= *t-option* for the LOG and EXP transformations is $e = 2.718\ldots$. The default parameter for SSPLINE is computed from SM=0. For the POWER transformation, you must specify the PARAMETER= *t-option*; there is no default.

SM=*n*

specifies a SAS/GRAPH-style smoothing parameter in the range 0 to 100. You can specify the SM= *t-option* only with the SSPLINE transformation. The smoothness of the function increases as the value of the smoothing parameter increases. By default, SM=0.

Spline t-options

The following *t-options* are available with the SPLINE and MSPLINE optimal transformations.

DEGREE=*n*
DEG=*n*

specifies the degree of the B-spline transformation. The degree must be a nonnegative integer. The defaults are DEGREE=3 for SPLINE variables and DEGREE=2 for MSPLINE variables.

The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

EVENLY
EVE

is used with the NKNOTS= *t-option* to space the knots evenly. The differences between adjacent knots are constant. If you specify NKNOTS=k, k knots are created at

$$\text{minimum} + i((\text{maximum} - \text{minimum})/(k+1))$$

for $i = 1, \ldots, k$. For example, if you specify

```
spline(X / knots=2 evenly)
```

and the variable X has a minimum of 4 and a maximum of 10, then the two interior knots are 6 and 8. Without the EVENLY *t-option*, the NKNOTS= *t-option* places knots at percentiles, so the knots are not evenly spaced.

KNOTS=*number-list* **|** *n* **TO** *m* **BY** *p*
KNO=*number-list* **|** *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. By default, there are no knots. The first time you specify a value in the knot list, it indicates a discontinuity in the nth (from DEGREE=n) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$th derivative of the transformation function at the value of the knot. Knots can be repeated any number of times for decreasing smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= *t-option* with the NKNOTS= *t-option*. You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 4613 in Chapter 75, "The TRANSREG Procedure.").

NKNOTS=*n*
NKN=*n*

creates n knots, the first at the $100/(n+1)$ percentile, the second at the $200/(n+1)$ percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the twenty-fifth percentile, the median, and the seventy-fifth percentile. By default, NKNOTS=0. The NKNOTS= *t-option* must be ≥ 0.

You cannot use the NKNOTS= *t-option* with the KNOTS= *t-option*. You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 4613 in Chapter 75, "The TRANSREG Procedure,").

Other t-options

The following *t-options* are available for all transformations.

NAME=*(variable-list)*
NAM=*(variable-list)*

renames variables as they are used in the TRANSFORM statement. This option allows a variable to be used more than once. For example, if the variable X is a character variable, then the following step stores both the original character variable X and a numeric variable XC that contains category numbers in the output data set.

```
proc prinqual data=A n=1 out=B;
   transform linear(Y) opscore(X / name=(XC));
   id X;
run;
```

REFLECT
REF

reflects the transformation

$$y = -(y - \bar{y}) + \bar{y}$$

after the iterations are completed and before the final standardization and results calculations.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When the TSTANDARD= option is specified in the PROC PRINQUAL statement, it specifies the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization just for selected variables.

WEIGHT Statement

WEIGHT *variable* **;**

When you use a WEIGHT statement, a weighted residual sum of squares is minimized. The WEIGHT statement has no effect on degrees of freedom or number of observations, but the weights affect most other calculations. The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than 0.

Details

Missing Values

PROC PRINQUAL can estimate missing values, subject to optional constraints, so that the covariance matrix is optimized. The procedure provides several approaches for handling missing data. When you specify the NOMISS option in the PROC PRINQUAL statement, observations with missing values are excluded from the analysis. Otherwise, missing data are estimated, using variable means as initial estimates. Missing values for OPSCORE character variables are treated the same as any other category during the initialization. See the section "Missing Values" on page 4599 in Chapter 75, "The TRANSREG Procedure," for more information on missing data estimation.

Controlling the Number of Iterations

Several options in the PROC PRINQUAL statement control the number of iterations performed. Iteration terminates when any one of the following conditions is satisfied:

- The number of iterations equals the value of the MAXITER= option.
- The average absolute change in variable scores from one iteration to the next is less than the value of the CONVERGE= option.
- The criterion change is less than the value of the CCONVERGE= option.

With the MTV method, the change in the proportion of variance criterion can become negative when the data have converged so that it is numerically impossible, within machine precision, to increase the criterion. Because the MTV algorithm is convergent, a negative criterion change is the result of very small amounts of rounding error. The MGV method displays the average squared multiple correlation (which is not the criterion being optimized), so the criterion change can become negative well before convergence. The MAC method criterion (average correlation) is never computed, so the CCONVERGE= option is ignored for METHOD=MAC. You can specify a negative value for either convergence option if you want to define convergence only in terms of the other convergence option.

With the MGV method, iterations minimize the generalized variance (determinant), but the generalized variance is not reported for two reasons. First, in most data sets, the generalized variance is almost always near zero (or will be after one or two iterations), which is its minimum. This does not mean that iteration is complete; it simply means that at least one multiple correlation is at or near one. The algorithm continues minimizing the determinant in $(m-1), (m-2)$ dimensions, and so on. Because the generalized variance is almost always near zero, it does not provide a good indication of how the iterations are progressing. The mean R^2 provides a better indication of convergence. The second reason for not reporting the generalized variance is that almost no additional time is required to compute R^2 values for each step. This is because the error sum of squares is a by-product of the algorithm at each step.

Computing the determinant at the end of each iteration adds more computations to an already computationally intensive algorithm.

You can increase the number of iterations to ensure convergence by increasing the value of the MAXITER= option and decreasing the value of the CONVERGE= option. Because the average absolute change in standardized variable scores seldom decreases below 1E−11, you typically do not specify a value for the CONVERGE= option less than 1E−8 or 1E−10. Most of the data changes occur during the first few iterations, but the data can still change after 50 or even 100 iterations. You can try different combinations of values for the CONVERGE= and MAXITER= options to ensure convergence without extreme overiteration. If the data do not converge with the default specifications, specify the REITERATE option, or try CONVERGE=1E−8 and MAXITER=50, or CONVERGE=1E−10 and MAXITER=200.

Performing a Principal Component Analysis of Transformed Data

PROC PRINQUAL produces an iteration history table that displays (for each iteration) the iteration number, the maximum and average absolute change in standardized variable scores computed over the iteratively transformed variables, the criterion being optimized, and the criterion change. In order to examine the results of the analysis in more detail, you can analyze the information in the output data set using other SAS procedures.

Specifically, use the PRINCOMP procedure to perform a components analysis on the transformed data. PROC PRINCOMP accepts the raw data from PROC PRINQUAL but issues a warning because the PROC PRINQUAL output data set has _NAME_ and _TYPE_ variables, but it is not a TYPE=CORR data set. You can ignore this warning.

If the output data set contains both scores and correlations, you must subset it for analysis with PROC PRINCOMP. Otherwise, the correlation observations are treated as ordinary observations and the PROC PRINCOMP results are incorrect. For example, consider the following statements:

```
proc prinqual data=a out=b correlations replace;
   transform spline(var1-var50 / nknots=3);
run;

proc princomp data=b;
   where _TYPE_='SCORE';
run;
```

Also note that the proportion of variance accounted for, as reported by PROC PRINCOMP, can exceed the proportion of variance accounted for in the last PROC PRINQUAL iteration. This is because PROC PRINQUAL reports the variance accounted for by the components analysis that generated the current scaling of the data, not a components analysis of the current scaling of the data.

Using the MAC Method

You can use the MAC algorithm alone by specifying METHOD=MAC, or you can use it as an initialization algorithm for METHOD=MTV and METHOD=MGV analyses by specifying the iteration option INITITER=. If any variables are negatively correlated, do not use the MAC algorithm with monotonic transformations (MONOTONE, UNTIE, and MSPLINE) because the signs of the correlations among the variables are not used when computing variable approximations. If an approximation is negatively correlated with the original variable, monotone constraints would make the optimally scaled variable a constant, which is not allowed (see the section "Avoiding Constant Transformations" on page 3672). When used with other transformations, the MAC algorithm can reverse the scoring of the variables. So, for example, if variable X is designated LOG(X) with METHOD=MAC and TSTANDARD=ORIGINAL, the final transformation (for example, TX) may not be LOG(X). If TX is not LOG(X), it has the same mean as LOG(X) and the same variance as LOG(X), and it is perfectly negatively correlated with LOG(X). PROC PRINQUAL displays a note for every variable that is reversed in this manner.

You can use the METHOD=MAC algorithm to reverse the scorings of some rating variables before a factor analysis. The correlations among bipolar ratings such as 'like - dislike', 'hot - cold', and 'fragile - monumental' are typically both positive and negative. If some items are reversed to say 'dislike - like', 'cold - hot', and 'monumental - fragile', some of the negative signs can be eliminated, and the factor pattern matrix would be cleaner. You can use PROC PRINQUAL with METHOD=MAC and LINEAR transformations to reverse some items, maximizing the average of the intercorrelations.

Output Data Set

The PRINQUAL procedure produces an output data set by default. By specifying the OUT=, APPROXIMATIONS, SCORES, REPLACE, and CORRELATIONS options in the PROC PRINQUAL statement, you can name this data set and control, to some extent, the contents of it.

Structure and Content

The output data set can have 16 different forms, depending on the specified combinations of the REPLACE, SCORES, APPROXIMATIONS, and CORRELATIONS options. You can specify any combination of these options. To illustrate, assume that the data matrix consists of N observations and m variables, and n components are computed. Then, define the following:

D the $N \times m$ matrix of original data with variable names that correspond to the names of the variables in the input data set. However, when you use the OPSCORE transformation on character variables, those variables are replaced by numeric variables that contain category numbers

T the $N \times m$ matrix of transformed data with variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

$\mathbf{S}$ the $N \times n$ matrix of component scores with variable names constructed from the value of the PREFIX= option and integers

$\mathbf{A}$ the $N \times m$ matrix of data approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

$\mathbf{R_{TD}}$ the $m \times m$ matrix of correlations between the transformed variables and the original variables with variable names that correspond to the names of the variables in the input data set. When missing values exist, casewise deletion is used to compute the correlations.

$\mathbf{R_{TT}}$ the $m \times m$ matrix of correlations among the transformed variables with the variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

$\mathbf{R_{TS}}$ the $m \times n$ matrix of correlations between the transformed variables and the principal component scores (component structure matrix) with variable names constructed from the value of the PREFIX= option and integers

$\mathbf{R_{TA}}$ the $m \times m$ matrix of correlations between the transformed variables and the variable approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

To create a data set WORK.A that contains all information, specify the following options in the PROC PRINQUAL statement

```
proc prinqual scores approximations correlations out=a;
```

and also use a TRANSFORM statement appropriate for your data. Then the WORK.A data set contains

D	T	S	A
R_{TD}	R_{TT}	R_{TS}	R_{TA}

To eliminate the bottom partitions that contain the correlations and component structure, do not specify the CORRELATIONS option. For example, use the following PROC PRINQUAL statement with an appropriate TRANSFORM statement.

```
proc prinqual scores approximations out=a;
```

Then the WORK.A data set contains

```
D T S A
```

If you use the following PROC PRINQUAL statement (with an appropriate TRANSFORM statement)

```
proc prinqual out=a;
```

this creates a data set WORK.A of the form

```
D T
```

To output transformed data and component scores only, specify the following options in the PROC PRINQUAL statement:

```
proc prinqual replace scores out=a;
```

Then the WORK.A data set contains

```
T S
```

TYPE *and* _NAME_ *Variables*

In addition to the preceding information, the output data set contains two character variables, the variable _TYPE_ (length 8) and the variable _NAME_ (length 32).

The _TYPE_ variable has the value 'SCORE' if the observation contains variables, transformed variables, components, or data approximations; the _TYPE_ variable has the value 'CORR' if the observation contains correlations or component structure.

By default, the _NAME_ variable has values 'ROW1', 'ROW2', and so on, for the observations with _TYPE_='SCORE'. If you use an ID statement, the variable _NAME_ contains the formatted ID variable for SCORES observations. The values of the variable _NAME_ for observations with _TYPE_='CORR' are the names of the transformed variables.

Certain procedures, such as PROC PRINCOMP, which can use the PROC PRINQUAL output data set, issue a warning that the PROC PRINQUAL data set contains _NAME_ and _TYPE_ variables but is not a TYPE=CORR data set. You can ignore this warning.

Variable Names

The TPREFIX=, APREFIX=, and PREFIX= options specify prefixes for the transformed and approximation variable names and for principal component score variables, respectively. PROC PRINQUAL constructs transformed and approximation variable names from a prefix and the first characters of the original variable name. The number of characters in the prefix plus the number of characters in the original variable name (including the final digits, if any) required to uniquely designate the new variables should not exceed 32. For example, if the APREFIX= parameter that you specify is one character, PROC PRINQUAL adds the first 31 characters of the original variable name; if your prefix is four characters, only the first 28 characters of the original variable name are added.

Effect of the TSTANDARD= and COVARIANCE Options

The values in the output data set are affected by the TSTANDARD= and COVARIANCE options. If you specify TSTANDARD=NOMISS, the NOMISS standardization is performed on the transformed data after the iterations have been completed, but before the output data set is created. The new means and variances are used in creating the output data set. Then, if you do not specify the COVARIANCE option, the data are transformed to mean zero and variance one. The principal component scores and data approximations are computed from the resulting matrix. The data are then linearly transformed to have the mean and variance specified by the TSTANDARD= option. The data approximations are transformed so that the means within each pair of a transformed variable and its approximation are the same. The ratio of the variance of a variable approximation to the variance of the corresponding transformed variable equals the proportion of the variance of the variable that is accounted for by the components model.

If you specify the COVARIANCE option and do not specify TSTANDARD=Z, you can input the transformed data to PROC PRINCOMP, again specifying the COVARIANCE option, to perform a components analysis of the results of PROC PRINQUAL. Similarly, if you do not specify the COVARIANCE option with PROC PRINQUAL and you input the transformed data to PROC PRINCOMP without the COVARIANCE option, you receive the same report. However, some combinations of PROC PRINQUAL options, such as COVARIANCE and TSTANDARD=Z, while valid, produce approximations and scores that cannot be reproduced by PROC PRINCOMP.

The component scores in the output data set are computed from the correlations among the transformed variables, or from the covariances if you specified the COVARIANCE option. The component scores are computed after the TSTANDARD=NOMISS transformation, if specified. The means of the component scores in the output data set are always zero. The variances equal the corresponding eigenvalues, unless you specify the STANDARD option; then the variances are set to one.

Avoiding Constant Transformations

There are times when the optimal scaling produces a constant transformed variable. This can happen with the MONOTONE, UNTIE, and MSPLINE transformations when the target is negatively correlated with the original input variable. It can happen with all transformations when the target is uncorrelated with the original input variable. When this happens, the procedure modifies the target to avoid a constant transformation. This strategy avoids certain nonoptimal solutions.

If the transformation is monotonic and a constant transformed variable results, the procedure multiplies the target by -1 and tries the optimal scaling again. If the transformation is not monotonic or if the multiplication by -1 did not help, the procedure tries using a random target. If the transformation is still constant, the previous nonconstant transformation is retained. When a constant transformation is avoided by any strategy, this message is displayed: “A constant transformation was avoided for *name*.”

Constant Variables

Constant and almost constant variables are zeroed and ignored.

Character OPSCORE Variables

Character OPSCORE variables are replaced by a numeric variable containing category numbers before the iterations, and the character values are discarded. Only the first eight characters are considered when determining category membership. If you want the original character variable in the output data set, give it a different name in the OPSCORE specificiation (OPSCORE(x / name=(x2)) and name the original variable on the ID statement (ID x;).

REITERATE Option Usage

You can use the REITERATE option to perform additional iterations when PROC PRINQUAL stops before the data have adequately converged. For example, suppose that you execute the following code:

```
proc prinqual data=A cor out=B;
   transform mspline(X1-X5);
run;
```

If the transformations do not converge in the default 30 iterations, you can perform more iterations without repeating the first 30 iterations.

```
proc prinqual data=B reiterate cor out=B;
   transform mspline(X1-X5);
run;
```

Note that a WHERE statement is not necessary to exclude the correlation observations. They are automatically excluded because their _TYPE_ variable value is not ’SCORE’.

You can also use the REITERATE option to specify starting values other than the original values for the transformations. Providing alternate starting points may avoid local optima. Here are two examples.

```
proc prinqual data=A out=B;
   transform rank(X1-X5);
run;

proc prinqual data=B reiterate out=C;
   /* Use ranks as the starting point. */
   transform monotone(X1-X5);
run;

data B;
   set A;
   array TXS[5] TX1-TX5;
   do j = 1 to 5;
      TXS[j] = normal(0);
   end;
run;

proc prinqual data=B reiterate out=C;
   /* Use a random starting point. */
   transform monotone(X1-X5);
run;
```

Note that divergence with the REITERATE option, particularly in the second iteration, is not an error since the initial transformation is not required to be a valid member of the transformation family. When you specify the REITERATE option, the iteration does not terminate when the criterion change is negative during the first ten iterations.

Passive Observations

Observations may be excluded from the analysis for several reasons, including zero weight, zero frequency, missing values in variables designated IDENTITY, or missing values with the NOMISS option specified. These observations are passive in that they do not contribute to determining transformations, R^2, total variance, and so on. However, some information can be computed for them, such as approximations, principal component scores, and transformed values. Passive observations in the output data set have a blank value for the variable _TYPE_.

Missing value estimates for passive observations may converge slowly with METHOD=MTV. In the following example, the missing value estimates should be 2, 5, and 8. Since the nonpassive observations do not change, the procedure converges in one iteration but the missing value estimates do not converge. The extra iterations produced by specifying CONVERGE=-1 and CCONVERGE=-1, as shown in the second PROC PRINQUAL step, generate the expected results.

```
data A;
   input X Y;
   datalines;
1 1
2 .
3 3
4 4
5 .
6 6
7 7
8 .
9 9
;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv;
   transform lin(X Y);
run;
```

```
proc print;
run;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv
   converge=-1 cconverge=-1;
   transform lin(X Y);
run;

proc print;
run;
```

Computational Resources

This section provides information on the computational resources required to run PROC PRINQUAL.

Let

N	$=$	number of observations
m	$=$	number of variables
n	$=$	number of principal components
k	$=$	maximum spline degree
p	$=$	maximum number of knots

- For the MTV algorithm, more than

  $$56m + 8Nm + 8\left(6N + (p + k + 2)(p + k + 11)\right)$$

  bytes of array space are required.
- For the MGV and MAC algorithms, more than $56m$ plus the maximum of the data matrix size and the optimal scaling work space bytes of array space are required. The data matrix size is $8Nm$ bytes. The optimal scaling work space requires less than $8\left(6N + (p + k + 2)(p + k + 11)\right)$ bytes.
- For the MTV and MGV algorithms, more than $56m + 4m(m + 1)$ bytes of array space are required.
- PROC PRINQUAL tries to store the original and transformed data in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time. The amount of memory for the preceding data formulas are underestimates of the amount of memory needed to handle most problems. These formulas give an absolute minimum amount of memory required. If a utility data set is used, and if memory could be used with perfect efficiency, then roughly the amount of memory stated previously would be needed. In reality, most problems require at least two or three times the minimum.

- PROC PRINQUAL sorts the data once. The sort time is roughly proportional to $mN^{3/2}$.
- For the MTV algorithm, the time required to compute the variable approximations is roughly proportional to $2Nm^2 + 5m^3 + nm^2$.
- For the MGV algorithm, one regression analysis per iteration is required to compute model parameter estimates. The time required for accumulating the crossproduct matrix is roughly proportional to Nm^2. The time required to compute the regression coefficients is roughly proportional to m^3. For each variable for each iteration, the swept crossproduct matrix is updated with time roughly proportional to m(N+m). The swept crossproduct matrix is updated for each variable with time roughly proportional to m^2, until computations are refreshed, requiring all sweeps to be performed again.
- The only computationally intensive part of the MAC algorithm is the optimal scaling, since variable approximations are simple averages.
- Each optimal scaling is a multiple regression problem, although some transformations are handled with faster special case algorithms. The number of regressors for the optimal scaling problems depends on the original values of the variable and the type of transformation. For each monotone spline transformation, an unknown number of multiple regressions is required to find a set of coefficients that satisfies the constraints. The B-spline basis is generated twice for each SPLINE and MSPLINE transformation for each iteration. The time required to generate the B-spline basis is roughly proportional to Nk^2.

Displayed Output

The main output from the PRINQUAL procedure is the output data set. However, the procedure does produce displayed output in the form of an iteration history table that includes the following:

- Iteration Number
- the criterion being optimized
- criterion change
- Maximum and Average Absolute Change in standardized variable scores computed over variables that can be iteratively transformed
- notes
- final convergence status

ODS Table Names

PROC PRINQUAL assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 59.2. ODS Tables Produced in PROC PRINQUAL

ODS Table Name	Description	Statement	Option
ConvergenceStatus	Convergence Status		default
Footnotes	Iteration History Footnotes		default
MAC	MAC Iteration History	PROC	METHOD=MAC
MGV	MGV Iteration History	PROC	METHOD=MGV
MTV	MTV Iteration History	PROC	METHOD=MTV

ODS Graphics (Experimental)

This section describes the use of ODS for creating graphics with the PRINQUAL procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release. To request a graph you must specify the ODS GRAPHICS statement in addition to the following option. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

The following table shows the available plot *option*.

Option	Plot Description
MDPREF	Multidimensional preference analysis

ODS Graph Names

PROC PRINQUAL assigns a name to the graph it creates using ODS. You can use this name to reference the graph when using ODS. The name is listed in Table 59.3.

To request a graph you must specify the ODS GRAPHICS statement in addition to the option indicated in Table 59.3. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Table 59.3. ODS Graphics Produced by PROC PRINQUAL

ODS Graph Name	Plot Description	Statement	Option
PrinqualPlot	Multidimensional preference analysis	PROC	MDPREF

Examples

Example 59.1. Multidimensional Preference Analysis of Cars Data

This example uses PROC PRINQUAL to perform a nonmetric multidimensional preference (MDPREF) analysis (Carroll 1972). MDPREF analysis is a principal component analysis of a data matrix with columns that correspond to people and rows that correspond to objects. The data are ratings or rankings of each person's preference for each object. The data are the transpose of the usual multivariate data matrix. (In other words, the columns are people instead of the more typical matrix where rows represent people.) The final result of an MDPREF analysis is a biplot (Gabriel 1981) of the resulting preference space. A biplot displays the judges and objects in a single plot by projecting them onto the plane in the transformed variable space that accounts for the most variance.

The data are ratings by 25 judges of their preference for each of 17 automobiles. The ratings are made on a 0 to 9 scale, with 0 meaning very weak preference and 9 meaning very strong preference for the automobile. These judgments were made in 1980 about that year's products. There are two additional variables that indicate the manufacturer and model of the automobile.

This example uses PROC PRINQUAL, PROC FACTOR, and the %PLOTIT macro. PROC FACTOR is used before PROC PRINQUAL to perform a principal component analysis of the raw judgments. PROC FACTOR is also used immediately after PROC PRINQUAL since PROC PRINQUAL is a scoring procedure that optimally scores the data but does not report the principal component analysis.

The %PLOTIT macro produces the biplot. For information on the %PLOTIT macro, see Appendix B, "Using the %PLOTIT Macro."

The scree plot, in the standard principal component analysis reported by PROC FACTOR, shows that two principal components should be retained for further use. (See the scree plot in Output 59.1.1 —there is a clear separation between the first two components and the remaining components.) There are nine eigenvalues that are precisely zero because there are nine fewer observations than variables in the data matrix. PROC PRINQUAL is then used to monotonically transform the raw judgments to maximize the proportion of variance accounted for by the first two principal components. The following statements create the data set and perform a principal component analysis of the original data. These statements produce Output 59.1.1.

```
title 'Preference Ratings for Automobiles Manufactured in 1980';

data CarPref;
   input Make $ 1-10 Model $ 12-22 @25 (Judge1-Judge25) (1.);
   datalines;
Cadillac   Eldorado     8007990491240508971093809
Chevrolet  Chevette     0051200423451043003515698
Chevrolet  Citation     4053305814161643544747795
Chevrolet  Malibu       6027400723121345545668658
Ford       Fairmont     2024006715021443530648655
Ford       Mustang      5007197705021101850657555
Ford       Pinto        0021000303030201500514078
Honda      Accord       5956897609699952998975078
Honda      Civic        4836709507488852567765075
Lincoln    Continental  7008990592230409962091909
Plymouth   Gran Fury    7006000434101107333458708
Plymouth   Horizon      3005005635461302444675655
Plymouth   Volare       4005003614021602754476555
Pontiac    Firebird     0107895613201206958265907
Volkswagen Dasher       4858696508877795377895000
Volkswagen Rabbit       4858509709695795487885000
```

```
Volvo      DL           9989998909999987989919000
;

* Principal Component Analysis of the Original Data;
options ls=80 ps=65;
proc factor data=CarPref nfactors=2 scree;
   ods select Eigenvalues ScreePlot;
   var Judge1-Judge25;
   title3 'Principal Components of Original Data';
run;
```

Output 59.1.1. Principal Component Analysis of Original Data

```
            Preference Ratings for Automobiles Manufactured in 1980

                     Principal Components of Original Data

                              The FACTOR Procedure
                      Initial Factor Method: Principal Components

         Eigenvalues of the Correlation Matrix: Total = 25  Average = 1

                  Eigenvalue    Difference    Proportion    Cumulative

             1    10.8857202     5.0349926        0.4354        0.4354
             2     5.8507276     3.8077964        0.2340        0.6695
             3     2.0429312     0.5207808        0.0817        0.7512
             4     1.5221504     0.3078035        0.0609        0.8121
             5     1.2143469     0.2564839        0.0486        0.8606
             6     0.9578630     0.2197345        0.0383        0.8989
             7     0.7381286     0.1497259        0.0295        0.9285
             8     0.5884027     0.2117186        0.0235        0.9520
             9     0.3766841     0.1091250        0.0151        0.9671
            10     0.2675591     0.0773893        0.0107        0.9778
            11     0.1901698     0.0463921        0.0076        0.9854
            12     0.1437776     0.0349382        0.0058        0.9911
            13     0.1088394     0.0607418        0.0044        0.9955
            14     0.0480977     0.0056610        0.0019        0.9974
            15     0.0424367     0.0202714        0.0017        0.9991
            16     0.0221653     0.0221653        0.0009        1.0000
            17     0.0000000     0.0000000        0.0000        1.0000
            18     0.0000000     0.0000000        0.0000        1.0000
            19     0.0000000     0.0000000        0.0000        1.0000
            20     0.0000000     0.0000000        0.0000        1.0000
            21     0.0000000     0.0000000        0.0000        1.0000
            22     0.0000000     0.0000000        0.0000        1.0000
            23     0.0000000     0.0000000        0.0000        1.0000
            24     0.0000000     0.0000000        0.0000        1.0000
            25     0.0000000                      0.0000        1.0000
```

Output 59.1.1. (continued)

```
            Preference Ratings for Automobiles Manufactured in 1980

                     Principal Components of Original Data

                              The FACTOR Procedure
                  Initial Factor Method: Principal Components

Scree Plot of Eigenvalues
     |
     |
     |
     |
     |
  12 +
     |
     |
     |
     |       1
     |
     |
  10 +
     |
     |
     |
     |
     |
     |
   8 +
E    |
i    |
g    |
e    |
n    |
v    |
a  6 +
l    |         2
u    |
e    |
s    |
     |
     |
   4 +
     |
     |
     |
     |
     |
     |
   2 +            3
     |
     |              4
     |                 5
     |                   6  7
     |                        8
     |                           9 0  1 2
   0 +                                     3 4  5 6  7 8  9 0  1 2  3 4  5
     |
     |
     |
     |
     -----+----+----+----+----+----+----+----+----+----+----+----+----+----+----
          0    2    4    6    8   10   12   14   16   18   20   22   24   26

                                        Number
```

To fit the nonmetric MDPREF model, you can use the PRINQUAL procedure. The MONOTONE option is specified in the TRANSFORM statement to request a nonmetric MDPREF analysis; alternatively, you can instead specify the IDENTITY option for a metric analysis. Several options are used in the PROC PRINQUAL statement. The option DATA=CarPref specifies the input data set, OUT=Results creates an output data set, and N=2 and the default METHOD=MTV transform the data to better fit a two-component model. The REPLACE option replaces the original data with the monotonically transformed data in the OUT= data set. The MDPREF option standardizes the component scores to variance one so that the geometry of the biplot is correct, and it creates two variables in the OUT= data set named Prin1 and Prin2. These variables contain the standardized principal component scores and structure matrix, which are used to make the biplot. If the variables in data matrix $\mathbf{X}$ are standardized to mean zero and variance one, and n is the number of rows in $\mathbf{X}$, then $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{W}'$ is the principal component model, where $\mathbf{X}'\mathbf{X}/(n-1) = \mathbf{W}\mathbf{\Lambda}\mathbf{W}'$. The $\mathbf{W}$ and $\mathbf{\Lambda}$ contain the eigenvectors and eigenvalues of the correlation matrix of $\mathbf{X}$. The first two columns of $\mathbf{V}$, the standardized component scores, and $\mathbf{W}\mathbf{\Lambda}^{1/2}$, which is the structure matrix, are output. The advantage of creating a biplot based on principal components is that coordinates do not depend on the sample size. The following statements transform the data and produce Output 59.1.2.

```
* Transform the Data to Better Fit a Two Component Model;
proc prinqual data=CarPref out=Results n=2 replace mdpref;
   id model;
   transform monotone(Judge1-Judge25);
   title2 'Multidimensional Preference (MDPREF) Analysis';
   title3 'Optimal Monotonic Transformation of Preference Data';
run;
```

Output 59.1.2. Transformation of Automobile Preference Data

```
            Preference Ratings for Automobiles Manufactured in 1980
                 Multidimensional Preference (MDPREF) Analysis
               Optimal Monotonic Transformation of Preference Data

                            The PRINQUAL Procedure

                     PRINQUAL MTV Algorithm Iteration History

Iteration    Average    Maximum     Proportion    Criterion
   Number     Change     Change    of Variance       Change    Note
----------------------------------------------------------------------------
        1    0.24994    1.28017        0.66946
        2    0.07223    0.36958        0.80194      0.13249
        3    0.04522    0.29026        0.81598      0.01404
        4    0.03096    0.25213        0.82178      0.00580
        5    0.02182    0.23045        0.82493      0.00315
        6    0.01602    0.19017        0.82680      0.00187
        7    0.01219    0.14748        0.82793      0.00113
        8    0.00953    0.11031        0.82861      0.00068
        9    0.00737    0.06461        0.82904      0.00043
       10    0.00556    0.04469        0.82930      0.00026
       11    0.00445    0.04087        0.82944      0.00014
       12    0.00381    0.03706        0.82955      0.00011
       13    0.00319    0.03348        0.82965      0.00009
       14    0.00255    0.02999        0.82971      0.00006
       15    0.00213    0.02824        0.82976      0.00005
       16    0.00183    0.02646        0.82980      0.00004
       17    0.00159    0.02472        0.82983      0.00003
       18    0.00139    0.02305        0.82985      0.00003
       19    0.00123    0.02145        0.82988      0.00002
       20    0.00109    0.01993        0.82989      0.00002
       21    0.00096    0.01850        0.82991      0.00001
       22    0.00086    0.01715        0.82992      0.00001
       23    0.00076    0.01588        0.82993      0.00001
       24    0.00067    0.01440        0.82994      0.00001
       25    0.00059    0.00871        0.82994      0.00001
       26    0.00050    0.00720        0.82995      0.00000
       27    0.00043    0.00642        0.82995      0.00000
       28    0.00037    0.00573        0.82995      0.00000
       29    0.00031    0.00510        0.82995      0.00000
       30    0.00027    0.00454        0.82995      0.00000    Not Converged

 WARNING: Failed to converge, however criterion change is less than 0.0001.
```

The iteration history displayed by PROC PRINQUAL indicates that the proportion of variance is increased from an initial 0.66946 to 0.82995. The proportion of variance accounted for by PROC PRINQUAL on the first iteration equals the cumulative proportion of variance shown by PROC FACTOR for the first two principal components. In this example, PROC PRINQUAL's initial iteration performs a standard principal component analysis of the raw data. The columns labeled Average Change, Maximum Change, and Variance Change contain values that always decrease, indicating that PROC PRINQUAL is improving the transformations at a monotonically decreasing rate over the iterations. This does not always happen, and when it does not, it suggests that the analysis may be converging to a degenerate solution. See Example 59.3 on page 3688 for a discussion of a degenerate solution. The algorithm does not converge in 30 iterations. However, the criterion change is small, indicating that more iterations are unlikely to have much effect on the results.

The second PROC FACTOR analysis is performed on the transformed data. The WHERE statement is used to retain only the monotonically transformed judgments. The scree plot shows that the first two eigenvalues are now much larger than the remaining smaller eigenvalues. The second eigenvalue has increased markedly at the expense of the next several eigenvalues. Two principal components seem to be necessary and sufficient to adequately describe these judges' preferences for these automobiles. The cumulative proportion of variance displayed by PROC FACTOR for the first two principal components is 0.83. The following statements perform the analysis and produce Output 59.1.3:

```
* Final Principal Component Analysis;
proc factor data=Results nfactors=2 scree;
   ods select Eigenvalues ScreePlot;
   var Judge1-Judge25;
   where _TYPE_='SCORE';
   title3 'Principal Components of Monotonically Transformed Data';
run;
```

Output 59.1.3. Principal Components of Transformed Data

```
          Preference Ratings for Automobiles Manufactured in 1980
               Multidimensional Preference (MDPREF) Analysis
           Principal Components of Monotonically Transformed Data

                            The FACTOR Procedure
                   Initial Factor Method: Principal Components

      Eigenvalues of the Correlation Matrix: Total = 25  Average = 1

                Eigenvalue    Difference    Proportion    Cumulative

          1    11.5959045     2.4429455        0.4638        0.4638
          2     9.1529589     7.9952554        0.3661        0.8300
          3     1.1577036     0.3072013        0.0463        0.8763
          4     0.8505023     0.1284323        0.0340        0.9103
          5     0.7220700     0.2613540        0.0289        0.9392
          6     0.4607160     0.0958339        0.0184        0.9576
          7     0.3648821     0.0877851        0.0146        0.9722
          8     0.2770970     0.1250945        0.0111        0.9833
          9     0.1520025     0.0506622        0.0061        0.9894
         10     0.1013403     0.0292763        0.0041        0.9934
         11     0.0720640     0.0200979        0.0029        0.9963
         12     0.0519661     0.0336675        0.0021        0.9984
         13     0.0182987     0.0027059        0.0007        0.9991
         14     0.0155927     0.0093669        0.0006        0.9997
         15     0.0062258     0.0055503        0.0002        1.0000
         16     0.0006755     0.0006755        0.0000        1.0000
         17     0.0000000     0.0000000        0.0000        1.0000
         18     0.0000000     0.0000000        0.0000        1.0000
         19     0.0000000     0.0000000        0.0000        1.0000
         20     0.0000000     0.0000000        0.0000        1.0000
         21     0.0000000     0.0000000        0.0000        1.0000
         22     0.0000000     0.0000000        0.0000        1.0000
         23     0.0000000     0.0000000        0.0000        1.0000
         24     0.0000000     0.0000000        0.0000        1.0000
         25     0.0000000                      0.0000        1.0000
```

Output 59.1.3. (continued)

```
            Preference Ratings for Automobiles Manufactured in 1980
                 Multidimensional Preference (MDPREF) Analysis
             Principal Components of Monotonically Transformed Data

                              The FACTOR Procedure
                  Initial Factor Method: Principal Components

Scree Plot of Eigenvalues
     |
     |
     |
     |
     |
  12 +
     |       1
     |
     |
     |
     |
     |
  10 +
     |
     |
     |         2
     |
     |
     |
   8 +
E    |
i    |
g    |
e    |
n    |
v    |
a  6 +
l    |
u    |
e    |
s    |
     |
     |
   4 +
     |
     |
     |
     |
     |
     |
   2 +
     |
     |
     |            3
     |              4  5
     |                   6
     |                      7 8  9
   0 +                             0  1 2  3 4  5 6  7 8  9 0  1 2  3 4  5
     |
     |
     |
     |
     -----+----+----+----+----+----+----+----+----+----+----+----+----+----+----
          0    2    4    6    8   10   12   14   16   18   20   22   24   26

                                        Number
```

The remainder of the example constructs the MDPREF biplot. A biplot is a plot that displays the relation between the row points and the columns of a data matrix. The rows of $\mathbf{V}$, the standardized component scores, and $\mathbf{W\Lambda}^{1/2}$, which is the structure matrix, contain enough information to reproduce $\mathbf{X}$. The (i, j) element of $\mathbf{X}$ is the product of row i of $\mathbf{V}$ and row j of $\mathbf{W\Lambda}^{1/2}$. If all but the first two columns of $\mathbf{V}$ and $\mathbf{W\Lambda}^{1/2}$ are discarded, the (i, j) element of $\mathbf{X}$ is approximated by the product of row i of $\mathbf{V}$ and row j of $\mathbf{W\Lambda}^{1/2}$.

Since the MDPREF analysis is based on a principal component model, the dimensions of the MDPREF biplot are the first two principal components. The first principal component is the longest dimension through the MDPREF biplot. The first principal component is overall preference, which is the most salient dimension in the preference judgments. One end points in the direction that is on the average preferred most by the judges, and the other end points in the least preferred direction. The second principal component is orthogonal to the first principal component, and it is the orthogonal direction that is the second most salient. The interpretation of the second dimension varies from example to example.

With an MDPREF biplot, it is geometrically appropriate to represent each automobile (object) by a point and each judge by a vector. The automobile points have coordinates that are the scores of the automobile on the first two principal components. The judge vectors emanate from the origin of the space and go through a point with coordinates that are the coefficients of the judge (variable) on the first two principal components.

The absolute length of a vector is arbitrary. However, the relative lengths of the vectors indicate fit, with the squared lengths being proportional to the communalities in the PROC FACTOR output. The direction of the vector indicates the direction that is most preferred by the individual judge, with preference increasing as the vector moves from the origin. Let $\mathbf{v}'$ be row i of $\mathbf{V}$, $\mathbf{u}'$ be row j of $\mathbf{U} = \mathbf{W\Lambda}^{1/2}$, $\|\mathbf{v}\|$ be the length of $\mathbf{v}$, $\|\mathbf{u}\|$ be the length of $\mathbf{u}$, and θ be the angle between $\mathbf{v}$ and $\mathbf{u}$. The predicted degree of preference that an individual judge has for an automobile is $\mathbf{u}'\mathbf{v} = \|\mathbf{u}\|\ \|\mathbf{v}\| \cos\theta$. Each car point can be orthogonally projected onto the vector. The projection of car i on vector j is $\mathbf{u}((\mathbf{u}'\mathbf{v})/(\mathbf{u}'\mathbf{u}))$ and the length of this projection is $\|\mathbf{v}\| \cos\theta$. The automobile that projects farthest along a vector in the direction it points is that judge's most preferred automobile, since the length of this projection, $\|\mathbf{v}\| \cos\theta$, differs from the predicted preference, $\|\mathbf{u}\|\ \|\mathbf{v}\| \cos\theta$, only by $\|\mathbf{u}\|$, which is constant within each judge.

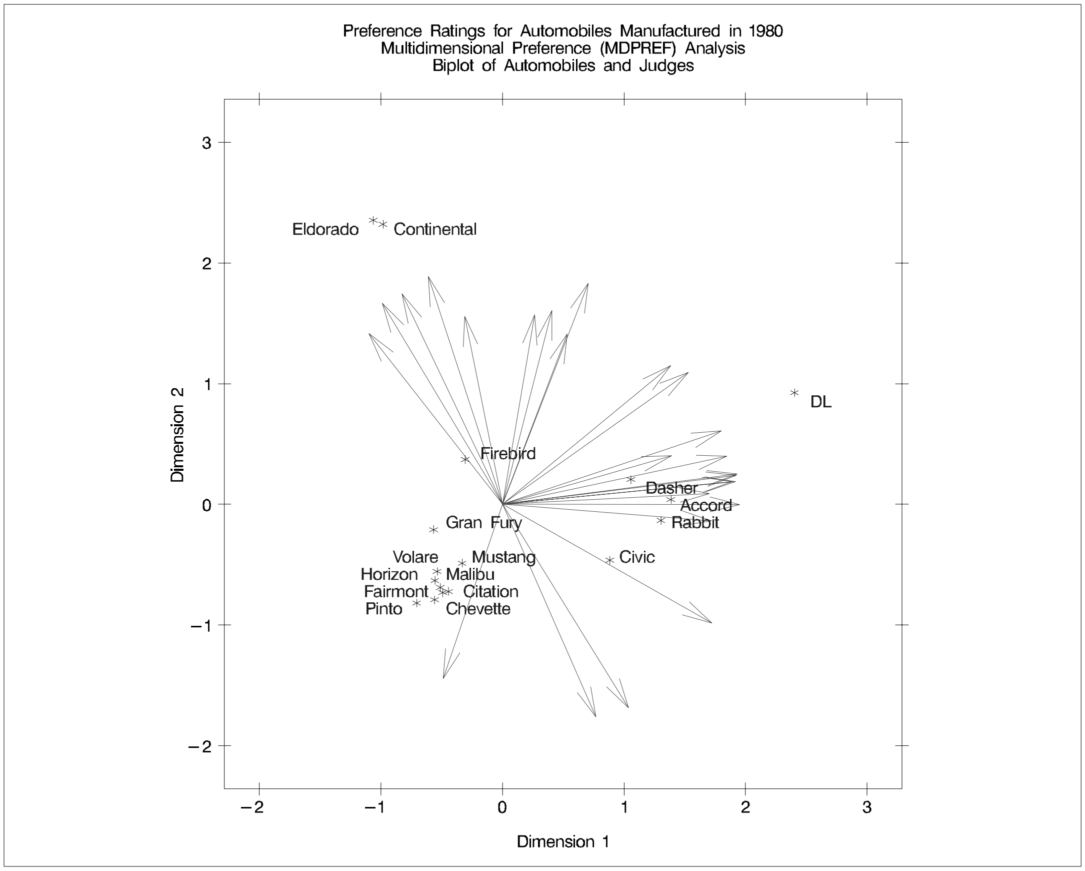
To interpret the biplot, look for directions through the plot that show a continuous change in some attribute of the automobiles, or look for regions in the plot that contain clusters of automobile points and determine what attributes the automobiles have in common. Those points that are tightly clustered in a region of the plot represent automobiles that have the same preference patterns across the judges. Those vectors that point in roughly the same direction represent judges who tend to have similar preference patterns.

The following statement constructs the biplot and produces Output 59.1.4:

```
title3 'Biplot of Automobiles and Judges';
%plotit(data=results, datatype=mdpref 2);
```

The DATATYPE=MDPREF 2 option indicates that the coordinates come from an MDPREF analysis, so the macro represents the scores as points and the structure as vectors, with the vectors stretched by a factor of two to make a better graphical display.

Output 59.1.4. Preference Ratings for Automobiles Manufactured in 1980



In the biplot, American automobiles are located on the left of the space, while European and Japanese automobiles are located on the right. At the top of the space are expensive American automobiles (Cadillac Eldorado, Lincoln Continental) while at the bottom are inexpensive ones (Pinto, Chevette). The first principal component differentiates American from imported automobiles, and the second arranges automobiles by price and other associated characteristics.

The two expensive American automobiles form a cluster, the sporty automobile (Firebird) is by itself, the Volvo DL is by itself, and the remaining imported autos form a cluster, as do the remaining American autos. It seems there are 5 prototypical automobiles in this set of 17, in terms of preference patterns among the 25 judges.

Most of the judges prefer the imported automobiles, especially the Volvo. There is also a fairly large minority that prefer the expensive cars, whether or not they are American (those with vectors that point towards one o'clock), or simply prefer expensive American automobiles (vectors that point towards eleven o'clock). There are two people who prefer anything except expensive American cars (five o'clock vectors), and one who prefers inexpensive American cars (seven o'clock vector).

Several vectors point toward the upper-right corner of the plot, toward a region with no cars. This is the region between the European and Japanese cars on the right and the luxury cars on the top. This suggests that there is a market for luxury Japanese and European cars.

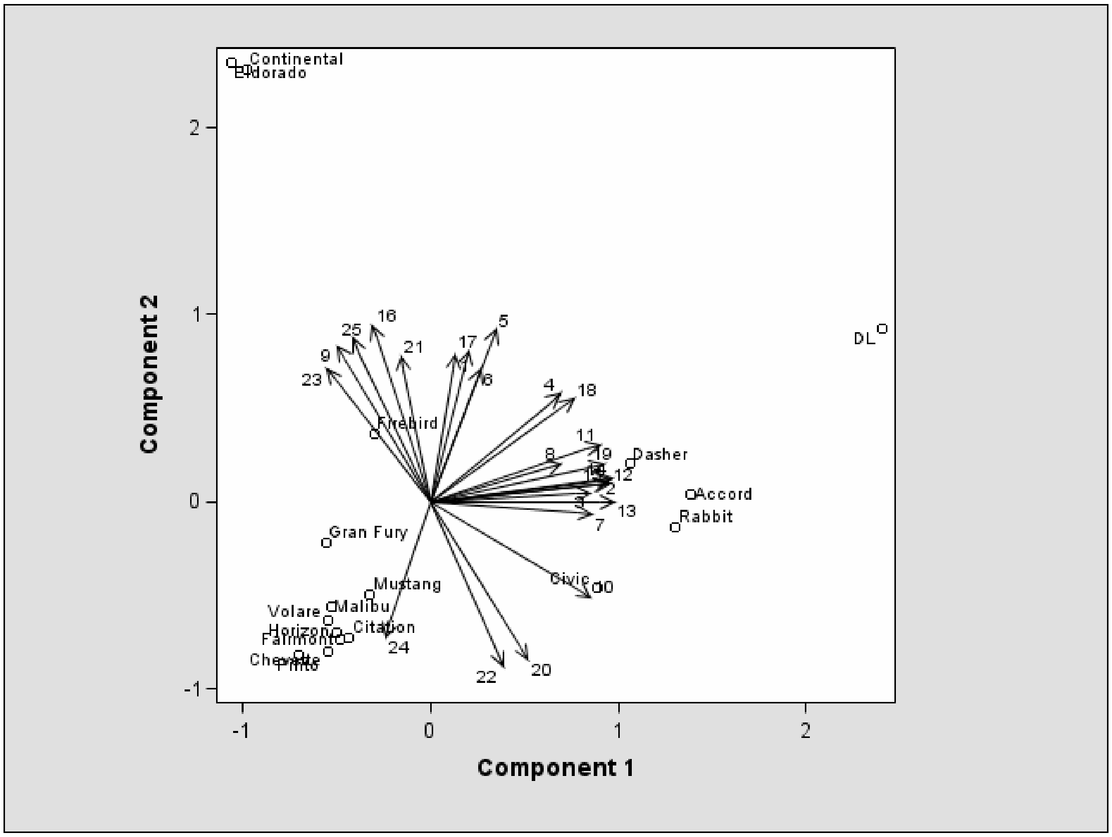
Example 59.2. Multidimensional Preference Analysis of Cars Data, ODS Graphics (Experimental)

The following graphical displays are requested by specifying the experimental ODS GRAPHICS statement. For general information about ODS graphics see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the PRINQUAL procedure, see the "ODS Graphics" section on page 3677.

```
title 'Preference Ratings for Automobiles Manufactured in 1980';

options validvarname=any;

data CarPref;
   input Make $ 1-10 Model $ 12-22 @25 ('1'n-'25'n) (1.);
   datalines;
Cadillac   Eldorado     8007990491240508971093809
Chevrolet  Chevette     0051200423451043003515698
Chevrolet  Citation     4053305814161643544747795
Chevrolet  Malibu       6027400723121345545668658
Ford       Fairmont     2024006715021443530648655
Ford       Mustang      5007197705021101850657555
Ford       Pinto        0021000303030201500514078
Honda      Accord       5956897609699952998975078
Honda      Civic        4836709507488852567765075
Lincoln    Continental  7008990592230409962091909
Plymouth   Gran Fury    7006000434101107333458708
Plymouth   Horizon      3005005635461302444675655
Plymouth   Volare       4005003614021602754476555
Pontiac    Firebird     0107895613201206958265907
Volkswagen Dasher       4858696508877795377895000
Volkswagen Rabbit       4858509709695795487885000
Volvo      DL           9899998909999987989919000
;

ods html;
ods graphics on;

proc prinqual data=CarPref out=Results n=2 replace mdpref maxiter=100;
   id model;
```

```
      transform monotone('1'n-'25'n);
      title2 'Multidimensional Preference (MDPREF) Analysis';
      title3 'Optimal Monotonic Transformation of Preference Data';
   run;

   ods graphics off;
   ods html close;
```

Output 59.2.1. Multidimensional Preference Analysis (Experimental)



Example 59.3. Principal Components of Basketball Rankings

The data in this example are 1985–1986 preseason rankings of 35 college basketball teams by 10 different news services. The services do not all rank the same teams or the same number of teams, so there are missing values in these data. Each of the 35 teams in the data set is ranked by at least one news service. One way of summarizing these data is with a principal component analysis, since the rankings should all be related to a single underlying variable, the first principal component.

You can use PROC PRINQUAL to estimate the missing ranks and compute scores for all observations. You can formulate a PROC PRINQUAL analysis that assumes that the observed ranks are ordinal variables and replaces the ranks with new numbers that are monotonic with the ranks and better fit the one principal component model.

The missing rank estimates need to be constrained since a news service would have positioned the unranked teams below the teams it ranked. PROC PRINQUAL should impose order constraints within the nonmissing values and between the missing and nonmissing values, but not within the missing values. PROC PRINQUAL has sophisticated missing data handling facilities; however, these facilities cannot directly handle this problem. The solution requires reformulating the problem.

By performing some preliminary data manipulations, specifying the N=1 option in the PROC PRINQUAL statement, and specifying the UNTIE transformation in the TRANSFORM statement, you can make the missing value estimates conform to the requirements. The PROC MEANS step finds the largest rank for each variable. The next DATA step replaces missing values with a value that is one larger than the largest observed rank. The N=1 option (in the PRINQUAL procedure) specifies that the variables should be transformed to make them as one-dimensional as possible. The UNTIE transformation in the TRANSFORM statement monotonically transforms the ranks, untying any ties in an optimal way. Because the only ties are for the values that replace the missing values, and because these values are larger than the observed values, the rescoring of the data satisfies the preceding requirements.

The following statements create the data set and perform the transformations discussed previously. These statements produce Output 59.3.1.

```
* Example 2: Basketball Data
*
* Preseason 1985 College Basketball Rankings
* (rankings of 35 teams by 10 news services)
*
* Note: (a) Various news services rank varying numbers of teams.
*       (b) Not all 35 teams are ranked by all news services.
*       (c) Each team is ranked by at least one service.
*       (d) Rank 20 is missing for UPI.;

title1 ’1985 Preseason College Basketball Rankings’;

data bballm;
   input School $13. CSN DurhamSun DurhamHerald WashingtonPost
         USA_Today SportMagazine InsideSports UPI AP
         SportsIllustrated;
   label CSN                 = ’Community Sports News (Chapel Hill, NC)’
         DurhamSun           = ’Durham Sun’
         DurhamHerald        = ’Durham Morning Herald’
         WashingtonPost      = ’Washington Post’
         USA_Today           = ’USA Today’
         SportMagazine       = ’Sport Magazine’
         InsideSports        = ’Inside Sports’
         UPI                 = ’United Press International’
         AP                  = ’Associated Press’
         SportsIllustrated = ’Sports Illustrated’
         ;
   format CSN--SportsIllustrated 5.1;
   datalines;
Louisville     1  8  1  9  8  9  6 10  9  9
```

```
Georgia Tech   2  2  4  3  1  1  1  2  1  1
Kansas         3  4  5  1  5 11  8  4  5  7
Michigan       4  5  9  4  2  5  3  1  3  2
Duke           5  6  7  5  4 10  4  5  6  5
UNC            6  1  2  2  3  4  2  3  2  3
Syracuse       7 10  6 11  6  6  5  6  4 10
Notre Dame     8 14 15 13 11 20 18 13 12  .
Kentucky       9 15 16 14 14 19 11 12 11 13
LSU           10  9 13  . 13 15 16  9 14  8
DePaul        11  . 21 15 20  . 19  .  . 19
Georgetown    12  7  8  6  9  2  9  8  8  4
Navy          13 20 23 10 18 13 15  . 20  .
Illinois      14  3  3  7  7  3 10  7  7  6
Iowa          15 16  .  . 23  .  . 14  . 20
Arkansas      16  .  .  . 25  .  .  .  . 16
Memphis State 17  . 11  . 16  8 20  . 15 12
Washington    18  .  .  .  .  .  . 17  .  .
UAB           19 13 10  . 12 17  . 16 16 15
UNLV          20 18 18 19 22  . 14 18 18  .
NC State      21 17 14 16 15  . 12 15 17 18
Maryland      22  .  .  . 19  .  .  . 19 14
Pittsburgh    23  .  .  .  .  .  .  .  .  .
Oklahoma      24 19 17 17 17 12 17  . 13 17
Indiana       25 12 20 18 21  .  .  .  .  .
Virginia      26  . 22  .  . 18  .  .  .  .
Old Dominion  27  .  .  .  .  .  .  .  .  .
Auburn        28 11 12  8 10  7  7 11 10 11
St. Johns     29  .  .  .  . 14  .  .  .  .
UCLA          30  .  .  .  .  .  . 19  .  .
St. Joseph’s   .  . 19  .  .  .  .  .  .  .
Tennessee      .  . 24  .  . 16  .  .  .  .
Montana        .  .  . 20  .  .  .  .  .  .
Houston        .  .  .  . 24  .  .  .  .  .
Virginia Tech  .  .  .  .  .  . 13  .  .  .
;

* Find maximum rank for each news service and replace
* each missing value with the next highest rank.;

proc means data=bballm noprint;
   output out=maxrank
      max=mcsn mdurs mdurh mwas musa mspom mins mupi map mspoi;
run;

data bball;
   set bballm;
   if _n_=1 then set maxrank;
   array services[10] CSN--SportsIllustrated;
   array maxranks[10] mcsn--mspoi;
   keep School CSN--SportsIllustrated;
   do i=1 to 10;
      if services[i]=. then services[i]=maxranks[i]+1;
      end;
run;
```

```
* Assume that the ranks are ordinal and that unranked teams
* would have been ranked lower than ranked teams.  Monotonically
* transform all ranked teams while estimating the unranked teams.
* Enforce the constraint that the missing ranks are estimated to
* be less than the observed ranks.  Order the unranked teams
* optimally within this constraint.  Do this so as to maximize
* the variance accounted for by one linear combination.  This
* makes the data as nearly rank one as possible, given the
* constraints.
*
* NOTE: The UNTIE transformation should be used with caution.
* If frequently produces degenerate results.;

proc prinqual data=bball out=tbball scores n=1 tstandard=z;
   title2 'Optimal Monotonic Transformation of Ranked Teams';
   title3 'with Constrained Estimation of Unranked Teams';
   transform untie(CSN -- SportsIllustrated);
   id School;
run;
```

Output 59.3.1. Transformation of Basketball Team Rankings

```
                  1985 Preseason College Basketball Rankings
               Optimal Monotonic Transformation of Ranked Teams
                with Constrained Estimation of Unranked Teams

                            The PRINQUAL Procedure

                   PRINQUAL MTV Algorithm Iteration History

Iteration    Average    Maximum     Proportion    Criterion
   Number     Change     Change    of Variance       Change    Note
----------------------------------------------------------------------------
        1    0.18563    0.76531        0.85850
        2    0.03225    0.14627        0.94362      0.08512
        3    0.02126    0.10530        0.94669      0.00307
        4    0.01467    0.07526        0.94801      0.00132
        5    0.01067    0.05282        0.94865      0.00064
        6    0.00800    0.03669        0.94899      0.00034
        7    0.00617    0.02862        0.94919      0.00020
        8    0.00486    0.02636        0.94932      0.00013
        9    0.00395    0.02453        0.94941      0.00009
       10    0.00327    0.02300        0.94947      0.00006
       11    0.00275    0.02166        0.94952      0.00005
       12    0.00236    0.02041        0.94956      0.00004
       13    0.00205    0.01927        0.94959      0.00003
       14    0.00181    0.01818        0.94962      0.00003
       15    0.00162    0.01719        0.94964      0.00002
       16    0.00147    0.01629        0.94966      0.00002
       17    0.00136    0.01546        0.94968      0.00002
       18    0.00128    0.01469        0.94970      0.00002
       19    0.00121    0.01398        0.94971      0.00001
       20    0.00115    0.01332        0.94973      0.00001
       21    0.00111    0.01271        0.94974      0.00001
       22    0.00105    0.01213        0.94975      0.00001
       23    0.00099    0.01155        0.94976      0.00001
       24    0.00095    0.01095        0.94977      0.00001
       25    0.00091    0.01038        0.94978      0.00001
       26    0.00088    0.00986        0.94978      0.00001
       27    0.00084    0.00936        0.94979      0.00001
       28    0.00081    0.00889        0.94980      0.00001
       29    0.00077    0.00846        0.94980      0.00000
       30    0.00073    0.00805        0.94980      0.00000    Not Converged

 WARNING: Failed to converge, however criterion change is less than 0.0001.
```

An alternative approach is to use the pairwise deletion option of the CORR procedure to compute the correlation matrix and then use PROC PRINCOMP or PROC FACTOR to perform the principal component analysis. This approach has several disadvantages. The correlation matrix may not be positive semidefinite (psd), an assumption required for principal component analysis. PROC PRINQUAL always produces a psd correlation matrix. Even with pairwise deletion, PROC CORR removes the six observations with only a single nonmissing value from this data set. Finally, it is still not possible to calculate scores on the principal components for those teams that have missing values.

It is possible to compute the composite ranking using PROC PRINCOMP and some preliminary data manipulations, similar to those discussed previously.

Chapter 58, "The PRINCOMP Procedure," contains an example where the average of the unused ranks in each poll is substituted for the missing values, and each observation is weighted by the number of nonmissing values. This method has much to recommend it. It is much faster and simpler than using PROC PRINQUAL. It is also much less prone to degeneracies and capitalization on chance. However, PROC PRINCOMP does not allow the nonmissing ranks to be monotonically transformed and the missing values untied to optimize fit.

PROC PRINQUAL monotonically transforms the observed ranks and estimates the missing ranks (within the constraints given previously) to account for almost 95 percent of the variance of the transformed data by just one dimension. PROC FACTOR is then used to report details of the principal component analysis of the transformed data. As shown by the Factor Pattern values in Output 59.3.2, nine of the ten news services have a correlation of 0.95 or larger with the scores on the first principal component after the data are optimally transformed. The scores are sorted and the composite ranking is displayed following the PROC FACTOR output. More confidence can be placed in the stability of the scores for the teams that are ranked by the majority of the news services than in scores for teams that are seldom ranked.

The monotonic transformations are plotted for each of the ten news services. These plots are the values of the raw ranks (with the missing ranks replaced by the maximum rank plus one) versus the rescored (transformed) ranks. The transformations are the step functions that maximize the fit of the data to the principal component model. Smoother transformations could be found by using MSPLINE transformations, but MSPLINE transformations would not correctly handle the missing data problem.

The following statements perform the final analysis and produce Output 59.3.2:

```
* Perform the Final Principal Component Analysis;
proc factor nfactors=1;
   var TCSN -- TSportsIllustrated;
   title4 'Principal Component Analysis';
run;

proc sort;
   by Prin1;
run;

* Display Scores on the First Principal Component;
proc print;
   title4 'Teams Ordered by Scores on First Principal Component';
   var School Prin1;
run;

* Plot the Transformations;
goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot;
   title;
   axis1 minor=none label=(angle=90 rotate=0)
```

```
         order=(-3 to 2 by 1);
   axis2 minor=none order=(0 to 40 by 10);
   plot TCSN*CSN                             / &opts name='prqex1';
   plot TDurhamSun*DurhamSun                 / &opts name='prqex2';
   plot TDurhamHerald*DurhamHerald           / &opts name='prqex3';
   plot TWashingtonPost*WashingtonPost       / &opts name='prqex4';
   plot TUSA_Today*USA_Today                 / &opts name='prqex5';
   plot TSportMagazine*SportMagazine         / &opts name='prqex6';
   plot TInsideSports*InsideSports           / &opts name='prqex7';
   plot TUPI*UPI                             / &opts name='prqex8';
   plot TAP*AP                               / &opts name='prqex9';
   plot TSportsIllustrated*SportsIllustrated / &opts name='prqex10';
   symbol1 c=blue;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:prqex1 2:prqex2 3:prqex3 4:prqex4;
   treplay 1:prqex5 2:prqex6 3:prqex7 4:prqex8;
   treplay 1:prqex9 3:prqex10;
run; quit;
```

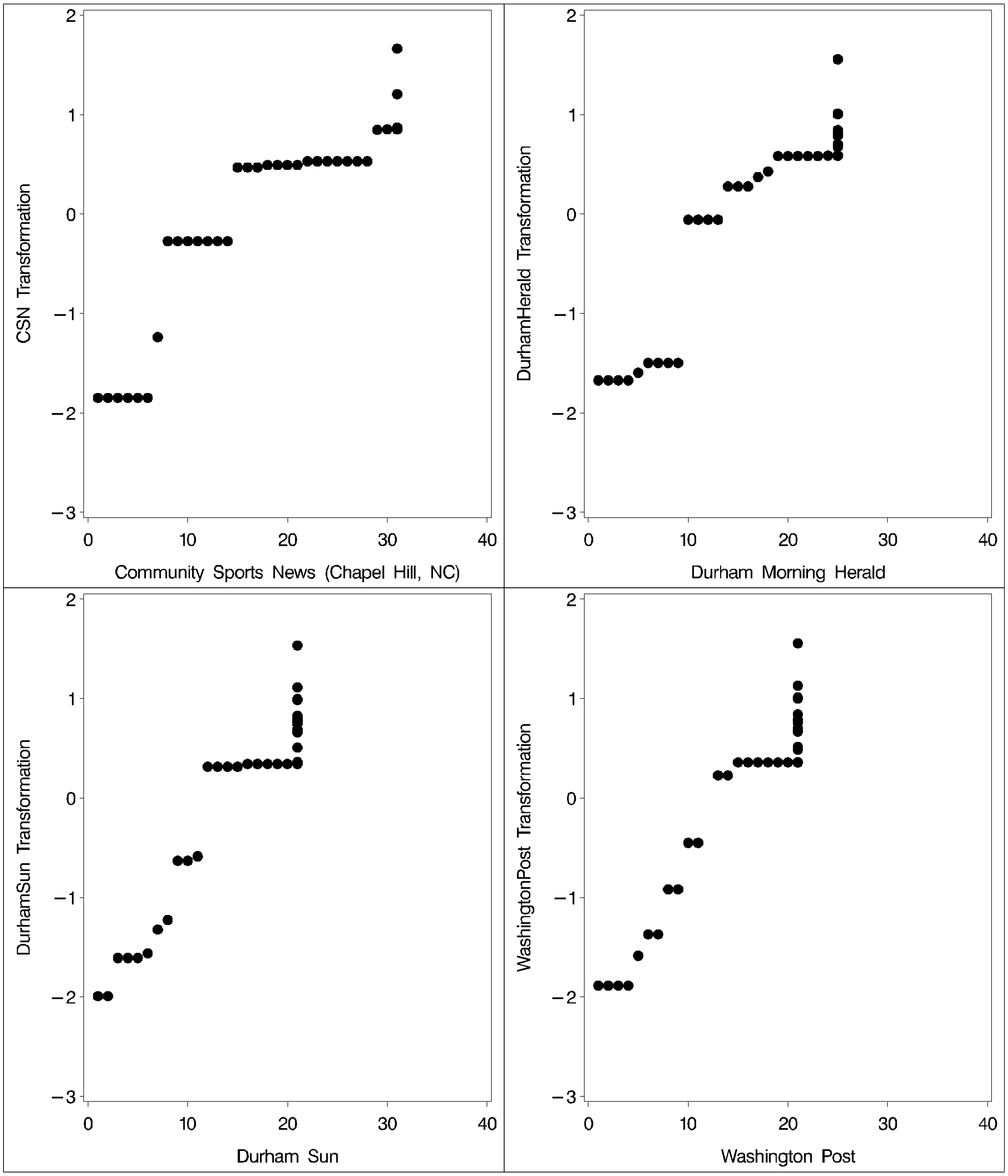
Output 59.3.2. Alternative Approach for Analyzing Basketball Rankings

```
                1985 Preseason College Basketball Rankings
             Optimal Monotonic Transformation of Ranked Teams
              with Constrained Estimation of Unranked Teams
                       Principal Component Analysis

                           The FACTOR Procedure
                 Initial Factor Method: Principal Components

                     Prior Communality Estimates: ONE



         Eigenvalues of the Correlation Matrix: Total = 10  Average = 1

                  Eigenvalue    Difference    Proportion    Cumulative

             1    9.49808040    9.27698055        0.9498        0.9498
             2    0.22109985    0.13434105        0.0221        0.9719
             3    0.08675881    0.01266762        0.0087        0.9806
             4    0.07409119    0.03048596        0.0074        0.9880
             5    0.04360523    0.00567364        0.0044        0.9924
             6    0.03793160    0.02098385        0.0038        0.9962
             7    0.01694775    0.00299099        0.0017        0.9979
             8    0.01395675    0.00982630        0.0014        0.9992
             9    0.00413045    0.00073249        0.0004        0.9997
            10    0.00339797                      0.0003        1.0000

              1 factor will be retained by the NFACTOR criterion.



                              Factor Pattern

                                                                 Factor1

     TCSN                    CSN Transformation                  0.91136
     TDurhamSun              DurhamSun Transformation            0.98887
     TDurhamHerald           DurhamHerald Transformation         0.97402
     TWashingtonPost         WashingtonPost Transformation       0.97408
     TUSA_Today              USA_Today Transformation            0.98867
     TSportMagazine          SportMagazine Transformation        0.95331
     TInsideSports           InsideSports Transformation         0.98521
     TUPI                    UPI Transformation                  0.98534
     TAP                     AP Transformation                   0.99590
     TSportsIllustrated      SportsIllustrated Transformation    0.98615


                      Variance Explained by Each Factor

                                 Factor1

                               9.4980804


              Final Communality Estimates: Total = 9.498080

                                      TDurham      TWashington
       TCSN      TDurhamSun            Herald             Post       TUSA_Today

   0.83057866      0.97785439      0.94870875       0.94882907       0.97747798

       TSport         TInside                                           TSports
     Magazine          Sports            TUPI              TAP      Illustrated

   0.90879058      0.97064640      0.97088804      0.99181626       0.97249026
```

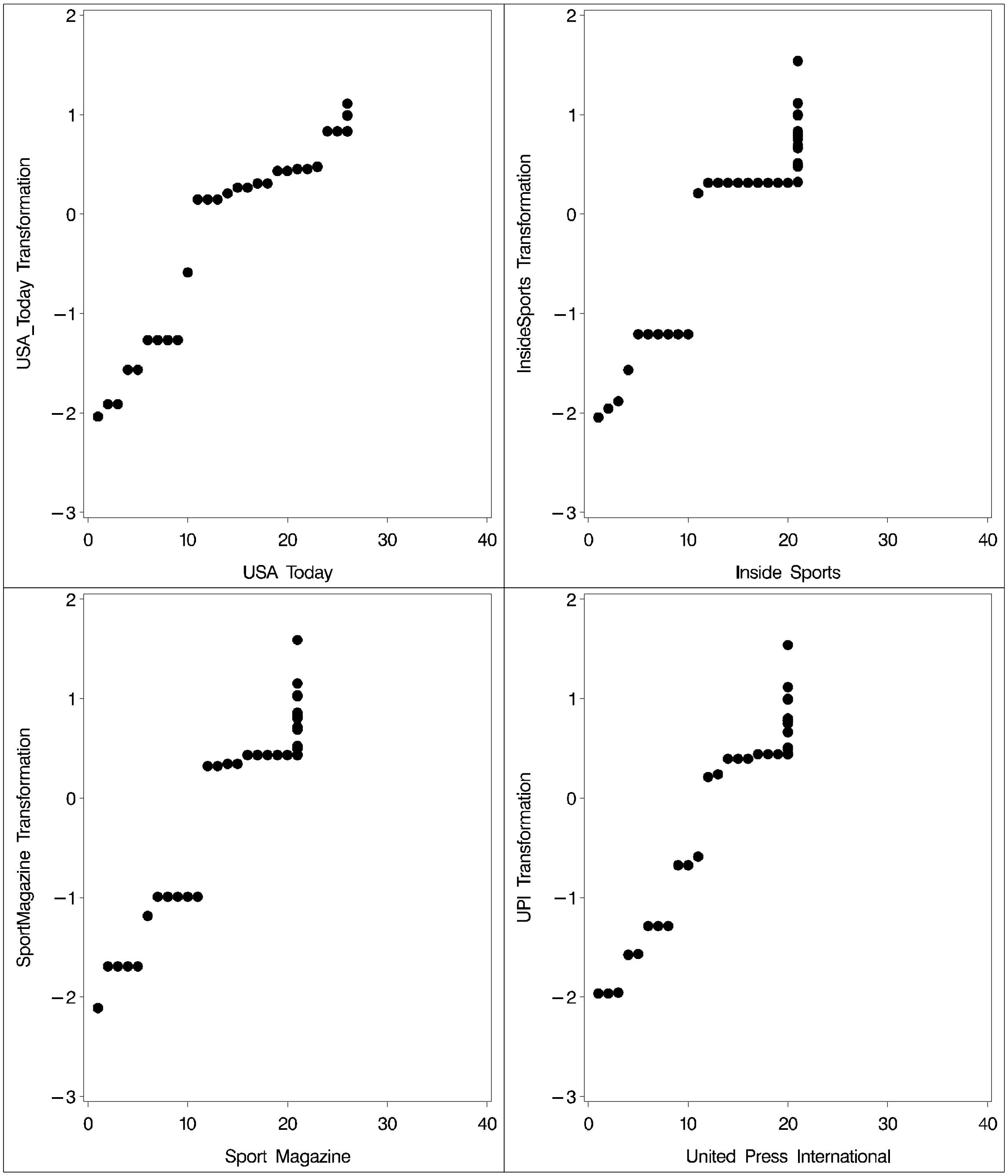
Output 59.3.2. (continued)

```
     1985 Preseason College Basketball Rankings
  Optimal Monotonic Transformation of Ranked Teams
   with Constrained Estimation of Unranked Teams
Teams Ordered by Scores on First Principal Component

          OBS    School             Prin1

            1    Georgia Tech     -6.20315
            2    UNC              -5.93314
            3    Michigan         -5.71034
            4    Kansas           -4.78699
            5    Duke             -4.75896
            6    Illinois         -4.19220
            7    Georgetown       -4.02861
            8    Louisville       -3.73087
            9    Syracuse         -3.47497
           10    Auburn           -1.78429
           11    LSU              -0.35928
           12    Memphis State     0.46737
           13    Kentucky          0.63661
           14    Notre Dame        0.71919
           15    Navy              0.76187
           16    UAB               0.98316
           17    DePaul            1.09891
           18    Oklahoma          1.12012
           19    NC State          1.15144
           20    UNLV              1.28766
           21    Iowa              1.45260
           22    Indiana           1.48123
           23    Maryland          1.54935
           24    Virginia          2.01385
           25    Arkansas          2.02718
           26    Washington        2.10878
           27    Tennessee         2.27770
           28    Virginia Tech     2.36103
           29    St. Johns         2.37387
           30    Montana           2.43502
           31    UCLA              2.52481
           32    Pittsburgh        3.00907
           33    Old Dominion      3.03324
           34    St. Joseph’s      3.39259
           35    Houston           4.69614
```

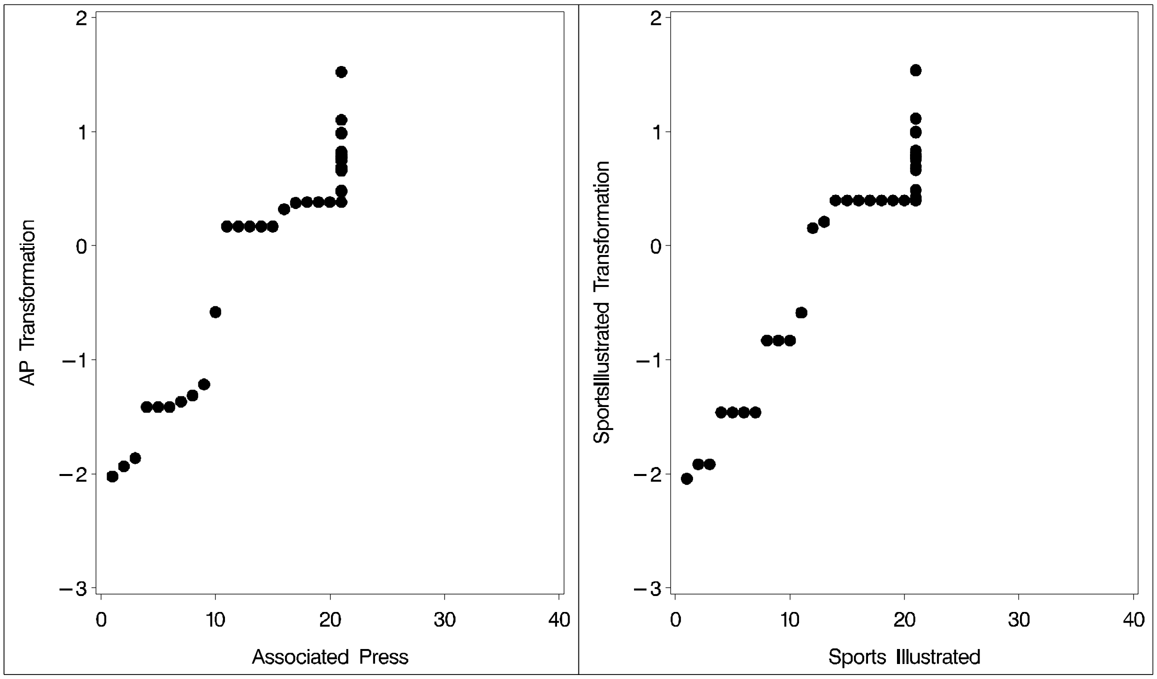
Output 59.3.3. Monotonic Transformation for Each News Service



Output 59.3.3. (continued)



Output 59.3.3. (continued)



The ordinary PROC PRINQUAL missing data handling facilities do not work for these data because they do not constrain the missing data estimates properly. If you code the missing ranks as missing and specify linear transformations, then you can compute least-squares estimates of the missing values without transforming the observed values. The first principal component then accounts for 92 percent of the variance after 20 iterations. However, Virginia Tech is ranked number 11 by its score even though it appeared in only one poll (InsideSports ranked it number 13, anchoring it firmly in the middle). Specifying monotone transformations is also inappropriate since they too allow unranked teams to move in between ranked teams.

With these data, the combination of monotone transformations and the freedom to score the missing ranks without constraint leads to degenerate transformations. PROC PRINQUAL tries to merge the 35 points into two points, producing a perfect fit in one dimension. There is evidence for this after 20 iterations when the Average Change, Maximum Change, and Variance Change values are all increasing, instead of the more stable decreasing change rate seen in the analysis shown. The change rates all stop increasing after 41 iterations, and it is clear by 70 or 80 iterations that one component will account for 100 percent of the transformed variables variance after sufficient iteration. While this may seem desirable (after all, it is a perfect fit), you should, in fact, be on guard when this happens. Whenever convergence is slow, the rates of change increase, or the final data perfectly fit the model, the solution is probably degenerating due to too few constraints on the scorings.

PROC PRINQUAL can account for 100 percent of the variance by scoring Montana and UCLA with one positive value on all variables and scoring all the other teams with one negative value on all variables. This inappropriate analysis suggests that all ranked teams are equally good except for two teams that are less good. Both of these two teams are ranked by only one news service, and their only nonmissing rank is last in the poll. This accounts for the degeneracy.

References

de Boor, C. (1978), *A Practical Guide to Splines*, New York: Springer Verlag.

Carroll, J.D. (1972), "Individual Differences and Multidimensional Scaling," in *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, eds. R.N. Shepard, A.K. Romney, and S.B. Nerlove, New York: Seminar Press.

Eckart, C. and Young, G. (1936), "The Approximation of One Matrix by Another of Lower Rank," *Psychometrika*, 1, 211–218.

Fisher, R. (1938), *Statistical Methods for Research Workers (10th Edition)*, Edinburgh: Oliver and Boyd Press.

Gabriel, K.R. (1981), "Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis," *Interpreting Multivariate Data*, ed. V. Barnett, London: John Wiley & Sons, Inc.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Goodnight, J.H. (1978), SAS Technical Report R-106, *The SWEEP Operator: Its Importance in Statistical Computing*, Cary, NC: SAS Institute Inc.

Harman, H.H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, 24, 498–520.

Kuhfeld, W.F., Sarle, W.S., and Young, F.W. (1985), "Methods of Generating Model Estimates in the PRINQUAL Macro," *SAS Users Group International Conference Proceedings: SUGI 10*, Cary, NC: SAS Institute Inc., 962–971.

Kruskal, J.B. (1964)," Multidimensional Scaling By Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, 29, 1–27.

Kruskal, J.B. and Shepard, R.N. (1974), "A Nonmetric Variety of Linear Factor Analysis," *Psychometrika*, 38, 123–157.

de Leeuw, J. (1985), (Personal Conversation).

de Leeuw, J. (1986), "Regression with Optimal Scaling of the Dependent Variable," Department of Data Theory, The University of Leiden, The Netherlands.

van Rijckeveorsel, J. (1982), “Canonical Analysis with B-Splines,” in *COMPUSTAT 1982*, Part I, eds. H. Caussinus, P. Ettinger, and R. Tomassone, Vienna: Wein, Physica Verlag.

Sarle, W.S. (1984), (Personal Conversation).

Siegel, S. (1956), *Nonparametric Statistics*, New York: McGraw Hill Book Co.

Smith, P.L. (1979), “Splines as a Useful and Convenient Statistical Tool,” *The American Statistician*, 33, 57–62.

Tenenhaus, M. and Vachette, J.L. (1977), “PRINQUAL: Un Programme d’Analyse en Composantes Principales d’un Ensemble de Variables Nominales ou Numeriques,” *Les Cahiers de Recherche #68*, CESA, Jouy-en-Josas, France.

Winsberg, S. and Ramsay, J.O. (1983), “Monotone Spline Transformations for Dimension Reduction,” *Psychometrika*, 48, 575–595.

Young, F.W. (1981), “Quantitative Analysis of Qualitative Data,” *Psychometrika*, 46, 357–388.

Young, F.W., Takane, Y., and de Leeuw, J. (1978), “The Principal Components of Mixed Measurement Level Multivariate Data: An Alternating Least Squares Method with Optimal Scaling Features,” *Psychometrika*, 43, 279–281.

Chapter 60
The PROBIT Procedure

Chapter Contents

Chapter 60
The PROBIT Procedure

Overview

The PROBIT procedure calculates maximum likelihood estimates of regression parameters and the natural (or threshold) response rate for quantal response data from biological assays or other discrete event data. This includes probit, logit, ordinal logistic, and extreme value (or gompit) regression models.

Probit analysis developed from the need to analyze qualitative (dichotomous or polytomous) dependent variables within the regression framework. Many response variables are binary by nature (yes/no), while others are measured ordinally rather than continuously (degree of severity). Collett (1991) and Agresti (1990), for example, have shown ordinary least squares (OLS) regression to be inadequate when the dependent variable is discrete. Probit or logit analyses are more appropriate in this case.

The PROBIT procedure computes maximum likelihood estimates of the parameters $\boldsymbol{\beta}$ and C of the probit equation using a modified Newton-Raphson algorithm. When the response Y is binary, with values 0 and 1, the probit equation is

$$p = \text{Pr}(Y = 0) = C + (1 - C)F(\mathbf{x}'\boldsymbol{\beta})$$

where

$\boldsymbol{\beta}$ is a vector of parameter estimates

F is a cumulative distribution function (the normal, logistic, or extreme value)

$\mathbf{x}$ is a vector of explanatory variables

p is the probability of a response

C is the natural (threshold) response rate

Notice that PROC PROBIT, by default, models the probability of the *lower* response levels. The choice of the distribution function F (normal for the probit model, logistic for the logit model, and extreme value or Gompertz for the gompit model) determines the type of analysis. For most problems, there is relatively little difference between the normal and logistic specifications of the model. Both distributions are symmetric about the value zero. The extreme value (or Gompertz) distribution, however, is not symmetric, approaching 0 on the left more slowly than it approaches 1 on the right. You can use the extreme value distribution where such asymmetry is appropriate.

For ordinal response models, the response, Y, of an individual or an experimental unit may be restricted to one of a (usually small) number, $k + 1(k \geq 1)$, of ordinal values, denoted for convenience by $1, \ldots, k, k + 1$. For example, the severity of coronary disease can be classified into three response categories as 1=no disease, 2=angina pectoris, and 3=myocardial infarction. The PROBIT procedure fits a common slopes cumulative model, which is a parallel lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$\Pr(Y \leq 1 \mid \mathbf{x}) = F(\mathbf{x}'\boldsymbol{\beta})$$

$$\Pr(Y \leq i \mid \mathbf{x}) = F(\alpha_i + \mathbf{x}'\boldsymbol{\beta}), \quad 2 \leq i \leq k$$

where $\alpha_2, \ldots, \alpha_k$ are $k-1$ intercept parameters. By default, the covariate vector **x** contains an overall intercept term.

You can set or estimate the natural (threshold) response rate C. Estimation of C can begin either from an initial value that you specify or from the rate observed in a control group. By default, the natural response rate is fixed at zero.

An observation in the data set analyzed by the PROBIT procedure may contain the response and explanatory values for one subject. Alternatively, it may provide the number of observed events from a number of subjects at a particular setting of the explanatory variables. In this case, PROC PROBIT models the probability of an event.

Getting Started

The following example illustrates how you can use the PROBIT procedure to compute the threshold response rate and regression parameter estimates for quantal response data.

Estimating the Natural Response Threshold Parameter

Suppose you want to test the effect of a drug at 12 dosage levels. You randomly divide 180 subjects into 12 groups of 15—one group for each dosage level. You then conduct the experiment and, for each subject, record the presence or absence of a positive response to the drug. You summarize the data by counting the number of subjects responding positively in each dose group. Your data set is as follows:

```
data study;
   input Dose Respond;
   Number = 15;
   datalines;
0     3
1.1   4
1.3   4
2.0   3
2.2   5
2.8   4
3.7   5
3.9   9
4.4   8
```

```
4.8  11
5.9  12
6.8  13
;
run;
```

The variable dose represents the amount of drug administered. The first group, receiving a dose level of 0, is the control group. The variable number represents the number of subjects in each group. All groups are equal in size; hence, number has the value 15 for all observations. The variable respond represents the number of subjects responding to the associated drug dosage.

You can model the probability of positive response as a function of dosage using the following statements:

```
proc probit data=study log10 optc;
   model respond/number=dose;
   output    out = new p = p_hat;
   predpplot var = dose  cfit = blue cframe=ligr inborder;
   inset;
   ippplot   var = dose  cfit = blue cframe=ligr inborder;
   inset;
run;
```

The DATA= option specifies that PROC PROBIT analyze the SAS data set study. The LOG10 option replaces the first continuous independent variable (dose) by its common logarithm. The OPTC option estimates the natural response rate. When you use the LOG10 option with the OPTC option, any observations with a dose value less than or equal to zero are used in the estimation as a control group.

The OUTPUT statement creates a new data set, new, that contains all the variables in the original data set, and a new variable, p_hat, that represents the predicted probabilities.

The MODEL statement specifies a proportional response using the variables respond and number in *events/trials* syntax. The variable dose is the stimulus or explanatory variable. The results from this analysis are displayed in the following figures.

```
                                Probit Procedure

                               Model Information

                Data Set                                 WORK.STUDY
                Events Variable                             Respond
                Trials Variable                              Number
                Number of Observations                           12
                Number of Events                                 81
                Number of Trials                                180
                Number of Events In Control Group                 3
                Number of Trials In Control Group                15
                Name of Distribution                         Normal
                Log Likelihood                         -104.3945783


 Algorithm converged.
```

Figure 60.1. Model Fitting Information for the PROBIT Procedure

Figure 60.1 displays background information about the model fit. Included are the name of the input data set, the response variables used, and the number of observations, events, and trials. The last line in Figure 60.1 shows the final value of the log-likelihood function.

Figure 60.2 displays the table of parameter estimates for the model. The parameter C, which is the natural response threshold or the proportion of individuals responding at zero dose, is estimated to be 0.2409. Since both the intercept and the slope coefficient have significant p-values (0.0020, 0.0010), you can write the model for

$$\text{Pr(response)} = C + (1 - C)F(\mathbf{x}'\beta)$$

as

$$\text{Pr(response)} = 0.2409 + 0.7591(\Phi(-4.1439 + 6.2308 \times \log_{10}\text{(dose)}))$$

where Φ is the normal cumulative distribution function.

```
                                Probit Procedure

                          Analysis of Parameter Estimates

                             Standard   95% Confidence     Chi-
    Parameter   DF Estimate     Error       Limits       Square Pr > ChiSq

    Intercept    1  -4.1438    1.3415  -6.7731  -1.5146    9.54     0.0020
    Log10(Dose)  1   6.2308    1.8996   2.5076   9.9539   10.76     0.0010
    _C_          1   0.2409    0.0523   0.1385   0.3433
```

Figure 60.2. Model Parameter Estimates for the PROBIT Procedure

Finally, PROC PROBIT specifies the resulting tolerance distribution by providing the mean MU and scale parameter SIGMA as well as the covariance matrix of the distribution parameters.

```
                              Probit Procedure

                  Probit Model in Terms of Tolerance Distribution

                                    MU          SIGMA

                              0.66506312    0.16049411


               Estimated Covariance Matrix for Tolerance Parameters

                               MU              SIGMA                _C_

          MU                0.001158         -0.000493           0.000954
          SIGMA            -0.000493          0.002394          -0.000999
          _C_               0.000954         -0.000999           0.002731
```

Figure 60.3. Tolerance Distribution Estimates for the PROBIT Procedure

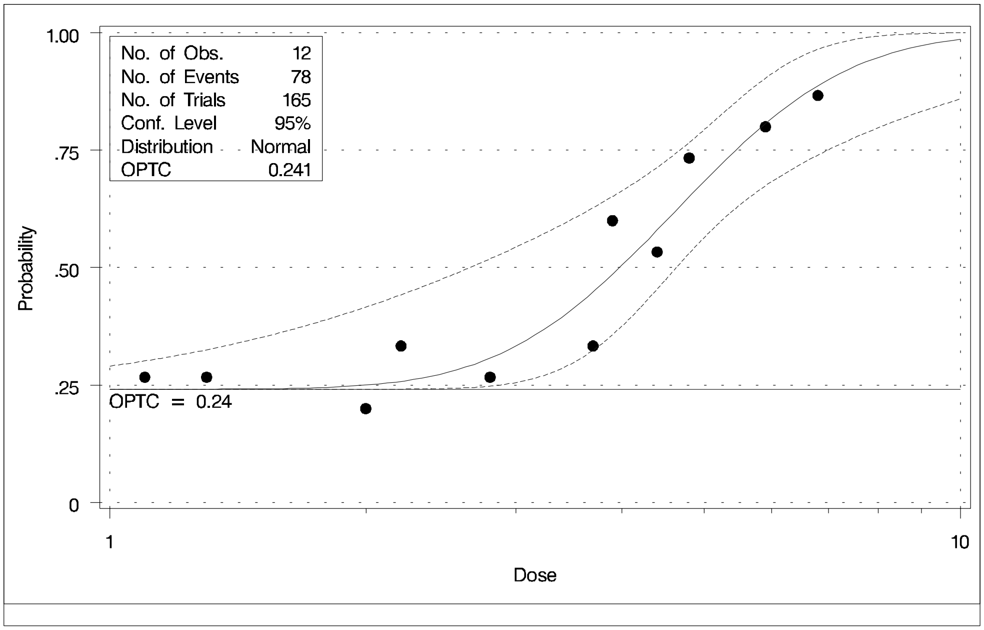


Figure 60.4. Plot of Observed and Fitted Probabilities versus Dose Level

The PREDPPLOT statement creates the plot in Figure 60.4, showing the relationship between dosage level, observed response proportions, and estimated probability values. The dashed lines represent pointwise confidence bands for the fitted probabilities, and a reference line is plotted at the estimated threshold value of .24.

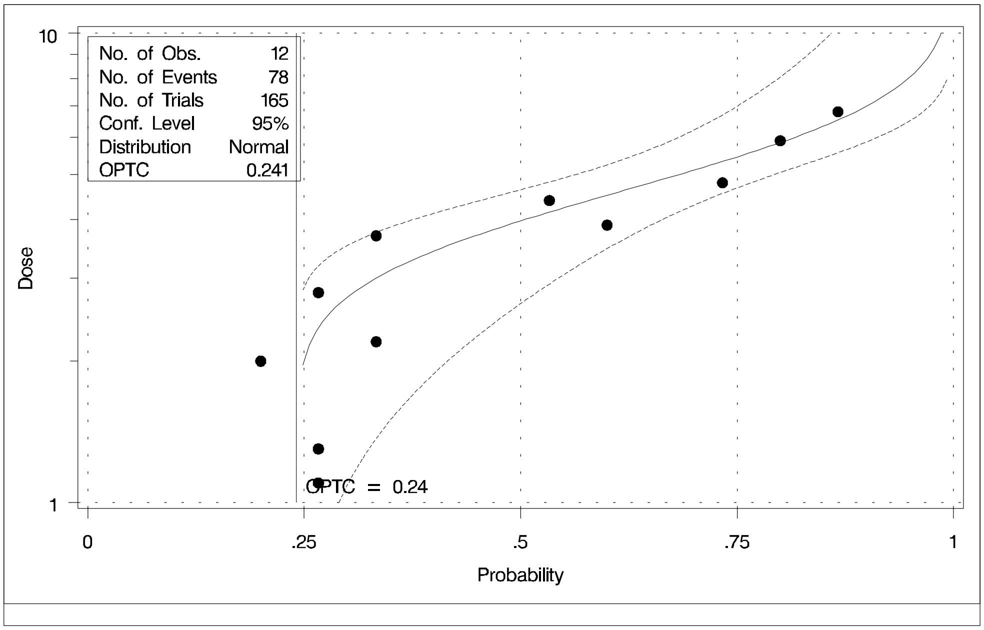


Figure 60.5. Inverse Predicted Probability Plot with Fiducial Limits

The IPPPLOT statement creates the plot in Figure 60.5, showing the inverse relationship between dosage level and observed response proportions/estimated probability values. The dashed lines represent pointwise fiducial limits for the predicted values of the dose variable, and a reference line is also plotted at the estimated threshold value of .24.

The INSET statement after each of these plot statements draws a box within the plot. In the inset box, summary information about the model fitting is printed.

Syntax

The following statements are available in PROC PROBIT.

PROC PROBIT < *options* > **;**
  MODEL *response=independents* < */ options* > **;**

  BY *variables* **;**
  CLASS *variables* **;**
  OUTPUT < **OUT=***SAS-data-set* > < *options* > **;**
  WEIGHT *variable* **;**

  CDFPLOT < *VAR = variable* > < *options* > **;**
  INSET < *keyword-list* > < */ options* > **;**
  IPPPLOT < *VAR = variable* > < *options* > **;**
  LPREDPLOT < *VAR = variable* > < *options* > **;**
  PREDPPLOT < *VAR = variable* > < *options* > **;**

A MODEL statement is required. Only a single MODEL statement can be used with one invocation of the PROBIT procedure. If multiple MODEL statements are present, only the last one is used. Main effects and higher-order terms can be specified in the MODEL statement, similar to the GLM procedure. If a CLASS statement is used, it must precede the MODEL statement.

The CDFPLOT, INSET, IPPPLOT, LPREDPLOT, and PREDPPLOT statements are used to produce graphical output. You can use any appropriate combination of the graphical statements after the MODEL statement.

PROC PROBIT Statement

PROC PROBIT < *options* > **;**

The PROC PROBIT statement starts the procedure. You can specify the following options in the PROC PROBIT statement.

COVOUT

writes the parameter estimate covariance matrix to the OUTEST= data set.

C=*rate*
OPTC

controls how the natural response is handled. Specify the OPTC option to request that the natural response rate C be estimated. Specify the C=*rate* option to set the natural response rate or to provide the initial estimate of the natural response rate. The natural response rate value must be a number between 0 and 1.

- If you specify neither the OPTC nor the C= option, a natural response rate of zero is assumed.
- If you specify both the OPTC and the C= option, the C= option should be a reasonable initial estimate of the natural response rate. For example, you could use the ratio of the number of responses to the number of subjects in a control group.
- If you specify the C= option but not the OPTC option, the natural response rate is set to the specified value and not estimated.
- If you specify the OPTC option but not the C= option, PROC PROBIT's action depends on the response variable, as follows:
  - If you specify either the LN or LOG10 option and some subjects have the first independent variable (dose) values less than or equal to zero, these subjects are treated as a control group. The initial estimate of C is then the ratio of the number of responses to the number of subjects in this group.
  - If you do not specify the LN or LOG10 option or if there is no control group, then one of the following occurs:
    - If all responses are greater than zero, the initial estimate of the natural response rate is the minimal response rate (the ratio of the number of responses to the number of subjects in a dose group) across all dose levels.

· If one or more of the responses is zero (making the response rate zero in that dose group), the initial estimate of the natural rate is the reciprocal of twice the largest number of subjects in any dose group in the experiment.

DATA=*SAS-data-set*

specifies the SAS data set to be used by PROC PROBIT. By default, the procedure uses the most recently created SAS data set.

GOUT=*graphics-catalog*

specifies a graphics catalog in which to save graphics output.

HPROB=*p*

specifies a minimum probability level for the Pearson chi-square to indicate a good fit. The default value is 0.10. The LACKFIT option must also be specified for this option to have any effect. For Pearson goodness-of-fit chi-square values with probability greater than the HPROB= value, the fiducial limits, if requested with the INVERSECL option, are computed using a critical value of 1.96. For chi-square values with probability less than the value of the HPROB= option, the critical value is a 0.95 two-sided quantile value taken from the t distribution with degrees of freedom equal to $(k-1) \times m - q$, where k is the number of levels for the response variable, m is the number of different sets of independent variable values, and q is the number of parameters fit in the model. Note that the HPROB= option can also appear in the MODEL statement.

INEST= *SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= *SAS-data-set*" on page 3757 for a detailed description of the contents of the INEST= data set.

INVERSECL

computes confidence limits for the values of the first continuous independent variable (such as dose) that yield selected response rates. If the algorithm fails to converge (this can happen when C is nonzero), missing values are reported for the confidence limits. See the section "Inverse Confidence Limits" on page 3761 for details. Note that the INVERSECL option can also appear in the MODEL statement.

LACKFIT

performs two goodness-of-fit tests (a Pearson chi-square test and a log-likelihood ratio chi-square test) for the fitted model.

To compute the test statistics, proper grouping of the observations into subpopulations is needed. You can use the AGGREGATE or AGGREGATE= option for this end. See the entry for the AGGREGATE and AGGREGATE= options under the MODEL statement. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

Note: This test is not appropriate if the data are very sparse, with only a few values at each set of the independent variable values.

If the Pearson chi-square test statistic is significant, then the covariance estimates and standard error estimates are adjusted.See the “Lack of Fit Tests” section on page 3759 for a description of the tests. Note that the LACKFIT option can also appear in the MODEL statement.

LOG
LN

analyzes the data by replacing the first continuous independent variable by its natural logarithm. This variable is usually the level of some treatment such as dosage. In addition to the usual output given by the INVERSECL option, the estimated dose values and 95% fiducial limits for dose are also displayed. If you specify the OPTC option, any observations with a dose value less than or equal to zero are used in the estimation as a control group. If you do not specify the OPTC option with the LOG or LN option, then any observations with the first continuous independent variable values less than or equal to zero are ignored.

LOG10

specifies an analysis like that of the LN or LOG option except that the common logarithm (log to the base 10) of the dose value is used rather than the natural logarithm.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, “Using the Output Delivery System.”

OPTC

controls how the natural response is handled. See the description of the C= option on page 3711 for details.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables specified in the CLASS statement, including the levels of the response variable. Response level ordering is important since PROC PROBIT always models the probability of response levels at the beginning of the ordering. See the section “Response Level Ordering” on page 3754 for further details. This ordering also determines which parameters in the model correspond to each level in the data. The following table shows how PROC PROBIT interprets values of the ORDER= option.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For the values FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide*.

OUTEST= SAS-data-set

specifies a SAS data set to contain the parameter estimates and, if the COVOUT option is specified, their estimated covariances. If you omit this option, the output data set is not created. The contents of the data set are described in the section "OUTEST= *SAS-data-set*" on page 3762.

XDATA= *SAS-data-set*

specifies an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement. If there are covariates specified in a MODEL statement, you specify fixed values for the effects in the MODEL statement by the XDATA= data set when predicted values and/or fiducial limits for a single continuous variable (dose variable) are required. These specified values for the effects in the MODEL statement are also used for generating plots. See the section "XDATA= *SAS-data-set*" on page 3763 for a detailed description of the contents of the XDATA= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PROBIT to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order on each of the BY variables, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PROBIT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CDFPLOT Statement

CDFPLOT <*var = variable*> <*options* >**;**

The CDFPLOT statement plots the predicted cumulative distribution function (CDF) of the multinomial response variable as a function of a single continuous independent variable (dose variable). You can only use this statement after a multinomial model statement.

VAR= *(variable)*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

The predicted cumulative distribution function is defined as

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}})$$

where $j = 1, \ldots, k$ are the indexes of the k levels of the multinomial response variable, *F* is the CDF of the distribution used to model the cumulative probabilities, $\hat{\mathbf{b}}$ is the vector of estimated parameters, $\mathbf{x}$ is the covariate vector, $\hat{a}_j$ are estimated ordinal intercepts with $\hat{a}_1 = 0$, and C is the threshold parameter, either known or estimated from the model. Let x_1 be the covariate corresponding to the dose variable and $\mathbf{x}_{-1}$ be the vector of the rest of the covariates. Let the corresponding estimated parameters be $\hat{b}_1$ and $\hat{\mathbf{b}}_{-1}$. Then

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + x_1\hat{b}_1 + \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})$$

To plot $\hat{F}_j$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

specify the levels of the multinomial response variable for which the cdf curves are requested, and add features to the plot. There are $k - 1$ curves for a k-level multinomial response variable (for the highest level, it is the constant line 1). You can specify any of them to be plotted by the LEVEL= option in the CDFPLOT statement. See the LEVEL= option for how to specify the levels.

An attached box on the right side of the plot is used to label these curves with the names of their levels. You can specify the color of this box using the CLABBOX= option.

You can use options in the CDFPLOT statement to

- superimpose specification limits
- specify the levels for which the cdf curves are requested
- specify graphical enhancements (such as color or text height)

Summary of Options

The following tables list all *options* by function. The “Dictionary of Options” on page 3718 describes each option in detail.

CDF Options

Table 60.1. Options for CDFPLOT

LEVEL=*character-list*	specifies the names of the levels for which the cdf curves are requested
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 60.2. Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CLABBOX=*color*	specifies color for label box
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 60.3. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies an ANNOTATE data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses cdf curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 60.4. Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 60.5. Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 60.6. Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 60.7. Options for Reference Lines

HREF< (INTERSECT)> =value-list	requests horizontal reference line
HREFLABELS= ('label1',. . .,'labeln')	specifies labels for HREF= lines
HREFLABPOS=n	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =value-list	requests vertical reference line
VREFLABELS= ('label1',. . .,'labeln')	specifies labels for VREF= lines
VREFLABPOS=n	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the CDFPLOT statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an ANNOTATE data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the cdf plot. The ANNOTATE= data set you specify in the CDFPLOT statement is used for all plots created by the statement.

CAXIS=*color*
CAXES=*color*
specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*
specifies the color for the fitted cdf curves. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*
specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*
specifies the color for grid lines. The default is the first color in the device color list.

CLABBOX=*color*
specifies the color for the area enclosed by the label box for cdf curves. This area is not shaded by default.

CHREF=*color*
CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION=’*string***’**
DES=’*string***’**

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the horizontal axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists are:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF <(INTERSECT)> =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'***,. . .,***'labeln'*
HREFLABEL=*'label1'***,. . .,***'labeln'*
HREFLAB=*'label1'***,. . .,***'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=n

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around cdf plots.

LEVEL= (*character-list***)**
ORDINAL= (*character-list***)**

specifies the names of the levels for which cdf curves are requested. Names should be quoted and separated by space. If there is no correct name provided, no cdf curve is plotted.

LFIT=*linetype*

specifies a line style for fitted curves. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*
LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOFIT

suppresses the fitted cdf curves.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=n

specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists are:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',. . .,'labeln'*
VREFLABEL=*'label1',. . .,'labeln'*
VREFLAB=*'label1',. . .,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=n

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VUPPER=*value*

specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=n
specifies line thickness for axes and frame. The default value is 1.

WFIT=n
specifies line thickness for fitted curves. The default value is 1.

WGRID=n
specifies line thickness for grids. The default value is 1.

WREFL=n
specifies line thickness for reference lines. The default value is 1.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Classification variables can be either character or numeric. If a single response variable is specified in the MODEL statement, it must also be specified in a CLASS statement.

Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in *SAS Language Reference: Dictionary*.

If the CLASS statement is used, it must appear before any of the MODEL statements.

INSET Statement

INSET <*keyword-list*> <*options*>**;**

The box or table of summary information produced on plots made with the CDFPLOT, IPPPLOT, LPREDPLOT, and PREDPPLOT statement is called an *inset*. You can use the INSET statement to customize both the information that is printed in the inset box and the appearance of the inset box. To supply the information that is displayed in the inset box, you specify *keywords* corresponding to the information you want shown. For example, the following statements produce a predicted probability plot with the number of trials, the number of events, the name of the distribution, and the estimated optimum natural threshold in the inset.

```
proc probit data=epidemic;
   model r/n = dose;
   predpplot ;
   inset nobs ntrials nevents dist optc;
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset containing the number of observations and the name of the distribution, labeled “Sample Size” and “Distribution” in the inset.

```
inset nobs='Sample Size' dist='Distribution';
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

The *options* control the appearance of the box.

If you specify more than one INSET statement, only the first one is used.

Keywords Used in the INSET Statement

The following tables list keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 60.8. Summary Statistics

NOBS	number of observations
NTRIALS	number of trials
NEVENTS	number of events
C	the user inputted threshold
OPTC	the estimated natural threshold
NRESPLEV	number of levels of the response variable

Table 60.9. General Information

CONFIDENCE	confidence coefficient for all confidence intervals or for the Weibayes fit
DIST	name of the distribution

Options Used in the INSET Statement

The following tables list the options available in the INSET statement.

Table 60.10. General Appearance Options

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text
HEADER=*'quoted string'*	specifies text for header or box title
NOFRAME	omits frame around box
POS= *value* <DATA | PERCENT>	determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in axis percent units or axis data units.
REFPOINT= *name*	specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. The REFPOINT= option specifies which corner of the inset frame you have specified with coordinates (x, y) and it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

Table 60.11. Color and Pattern Options

CFILL=*color*	specifies color for filling box
CFILLH=*color*	specifies color for filling box header
CFRAME=*color*	specifies color for frame
CHEADER=*color*	specifies color for text in header
CTEXT=*color*	specifies color for text

IPPPLOT Statement

IPPPLOT <*var* = *variable*> <*options*>**;**

The IPPPLOT statement plots the inverse of the predicted probability against a single continuous variable (dose variable) in the MODEL statement for the binomial model. You can only use this statement after a binomial model statement. The confidence limits for the predicted values of the dose variable are the computed fiducial limits, not the inverse of the confidence limits of the predicted probabilities. Refer to the section "Inverse Confidence Limits" on page 3761 for more details.

VAR= *(variable)*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

For the binomial model, the response variable is a probability. An estimate of the dose level $\hat{x}_1$ needed for a response of p is given by

$$\hat{x}_1 = (F^{-1}(p) - \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})/\hat{b}_1$$

where F is the cumulative distribution function used to model the probability, $\mathbf{x}_{-1}$ is the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ is the vector of the estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ is the estimated parameter for the dose variable of interest.

To plot $\hat{x}_1$ as a function of p, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

add features to the plot.

You can use options in the IPPPLOT statement to

- superimpose specification limits
- suppress or add the observed data points on the plot
- suppress or add the fiducial limits on the plot
- specify graphical enhancements (such as color or text height)

Summary of Options

The following tables list all *options* by function. The "Dictionary of Options" on page 3728 describes each option in detail.

IPP Options

Table 60.12. Plot Layout Options for IPPPLOT

NOCONF	suppresses fiducial limits
NODATA	suppresses observed data points on the plot
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 60.13. Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 60.14. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE= *SAS-data-set*	specifies an ANNOTATE data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 60.15. Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 60.16. Options for Reference Lines

HREF<(INTERSECT)> =value-list	requests horizontal reference line
HREFLABELS= ('label1',. . .,'labeln')	specifies labels for HREF= lines
HREFLABPOS=n	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =value-list	requests vertical reference line
VREFLABELS= ('label1',. . .,'labeln')	specifies labels for VREF= lines
VREFLABPOS=n	specifies horizontal position of labels for VREF= lines

Table 60.17. Graphics Catalog Options

DESCRIPTION='*string*'	specifies description for graphics catalog member
NAME='*string*'	specifies name for plot in graphics catalog

Table 60.18. Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the IPPPLOT statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an ANNOTATE data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the ipp plot. The ANNOTATE= data set you specify in the IPPPLOT statement is used for all plots created by the statement.

CAXIS=*color*
CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted ipp curves. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*
specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*
specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*
CH=*color*
specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*
specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*
specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**
DES='*string***'**
specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*
specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2***<***by value3***>**
specifies tick mark values for the horizontal axis. *value1, value2,* and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists are:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*
specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'*,. . .,*'labeln'*
HREFLABEL=*'label1'*,. . .,*'labeln'*
HREFLAB=*'label1'*,. . .,*'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around ipp plots.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*) and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*
LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF

suppresses fiducial limits from the plot.

NODATA

suppresses observed data points from the plot.

NOFIT

suppresses the fitted ipp curves.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=n

specifies the vertical position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	bottom

VAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists are:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',. . .,'labeln'*
VREFLABEL=*'label1',. . .,'labeln'*
VREFLAB=*'label1',. . .,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=n

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VUPPER=*value*

specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=n

specifies line thickness for axes and frame. The default value is 1.

WFIT=n

specifies line thickness for fitted curves. The default value is 1.

WGRID=n

specifies line thickness for grids. The default value is 1.

WREFL=n

specifies line thickness for reference lines. The default value is 1.

LPREDPLOT Statement

LPREDPLOT *<var = variable> <options>***;**

The LPREDPLOT statement plots the linear predictor $\mathbf{x}'\mathbf{b}$ against a single continuous variable (dose variable) in the MODEL statement for either the binomial model or the multinomial model. The confidence limits for the predicted values are only available for the binomial model.

VAR= *(variable)*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement for which the linear predictor plot is plotted. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

Let x_1 be the covariate of the dose variable, $\mathbf{x}_{-1}$ be the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ be the vector of estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ be the estimated parameter for the dose variable of interest.

To plot $\hat{\mathbf{x}}'\mathbf{b}$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

add features to the plot.

For the multinomial model, you can use the LEVEL= option to specify the levels for which the linear predictor lines are plotted. The lines are labeled by the names of their levels in the middle.

You can use options in the LPREDPLOT statement to

- superimpose specification limits

- suppress or add the observed data points on the plot for the binomial model
- suppress or add the confidence limits for the binomial model
- specify the levels for which the linear predictor lines are requested for the multinomial model
- specify graphical enhancements (such as color or text height)

Summary of Options

The following tables list all *options* by function. The "Dictionary of Options" on page 3736 describes each option in detail.

LPRED Options

Table 60.19. Plot Layout Options for LPREDPLOT

LEVEL=*character-list*	specifies the names of the levels for which the linear predictor lines are requested (only for the multinomial model)
NOCONF	suppresses fiducial limits (only for the binomial model)
NODATA	suppresses observed data points on the plot (only for the binomial model)
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 60.20. Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 60.21. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE= *SAS-data-set*	specifies an ANNOTATE data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 60.22. Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 60.23. Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 60.24. Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 60.25. Options for Reference Lines

HREF<(INTERSECT)> =value-list	requests horizontal reference line
HREFLABELS= ('label1',. . .,'labeln')	specifies labels for HREF= lines
HREFLABPOS=n	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =value-list	requests vertical reference line
VREFLABELS= ('label1',. . .,'labeln')	specifies labels for VREF= lines
VREFLABPOS=n	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the LPREDPLOT statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an ANNOTATE data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the lpred plot. The ANNOTATE= data set you specify in the LPREDPLOT statement is used for all plots created by the statement.

CAXIS=*color*
CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted lpred lines. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*
CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**
DES='*string***'**

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the horizontal axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists are:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'***,**. . .**,***'labeln'*
HREFLABEL=*'label1'***,**. . .**,***'labeln'*
HREFLAB=*'label1'***,**. . .**,***'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=n

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around lpred plots.

LEVEL= (*character-list***)**
ORDINAL= (*character-list***)**

specifies the names of the levels for which linear predictor lines are requested. Names should be quoted and separated by space. If there is no correct name provided, no lpred line is plotted.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*) and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*
LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF

suppresses confidence limits from the plot. This only works for the binomial model. Confidence limits are not plotted for the multinomial model.

NODATA

suppresses observed data points from the plot. This only works for the binomial model. Data points are not plotted for the multinomial model.

NOFIT

suppresses the fitted lpred lines.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=n

specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists are:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',...,'labeln'*
VREFLABEL=*'label1',...,'labeln'*
VREFLAB=*'label1',...,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=n

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VUPPER=*number*

specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *number* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=n

specifies line thickness for axes and frame. The default value is 1.

WFIT=n

specifies line thickness for fitted lines. The default value is 1.

WGRID=n

specifies line thickness for grids. The default value is 1.

WREFL=n

specifies line thickness for reference lines. The default value is 1.

MODEL Statement

<label:> **MODEL** *response=effects* < */ options* > **;**

<label:> **MODEL** *events/trials=effects* < */ options* > **;**

The MODEL statement names the variables used as the response and the independent variables. Additionally, you can specify the distribution used to model the response, as well as other options. Only a single MODEL statement can be used with one invocation of the PROBIT procedure. If multiple MODEL statements are present, only the last is used. Main effects and interaction terms can be specified in the MODEL statement, similar to the GLM procedure.

The optional *label* is used to label output from the matching MODEL statement.

The *response* can be a single variable with a value that is used to indicate the level of the observed response. Such a response variable must be listed in the CLASS statement. For example, the response might be a variable called Symptoms that takes on the values 'None,' 'Mild,' or 'Severe.' Note that, for dichotomous response variables, the probability of the lower sorted value is modeled by default (see the "Details" section beginning on page 3754). Because the model fit by the PROBIT procedure requires ordered response levels, you may need to use either the ORDER=DATA option in the PROC PROBIT statement or a numeric coding of the response to get the desired ordering of levels.

Alternatively, the response can be specified as a pair of variable names separated by a slash (/). The value of the first variable, *events*, is the number of positive responses (or events). The value of the second variable, *trials*, is the number of trials. Both variables must be numeric and non-negative, and the ratio of the first variable value to the second variable value must be between 0 and 1, inclusive. For example, the variables might be hits, a variable containing the number of hits for a baseball player, and AtBats, a variable containing the number of times at bat. A model for hitting proportion (batting average) as a function of age could be specified as

```
   model hits/AtBats=age;
```

The *effects* following the equal sign are the covariates in the model. Higher-order effects, such as interactions and nested terms, are allowed in the list, similar to the GLM procedure. Variable names and combinations of variable names representing higher-order terms are allowed to appear in this list. Class variables can be used as effects, and indicator variables are generated for the class levels. If you do not specify any covariates following the equal sign, an intercept-only model is fit.

The following options are available in the MODEL statement.

AGGREGATE
AGGREGATE= *(variable-list)*

specifies the subpopulations on which the Pearson chi-square test statistic and the log-likelihood ratio chi-square test statistic (deviance) are calculated if the LACKFIT option is specified. See the section "Rescaling the Covariance Matrix" on page 3760 for details of Pearson's chi-square and deviance calculations.

Observations with common values in the given list of variables are regarded as coming from the same subpopulation. Variables in the list can be any variables in the input data set. Specifying the AGGREGATE option is equivalent to specifying the AGGREGATE= option with a variable list that includes all independent variables in the MODEL statement. The PROBIT procedure sorts the input data set according to the variables specified in this list. Information for the sorted data set is reported in the "Response-Covariate Profile" table.

The deviance and Pearson goodness-of-fit statistics are calculated if the LACKFIT option is specified in the MODEL statement. The calculated results are reported in the "Goodness-of-Fit" table. If the Pearson chi-square test is significant with the test level specified by the HPROB= option, the fiducial limits, if required with the INVERSECL option in the MODEL statement, are modified (see the section "Inverse Confidence Limits" on page 3761 for details). Also, the covariance matrix is re-scaled by the dispersion parameter when the SCALE= option is specified.

ALPHA=*value*

sets the significance level for the confidence intervals for regression parameters, fiducial limits for the predicted values, and confidence intervals for the predicted probabilities. The value must be between 0 and 1. The default value is ALPHA=0.05.

CONVERGE=*value*

specifies the convergence criterion. Convergence is declared when the maximum change in the parameter estimates between Newton-Raphson steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change.

By default, CONVERGE=1.0E-8.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COVB

displays the estimated covariance matrix of the parameter estimates.

DISTRIBUTION=*distribution-type*
DIST=*distribution-type*
D=*distribution-type*

specifies the cumulative distribution function used to model the response probabilities. The distributions are described in the "Details" section beginning on page 3754. Valid values for *distribution-type* are

NORMAL the normal distribution for the probit model

LOGISTIC the logistic distribution for the logit model

EXTREMEVALUE | EXTREME | GOMPERTZ the extreme value, or Gompertz distribution for the gompit model

By default, DISTRIBUTION=NORMAL.

HPROB=*p*

specifies a minimum probability level for the Pearson chi-square to indicate a good fit. The default value is 0.10. The LACKFIT option must also be specified for this option to have any effect. For Pearson goodness-of-fit chi-square values with probability greater than the HPROB= value, the fiducial limits, if requested with the INVERSECL option, are computed using a critical value of 1.96. For chi-square values with probability less than the value of the HPROB= option, the critical value is a 0.95 two-sided quantile value taken from the t distribution with degrees of freedom equal to $(k-1)\times m-q$, where k is the number of levels for the response variable, m is the number of different sets of independent variable values, and q is the number of parameters fit in the model. If you specify the HPROB= option in both the PROC PROBIT and MODEL statements, the MODEL statement option takes precedence.

INITIAL=*values*

sets initial values for the parameters in the model other than the intercept. The values must be given in the order in which the variables are listed in the MODEL statement. If some of the independent variables listed in the MODEL statement are classification variables, then there must be as many values given for that variable as there are classification levels minus 1. The INITIAL option can be specified as follows.

Type of List	Specification
list separated by blanks	`initial=3 4 5`
list separated by commas	`initial=3,4,5`

By default, all parameters have initial estimates of zero.

Note: The INITIAL= option is overwritten by the INEST= option in the PROC PROBIT statement.

INTERCEPT=*value*

initializes the intercept parameter to *value*. By default, INTERCEPT=0.

INVERSECL

computes confidence limits for the values of the first continuous independent variable (such as dose) that yield selected response rates. If the algorithm fails to converge (this can happen when C is nonzero), missing values are reported for the confidence limits. See the section "Inverse Confidence Limits" on page 3761 for details.

ITPRINT

displays the iteration history, the final evaluation of the gradient, and the second derivative matrix (Hessian).

LACKFIT

performs two goodness-of-fit tests (a Pearson chi-square test and a log-likelihood ratio chi-square test) for the fitted model.

To compute the test statistics, proper grouping of the observations into subpopulations is needed. You can use the AGGREGATE or AGGREGATE= option to this end. See the entry for the AGGREGATE and AGGREGATE= options under the MODEL statement. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

Note: This test is not appropriate if the data are very sparse, with only a few values at each set of the independent variable values.

If the Pearson chi-square test statistic is significant, then the covariance estimates and standard error estimates are adjusted.See the section "Lack of Fit Tests" on page 3759 for a description of the tests. Note that the LACKFIT option can also appear in the PROC PROBIT statement. See the section "PROC PROBIT Statement" on page 3711 for details.

MAXITER=*value*
MAXIT=*value*

specifies the maximum number of iterations to be performed in estimating the parameters. By default, MAXITER=50.

NOINT

fits a model with no intercept parameter. If the INTERCEPT= option is also specified, the intercept is fixed at the specified value; otherwise, it is set to zero. This is most useful when the response is binary. When the response has k levels, then $k - 1$ intercept parameters are fit. The NOINT option sets the intercept parameter corresponding to the lowest response level equal to zero. A Lagrange multiplier, or score, test for the restricted model is computed when the NOINT option is specified.

SCALE= *scale*

enables you to specify the method for estimating the dispersion parameter. To correct for overdispersion or underdispersion, the covariance matrix is multiplied by the estimate of the dispersion parameter. Valid values for *scale* are as follows:

D | DEVIANCE	specifies that the dispersion parameter be estimated by the deviance divided by its degrees of freedom.
P | PEARSON	specifies that the dispersion parameter be estimated by the Pearson chi-square statistic divided by its degrees of freedom. This is set as the default.

You can use the AGGREGATE= option to define the subpopulations for calculating the Pearson chi-square statistic and the deviance.

The “Goodness-of-Fit ” table includes the Pearson chi-square statistic, the deviance, their degrees of freedom, the ratio of each statistic divided by its degrees of freedom, and the corresponding p-value.

SINGULAR=*value*

specifies the singularity criterion for determining linear dependencies in the set of independent variables. The sum of squares and cross-products matrix of the independent variables is formed and swept. If the relative size of a pivot becomes less than the value specified, then the variable corresponding to the pivot is considered to be linearly dependent on the previous set of variables considered. By default, SINGULAR=1E−12.

OUTPUT Statement

OUTPUT <*OUT=SAS-data-set*> <*keyword=name*. . .*keyword=name*>**;**

The OUTPUT statement creates a new SAS data set containing all variables in the input data set and, optionally, the fitted probabilities, the estimate of $\mathbf{x}'\beta$, and the estimate of its standard error. Estimates of the probabilities, $\mathbf{x}'\beta$, and the standard errors are computed for observations with missing response values as long as the values of all the explanatory variables are nonmissing. This enables you to compute these statistics for additional settings of the explanatory variables that are of interest but for which responses are not observed.

You can specify multiple OUTPUT statements. Each OUTPUT statement creates a new data set and applies only to the preceding MODEL statement. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets).

Details on the specifications in the OUTPUT statement are as follows:

keyword=name specifies the statistics to include in the output data set and assigns names to the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

PROB | P cumulative probability estimates

$$p = C + (1 - C)F(a_j + \mathbf{x}'\boldsymbol{\beta})$$

STD standard error estimates of $a_j + \mathbf{x}'\mathbf{b}$

XBETA estimates of $a_j + \mathbf{x}'\beta$

OUT=*SAS-data-set* names the output data set. By default, the new data set is named using the DATAn convention.

When the *single variable response* syntax is used, the _LEVEL_ variable is added to the output data set, and there are $k - 1$ output observations for each input observation, where k is the number of response levels. There is no observation output corresponding to the highest response level. For each of the $k - 1$ observations, the PROB variable contains the fitted probability of obtaining a response level up to the level indicated by the _LEVEL_ variable, the XBETA variable contains $a_j + \mathbf{x}'\mathbf{b}$, where j references the levels ($a_1 = 0$), and the STD variable contains the standard error estimate of the XBETA variable. See the “Details” section, which follows, for the formulas for the parameterizations.

PREDPPLOT Statement

PREDPPLOT *<var = variable> <options>***;**

The PREDPPLOT statement plots the predicted probability against a single continuous variable (dose variable) in the MODEL statement for both the binomial model and the multinomial model. Confidence limits are only available for the binomial model. An attached box on the right side of the plot is used to label predicted probability curves with the names of their levels for the multinomial model. You can specify the color of this box using the CLABBOX= option.

VAR= *(variable)*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

The predicted probability is

$$\hat{p} = C + (1 - C)F(\mathbf{x}'\hat{\mathbf{b}})$$

for the binomial model and

$$\begin{aligned}
\hat{p}_1 &= C + (1 - C)F(\mathbf{x}'\hat{\mathbf{b}}) \\
\hat{p}_j &= (1 - C)(F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}}) - F(\hat{a}_{j-1} + \mathbf{x}'\hat{\mathbf{b}})) \quad j = 2, \ldots, k - 1 \\
\hat{p}_k &= (1 - C)(1 - F(\hat{a}_{k-1} + \mathbf{x}'\hat{\mathbf{b}}))
\end{aligned}$$

for the multinomial model with k response levels, where F is the cumulative distribution function used to model the probability, $\mathbf{x}'$ is the vector of the covariates, $\hat{a}_j$ are the estimated ordinal intercepts with $\hat{a}_1 = 0$, C is the threshold parameter, either known or estimated from the model, and $\hat{\mathbf{b}}'$ is the vector of estimated parameters.

To plot $\hat{p}$ (or $\hat{p}_j$) as a function of a continuous variable x_1, the remaining covariates $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

enable you to plot the observed data and add features to the plot.

You can use options in the PREDPPLOT statement to

- superimpose specification limits
- suppress or add observed data points for the binomial model
- suppress or add confidence limits for the binomial model
- specify the levels for which predicted probability curves are requested for the multinomial model
- specify graphical enhancements (such as color or text height)

Summary of Options

The following tables list all *options* by function. The "Dictionary of Options" on page 3749 describes each option in detail.

PREDPPLOT Options

Table 60.26. Plot Layout Options for PREDPPLOT

LEVEL=*character-list*	specifies the names of the levels for which the predicted probability curves are requested (only for the multinomial model)
NOCONF	suppresses confidence limits
NODATA	suppresses observed data points on the plot
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 60.27. Color Options

CAXIS=*color*	specifies color for the axes
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CLABBOX=*color*	specifies color for label box
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 60.28. Options to Enhance Plots Produced on Graphics Devices

ANNOTATE= *SAS-data-set*	specifies an ANNOTATE data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 60.29. Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 60.30. Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 60.31. Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 60.32. Options for Reference Lines

HREF<(INTERSECT)> =value-list	requests horizontal reference line
HREFLABELS= ('label1',. . .,'labeln')	specifies labels for HREF= lines
HREFLABPOS=n	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =value-list	requests vertical reference line
VREFLABELS= ('label1',. . .,'labeln')	specifies labels for VREF= lines
VREFLABPOS=n	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the PREDPPLOT statement.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*
specifies an ANNOTATE data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the predicted probability plot. The ANNOTATE= data set you specify in the PREDPPLOT statement is used for all plots created by the statement.

CAXIS=*color*
CAXES=*color*
specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*
specifies the color for the fitted predicted probability curves. The default is the first color in the device color list.

CFRAME=*color*
CFR=*color*
specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*
specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*
CH=*color*
specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**
DES='*string***'**

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the horizontal axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists are:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas.

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies the offset for the horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'***,***. . .***,***'labeln'*
HREFLABEL=*'label1'***,***. . .***,***'labeln'*
HREFLAB=*'label1'***,***. . .***,***'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around predicted probability plots.

LEVEL= (*character-list***)**
ORDINAL= (*character-list***)**

specifies the names of the levels for which predicted probability curves are requested. Names should be quoted and separated by space. If there is no correct name provided, no fitted probability curve is plotted.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype* = *1*) and confidence limits are drawn by connecting dashed lines (*linetype* = *3*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*
LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*
LV=*linetype*
specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*
specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF
suppresses confidence limits from the plot. This only works for the binomial model. Confidence limits are not plotted for the multinomial model.

NODATA
suppresses observed data points from the plot. This only works for the binomial model. The data points are not plotted for the multinomial model.

NOFIT
suppresses the fitted predicted probability curves.

NOFRAME
suppresses the frame around plotting areas.

NOGRID
suppresses grid lines.

NOHLABEL
suppresses horizontal labels.

NOHTICK
suppresses horizontal tick marks.

NOTHRESH
suppresses the threshold line.

NOVLABEL
suppresses vertical labels.

NOVTICK
suppresses vertical tick marks.

THRESHLABPOS=n
specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VAXIS=*value1 to value2***<***by value3***>**

specifies tick mark values for the vertical axis. *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists are:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',. . .,'labeln'*
VREFLABEL=*'label1',. . .,'labeln'*
VREFLAB=*'label1',. . .,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=n

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VUPPER=*value*

specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=n

specifies line thickness for axes and frame. The default value is 1.

WFIT=n

specifies line thickness for fitted curves. The default value is 1.

WGRID=n

specifies line thickness for grids. The default value is 1.

WREFL=n

specifies line thickness for reference lines. The default value is 1.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement can be used with PROC PROBIT to weight each observation by the value of the variable specified. The contribution of each observation to the likelihood function is multiplied by the value of the weight variable. Observations with zero, negative, or missing weights are not used in model estimation.

Details

Missing Values

PROC PROBIT does not use any observations having missing values for any of the independent variables, the response variables, or the weight variable. If only the response variables are missing, statistics requested in the OUTPUT statement are computed.

Response Level Ordering

For binary response data, PROC PROBIT fits the following model by default,

$$\Phi^{-1}\left(\frac{p-C}{1-C}\right) = \mathbf{x}'\boldsymbol{\beta}$$

where p is the probability of the response level identified as the first level in the “Weighted Frequency Counts for the Ordered Response Categories” table in the output and Φ is the normal cumulative distribution function. By default, the covariate vector **x** contains an intercept term. This is sometimes called Abbot’s formula.

Because of the symmetry of the normal (and logistic) distribution, the effect of reversing the order of the two response values is to change the signs of $\boldsymbol{\beta}$ in the preceding equation.

By default, response levels appear in ascending, sorted order (that is, the lowest level appears first and then the next lowest, and so on). There are a number of ways that you can control the sort order of the response categories and, therefore, which level is assigned the first ordered level. One of the most common sets of response levels is {0,1}, with 1 representing the event with the probability that is to be modeled.

Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and EXPOSURE is the explanatory variable. By default, PROC PROBIT assigns the first ordered level to response level 0, causing the probability of the nonevent to be modeled. There are several ways to change this.

Besides recoding the variable Y, you can

- assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. For this example, Y=0 could be assigned formatted value ‘nonevent’ and Y=1 could be assigned formatted value ‘event.’ Since ORDER=FORMATTED by default, Y=1 becomes the first ordered level. See Example 60.3 for an illustration of this method.

```
proc format;
   value disease 1='event' 0='nonevent';
run;
proc probit;
   model y=exposure;
   format y disease.;
run;
```

- arrange the input data set so that Y=1 appears first and use the ORDER=DATA option in the PROC PROBIT statement. Since ORDER=DATA sorts levels in order of their appearance in the data set, Y=1 becomes the first ordered level. Note that this option causes class variables to be sorted by their order of appearance in the data set, also.

Computational Method

The log-likelihood function is maximized by means of a ridge-stabilized Newton-Raphson algorithm. Initial regression parameter estimates are set to zero. The INITIAL= and INTERCEPT= options in the MODEL statement can be used to give nonzero initial estimates.

The log-likelihood function, L, is computed as

$$L = \sum_i w_i \ln(p_i)$$

where the sum is over the observations in the data set, w_i is the weight for the ith observation, and p_i is the modeled probability of the observed response. In the case of the events/trials syntax in the MODEL statement, each observation contributes two terms corresponding to the probability of the event and the probability of its complement:

$$L = \sum_i w_i [r_i \ln(p_i) + (n_i - r_i) \ln(1 - p_i)]$$

where r_i is the number of events and n_i is the number of trials for observation i. This log-likelihood function differs from the log-likelihood function for a binomial or multinomial distribution by additive terms consisting of the log of binomial or multinomial coefficients. These terms are parameter-independent and do not affect the model estimation or the standard errors and tests.

The estimated covariance matrix, $\mathbf{V}$, of the parameter estimates is computed as the negative inverse of the information matrix of second derivatives of L with respect to the parameters evaluated at the final parameter estimates. Thus, the estimated covariance matrix is derived from the observed information matrix rather than the expected information matrix (these are generally not the same). The standard error estimates for the parameter estimates are taken as the square roots of the corresponding diagonal elements of $\mathbf{V}$.

If convergence of the maximum likelihood estimates is attained, a Type III chi-square test statistic is computed for each effect, testing whether there is any contribution from any of the levels of the effect. This statistic is computed as a quadratic form in the appropriate parameter estimates using the corresponding submatrix of the asymptotic covariance matrix estimate. Refer to Chapter 32, “The GLM Procedure,” and Chapter 11, “The Four Types of Estimable Functions,” for more information about Type III estimable functions.

The asymptotic covariance matrix is computed as the inverse of the observed information matrix. Note that if the NOINT option is specified and class variables are used, the first class variable contains a contribution from an intercept term. The results are displayed in an ODS table named **Type3Analysis**.

Chi-square tests for individual parameters are Wald tests based on the observed information matrix and the parameter estimates. If an effect has a single degree of freedom in the parameter estimates table, the chi-square test for this parameter is equivalent to the Type III test for this effect.

In releases previous to Version 8.2, a multiple degree of freedom statistic was computed for each effect to test for contribution from any level of the effect. In general, the Type III test statistic in a main effect only model (no interaction terms) will be equal to the previously computed effect statistic, unless there are collinearities among the effects. If there are collinearities, the Type III statistic will adjust for them, and the value of the Type III statistic and the number of degrees of freedom might not be equal to those of the previous effect statistic.

The theory behind these tests assumes large samples. If the samples are not large, it may be better to base the tests on log-likelihood ratios. These changes in log likelihood can be obtained by fitting the model twice, once with all the parameters of interest and once leaving out the parameters to be tested. Refer to Cox and Oakes (1984) for a discussion of the merits of some possible test methods.

If some of the independent variables are perfectly correlated with the response pattern, then the theoretical parameter estimates may be infinite. Although fitted probabilities of 0 and 1 are not especially pathological, infinite parameter estimates are required to yield these probabilities. Due to the finite precision of computer arithmetic, the actual parameter estimates are not infinite. Indeed, since the tails of the distributions allowed in the PROBIT procedure become small rapidly, an argument to the cumulative distribution function of around 20 becomes effectively infinite. In the case of such parameter estimates, the standard error estimates and the corresponding chi-square tests are not trustworthy.

Distributions

The distributions, $F(x)$, allowed in the PROBIT procedure are specified with the DISTRIBUTION= option in the model statement. The cumulative distribution functions for the available distributions are

$$\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) \, dz \text{(normal)}$$

$$\frac{1}{1+e^{-x}}\text{(logistic)}$$

$$1 - e^{-e^{x}}\text{(extreme value or Gompertz)}$$

The variances of these three distributions are not all equal to 1, and their means are not all equal to zero. Their means and variances are shown in the following table, where γ is the Euler constant.

Distribution	**Mean**	**Variance**
Normal	0	1
Logistic	0	$\pi^2/3$
extreme value or Gompertz	$-\gamma$	$\pi^2/6$

When comparing parameter estimates using different distributions, you need to take into account the different scalings and, for the extreme value (or Gompertz) distribution, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates from the logistic model should be about $\pi/\sqrt{3}$ larger than the estimates from the probit model.

INEST= *SAS-data-set*

The INEST= data set names a SAS data set that specifies initial estimates for all the parameters in the model.

The INEST= data set must contain the intercept variables (named Intercept for binary response model and Intercept, Intercept2, Intercept3, and so forth, for multinomial response models) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in a BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values. Combining the INEST= data set and the option MAXIT= in the MODEL statement, partial scoring can be done, such as predicting on a validation data set by using the model built from a training data set.

You can specify starting values for the iterative algorithm in the INEST= data set. This data set overwrites the INITIAL= option in the MODEL statement, which is a little difficult to use for models with multilevel interaction effects. The INEST= data set has the same structure as the "OUTEST= *SAS-data-set*" on page 3762, but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization.

Model Specification

For a two-level response, the probability that the lesser response occurs is modeled by the probit equation as

$$p = C + (1 - C)F(\mathbf{x}'\mathbf{b})$$

The probability of the other (complementary) event is $1 - p$.

For a multilevel response with outcomes labeled l_i for $i = 1, 2, \ldots, k$, the probability, p_j, of observing level l_j is as follows.

$$\begin{aligned}
p_1 &= C + (1 - C)F(\mathbf{x}'\mathbf{b}) \\
p_2 &= (1 - C)\left(F(a_2 + \mathbf{x}'\mathbf{b}) - F(\mathbf{x}'\mathbf{b})\right) \\
&\vdots \\
p_j &= (1 - C)\left(F(a_j + \mathbf{x}'\mathbf{b}) - F(a_{j-1} + \mathbf{x}'\mathbf{b})\right) \\
&\vdots \\
p_k &= (1 - C)(1 - F(a_{k-1} + \mathbf{x}'\mathbf{b}))
\end{aligned}$$

Thus, for a k-level response, there are $k - 2$ additional parameters, $a_2, a_3, \ldots, a_{k-1}$, estimated. These parameters are denoted by Intercept$j$, $j = 2, 3, \ldots, k - 1$ in the output.

An intercept parameter is always added to the set of independent variables as the first term in the model unless the NOINT option is specified in the MODEL statement. If a classification variable taking on k levels is used as one of the independent variables, a set of k indicator variables is generated to model the effect of this variable. Because of the presence of the intercept term, there are at most $k - 1$ degrees of freedom for this effect in the model.

Lack of Fit Tests

Two goodness-of-fit tests can be requested from the PROBIT procedure: a Pearson chi-square test and a log-likelihood ratio chi-square test.

To compute the test statistics, you can use the AGGREGATE or AGGREGATE= option grouping the observations into subpopulations. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes that each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

If the Pearson goodness-of-fit chi-square test is requested and the p-value for the test is too small, variances and covariances are adjusted by a heterogeneity factor (the goodness-of-fit chi-square divided by its degrees of freedom) and a critical value from the t distribution is used to compute the fiducial limits. The Pearson chi-square test statistic is computed as

$$\chi_P^2 = \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{(r_{ij} - n_i \hat{p}_{ij})^2}{n_i \hat{p}_{ij}}$$

where the sum on i is over grouping, the sum on j is over levels of response, the r_{ij} is the frequency of response level j for the ith grouping, n_i is the total frequency for the ith grouping, and $\hat{p}_{ij}$ is the fitted probability for the jth level at the ith grouping.

The likelihood ratio chi-square test statistic is computed as

$$\chi_D^2 = 2 \sum_{i=1}^{m} \sum_{j=1}^{k} r_{ij} \ln \left(\frac{r_{ij}}{n_i \hat{p}_{ij}} \right)$$

This quantity is sometimes called the deviance. If the modeled probabilities fit the data, these statistics should be approximately distributed as chi-square with degrees of freedom equal to $(k-1) \times m - q$, where k is the number of levels of the multinomial or binomial response, m is the number of sets of independent variable values (covariate patterns), and q is the number of parameters fit in the model.

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the groupings. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

Rescaling the Covariance Matrix

One way of correcting overdispersion is to multiply the covariance matrix by a dispersion parameter. You can supply the value of the dispersion parameter directly, or you can estimate the dispersion parameter based on either the Pearson chi-square statistic or the deviance for the fitted model.

The Pearson chi-square statistic χ^2_P and the deviance χ^2_D are defined in the section "Lack of Fit Tests" on page 3759. If the SCALE= option is specified in the MODEL statement, the dispersion parameter is estimated by

$$\widehat{\sigma}^2 = \begin{cases} \chi^2_P/(m(k-1)-q) & \text{SCALE=PEARSON} \\ \chi^2_D/(m(k-1)-q) & \text{SCALE=DEVIANCE} \\ (constant)^2 & \text{SCALE=}\textit{constant} \end{cases}$$

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the subpopulations. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulation profiles. If you do not specify this option, each observation is regarded as coming from a separate subpopulation. For *events/trials* syntax, each observation represents n Bernoulli trials, where n is the value of the *trials* variable; for *single-trial* syntax, each observation represents a single trial. Without the AGGREGATE (or AGGREGATE=) option, the Pearson chi-square statistic and the deviance are calculated only for *events/trials* syntax.

Note that the parameter estimates are not changed by this method. However, their standard errors are adjusted for overdispersion, affecting their significance tests.

Tolerance Distribution

For a single independent variable, such as a dosage level, the models for the probabilities can be justified on the basis of a population with mean μ and scale parameter σ of tolerances for the subjects. Then, given a dose x, the probability, P, of observing a response in a particular subject is the probability that the subject's tolerance is less than the dose or

$$P = F\left(\frac{x-\mu}{\sigma}\right)$$

Thus, in this case, the intercept parameter, $\mathbf{b}_0$, and the regression parameter, $\mathbf{b}_1$, are related to μ and σ by

$$\mathbf{b}_1 = \frac{1}{\sigma}$$

$$\mathbf{b}_0 = -\frac{\mu}{\sigma}$$

Note: The parameter σ is not equal to the standard deviation of the population of tolerances for the logistic and extreme value distributions.

Inverse Confidence Limits

In bioassay problems, estimates of the values of the independent variables that yield a desired response are often needed. For instance, the value yielding a 50% response rate (called the ED50 or LD50) is often used. The INVERSECL option requests that confidence limits be computed for the value of the independent variable that yields a specified response. These limits are computed only for the first continuous variable effect in the model. The other variables are set either at their mean values if they are continuous or at the reference (last) level if they are discrete variables. For a discussion of inverse confidence limits, refer to Hubert, Bohidar, and Peace (1988).

For the PROBIT procedure, the response variable is a probability. An estimate of the first continuous variable value needed to achieve a response of p is given by

$$\hat{x}_1 = \frac{1}{b_1}\left(F^{-1}(p) - \mathbf{x}^{*\prime}\mathbf{b}^*\right)$$

where F is the cumulative distribution function used to model the probability, $\mathbf{x}^*$ is the vector of independent variables excluding the first one, which can be specified by the XDATA= option described in the section "XDATA= *SAS-data-set*" on page 3763, $\mathbf{b}^*$ is the vector of parameter estimates excluding the first one, and b_1 is the estimated regression coefficient for the independent variable of interest. Note that, for both binary and ordinal models, the INVERSECL option provides estimates of the value of x_1 yielding $\Pr(\text{first response level}) = p$, for various values of p.

This estimator is given as a ratio of random variables, for example, $r = a/b$. Confidence limits for this ratio can be computed using Fieller's theorem. A brief description of this theorem follows. Refer to Finney (1971) for a more complete description of Fieller's theorem.

If the random variables a and b are thought to be distributed as jointly normal, then for any fixed value r the following probability statement holds if z is an $\alpha/2$ quantile from the standard normal distribution and $\mathbf{V}$ is the variance-covariance matrix of a and b.

$$\Pr\left((a - rb)^2 > z^2(V_{aa} - 2rV_{ab} + r^2V_{bb})\right) = \alpha$$

Usually the inequality can be solved for r to yield a confidence interval. The PROBIT procedure uses a value of 1.96 for z, corresponding to an α value of 0.05, unless the goodness-of-fit p-value is less than the specified value of the HPROB= option. When this happens, the covariance matrix is scaled by the heterogeneity factor, and a t distribution quantile is used for z.

It is possible for the roots of the equation for r to be imaginary or for the confidence interval to be all points outside of an interval. In these cases, the limits are set to missing by the PROBIT procedure.

Although the normal and logistic distribution give comparable fitted values of p if the empirically observed proportions are not too extreme, they can give appreciably different values when extrapolated into the tails. Correspondingly, the estimates of the confidence limits and dose values can be different for the two distributions even when they agree quite well in the body of the data. Extrapolation outside of the range of the actual data are often sensitive to model assumptions, and caution is advised if extrapolation is necessary.

OUTEST= *SAS-data-set*

The OUTEST= data set contains parameter estimates and the log likelihood for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different modeling using the PROBIT procedure. If you specify the COVOUT option, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates.

The OUTEST= data set contains each variable used as a dependent or independent variable in any MODEL statement. One observation consists of parameter values for the model with the dependent variable having the value -1. If you specify the COVOUT option, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL a character variable containing the label of the MODEL statement, if present, or blank otherwise

NAME a character variable containing the name of the dependent variable for the parameter estimates observations or the name of the row for the covariance matrix estimates

TYPE a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates

DIST a character variable containing the name of the distribution modeled

LNLIKE a numeric variable containing the last computed value of the log likelihood

_C_ a numeric variable containing the estimated threshold parameter

INTERCEPT a numeric variable containing the intercept parameter estimates and covariances

Any BY variables specified are also added to the OUTEST= data set.

XDATA= *SAS-data-set*

The XDATA= data set is used for specifiying values for the effects in the MODEL statement when predicted values and/or fiducial limits for a single continuous variable (dose variable) are required. It is also used for plots specified by the CDFPLOT, IPPPLOT, LPREDPLOT, and PREDPPLOT statement.

The XDATA= data names a SAS data set that contains user input values for all the independent variables in the MODEL statement and the variables in the CLASS statement. The XDATA= data set has the same structure as the DATA= data set but is not required to have all the variables or observations that appear in the DATA= data set.

The XDATA= data set must contain all the independent variables in the MODEL statement and variables in the CLASS statement. Even though variables in the CLASS statement may not be used in the MODEL statement, valid values are required for these variables in the XDATA= data set. Missing values are not allowed. For independent variables in the MODEL statement, although the dose variable's value is not used in the computing of predicted values and/or fiducial limits for the dose variable, missing values are not allowed in the XDATA= data set for any of the independent variables. Missing values are allowed for the dependent variables and other variables if they are included in the XDATA= data set and not listed in the CLASS statement.

If BY processing is used, the XDATA= data set should also include the BY variables, and there must be at least one valid observation for each BY group. If there is more than one valid observation in one BY group, the last one read is used for that BY group.

If there is no XDATA= data set in the PROC PROBIT statement, by default, the PROBIT procedure will use overall mean for effects containing continuous variable (or variables) and the highest level of a single classification variable as reference level. The rules are summarized as follows:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

Displayed Output

If you request the iteration history (ITPRINT), PROC PROBIT displays

- the current value of the log likelihood
- the ridging parameter for the modified Newton-Raphson optimization process
- the current estimate of the parameters
- the current estimate of the parameter C for a natural (threshold) model
- the values of the gradient and the Hessian on the last iteration

If you include CLASS variables, PROC PROBIT displays

- the numbers of levels for each CLASS variable
- the (ordered) values of the levels
- the number of observations used

After the model is fit, PROC PROBIT displays

- the name of the input data set
- the name of the dependent variables
- the number of observations used
- the number of events and the number of trials
- the final value of the log-likelihood function
- the parameter estimates
- the standard error estimates of the parameter estimates
- approximate chi-square test statistics for the test

If you specify the COVB or CORRB options, PROC PROBIT displays

- the estimated covariance matrix for the parameter estimates
- the estimated correlation matrix for the parameter estimates

If you specify the LACKFIT option, PROC PROBIT displays

- a count of the number of levels of the response and the number of distinct sets of independent variables
- a goodness-of-fit test based on the Pearson chi-square
- a goodness-of-fit test based on the likelihood-ratio chi-square

If you specify only one independent variable, the normal distribution is used to model the probabilities, and the response is binary, PROC PROBIT displays

- the mean MU of the stimulus tolerance
- the scale parameter SIGMA of the stimulus tolerance
- the covariance matrix for MU, SIGMA, and the natural response parameter C

If you specify the INVERSECL options, PROC PROBIT also displays

- the estimated dose along with the 95% fiducial limits for probability levels 0.01 to 0.10, 0.15 to 0.85 by 0.05, and 0.90 to 0.99

ODS Table Names

PROC PROBIT assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 60.33. ODS Tables Produced in PROC PROBIT

ODS Table Name	Description	Statement	Option
ClassLevels	Class variable levels	CLASS	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
CovTolerance	Covariance matrix for location and scale	MODEL	default*
GoodnessOfFit	Goodness of fit tests	MODEL	LACKFIT
IterHistory	Iteration history	MODEL	ITPRINT
LagrangeStatistics	Lagrange statistics	MODEL	NOINT
LastGrad	Last evaluation of the gradient	MODEL	ITPRINT
LastHess	Last evaluation of the Hessian	MODEL	ITPRINT
LogProbitAnalysis	Probit analysis for log dose	MODEL	INVERSECL
ModelInfo	Model information	MODEL	default
MuSigma	Location and scale	MODEL	default*
NObs	Observations Summary	PROC	default
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ProbitAnalysis	Probit analysis for linear dose	MODEL	INVERSECL
ResponseLevels	Response-covariate profile	MODEL	LACKFIT
ResponseProfiles	Counts for ordinal data	MODEL	default
Type3Analysis	Type 3 tests	MODEL	default*

* Depends on data.

Examples

Example 60.1. Dosage Levels

In this example, Dose is a variable representing the level of a stimulus, N represents the number of subjects tested at each level of the stimulus, and Response is the number of subjects responding to that level of the stimulus. Both probit and logit response models are fit to the data. The LOG10 option in the PROC PROBIT statement requests that the log base 10 of Dose is used as the independent variable. Specifically, for a given level of Dose, the probability p of a positive response is modeled as

$$p = \Pr(\textsf{Response}) = F\left(b_0 + b_1 \times \log_{10}(\textsf{Dose})\right)$$

The probabilities are estimated first using the normal distribution function (the default) and then using the logistic distribution function. Note that, in this model specification, the natural rate is assumed to be zero.

The LACKFIT option specifies lack-of-fit tests and the INVERSECL option specifies inverse confidence limits.

In the DATA step that reads the data, a number of observations are generated that have a missing value for the response. Although the PROBIT procedure does not use the observations with the missing values to fit the model, it does give predicted values for all nonmissing sets of independent variables. These data points fill in the plot of fitted and observed values in the logistic model displayed in Output 60.1.2. The plot, requested with the PREDPPLOT statement, displays the estimated logistic cumulative distribution function and the observed response rates. The VAR= DOSE option specifies the horizontal axis variable in the plot.

The following statements produce Output 60.1.1:

```
data a;
   infile cards eof=eof;
   input Dose N Response;
   Observed= Response/N;
   output;
   return;
eof: do Dose=0.5 to 7.5 by 0.25;
        output;
     end;
   datalines;
1 10 1
2 12 2
3 10 4
4 10 5
5 12 8
6 10 8
7 10 10
;
```

```
proc probit log10;
   model Response/N=Dose / lackfit inversecl itprint;
   output out=B p=Prob std=std xbeta=xbeta;
   title 'Output from Probit Procedure';
run;

symbol v=dot c=white;
proc probit log10;
   model Response/N=Dose /  d=logistic inversecl;
   predpplot var = dose  cfit = blue cframe=ligr inborder;
   output out=B p=Prob std=std xbeta=xbeta;
   title 'Output from Probit Procedure';
run;
```

Output 60.1.1. Dosage Levels: PROC PROBIT

```
                         Output from Probit Procedure

                               Probit Procedure

                    Iteration History for Parameter Estimates

    Iter     Ridge     Loglikelihood        Intercept     Log10(Dose)

       0         0        -51.292891                0               0
       1         0        -37.881166     -1.355817008     2.635206083
       2         0        -37.286169     -1.764939171    3.3408954936
       3         0        -37.280389     -1.812147863    3.4172391614
       4         0        -37.280388     -1.812704962     3.418117919
       5         0        -37.280388     -1.812704962     3.418117919
```

```
                        Output from Probit Procedure

                              Probit Procedure

                             Model Information

                     Data Set                        WORK.B
                     Events Variable               Response
                     Trials Variable                      N
                     Number of Observations               7
                     Number of Events                    38
                     Number of Trials                    74
                     Missing Values                      29
                     Name of Distribution            Normal
                     Log Likelihood            -37.28038802


                Last Evaluation of the Negative of the Gradient

                            Intercept   Log10(Dose)

                          3.434907E-7   -2.09809E-8


                 Last Evaluation of the Negative of the Hessian

                                  Intercept   Log10(Dose)

                    Intercept    36.005280383  20.152675982
                    Log10(Dose)  20.152675982  13.078826305


                           Goodness-of-Fit Tests

         Statistic                        Value       DF    Pr > ChiSq

         Pearson Chi-Square              3.6497        5        0.6009
         L.R.    Chi-Square              4.6381        5        0.4616


                        Response-Covariate Profile

                        Response Levels              2
                        Number of Covariate Values    7
```

The p-values in the Goodness-of-Fit table of 0.6009 for the Pearson chi-square and 0.4616 for the likelihood ratio chi-square indicate an adequate fit for the model fit with the normal distribution.

```
                     Output from Probit Procedure

                           Probit Procedure

                   Analysis of Parameter Estimates

                        Standard   95% Confidence     Chi-
Parameter   DF Estimate    Error       Limits       Square Pr > ChiSq

Intercept    1  -1.8127   0.4493  -2.6934  -0.9320   16.27     <.0001
Log10(Dose)  1   3.4181   0.7455   1.9569   4.8794   21.02     <.0001


           Probit Model in Terms of Tolerance Distribution

                               MU         SIGMA

                       0.53032254    0.29255866


                     Estimated Covariance Matrix
                      for Tolerance Parameters

                                   MU             SIGMA

              MU             0.002418         -0.000409
              SIGMA         -0.000409          0.004072
```

Tolerance distribution parameter estimates for the normal distribution indicate a mean tolerance for the population of 0.5303.

```
                          Output from Probit Procedure

                                Probit Procedure

                         Probit Analysis on Log10(Dose)

          Probability          Log10(Dose)       95% Fiducial Limits

                 0.01             -0.15027      -0.69518       0.07710
                 0.02             -0.07052      -0.55766       0.13475
                 0.03             -0.01992      -0.47064       0.17156
                 0.04              0.01814      -0.40534       0.19941
                 0.05              0.04911      -0.35233       0.22218
                 0.06              0.07546      -0.30731       0.24165
                 0.07              0.09857      -0.26793       0.25881
                 0.08              0.11926      -0.23273       0.27425
                 0.09              0.13807      -0.20080       0.28837
                 0.10              0.15539      -0.17147       0.30142
                 0.15              0.22710      -0.05086       0.35631
                 0.20              0.28410       0.04369       0.40124
                 0.25              0.33299       0.12343       0.44116
                 0.30              0.37690       0.19348       0.47857
                 0.35              0.41759       0.25658       0.51504
                 0.40              0.45620       0.31429       0.55182
                 0.45              0.49356       0.36754       0.58999
                 0.50              0.53032       0.41693       0.63057
                 0.55              0.56709       0.46296       0.67451
                 0.60              0.60444       0.50618       0.72271
                 0.65              0.64305       0.54734       0.77603
                 0.70              0.68374       0.58745       0.83550
                 0.75              0.72765       0.62776       0.90265
                 0.80              0.77655       0.66999       0.98008
                 0.85              0.83354       0.71675       1.07279
                 0.90              0.90525       0.77313       1.19191
                 0.91              0.92257       0.78646       1.22098
                 0.92              0.94139       0.80083       1.25265
                 0.93              0.96208       0.81653       1.28759
                 0.94              0.98519       0.83394       1.32672
                 0.95              1.01154       0.85367       1.37149
                 0.96              1.04250       0.87669       1.42424
                 0.97              1.08056       0.90480       1.48928
                 0.98              1.13116       0.94189       1.57602
                 0.99              1.21092       0.99987       1.71321
```

The LD50 (ED50 for log dose) is 0.5303, the dose corresponding to a probability of 0.5. This is the same as the mean tolerance for the normal distribution.

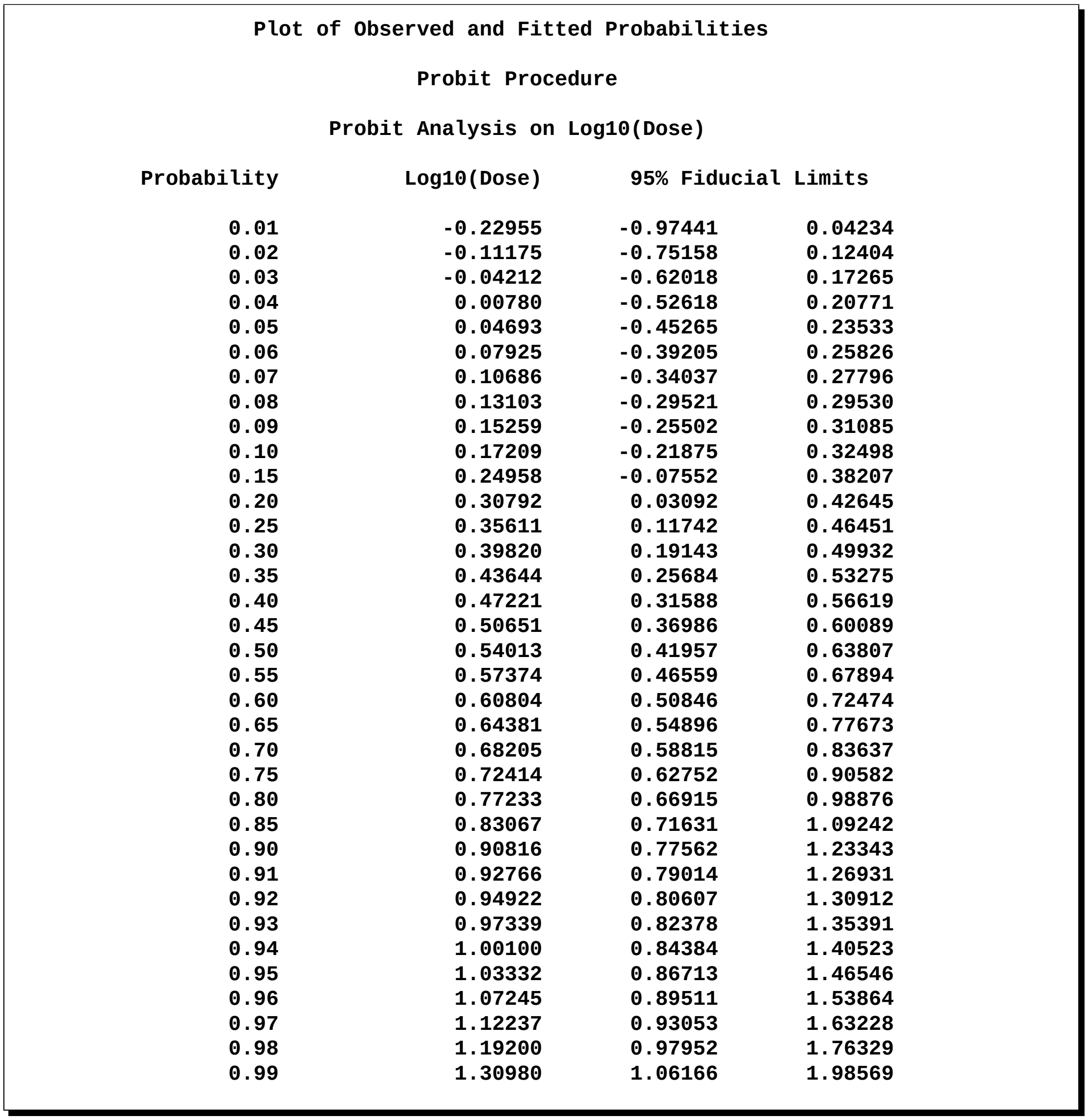
```
                     Output from Probit Procedure

                           Probit Procedure

                       Probit Analysis on Dose

       Probability          Dose       95% Fiducial Limits

              0.01       0.70750       0.20175       1.19427
              0.02       0.85012       0.27691       1.36380
              0.03       0.95517       0.33834       1.48444
              0.04       1.04266       0.39324       1.58274
              0.05       1.11971       0.44429       1.66793
              0.06       1.18976       0.49282       1.74443
              0.07       1.25478       0.53960       1.81473
              0.08       1.31600       0.58515       1.88042
              0.09       1.37427       0.62980       1.94252
              0.10       1.43019       0.67380       2.00181
              0.15       1.68696       0.88950       2.27147
              0.20       1.92353       1.10584       2.51906
              0.25       2.15276       1.32870       2.76161
              0.30       2.38180       1.56128       3.01000
              0.35       2.61573       1.80543       3.27374
              0.40       2.85893       2.06200       3.56306
              0.45       3.11573       2.33098       3.89038
              0.50       3.39096       2.61175       4.27138
              0.55       3.69051       2.90374       4.72619
              0.60       4.02199       3.20759       5.28090
              0.65       4.39594       3.52651       5.97077
              0.70       4.82770       3.86765       6.84706
              0.75       5.34134       4.24385       7.99189
              0.80       5.97787       4.67724       9.55169
              0.85       6.81617       5.20900      11.82480
              0.90       8.03992       5.93105      15.55653
              0.91       8.36704       6.11584      16.63320
              0.92       8.73752       6.32165      17.89163
              0.93       9.16385       6.55431      19.39034
              0.94       9.66463       6.82245      21.21881
              0.95      10.26925       7.13949      23.52275
              0.96      11.02811       7.52816      26.56066
              0.97      12.03830       8.03149      30.85201
              0.98      13.52585       8.74763      37.67206
              0.99      16.25233       9.99709      51.66627
```

The ED50 for dose is 3.39 with a 95% confidence interval of (2.61, 4.27).

```
                    Plot of Observed and Fitted Probabilities

                                 Probit Procedure

                                Model Information

                      Data Set                          WORK.A
                      Events Variable                 Response
                      Trials Variable                        N
                      Number of Observations                 7
                      Number of Events                      38
                      Number of Trials                      74
                      Missing Values                        29
                      Name of Distribution            Logistic
                      Log Likelihood              -37.11065336


 Algorithm converged.
```

```
                    Plot of Observed and Fitted Probabilities

                                 Probit Procedure

                          Analysis of Parameter Estimates

                            Standard   95% Confidence     Chi-
    Parameter   DF Estimate    Error       Limits       Square Pr > ChiSq

    Intercept    1  -3.2246   0.8861  -4.9613  -1.4880   13.24     0.0003
    Log10(Dose)  1   5.9702   1.4492   3.1299   8.8105   16.97     <.0001
```

The regression parameter estimates for the logistic model of -3.22 and 5.97 are approximately $\pi/\sqrt{3}$ times as large as those for the normal model.



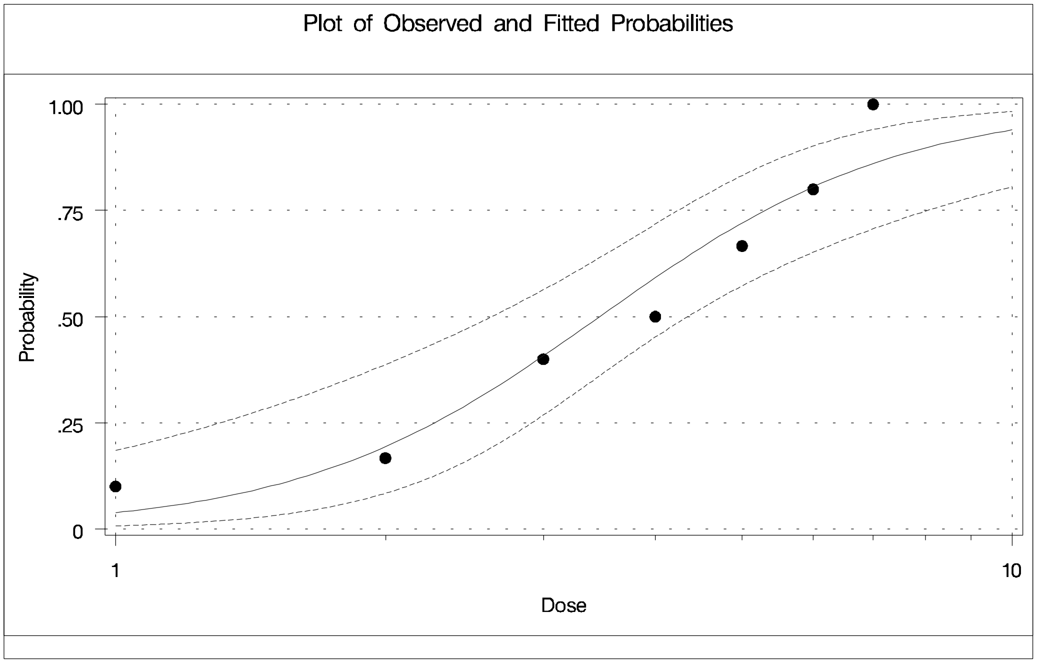
```
                    Plot of Observed and Fitted Probabilities

                                Probit Procedure

                          Probit Analysis on Log10(Dose)

           Probability          Log10(Dose)       95% Fiducial Limits

                  0.01             -0.22955      -0.97441       0.04234
                  0.02             -0.11175      -0.75158       0.12404
                  0.03             -0.04212      -0.62018       0.17265
                  0.04              0.00780      -0.52618       0.20771
                  0.05              0.04693      -0.45265       0.23533
                  0.06              0.07925      -0.39205       0.25826
                  0.07              0.10686      -0.34037       0.27796
                  0.08              0.13103      -0.29521       0.29530
                  0.09              0.15259      -0.25502       0.31085
                  0.10              0.17209      -0.21875       0.32498
                  0.15              0.24958      -0.07552       0.38207
                  0.20              0.30792       0.03092       0.42645
                  0.25              0.35611       0.11742       0.46451
                  0.30              0.39820       0.19143       0.49932
                  0.35              0.43644       0.25684       0.53275
                  0.40              0.47221       0.31588       0.56619
                  0.45              0.50651       0.36986       0.60089
                  0.50              0.54013       0.41957       0.63807
                  0.55              0.57374       0.46559       0.67894
                  0.60              0.60804       0.50846       0.72474
                  0.65              0.64381       0.54896       0.77673
                  0.70              0.68205       0.58815       0.83637
                  0.75              0.72414       0.62752       0.90582
                  0.80              0.77233       0.66915       0.98876
                  0.85              0.83067       0.71631       1.09242
                  0.90              0.90816       0.77562       1.23343
                  0.91              0.92766       0.79014       1.26931
                  0.92              0.94922       0.80607       1.30912
                  0.93              0.97339       0.82378       1.35391
                  0.94              1.00100       0.84384       1.40523
                  0.95              1.03332       0.86713       1.46546
                  0.96              1.07245       0.89511       1.53864
                  0.97              1.12237       0.93053       1.63228
                  0.98              1.19200       0.97952       1.76329
                  0.99              1.30980       1.06166       1.98569
```

```
                    Plot of Observed and Fitted Probabilities

                               Probit Procedure

                            Probit Analysis on Dose

              Probability           Dose       95% Fiducial Limits

                     0.01        0.58945       0.10607       1.10241
                     0.02        0.77312       0.17718       1.33058
                     0.03        0.90757       0.23978       1.48817
                     0.04        1.01813       0.29773       1.61327
                     0.05        1.11413       0.35266       1.71922
                     0.06        1.20018       0.40546       1.81244
                     0.07        1.27896       0.45670       1.89654
                     0.08        1.35218       0.50675       1.97379
                     0.09        1.42100       0.55588       2.04572
                     0.10        1.48625       0.60430       2.11339
                     0.15        1.77656       0.84038       2.41030
                     0.20        2.03199       1.07379       2.66961
                     0.25        2.27043       1.31046       2.91416
                     0.30        2.50152       1.55393       3.15736
                     0.35        2.73172       1.80652       3.40996
                     0.40        2.96627       2.06957       3.68292
                     0.45        3.21006       2.34345       3.98927
                     0.50        3.46837       2.62768       4.34578
                     0.55        3.74746       2.92138       4.77466
                     0.60        4.05546       3.22451       5.30573
                     0.65        4.40366       3.53961       5.98041
                     0.70        4.80891       3.87391       6.86079
                     0.75        5.29836       4.24155       8.05044
                     0.80        5.92009       4.66820       9.74455
                     0.85        6.77126       5.20365      12.37149
                     0.90        8.09391       5.96508      17.11715
                     0.91        8.46559       6.16800      18.59129
                     0.92        8.89644       6.39837      20.37592
                     0.93        9.40575       6.66469      22.58957
                     0.94       10.02317       6.97977      25.42292
                     0.95       10.79732       7.36428      29.20549
                     0.96       11.81534       7.85438      34.56521
                     0.97       13.25466       8.52173      42.88232
                     0.98       15.55972       9.53941      57.98207
                     0.99       20.40815      11.52549      96.75820
```

Both the ED50 and the LD50 are similar to those for the normal model.

The statement PREDPPLOT creates the plot of observed and fitted probabilities in Output 60.1.2. The dashed line represent pointwise confidence bands for the probabilities.

Output 60.1.2. Plot of Observed and Fitted Probabilities



Example 60.2. Multilevel Response

In this example, two preparations, a standard preparation and a test preparation, are each given at several dose levels to groups of insects. The symptoms are recorded for each insect within each group, and two multilevel probit models are fit. Because the natural sort order of the three levels is not the same as the response order, the ORDER=DATA option is specified in the PROC PROBIT statement to get the desired order.

The following statements produce Output 60.2.1:

```
data multi;
   input Prep $ Dose Symptoms $ N;
   LDose=log10(Dose);
   if Prep='test' then PrepDose=LDose;
   else PrepDose=0;
   datalines;
stand     10      None       33
stand     10      Mild        7
stand     10      Severe     10
stand     20      None       17
stand     20      Mild       13
stand     20      Severe     17
stand     30      None       14
stand     30      Mild        3
stand     30      Severe     28
stand     40      None        9
stand     40      Mild        8
stand     40      Severe     32
```

```
test      10      None       44
test      10      Mild        6
test      10      Severe      0
test      20      None       32
test      20      Mild       10
test      20      Severe     12
test      30      None       23
test      30      Mild        7
test      30      Severe     21
test      40      None       16
test      40      Mild        6
test      40      Severe     19
;

proc probit order=data;
   class Prep Symptoms;
   nonpara: model Symptoms=Prep LDose PrepDose / lackfit;
   weight N;
   title 'Probit Models for Symptom Severity';
run;

proc probit order=data;
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   weight N;
   title 'Probit Models for Symptom Severity';
run;
```

The first model allows for nonparallelism between the dose response curves for the two preparations by inclusion of an interaction between Prep and LDose. The interaction term is labeled PrepDose in the “Analysis of Parameter Estimates” table. The results of this first model indicate that the parameter for the interaction term is not significant, having a Wald chi-square of 0.73. Also, since the first model is a generalization of the second, a likelihood ratio test statistic for this same parameter can be obtained by multiplying the difference in log likelihoods between the two models by 2. The value obtained, $2 \times (-345.94 - (-346.31))$, is 0.73. This is in close agreement with the Wald chi-square from the first model. The lack-of-fit test statistics for the two models do not indicate a problem with either fit.

Output 60.2.1. Multilevel Response: PROC PROBIT

```
                  Probit Models for Symptom Severity

                          Probit Procedure

                      Class Level Information

              Name           Levels    Values

              Prep                2    stand test
              Symptoms            3    None Mild Severe
```

```
                        Probit Models for Symptom Severity

                                 Probit Procedure

                                Model Information

                      Data Set                       WORK.MULTI
                      Dependent Variable               Symptoms
                      Weight Variable                         N
                      Number of Observations                 23
                      Missing Values                          1
                      Name of Distribution               Normal
                      Log Likelihood               -345.9401767
```

```
                        Probit Models for Symptom Severity

                                 Probit Procedure

                         Analysis of Parameter Estimates

                                    Standard   95% Confidence     Chi-
   Parameter         DF Estimate     Error        Limits        Square Pr > ChiSq

   Intercept          1   3.8080    0.6252   2.5827   5.0333    37.10     <.0001
   Intercept2         1   0.4684    0.0559   0.3589   0.5780    70.19     <.0001
   Prep       stand   1  -1.2573    0.8190  -2.8624   0.3479     2.36     0.1247
   Prep       test    0   0.0000    0.0000   0.0000   0.0000      .         .
   LDose              1  -2.1512    0.3909  -2.9173  -1.3851    30.29     <.0001
   PrepDose           1  -0.5072    0.5945  -1.6724   0.6580     0.73     0.3935
```

```
                        Probit Models for Symptom Severity

                                 Probit Procedure

                             Class Level Information

                    Name           Levels    Values

                    Prep                2    stand test
                    Symptoms            3    None Mild Severe
```

```
                       Probit Models for Symptom Severity

                                Probit Procedure

                               Model Information

                      Data Set                       WORK.MULTI
                      Dependent Variable               Symptoms
                      Weight Variable                         N
                      Number of Observations                 23
                      Missing Values                          1
                      Name of Distribution               Normal
                      Log Likelihood                -346.306141
```

```
                       Probit Models for Symptom Severity

                                Probit Procedure

                        Analysis of Parameter Estimates

                                Standard   95% Confidence     Chi-
    Parameter         DF Estimate    Error       Limits       Square Pr > ChiSq

    Intercept          1   3.4148   0.4126   2.6061   4.2235   68.50     <.0001
    Intercept2         1   0.4678   0.0558   0.3584   0.5772   70.19     <.0001
    Prep       stand   1  -0.5675   0.1259  -0.8142  -0.3208   20.33     <.0001
    Prep       test    0   0.0000   0.0000   0.0000   0.0000     .         .
    LDose              1  -2.3721   0.2949  -2.9502  -1.7940   64.68     <.0001
```

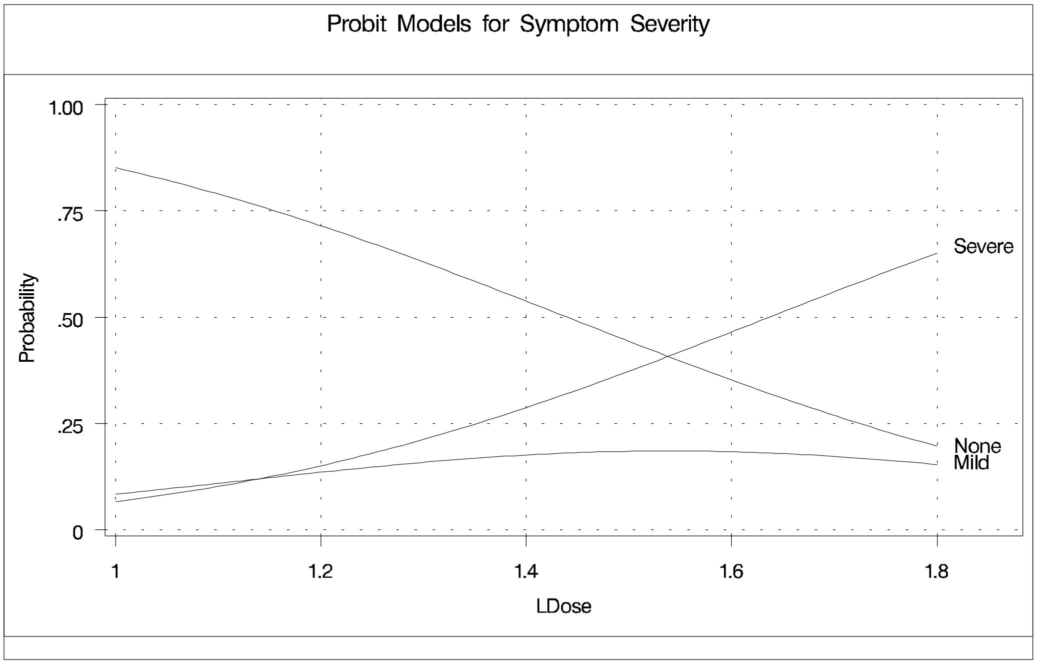
The negative coefficient associated with LDose indicates that the probability of having no symptoms (Symptoms='None') or no or mild symptoms (Symptoms='None' or Symptoms='Mild') decreases as LDose increases; that is, the probability of a severe symptom increases with LDose. This association is apparent for both treatment groups.

The negative coefficient associated with the standard treatment group (Prep = stand) indicates that the standard treatment is associated with more severe symptoms across all Ldose values.

The following statements use the PREDPPLOT statement to create the plot shown in Output 60.2.2 of the probabilities of the response taking on individual levels as a function of LDose. Since there are two covariates, LDose and Prep, the value of the CLASS variable Prep is fixed at the highest level, test. Although not shown here, the CDFPLOT statement creates similar plots of the cumulative response probabilities, instead of individual response level probabilities.

```
proc probit data=multi order=data;
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   predpplot var=ldose  level=("None" "Mild" "Severe")
             cfit=blue cframe=ligr inborder noconf ;
   weight N;
   title 'Probit Models for Symptom Severity';
run;
```

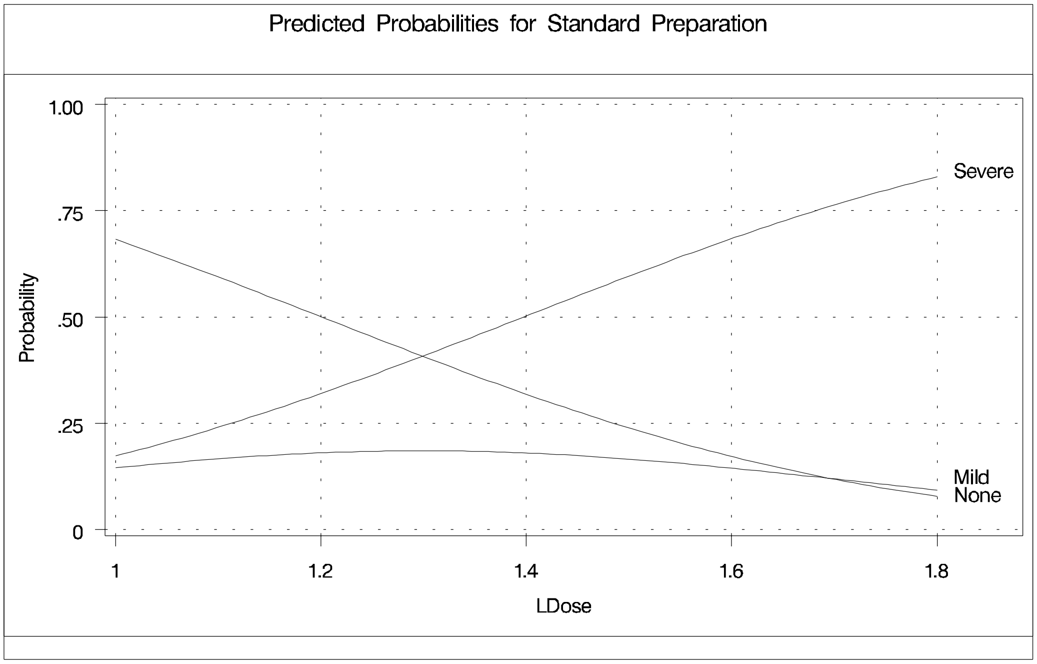
Output 60.2.2. Plot of Predicted Probilities for the Test Preparation Group



The following statements use the XDATA= data set to create a plot of the predicted probabilities with Prep set to the stand level. The resulting plot is shown in Output 60.2.3.

```
data xrow;
   input Prep $ Dose Symptoms $ N;
   LDose=log10(Dose);
   datalines;
stand     40      Severe     32
run;

proc probit data=multi order=data xdata=xrow;
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   predpplot var=ldose  level=("None" "Mild" "Severe")
             cfit=blue cframe=ligr inborder noconf ;
   weight N;
   title 'Predicted Probabilities for Standard Preparation';
run;
```

Output 60.2.3. Plot of Predicted Probabilities for the Standard Preparation Group



Example 60.3. Logistic Regression

In this example, a series of people are questioned as to whether or not they would subscribe to a new newspaper. For each person, the variables sex (Female, Male), age, and subs (1=yes,0=no) are recorded. The PROBIT procedure is used to fit a logistic regression model to the probability of a positive response (subscribing) as a function of the variables sex and age. Specifically, the probability of subscribing is modeled as

$$p = \Pr(\text{subs} = 1) = F\left(b_0 + b_1 \times \text{sex} + b_2 \times \text{age}\right)$$

where F is the cumulative logistic distribution function.

By default, the PROBIT procedure models the probability of the lower response level for binary data. One way to model $\Pr(\text{subs} = 1)$ is to format the response variable so that the formatted value corresponding to subs=1 is the lower level. The following statements format the values of subs as 1 = 'accept' and 0 = 'reject', so that PROBIT models $\Pr(\text{accept}) = \Pr(\text{subs} = 1)$.

The following statements produce Output 60.3.1:

```
data news;
   input sex $ age subs;
   datalines;
Female     35    0
Male       44    0
Male       45    1
```

```
Female     47    1
Female     51    0
Female     47    0
Male       54    1
Male       47    1
Female     35    0
Female     34    0
Female     48    0
Female     56    1
Male       46    1
Female     59    1
Female     46    1
Male       59    1
Male       38    1
Female     39    0
Male       49    1
Male       42    1
Male       50    1
Female     45    0
Female     47    0
Female     30    1
Female     39    0
Female     51    0
Female     45    0
Female     43    1
Male       39    1
Male       31    0
Female     39    0
Male       34    0
Female     52    1
Female     46    0
Male       58    1
Female     50    1
Female     32    0
Female     52    1
Female     35    0
Female     51    0
;

proc format;
   value subscrib 1 = ’accept’ 0 = ’reject’;
run;

proc probit;
   class subs sex;
   model subs=sex age / d=logistic itprint;
   format subs subscrib.;
   title ’Logistic Regression of Subscription Status’;
run;
```

Output 60.3.1. Logistic Regression: PROC PROBIT

```
                   Logistic Regression of Subscription Status

                                Probit Procedure

                            Class Level Information

                       Name      Levels    Values

                       subs           2    accept reject
                       sex            2    Female Male

PROC PROBIT is modeling the probabilities of levels of subs having LOWER
Ordered Values in the response profile table.
```

```
                   Logistic Regression of Subscription Status

                                Probit Procedure

                    Iteration History for Parameter Estimates

 Iter    Ridge    Loglikelihood       Intercept       sexFemale             age

    0        0       -27.725887               0               0               0
    1        0       -20.142659    -3.634567629    -1.648455751    0.1051634384
    2        0        -19.52245    -5.254865196    -2.234724956    0.1506493473
    3        0       -19.490439    -5.728485385    -2.409827238    0.1639621828
    4        0       -19.490303     -5.76187293    -2.422349862    0.1649007124
    5        0       -19.490303      -5.7620267    -2.422407743    0.1649050312
    6        0       -19.490303      -5.7620267    -2.422407743    0.1649050312


                               Model Information

                     Data Set                      WORK.NEWS
                     Dependent Variable                 subs
                     Number of Observations               40
                     Name of Distribution           Logistic
                     Log Likelihood             -19.49030281

PROC PROBIT is modeling the probabilities of levels of subs having LOWER
Ordered Values in the response profile table.
```

```
                   Logistic Regression of Subscription Status

                                Probit Procedure

PROC PROBIT is modeling the probabilities of levels of subs having LOWER
Ordered Values in the response profile table.


                Last Evaluation of the Negative of the Gradient

                       Intercept     sexFemale           age

                    -5.95457E-12  8.768328E-10  -1.636696E-8


                Last Evaluation of the Negative of the Hessian

                            Intercept     sexFemale           age

              Intercept  6.4597397447  4.6042218284  292.04051848
              sexFemale  4.6042218284  4.6042218284  216.20829515
              age        292.04051848  216.20829515  13487.329973


  Algorithm converged.
```

```
                   Logistic Regression of Subscription Status

                                Probit Procedure

PROC PROBIT is modeling the probabilities of levels of subs having LOWER
Ordered Values in the response profile table.


                        Analysis of Parameter Estimates

                                Standard   95% Confidence     Chi-
  Parameter         DF Estimate    Error       Limits       Square Pr > ChiSq

  Intercept          1  -5.7620   2.7635 -11.1783  -0.3458    4.35     0.0371
  sex        Female  1  -2.4224   0.9559  -4.2959  -0.5489    6.42     0.0113
  sex        Male    0   0.0000   0.0000   0.0000   0.0000     .        .
  age                1   0.1649   0.0652   0.0371   0.2927    6.40     0.0114
```

From Output 60.3.1, there appears to be an effect due to both the variables sex and age. The positive coefficient for age indicates that older people are more likely to subscribe than younger people. The negative coefficient for sex indicates that females are less likely to subscribe than males.

Example 60.4. An Epidemiology Study

The data, which is from an epidemiology study, consists of five variables: the number, r, of individuals surviving after an epidemic, out of n treated, for combinations of medicine dosage (dose), treatment (treat = A, B), and sex (sex = 0(Female), 1(Male)).

To see if the two treatments have different effects on male and female individual survival rate, the interaction term between the two variables treat and sex is included in the model.

The following invocation of PROC PROBIT fits the binary probit model to the grouped data:

```
data epidemic;
   input treat$ dose n r sex;
   label dose = Dose;
   datalines;
A  2.17 142 142  0
A   .57 132  47  1
A  1.68 128 105  1
A  1.08 126 100  0
A  1.79 125 118  0
B  1.66 117 115  1
B  1.49 127 114  0
B  1.17  51  44  1
B  2.00 127 126  0
B   .80 129 100  1
;

data xval;
   input treat $ dose sex ;
   datalines;
B  2.  1
;

title 'Epidemiology Study';
proc probit optc lackfit covout data = epidemic
                        outest = out1 xdata = xval;
   class treat sex;
   model r/n = dose treat sex sex*treat/corrb covb inversecl;
   output out = out2 p =p;

   predpplot
            var = dose
            font = swiss
            vref(intersect) =  .6667
            vreflab = 'two thirds'
            vreflabpos = 2
            cfit=blue
            cframe=ligr
            ;
   inset    /
```

```
               cfill = white
               ctext = blue
               pos = se ;

      ippplot
               font = swiss
               href(intersect) =  .75
               hreflab = ’three quarters’
               vreflabpos = 2
               threshlabpos = 2
               cfit=blue
               cframe=ligr
               ;
      inset    /
               cfill = white
               ctext = blue;

      lpredplot
               font = swiss
               vref(intersect) =  1.
               vreflab = ’unit probit’
               vreflabpos = 2
               cfit=blue
               cframe=ligr
               ;
      inset    /
               cfill = white
               ctext = blue;
   run;
```

The results of this analysis are shown in the following tables and figures.

Beginning with SAS Release 8.2, the PROBIT procedure does not support multiple MODEL statements. Only the last one is used if there is more than one MODEL statement in one invocation of the PROBIT procedure.

Output 60.4.1. Class Level Information

```
                               Epidemiology Study

                                Probit Procedure

                            Class Level Information

                          Name       Levels    Values

                          treat           2    A B
                          sex             2    0 1
```

Output 60.4.1 displays the table of level information for *all* classification variables in the CLASS statement.

Output 60.4.2. Parameter Information

```
                              Epidemiology Study

                               Probit Procedure

                            Parameter Information

                 Parameter      Effect       treat    sex

                 Intercept      Intercept
                 dose           dose
                 treatA         treat        A
                 treatB         treat        B
                 sex0           sex                   0
                 sex1           sex                   1
                 treatAsex0     treat*sex    A        0
                 treatAsex1     treat*sex    A        1
                 treatBsex0     treat*sex    B        0
                 treatBsex1     treat*sex    B        1
```

Output 60.4.2 displays the table of parameter information for the effects in the MODEL statement. The name of a parameter is generated from combining the variable names and level names in the effect. The maximum length of a parameter name is 32. The name of the effects are specified in the MODEL statement. The length of names of effects can be specified by the NAMELEN= option in the PROC PROBIT statement, with the default length 20.

Output 60.4.3. Model Information

```
                              Epidemiology Study

                               Probit Procedure

                              Model Information

                 Data Set                    WORK.EPIDEMIC
                 Events Variable                         r
                 Trials Variable                         n
                 Number of Observations                 10
                 Number of Events                     1011
                 Number of Trials                     1204
                 Name of Distribution               Normal
                 Log Likelihood               -387.2467391


  Algorithm converged.
```

Output 60.4.3 displays background information about the model fit. Included are the name of the input data set, the response variables used, and the number of observations, events, and trials. The table also includes the status of the convergence of the model fitting algorithm and the final value of log-likelihood function.

Output 60.4.4. Goodness-of-Fit Tests and Response-Covariate Profile

```
                                  Epidemiology Study

                                   Probit Procedure

                                 Goodness-of-Fit Tests

      Statistic                         Value       DF     Pr > ChiSq

      Pearson Chi-Square               4.9317        4         0.2944
      L.R.    Chi-Square               5.7079        4         0.2220


                               Response-Covariate Profile

                             Response Levels               2
                             Number of Covariate Values    10
```

Output 60.4.4 displays the table of goodness-of-fit tests requested with the LACKFIT option in the PROC PROBIT statement. Two goodness-of-fit statistics, the Pearson chi-square statistic and the likelihood ratio chi-square statistic, are computed. The grouping method for computing these statistics can be specified by the AGGREGATE= option. The details can be found in the AGGREGATE= option and an example can be found in the second part of this example. By default, the PROBIT procedure uses the covariates in the MODEL statement to do grouping. Observations with the same values of the covariates in the MODEL statement are grouped into cells and the two statistics are computed according to these cells. The total number of cells, and the number of levels for the response variable are reported next in the "Response-Covariate Profile."

In this example, neither the Pearson chi-square nor the log-likelihood ratio chi-square tests are significant at the 0.1 level, which is the default test level used by the PROBIT procedure. That means that the model, which includes the interaction of treat and sex, is suitable for this epidemiology data set. (Further investigation shows that models without the interaction of treat and sex are not acceptable by either test.)

Output 60.4.5. Type III Tests

```
                                  Epidemiology Study

                                   Probit Procedure

                              Type III Analysis of Effects

                                                Wald
               Effect       DF    Chi-Square    Pr > ChiSq

               dose          1       42.1691        <.0001
               treat         1       16.1421        <.0001
               sex           1        1.7710        0.1833
               treat*sex     1       13.9343        0.0002
```

Output 60.4.5 displays the Type III test results for all effects specified in the MODEL statement, which include the degrees of freedom for the effect, the Wald Chi-Square test statistic, and the p-value.

Output 60.4.6. Analysis of Parameter Estimates

```
                              Epidemiology Study

                               Probit Procedure

                        Analysis of Parameter Estimates

                                  Standard   95% Confidence     Chi-
 Parameter      DF Estimate     Error        Limits       Square Pr > ChiSq

 Intercept       1  -0.8871    0.3632  -1.5991  -0.1752     5.96     0.0146
 dose            1   1.6774    0.2583   1.1711   2.1837    42.17     <.0001
 treat      A    1  -1.2537    0.2616  -1.7664  -0.7410    22.97     <.0001
 treat      B    0   0.0000    0.0000   0.0000   0.0000      .         .
 sex        0    1  -0.4633    0.2289  -0.9119  -0.0147     4.10     0.0429
 sex        1    0   0.0000    0.0000   0.0000   0.0000      .         .
 treat*sex A 0   1   1.2899    0.3456   0.6126   1.9672    13.93     0.0002
 treat*sex A 1   0   0.0000    0.0000   0.0000   0.0000      .         .
 treat*sex B 0   0   0.0000    0.0000   0.0000   0.0000      .         .
 treat*sex B 1   0   0.0000    0.0000   0.0000   0.0000      .         .
 _C_             1   0.2735    0.0946   0.0881   0.4589
```

Output 60.4.6 displays the table of parameter estimates for the model. The PROBIT procedure displays information for all the parameters of an effect. Degenerate parameters are indicated by 0 degree of freedom. Confidence intervals are computed for all parameters with non-zero degrees of freedom, including the natural threshold C if the OPTC option is specified in the PROC PROBIT statement. The confidence level can be specified by the ALPHA= option in the MODEL statement. The default confidence level is 95%.

From this table, you can see the following results:

- dose has significant positive effect on the survival rate.
- Individuals under treatment A have a lower survival rate.
- Male individuals have a higher survival rate.
- Female individuals under treatment A have a higher survival rate.

Output 60.4.7. Estimated Covariance Matrix

```
                              Epidemiology Study

                               Probit Procedure

                          Estimated Covariance Matrix

               Intercept          dose        treatA          sex0    treatAsex0

Intercept       0.131944     -0.087353      0.053551      0.030285     -0.067056
dose           -0.087353      0.066723     -0.047506     -0.034081      0.058620
treatA          0.053551     -0.047506      0.068425      0.036063     -0.075323
sex0            0.030285     -0.034081      0.036063      0.052383     -0.063599
treatAsex0     -0.067056      0.058620     -0.075323     -0.063599      0.119408
_C_            -0.028073      0.018196     -0.017084     -0.008088      0.019134

                           Estimated Covariance Matrix

                                                   _C_

                            Intercept        -0.028073
                            dose              0.018196
                            treatA           -0.017084
                            sex0             -0.008088
                            treatAsex0        0.019134
                            _C_               0.008948
```

Output 60.4.8. Estimated Correlation Matrix

```
                              Epidemiology Study

                               Probit Procedure

                          Estimated Correlation Matrix

               Intercept          dose        treatA          sex0    treatAsex0

Intercept       1.000000     -0.930998      0.563595      0.364284     -0.534227
dose           -0.930998      1.000000     -0.703083     -0.576477      0.656744
treatA          0.563595     -0.703083      1.000000      0.602359     -0.833299
sex0            0.364284     -0.576477      0.602359      1.000000     -0.804154
treatAsex0     -0.534227      0.656744     -0.833299     -0.804154      1.000000
_C_            -0.817027      0.744699     -0.690420     -0.373565      0.585364

                          Estimated Correlation Matrix

                                                   _C_

                            Intercept        -0.817027
                            dose              0.744699
                            treatA           -0.690420
                            sex0             -0.373565
                            treatAsex0        0.585364
                            _C_               1.000000
```

Output 60.4.7 and Output 60.4.8 display tables of estimated covariance matrix and estimated correlation matrix for estimated parameters with a non-zero degree of freedom, respectively. They are computed by the inverse of the Hessian matrix of the estimated parameters.

Output 60.4.9. Probit Analysis on Dose

```
                              Epidemiology Study

                               Probit Procedure

                            Probit Analysis on dose

              Probability          dose       95% Fiducial Limits

                     0.01      -0.85801      -1.81301      -0.33743
                     0.02      -0.69549      -1.58167      -0.21116
                     0.03      -0.59238      -1.43501      -0.13093
                     0.04      -0.51482      -1.32476      -0.07050
                     0.05      -0.45172      -1.23513      -0.02130
                     0.06      -0.39802      -1.15888       0.02063
                     0.07      -0.35093      -1.09206       0.05742
                     0.08      -0.30877      -1.03226       0.09039
                     0.09      -0.27043      -0.97790       0.12040
                     0.10      -0.23513      -0.92788       0.14805
                     0.15      -0.08900      -0.72107       0.26278
                     0.20       0.02714      -0.55706       0.35434
                     0.25       0.12678      -0.41669       0.43322
                     0.30       0.21625      -0.29095       0.50437
                     0.35       0.29917      -0.17477       0.57064
                     0.40       0.37785      -0.06487       0.63387
                     0.45       0.45397       0.04104       0.69546
                     0.50       0.52888       0.14481       0.75654
                     0.55       0.60380       0.24800       0.81819
                     0.60       0.67992       0.35213       0.88157
                     0.65       0.75860       0.45879       0.94803
                     0.70       0.84151       0.56985       1.01942
                     0.75       0.93099       0.68770       1.09847
                     0.80       1.03063       0.81571       1.18970
                     0.85       1.14677       0.95926       1.30171
                     0.90       1.29290       1.12867       1.45386
                     0.91       1.32819       1.16747       1.49273
                     0.92       1.36654       1.20867       1.53590
                     0.93       1.40870       1.25284       1.58450
                     0.94       1.45579       1.30084       1.64012
                     0.95       1.50949       1.35397       1.70515
                     0.96       1.57258       1.41443       1.78353
                     0.97       1.65015       1.48626       1.88238
                     0.98       1.75326       1.57833       2.01720
                     0.99       1.91577       1.71776       2.23537
```

Output 60.4.9 displays the computed values and fiducial limits for the first single continuous variable dose in the MODEL statement, given the probability levels, without the effect of the natural threshold, and when the option INSERSECL in the MODEL statement is specified. If there is no single continuous variable in the MODEL specification but the INVERSECL option is specified, an error is reported. If the XDATA= option is used to input a data set for the independent variables in the MODEL statement, the PROBIT procedure uses these values for the independent variables other than the single continuous variable. Missing values are not permitted in the XDATA= data set for the independent variables, although the value for the single continuous variable is not used in the computing of the fiducial limits. A suitable valid value should be given. In the data set xval created by the SAS statements on page 3784, Dose = 2.

See the section "XDATA= *SAS-data-set*" on page 3763 for the default values for those effects other than the single continuous variable, for which the fiducial limits are computed.

In this example, there are two classification variables, treat and sex. Fiducial limits for the dose variable are computed for the highest level of the classification variables, treat = B and sex = 1, which is the default specification. Since these are the default values, you would get the same values and fiducial limits if you did not specify the XDATA= option in this example. The confidence level for the fiducial limits can be specified by the ALPHA= option in the MODEL statement. The default level is 95%.

If a LOG10 or LOG option is used in the PROC PROBIT statement, the values and the fiducial limits are computed for both the single continuous variable and its logarithm.

Output 60.4.10. Outest Data Set for Epidemiology Study

```
Obs _MODEL_ _NAME_     _TYPE_ _DIST_  _STATUS_    _LNLIKE_     r    Intercept

  1         r          PARMS  Normal 0 Converged -387.247 -1.00000  -0.88714
  2         Intercept  COV    Normal 0 Converged -387.247 -0.88714   0.13194
  3         dose       COV    Normal 0 Converged -387.247  1.67739  -0.08735
  4         treatA     COV    Normal 0 Converged -387.247 -1.25367   0.05355
  5         treatB     COV    Normal 0 Converged -387.247  0.00000   0.00000
  6         sex0       COV    Normal 0 Converged -387.247 -0.46329   0.03029
  7         sex1       COV    Normal 0 Converged -387.247  0.00000   0.00000
  8         treatAsex0 COV    Normal 0 Converged -387.247  1.28991  -0.06706
  9         treatAsex1 COV    Normal 0 Converged -387.247  0.00000   0.00000
 10         treatBsex0 COV    Normal 0 Converged -387.247  0.00000   0.00000
 11         treatBsex1 COV    Normal 0 Converged -387.247  0.00000   0.00000
 12         _C_        COV    Normal 0 Converged -387.247  0.27347  -0.02807

                         treat                 treat  treat treat treat
Obs   dose    treatA       B      sex0    sex1    Asex0  Asex1 Bsex0 Bsex1     _C_

  1  1.67739 -1.25367   0    -0.46329    0    1.28991   0      0     0     0.27347
  2 -0.08735  0.05355   0     0.03029    0   -0.06706   0      0     0    -0.02807
  3  0.06672 -0.04751   0    -0.03408    0    0.05862   0      0     0     0.01820
  4 -0.04751  0.06843   0     0.03606    0   -0.07532   0      0     0    -0.01708
  5  0.00000  0.00000   0     0.00000    0    0.00000   0      0     0     0.00000
  6 -0.03408  0.03606   0     0.05238    0   -0.06360   0      0     0    -0.00809
  7  0.00000  0.00000   0     0.00000    0    0.00000   0      0     0     0.00000
  8  0.05862 -0.07532   0    -0.06360    0    0.11941   0      0     0     0.01913
  9  0.00000  0.00000   0     0.00000    0    0.00000   0      0     0     0.00000
 10  0.00000  0.00000   0     0.00000    0    0.00000   0      0     0     0.00000
 11  0.00000  0.00000   0     0.00000    0    0.00000   0      0     0     0.00000
 12  0.01820 -0.01708   0    -0.00809    0    0.01913   0      0     0     0.00895
```

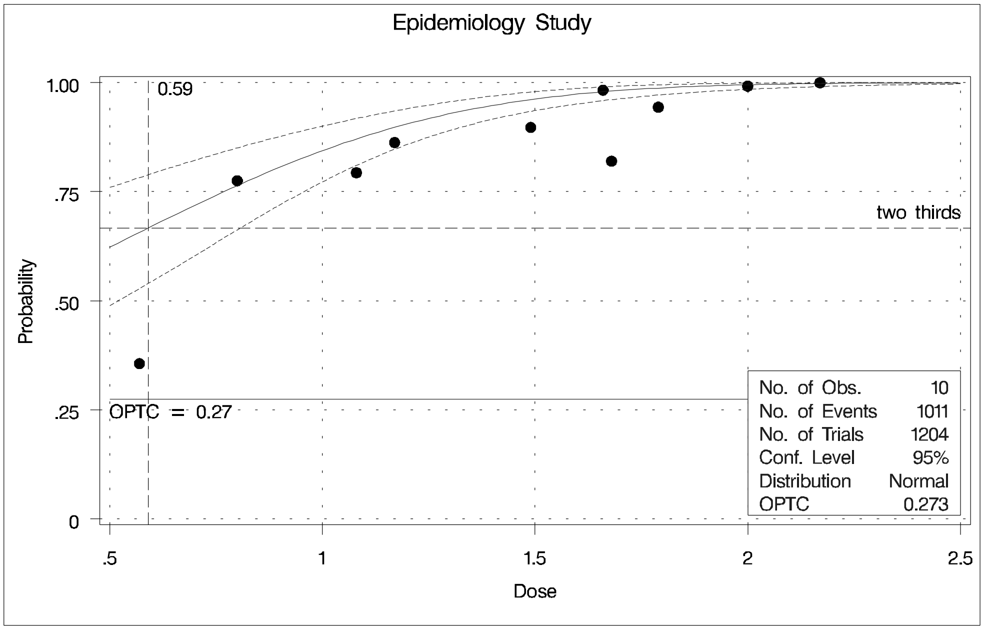
Output 60.4.10 displays the OUTEST= data set. All parameters for an effect are included.

The following three outputs, Output 60.4.11, Output 60.4.12, and Output 60.4.13, are generated from the three plot statements. The first plot, specified with the PREDPPLOT statement, is the plot of the predicted probability against the single continuous variable Dose, which is specified by the VAR= option in the PREDPPLOT statement. This single continuous variable must be in the MODEL statement. If the VAR= option is not used, the first single continuous variable in the MODEL statement is used. In this example, you would get the same plot if the VAR = dose was not used in the PREDPPLOT statement. You can specify values of other independent variables in the MODEL statement using an XDATA= data set, or by using the default values.

The second plot, specified with the IPPPLOT statement, is the inverse of the predicted probability plot with the fiducial limits. It should be pointed out that the fiducial limits are *not* just the inverse of the confidence limits in the predicted probability plot; see the section "Inverse Confidence Limits" on page 3761 for the computation of these limits. The third plot, specified with the LPREDPLOT statement, is the plot of the linear predictor $\mathbf{x}'\beta$ against the first single continuous variable (or the single continuous variable specified by the VAR= option) with the Wald confidence intervals.

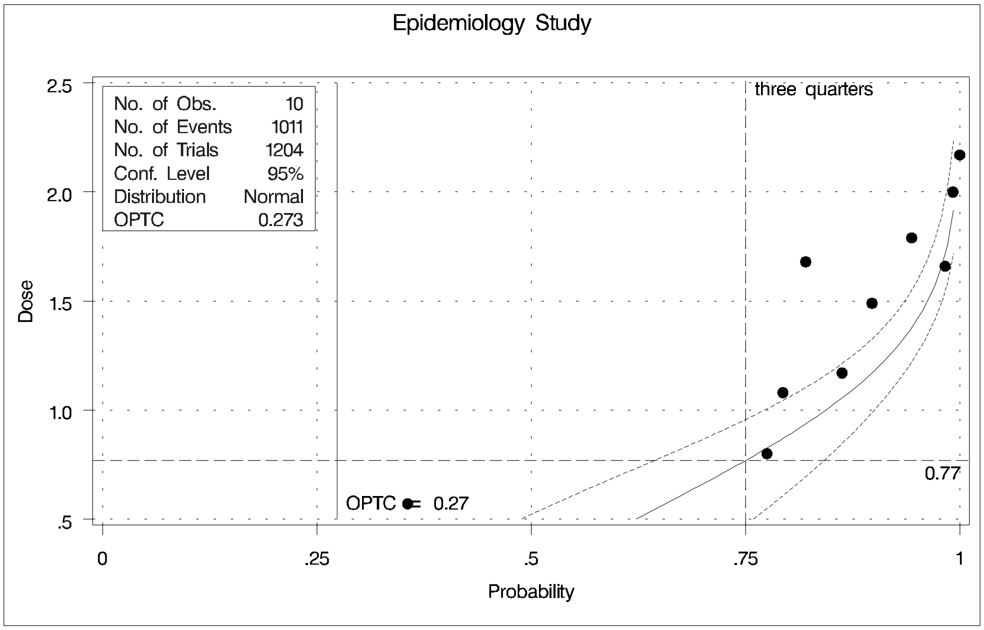
After each plot statement, an optional INSET statement is used to draw a box within the plot (inset box). In the inset box, information about the model fitting can be specified. See "INSET Statement" on page 3723 for more detail.

Output 60.4.11. Predicted Probability Plot

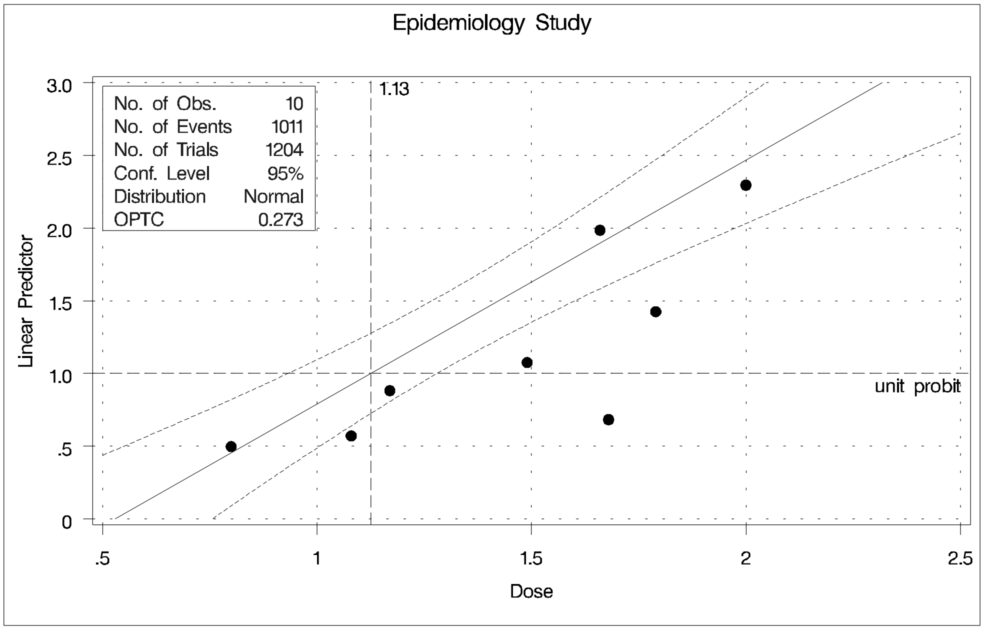


No. of Obs.	10
No. of Events	1011
No. of Trials	1204
Conf. Level	95%
Distribution	Normal
OPTC	0.273

Output 60.4.12. Inverse Predicted Probability Plot



Output 60.4.13. Linear Predictor Plot



Combining INEST= data set and the MAXIT= option in the MODEL statement, the PROBIT procedure can do prediction, if the parameterizations for the models used for the training data and the validation data are exactly the same.

After the first invocation of PROC PROBIT, you have the estimated parameters and their covariance matrix in the data set OUTEST = Out1, and the fitted probabilities for the training data set epidemic in the data set OUTPUT = Out2. See Output 60.4.10 on page 3791 for the data set Out1 and Output 60.4.14 on page 3795 for the data set Out2.

The validation data are collected in data set validate. The second invocation of PROC PROBIT simply passes the estimated parameters from the training data set epidemic to the validation data set validate for prediction. The predicted probabilities are stored in the data set OUTPUT = Out3 (see Output 60.4.15 on page 3795). The third invocation of PROC PROBIT passes the estimated parameters as initial values for a new fit of the validation data set using the same model. Predicted probabilities are stored in the data set OUTPUT = Out4 (see Output 60.4.16 on page 3795). Goodness-of-Fit tests are computed based on the cells grouped by the AGGREGATE= group variable. Results are shown in Output 60.4.17 on page 3796.

```
data validate;
   input treat $ dose  sex  n r group;
   datalines;
B  2.0  0  44 43  1
B  2.0  1  54 52  2
B  1.5  1  36 32  3
B  1.5  0  45 40  4
A  2.0  0  66 64  5
A  2.0  1  89 89  6
A  1.5  1  45 39  7
A  1.5  0  66 60  8
B  2.0  0  44 44  1
B  2.0  1  54 54  2
B  1.5  1  36 30  3
B  1.5  0  45 41  4
A  2.0  0  66 65  5
A  2.0  1  89 88  6
A  1.5  1  45 38  7
A  1.5  0  66 59  8
;

proc probit optc data = validate inest = out1;
   class treat sex;
   model r/n = dose treat sex sex*treat  / maxit = 0 ;
   output out = out3 p =p;
run ;

proc probit optc lackfit data = validate inest = out1;
   class treat sex;
   model r/n = dose treat sex sex*treat / aggregate = group ;
   output out = out4 p =p;
run ;
```

Output 60.4.14. Out2

```
              Obs    treat    dose      n      r     sex        p

                1      A      2.17    142    142      0     0.99272
                2      A      0.57    132     47      1     0.35925
                3      A      1.68    128    105      1     0.81899
                4      A      1.08    126    100      0     0.77517
                5      A      1.79    125    118      0     0.96682
                6      B      1.66    117    115      1     0.97901
                7      B      1.49    127    114      0     0.90896
                8      B      1.17     51     44      1     0.89749
                9      B      2.00    127    126      0     0.98364
               10      B      0.80    129    100      1     0.76414
```

Output 60.4.15. Out3

```
          Obs    treat    dose    sex     n     r    group       p

            1      B       2.0     0     44    43      1      0.98364
            2      B       2.0     1     54    52      2      0.99506
            3      B       1.5     1     36    32      3      0.96247
            4      B       1.5     0     45    40      4      0.91145
            5      A       2.0     0     66    64      5      0.98500
            6      A       2.0     1     89    89      6      0.91835
            7      A       1.5     1     45    39      7      0.74300
            8      A       1.5     0     66    60      8      0.91666
            9      B       2.0     0     44    44      1      0.98364
           10      B       2.0     1     54    54      2      0.99506
           11      B       1.5     1     36    30      3      0.96247
           12      B       1.5     0     45    41      4      0.91145
           13      A       2.0     0     66    65      5      0.98500
           14      A       2.0     1     89    88      6      0.91835
           15      A       1.5     1     45    38      7      0.74300
           16      A       1.5     0     66    59      8      0.91666
```

Output 60.4.16. Out4

```
          Obs    treat    dose    sex     n     r    group       p

            1      B       2.0     0     44    43      1      0.98954
            2      B       2.0     1     54    52      2      0.98262
            3      B       1.5     1     36    32      3      0.86187
            4      B       1.5     0     45    40      4      0.90095
            5      A       2.0     0     66    64      5      0.98768
            6      A       2.0     1     89    89      6      0.98614
            7      A       1.5     1     45    39      7      0.88075
            8      A       1.5     0     66    60      8      0.88964
            9      B       2.0     0     44    44      1      0.98954
           10      B       2.0     1     54    54      2      0.98262
           11      B       1.5     1     36    30      3      0.86187
           12      B       1.5     0     45    41      4      0.90095
           13      A       2.0     0     66    65      5      0.98768
           14      A       2.0     1     89    88      6      0.98614
           15      A       1.5     1     45    38      7      0.88075
           16      A       1.5     0     66    59      8      0.88964
```

Output 60.4.17. Goodness-of-Fit Table

```
                              Probit Procedure

                           Goodness-of-Fit Tests

        Statistic                         Value       DF    Pr > ChiSq

        Pearson Chi-Square               2.8101        2        0.2454
        L.R.    Chi-Square               2.8080        2        0.2456
```

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Collett, D (1991), *Modelling Binary Data*, London: Chapman and Hall.

Cox, D.R. (1970), *Analysis of Binary Data*, London: Chapman and Hall.

Cox, D.R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman and Hall.

Finney, D.J. (1971), *Probit Analysis*, Third Edition, London: Cambridge University Press.

Hubert, J.J., Bohidar, N.R., and Peace, K.E. (1988), “Assessment of Pharmacological Activity,” *Biopharmaceutical Statistics for Drug Development*, ed. K.E. Peace, New York: Marcel Dekker.

Chapter 61
The REG Procedure

Chapter Contents

Chapter 61
The REG Procedure

Overview

The REG procedure is one of many regression procedures in the SAS System. It is a general-purpose procedure for regression, while other SAS regression procedures provide more specialized applications. Other SAS/STAT procedures that perform at least one type of regression analysis are the CATMOD, GENMOD, GLM, LOGISTIC, MIXED, NLIN, ORTHOREG, PROBIT, RSREG, and TRANSREG procedures. SAS/ETS procedures are specialized for applications in time-series or simultaneous systems. These other SAS/STAT regression procedures are summarized in Chapter 2, "Introduction to Regression Procedures," which also contains an overview of regression techniques and defines many of the statistics computed by PROC REG and other regression procedures.

PROC REG provides the following capabilities:

- multiple MODEL statements
- nine model-selection methods
- interactive changes both in the model and the data used to fit the model
- linear equality restrictions on parameters
- tests of linear hypotheses and multivariate hypotheses
- collinearity diagnostics
- predicted values, residuals, studentized residuals, confidence limits, and influence statistics
- correlation or crossproduct input
- requested statistics available for output through output data sets
- experimental ODS graphics are now available. For more information, see the "ODS Graphics" section on page 3922. These plots are available in addition to the line-printer and the traditional high resolution plots currently available in PROC REG.
- plots
  - plot model fit summary statistics and diagnostic statistics
  - produce normal quantile-quantile (Q-Q) and probability-probability (P-P) plots for statistics such as residuals
  - specify special shorthand options to plot ridge traces, confidence intervals, and prediction intervals
  - display the fitted model equation, summary statistics, and reference lines on the plot

  - control the graphics appearance with PLOT statement options and with global graphics statements including the TITLE, FOOTNOTE, NOTE, SYMBOL, and LEGEND statements
  - "paint" or highlight line-printer scatter plots
  - produce partial regression leverage line-printer plots

Nine model-selection methods are available in PROC REG. In the simplest method, PROC REG fits the complete model that you specify. The other eight methods involve various ways of including or excluding variables from the model. You specify these methods with the SELECTION= option in the MODEL statement.

The methods are identified in the following list and are explained in detail in the "Model-Selection Methods" section on page 3873.

NONE no model selection. This is the default. The complete model specified in the MODEL statement is fit to the data.

FORWARD forward selection. This method starts with no variables in the model and adds variables.

BACKWARD backward elimination. This method starts with all variables in the model and deletes variables.

STEPWISE stepwise regression. This is similar to the FORWARD method except that variables already in the model do not necessarily stay there.

MAXR forward selection to fit the best one-variable model, the best two-variable model, and so on. Variables are switched so that R^2 is maximized.

MINR similar to the MAXR method, except that variables are switched so that the increase in R^2 from adding a variable to the model is minimized.

RSQUARE finds a specified number of models with the highest R^2 in a range of model sizes.

ADJRSQ finds a specified number of models with the highest adjusted R^2 in a range of model sizes.

CP finds a specified number of models with the lowest C_p in a range of model sizes.

Getting Started

Simple Linear Regression

Suppose that a response variable Y can be predicted by a linear function of a regressor variable X. You can estimate β_0, the intercept, and β_1, the slope, in

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$. Fitting this model with the REG procedure requires only the following MODEL statement, where y is the outcome variable and x is the regressor variable.

```
proc reg;
   model y=x;
run;
```

For example, you might use regression analysis to find out how well you can predict a child's weight if you know that child's height. The following data are from a study of nineteen children. Height and weight are measured for each child.

```
title 'Simple Linear Regression';
data Class;
   input Name $ Height Weight Age @@;
   datalines;
Alfred  69.0 112.5 14  Alice  56.5  84.0 13  Barbara 65.3  98.0 13
Carol   62.8 102.5 14  Henry  63.5 102.5 14  James   57.3  83.0 12
Jane    59.8  84.5 12  Janet  62.5 112.5 15  Jeffrey 62.5  84.0 13
John    59.0  99.5 12  Joyce  51.3  50.5 11  Judy    64.3  90.0 14
Louise  56.3  77.0 12  Mary   66.5 112.0 15  Philip  72.0 150.0 16
Robert  64.8 128.0 12  Ronald 67.0 133.0 15  Thomas  57.5  85.0 11
William 66.5 112.0 15
;
```

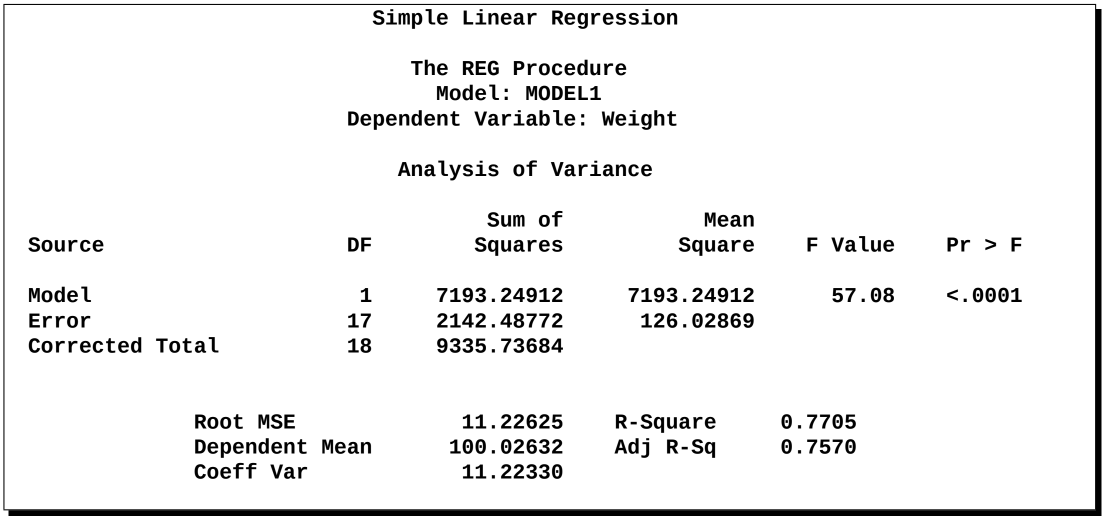
The equation of interest is

$$\text{Weight} = \beta_0 + \beta_1 \text{Height} + \epsilon$$

The variable Weight is the response or dependent variable in this equation, and β_0 and β_1 are the unknown parameters to be estimated. The variable Height is the regressor or independent variable, and ϵ is the unknown error. The following commands invoke the REG procedure and fit this model to the data.

```
proc reg;
   model Weight = Height;
run;
```

Figure 61.1 includes some information concerning model fit.



```
                          Simple Linear Regression

                             The REG Procedure
                               Model: MODEL1
                         Dependent Variable: Weight

                            Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                     1     7193.24912     7193.24912      57.08    <.0001
Error                    17     2142.48772      126.02869
Corrected Total          18     9335.73684


             Root MSE             11.22625    R-Square     0.7705
             Dependent Mean      100.02632    Adj R-Sq     0.7570
             Coeff Var            11.22330
```

Figure 61.1. ANOVA Table

The F statistic for the overall model is highly significant (F=57.076, p<0.0001), indicating that the model explains a significant portion of the variation in the data.

The degrees of freedom can be used in checking accuracy of the data and model. The model degrees of freedom are one less than the number of parameters to be estimated. This model estimates two parameters, β_0 and β_1; thus, the degrees of freedom should be $2 - 1 = 1$. The corrected total degrees of freedom are always one less than the total number of observations in the data set, in this case $19 - 1 = 18$.

Several simple statistics follow the ANOVA table. The Root MSE is an estimate of the standard deviation of the error term. The coefficient of variation, or Coeff Var, is a unitless expression of the variation in the data. The R-Square and Adj R-Square are two statistics used in assessing the fit of the model; values close to 1 indicate a better fit. The R-Square of 0.77 indicates that Height accounts for 77% of the variation in Weight.

The "Parameter Estimates" table shown in Figure 61.2 contains the estimates of β_0 and β_1. The table also contains the t statistics and the corresponding p-values for testing whether each parameter is significantly different from zero. The p-values ($t = -4.43$, $p = 0.0004$ and $t = 7.55$, $p < 0.0001$) indicate that the intercept and Height parameter estimates, respectively, are highly significant.

```
                         Simple Linear Regression

                            The REG Procedure
                              Model: MODEL1
                        Dependent Variable: Weight

                           Parameter Estimates

                           Parameter       Standard
    Variable     DF         Estimate          Error    t Value    Pr > |t|

    Intercept     1       -143.02692       32.27459      -4.43      0.0004
    Height        1          3.89903        0.51609       7.55      <.0001
```

Figure 61.2. Parameter Estimates

From the parameter estimates, the fitted model is

$$\text{Weight} = -143.0 + 3.9 \times \text{Height}$$

The REG procedure can be used interactively. After you specify a model with the MODEL statement and submit the PROC REG statements, you can submit further statements without reinvoking the procedure. The following command can now be issued to request a plot of the residual versus the predicted values, as shown in Figure 61.3.

```
plot r.*p.; run;
```

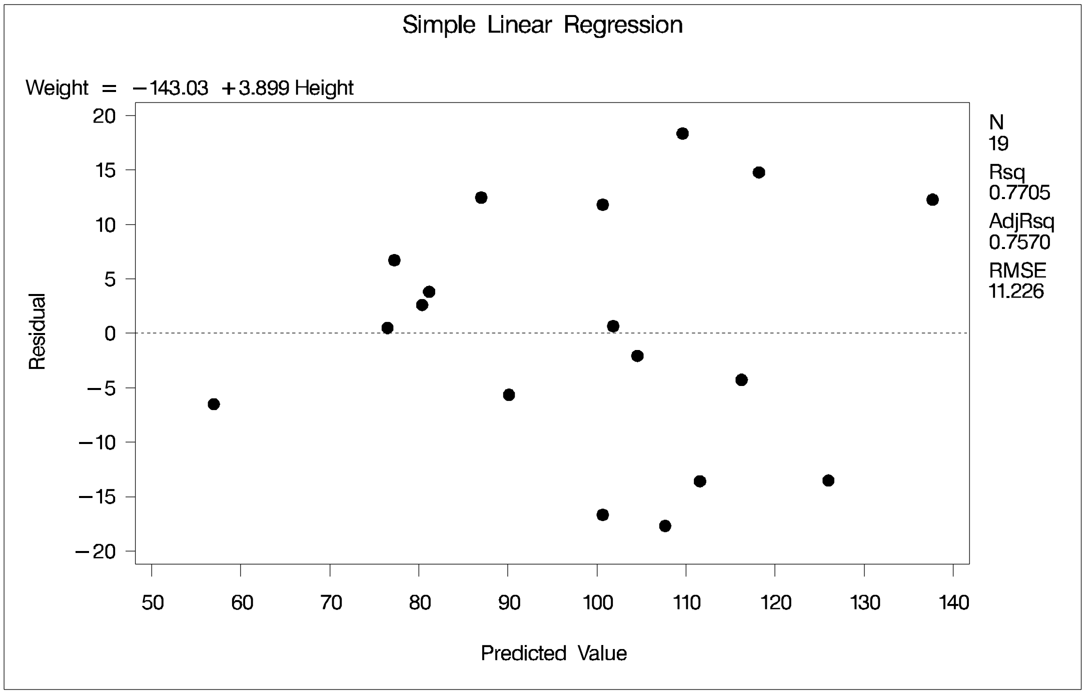


Figure 61.3. Plot of Residual vs. Predicted Values

A trend in the residuals would indicate nonconstant variance in the data. Figure 61.3 may indicate a slight trend in the residuals; they appear to increase slightly as the predicted values increase. A fan-shaped trend may indicate the need for a variance-stabilizing transformation. A curved trend (such as a semi-circle) may indicate the need for a quadratic term in the model. Since these residuals have no apparent trend, the analysis is considered to be acceptable.

Polynomial Regression

Consider a response variable Y that can be predicted by a polynomial function of a regressor variable X. You can estimate β_0, the intercept, β_1, the slope due to X, and β_2, the slope due to X^2, in

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$.

Consider the following example on population growth trends. The population of the United States from 1790 to 2000 is fit to linear and quadratic functions of time. Note that the quadratic term, YearSq, is created in the DATA step; this is done since polynomial effects such as Year*Year cannot be specified in the MODEL statement in PROC REG. The data are as follows:

```
data USPopulation;
   input Population @@;
   retain Year 1780;
   Year=Year+10;
   YearSq=Year*Year;
   Population=Population/1000;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
226542 248710 281422
;
```

The following statements begin the analysis. (Influence diagnostics and autocorrelation information for the full model are shown in Figure 61.43 on page 3900 and Figure 61.57 on page 3916.)

```
symbol1 c=blue;
proc reg data=USPopulation;
   var YearSq;
   model Population=Year / r cli clm;
   plot r.*p. / cframe=ligr;
run;
```

The DATA option ensures that the procedure uses the intended data set. Any variable that you might add to the model but that is not included in the first MODEL statement must appear in the VAR statement. In the MODEL statement, three options are specified: R requests a residual analysis to be performed, CLI requests 95% confidence limits for an individual value, and CLM requests these limits for the expected value of the dependent variable. You can request specific $100(1-\alpha)\%$ limits with the ALPHA= option in the PROC REG or MODEL statement. A plot of the residuals against the predicted values is requested by the PLOT statement.

The ANOVA table is displayed in Figure 61.4.

```
                                The REG Procedure
                                  Model: MODEL1
                          Dependent Variable: Population

                              Analysis of Variance

                                         Sum of           Mean
Source                     DF           Squares         Square    F Value    Pr > F

Model                       1            146869         146869     228.92    <.0001
Error                      20             12832      641.58160
Corrected Total            21            159700


             Root MSE              25.32946    R-Square     0.9197
             Dependent Mean        94.64800    Adj R-Sq     0.9156
             Coeff Var             26.76175


                              Parameter Estimates

                             Parameter       Standard
    Variable      DF          Estimate          Error    t Value    Pr > |t|

    Intercept      1       -2345.85498      161.39279     -14.54      <.0001
    Year           1           1.28786        0.08512      15.13      <.0001
```

Figure 61.4. ANOVA Table and Parameter Estimates

The Model F statistic is significant (F=228.92, p<0.0001), indicating that the model accounts for a significant portion of variation in the data. The R-Square indicates that the model accounts for 92% of the variation in population growth. The fitted equation for this model is

$$\text{Population} = -2345.85 + 1.29 \times \text{Year}$$

Figure 61.5 shows the confidence limits for both individual and expected values resulting from the CLM and CLI options.

```
                              The REG Procedure
                                Model: MODEL1
                        Dependent Variable: Population

                              Output Statistics

       Dependent Predicted    Std Error
   Obs  Variable     Value Mean Predict     95% CL Mean        95% CL Predict

     1    3.9290  -40.5778      10.4424  -62.3602  -18.7953  -97.7280   16.5725
     2    5.3080  -27.6991       9.7238  -47.9826   -7.4156  -84.2950   28.8968
     3    7.2390  -14.8205       9.0283  -33.6533    4.0123  -70.9128   41.2719
     4    9.6380   -1.9418       8.3617  -19.3841   15.5004  -57.5827   53.6991
     5   12.8660   10.9368       7.7314   -5.1906   27.0643  -44.3060   66.1797
     6   17.0690   23.8155       7.1470    8.9070   38.7239  -31.0839   78.7148
     7   23.1910   36.6941       6.6208   22.8834   50.5048  -17.9174   91.3056
     8   31.4430   49.5727       6.1675   36.7075   62.4380   -4.8073  103.9528
     9   39.8180   62.4514       5.8044   50.3436   74.5592    8.2455  116.6573
    10   50.1550   75.3300       5.5491   63.7547   86.9053   21.2406  129.4195
    11   62.9470   88.2087       5.4170   76.9090   99.5084   34.1776  142.2398
    12   75.9940  101.0873       5.4170   89.7876  112.3870   47.0562  155.1184
    13   91.9720  113.9660       5.5491  102.3907  125.5413   59.8765  168.0554
    14  105.7100  126.8446       5.8044  114.7368  138.9524   72.6387  181.0505
    15  122.7750  139.7233       6.1675  126.8580  152.5885   85.3432  194.1033
    16  131.6690  152.6019       6.6208  138.7912  166.4126   97.9904  207.2134
    17  151.3250  165.4805       7.1470  150.5721  180.3890  110.5812  220.3799
    18  179.3230  178.3592       7.7314  162.2317  194.4866  123.1163  233.6020
    19  203.2110  191.2378       8.3617  173.7956  208.6801  135.5969  246.8787
    20  226.5420  204.1165       9.0283  185.2837  222.9493  148.0241  260.2088
    21  248.7100  216.9951       9.7238  196.7116  237.2786  160.3992  273.5910
    22  281.4220  229.8738      10.4424  208.0913  251.6562  172.7235  287.0240
```

Figure 61.5. Confidence Limits

The observed dependent variable is displayed for each observation along with its predicted value from the regression equation and the standard error of the mean predicted value. The 95% CL Mean columns are the confidence limits for the expected value of each observation. The 95% CL Predict columns are the confidence limits for the individual observations.

Figure 61.6 displays the residual analysis requested by the R option.

```
                               Output Statistics

                       Std Error     Student                          Cook's
   Obs   Residual       Residual    Residual      -2-1 0 1 2               D

     1    44.5068         23.077       1.929     |      |***   |       0.381
     2    33.0071         23.389       1.411     |      |**    |       0.172
     3    22.0595         23.666       0.932     |      |*     |       0.063
     4    11.5798         23.909       0.484     |      |      |       0.014
     5     1.9292         24.121      0.0800     |      |      |       0.000
     6    -6.7465         24.300      -0.278     |      |      |       0.003
     7   -13.5031         24.449      -0.552     |     *|      |       0.011
     8   -18.1297         24.567      -0.738     |     *|      |       0.017
     9   -22.6334         24.655      -0.918     |     *|      |       0.023
    10   -25.1750         24.714      -1.019     |    **|      |       0.026
    11   -25.2617         24.743      -1.021     |    **|      |       0.025
    12   -25.0933         24.743      -1.014     |    **|      |       0.025
    13   -21.9940         24.714      -0.890     |     *|      |       0.020
    14   -21.1346         24.655      -0.857     |     *|      |       0.020
    15   -16.9483         24.567      -0.690     |     *|      |       0.015
    16   -20.9329         24.449      -0.856     |     *|      |       0.027
    17   -14.1555         24.300      -0.583     |     *|      |       0.015
    18     0.9638         24.121      0.0400     |      |      |       0.000
    19    11.9732         23.909       0.501     |      |*     |       0.015
    20    22.4255         23.666       0.948     |      |*     |       0.065
    21    31.7149         23.389       1.356     |      |**    |       0.159
    22    51.5482         23.077       2.234     |      |****  |       0.511


               Sum of Residuals                            0
               Sum of Squared Residuals                12832
               Predicted Residual SS (PRESS)           16662
```

Figure 61.6. Residual Analysis

The residual, its standard error, and the studentized residuals are displayed for each observation. The studentized residual is the residual divided by its standard error. The magnitude of each studentized residual is shown in a plot. Studentized residuals follow a t distribution and can be used to identify outlying or extreme observations. Asterisks (*) extending beyond the dashed lines indicate that the residual is more than three standard errors from zero. Many observations having absolute studentized residuals greater than 2 may indicate an inadequate model. The wave pattern seen in this plot is also an indication that the model is inadequate; a quadratic term may be needed or autocorrelation may be present in the data. Cook's D is a measure of the change in the predicted values upon deletion of that observation from the data set; hence, it measures the influence of the observation on the estimated regression coefficients. A fairly close agreement between the PRESS statistic (see Table 61.6 on page 3897) and the Sum of Squared Residuals indicates that the MSE is a reasonable measure of the predictive accuracy of the fitted model (Neter, Wasserman, and Kutner, 1990).

A plot of the residuals versus predicted values is shown in Figure 61.7.

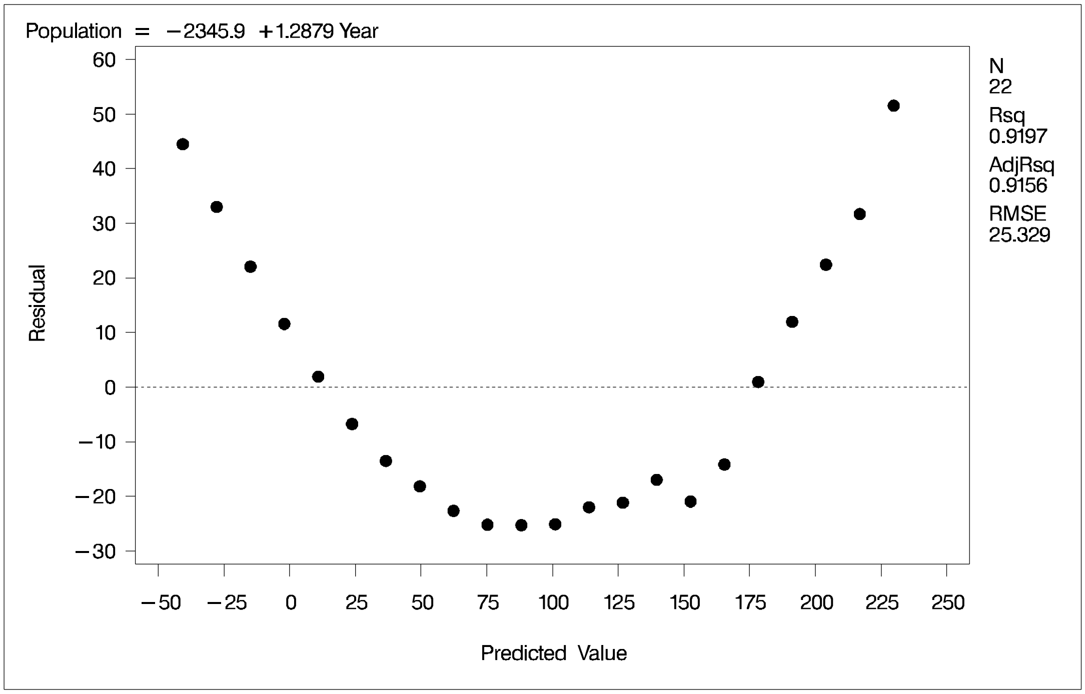


Figure 61.7. Plot of Residual vs. Predicted Values

The wave pattern of the studentized residual plot is seen here again. The semi-circle shape indicates an inadequate model; perhaps additional terms (such as the quadratic) are needed, or perhaps the data need to be transformed before analysis. If a model fits well, the plot of residuals against predicted values should exhibit no apparent trends.

Using the interactive feature of PROC REG, the following commands add the variable YearSq to the independent variables and refit the model.

```
add YearSq;
print;
plot / cframe=ligr;
run;
```

The ADD statement requests that YearSq be added to the model, and the PRINT command displays the ANOVA table for the new model. The PLOT statement with no variables recreates the most recent plot requested, in this case a plot of residual versus predicted values.

Figure 61.8 displays the ANOVA table and estimates for the new model.

```
                              The REG Procedure
                               Model: MODEL1.1
                        Dependent Variable: Population

                             Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                     2         159529          79765    8864.19    <.0001
Error                    19      170.97193        8.99852
Corrected Total          21         159700


             Root MSE              2.99975    R-Square     0.9989
             Dependent Mean       94.64800    Adj R-Sq     0.9988
             Coeff Var             3.16938


                             Parameter Estimates

                            Parameter       Standard
    Variable      DF         Estimate          Error    t Value    Pr > |t|

    Intercept      1            21631      639.50181      33.82      <.0001
    Year           1        -24.04581        0.67547     -35.60      <.0001
    YearSq         1          0.00668     0.00017820      37.51      <.0001
```

Figure 61.8. ANOVA Table and Parameter Estimates

The overall F statistic is still significant (F=8864.19, p<0.0001). The R-square has increased from 0.9197 to 0.9989, indicating that the model now accounts for 99.9% of the variation in Population. All effects are significant with p<0.0001 for each effect in the model.

The fitted equation is now

$$\text{Population} = 21631 - 24.046 \times \text{Year} + 0.0067 \times \text{Yearsq}$$

The confidence limits and residual analysis for the second model are displayed in Figure 61.9.

```
                          The REG Procedure
                           Model: MODEL1.1
                   Dependent Variable: Population

                          Output Statistics

     Dependent Predicted    Std Error
Obs  Variable      Value Mean Predict     95% CL Mean        95% CL Predict

  1    3.9290    6.2127       1.7565    2.5362    9.8892   -1.0631   13.4884
  2    5.3080    5.7226       1.4560    2.6751    8.7701   -1.2565   12.7017
  3    7.2390    6.5694       1.2118    4.0331    9.1057   -0.2021   13.3409
  4    9.6380    8.7531       1.0305    6.5963   10.9100    2.1144   15.3918
  5   12.8660   12.2737       0.9163   10.3558   14.1916    5.7087   18.8386
  6   17.0690   17.1311       0.8650   15.3207   18.9415   10.5968   23.6655
  7   23.1910   23.3254       0.8613   21.5227   25.1281   16.7932   29.8576
  8   31.4430   30.8566       0.8846   29.0051   32.7080   24.3107   37.4024
  9   39.8180   39.7246       0.9163   37.8067   41.6425   33.1597   46.2896
 10   50.1550   49.9295       0.9436   47.9545   51.9046   43.3476   56.5114
 11   62.9470   61.4713       0.9590   59.4641   63.4785   54.8797   68.0629
 12   75.9940   74.3499       0.9590   72.3427   76.3571   67.7583   80.9415
 13   91.9720   88.5655       0.9436   86.5904   90.5405   81.9836   95.1473
 14  105.7100  104.1178       0.9163  102.2000  106.0357   97.5529  110.6828
 15  122.7750  121.0071       0.8846  119.1556  122.8585  114.4612  127.5529
 16  131.6690  139.2332       0.8613  137.4305  141.0359  132.7010  145.7654
 17  151.3250  158.7962       0.8650  156.9858  160.6066  152.2618  165.3306
 18  179.3230  179.6961       0.9163  177.7782  181.6139  173.1311  186.2610
 19  203.2110  201.9328       1.0305  199.7759  204.0896  195.2941  208.5715
 20  226.5420  225.5064       1.2118  222.9701  228.0427  218.7349  232.2779
 21  248.7100  250.4168       1.4560  247.3693  253.4644  243.4378  257.3959
 22  281.4220  276.6642       1.7565  272.9877  280.3407  269.3884  283.9400

                           Output Statistics

                     Std Error     Student                            Cook's
     Obs  Residual    Residual    Residual      -2-1 0 1 2                 D

       1   -2.2837       2.432      -0.939    |      *|      |        0.153
       2   -0.4146       2.623      -0.158    |       |      |        0.003
       3    0.6696       2.744       0.244    |       |      |        0.004
       4    0.8849       2.817       0.314    |       |      |        0.004
       5    0.5923       2.856       0.207    |       |      |        0.001
       6   -0.0621       2.872     -0.0216    |       |      |        0.000
       7   -0.1344       2.873     -0.0468    |       |      |        0.000
       8    0.5864       2.866       0.205    |       |      |        0.001
       9    0.0934       2.856      0.0327    |       |      |        0.000
      10    0.2255       2.847      0.0792    |       |      |        0.000
      11    1.4757       2.842       0.519    |       |*     |        0.010
      12    1.6441       2.842       0.578    |       |*     |        0.013
      13    3.4065       2.847       1.196    |       |**    |        0.052
      14    1.5922       2.856       0.557    |       |*     |        0.011
      15    1.7679       2.866       0.617    |       |*     |        0.012
      16   -7.5642       2.873      -2.632    | *****|      |        0.208
      17   -7.4712       2.872      -2.601    | *****|      |        0.205
      18   -0.3731       2.856      -0.131    |       |      |        0.001
      19    1.2782       2.817       0.454    |       |      |        0.009
      20    1.0356       2.744       0.377    |       |      |        0.009
      21   -1.7068       2.623      -0.651    |      *|      |        0.044
      22    4.7578       2.432       1.957    |       |***   |        0.666


                 Sum of Residuals                -4.4596E-11
                 Sum of Squared Residuals          170.97193
                 Predicted Residual SS (PRESS)     237.71229
```

Figure 61.9. Confidence Limits and Residual Analysis

The plot of the studentized residuals shows that the wave structure is gone. The PRESS statistic is much closer to the Sum of Squared Residuals now, and both statistics have been dramatically reduced. Most of the Cook's D statistics have also been reduced.

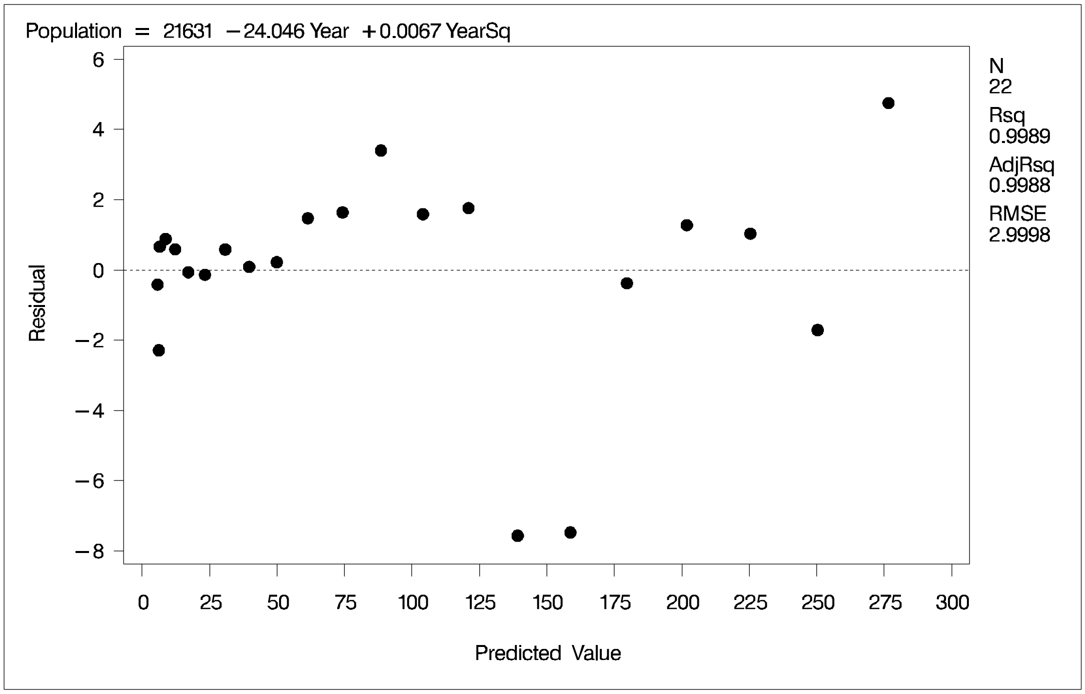


Figure 61.10. Plot of Residual vs. Predicted Values

The plot of residuals versus predicted values seen in Figure 61.10 has improved since a major trend is no longer visible.

To create a plot of the observed values, predicted values, and confidence limits against Year all on the same plot and to exert some control over the look of the resulting plot, you can submit the following statements.

```
symbol1 v=dot     c=yellow h=.3;
symbol2 v=square  c=red;
symbol3 f=simplex c=blue  h=2 v='-';
symbol4 f=simplex c=blue  h=2 v='-';
plot (Population predicted. u95. l95.)*Year
     / overlay cframe=ligr;
run;
```

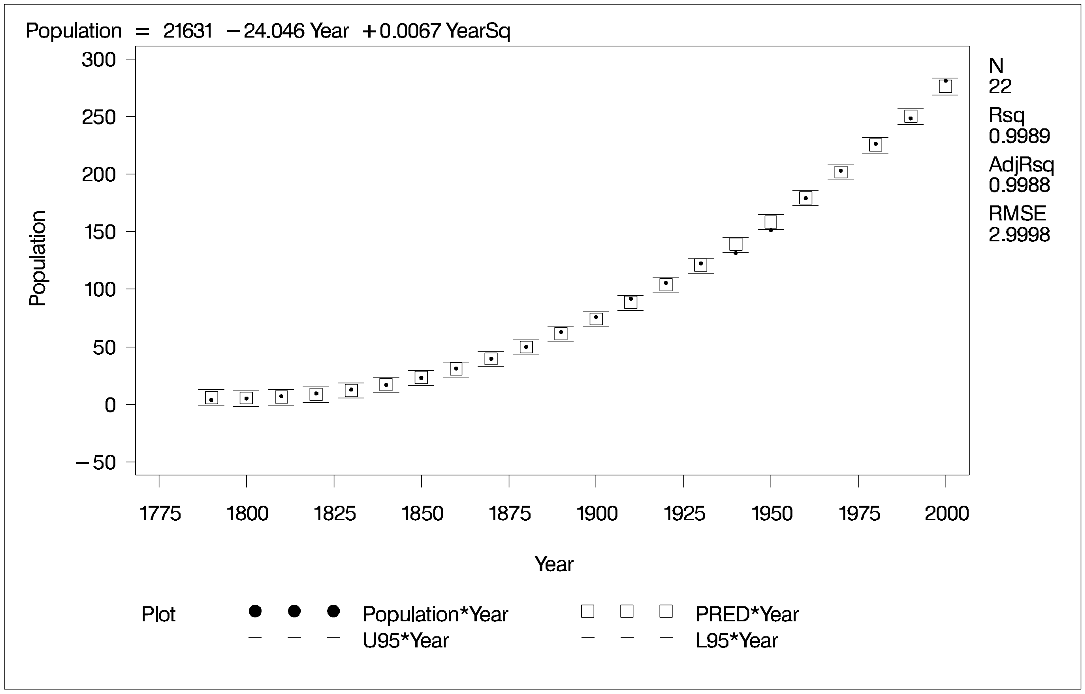


Figure 61.11. Plot of Population vs Year with Confidence Limits

The SYMBOL statements requests that the actual data be displayed as dots, the predicted values as squares, and the upper and lower 95% confidence limits for an individual value (sometimes called a *prediction interval*) as dashes. PROC REG provides the short-hand commands CONF and PRED to request confidence and prediction intervals for simple regression models; see the "PLOT Statement" section on page 3839 for details.

To complete an analysis of these data, you may want to examine influence statistics and, since the data are essentially time series data, examine the Durbin-Watson statistic. You might also want to examine other residual plots, such as the residuals vs. regressors.

Using PROC REG Interactively

PROC REG can be used interactively. After you specify a model with a MODEL statement and run REG with a RUN statement, a variety of statements can be executed without reinvoking REG.

The "Interactive Analysis" section on page 3869 describes which statements can be used interactively. These interactive statements can be executed singly or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement can be repeated. This is an important difference from the GLM procedure, which allows only one MODEL statement.

If you use REG interactively, you can end the REG procedure with a DATA step, another PROC step, an ENDSAS statement, or with a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you are using REG interactively, additional RUN statements do not end REG but tell the procedure to execute additional statements.

When a BY statement is used with PROC REG, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

When using REG interactively, you can fit a model, perform diagnostics, then refit the model, and perform diagnostics on the refitted model. Most of the interactive statements implicitly refit the model; for example, if you use the ADD statement to add a variable to the model, the regression equation is automatically recomputed. The two exceptions to this automatic recomputing are the PAINT and REWEIGHT statements. These two statements do not cause the model to be refitted. To do so, you can follow these statements either with a REFIT statement, which causes the model to be explicitly recomputed, or with another interactive statement that causes the model to be implicitly recomputed.

Syntax

The following statements are available in PROC REG.

```
PROC REG < options > ;
    < label: > MODEL dependents=<regressors> < / options > ;
    BY variables ;
    FREQ variable ;
    ID variables ;
    VAR variables ;
    WEIGHT variable ;
    ADD variables ;
    DELETE variables ;
    < label: > MTEST <equation, . . . ,equation> < / options > ;
    OUTPUT < OUT=SAS-data-set > keyword=names
        < . . . keyword=names > ;
    PAINT <condition | ALLOBS>
        < / options > | < STATUS | UNDO> ;
    PLOT <yvariable*xvariable> <=symbol>
        < . . .yvariable*xvariable> <=symbol> < / options > ;
    PRINT < options > < ANOVA > < MODELDATA > ;
    REFIT;
    RESTRICT equation, . . . ,equation ;
    REWEIGHT <condition | ALLOBS>
        < / options > | < STATUS | UNDO> ;
    < label: > TEST equation,<, . . .,equation> < / option > ;
```

Although there are numerous statements and options available in PROC REG, many analyses use only a few of them. Often you can find the features you need by looking at an example or by scanning this section.

In the preceding list, brackets denote optional specifications, and vertical bars denote a choice of one of the specifications separated by the vertical bars. In all cases, *label* is optional.

The PROC REG statement is required. To fit a model to the data, you must specify the MODEL statement. If you want to use only the options available in the PROC REG statement, you do not need a MODEL statement, but you must use a VAR statement. (See the example in the "OUTSSCP= Data Sets" section on page 3868.) Several MODEL statements can be used. In addition, several MTEST, OUTPUT, PAINT, PLOT, PRINT, RESTRICT, and TEST statements can follow each MODEL statement. The ADD, DELETE, and REWEIGHT statements are used interactively to change the regression model and the data used in fitting the model. The ADD, DELETE, MTEST, OUTPUT, PLOT, PRINT, RESTRICT, and TEST statements implicitly refit the model; changes made to the model are reflected in the results from these statements. The REFIT statement is used to refit the model explicitly and is most helpful when it follows PAINT and REWEIGHT statements, which do not refit the model. The BY, FREQ, ID, VAR, and WEIGHT statements are optionally specified once for the entire PROC step, and they must appear before the first RUN statement.

When TYPE=CORR, TYPE=COV, or TYPE=SSCP data sets are used as input data sets to PROC REG, statements and options that require the original data are not available. Specifically, the OUTPUT, PAINT, PLOT, and REWEIGHT statements and the MODEL and PRINT statement options P, R, CLM, CLI, DW, DWPROB, INFLUENCE, and PARTIAL are disabled.

You can specify the following statements with the REG procedure in addition to the PROC REG statement:

ADD adds independent variables to the regression model.

BY specifies variables to define subgroups for the analysis.

DELETE deletes independent variables from the regression model.

FREQ specifies a frequency variable.

ID names a variable to identify observations in the tables.

MODEL specifies the dependent and independent variables in the regression model, requests a model selection method, displays predicted values, and provides details on the estimates (according to which options are selected).

MTEST performs multivariate tests across multiple dependent variables.

OUTPUT creates an output data set and names the variables to contain predicted values, residuals, and other diagnostic statistics.

PAINT paints points in scatter plots.

PLOT generates scatter plots.

PRINT displays information about the model and can reset options.

REFIT refits the model.

RESTRICT places linear equality restrictions on the parameter estimates.

REWEIGHT excludes specific observations from analysis or changes the weights of observations used.

TEST performs an F test on linear functions of the parameters.

VAR lists variables for which crossproducts are to be computed, variables that can be interactively added to the model, or variables to be used in scatter plots.

WEIGHT declares a variable to weight observations.

PROC REG Statement

PROC REG < *options* > **;**

The PROC REG statement is required. If you want to fit a model to the data, you must also use a MODEL statement. If you want to use only the PROC REG options, you do not need a MODEL statement, but you must use a VAR statement. If you do not use a MODEL statement, then the COVOUT and OUTEST= options are not available.

Table 61.1 lists the options you can use with the PROC REG statement. Note that any option specified in the PROC REG statement applies to all MODEL statements.

Table 61.1. PROC REG Statement Options

Option	**Description**
Data Set Options	
DATA=	names a data set to use for the regression
OUTEST=	outputs a data set that contains parameter estimates and other model fit summary statistics
OUTSSCP=	outputs a data set that contains sums of squares and crossproducts
COVOUT	outputs the covariance matrix for parameter estimates to the OUTEST= data set
EDF	outputs the number of regressors, the error degrees of freedom, and the model R^2 to the OUTEST= data set
OUTSTB	outputs standardized parameter estimates to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
OUTSEB	outputs standard errors of the parameter estimates to the OUTEST= data set
OUTVIF	outputs the variance inflation factors to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
PCOMIT=	performs incomplete principal component analysis and outputs estimates to the OUTEST= data set
PRESS	outputs the PRESS statistic to the OUTEST= data set
RIDGE=	performs ridge regression analysis and outputs estimates to the OUTEST= data set
RSQUARE	same effect as the EDF option

Table 61.1. (continued)

Option	Description
TABLEOUT	outputs standard errors, confidence limits, and associated test statistics of the parameter estimates to the OUTEST= data set
High Resolution Graphics Options	
ANNOTATE=	specifies an annotation data set
GOUT=	specifies the graphics catalog in which graphics output is saved
Display Options	
CORR	displays correlation matrix for variables listed in MODEL and VAR statements
SIMPLE	displays simple statistics for each variable listed in MODEL and VAR statements
USCCP	displays uncorrected sums of squares and crossproducts matrix
ALL	displays all statistics (CORR, SIMPLE, and USSCP)
NOPRINT	suppresses output
LINEPRINTER	creates plots requested as line printer plot
Other Options	
ALPHA=	sets significance value for confidence and prediction intervals and tests
SINGULAR=	sets criterion for checking for singularity

Following are explanations of the options that you can specify in the PROC REG statement (in alphabetical order). Note that any option specified in the PROC REG statement applies to all MODEL statements.

ALL

requests the display of many tables. Using the ALL option in the PROC REG statement is equivalent to specifying ALL in every MODEL statement. The ALL option also implies the CORR, SIMPLE, and USSCP options.

ALPHA=*number*

sets the significance level used for the construction of confidence intervals. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals. This option affects the PROC REG option TABLEOUT; the MODEL options CLB, CLI, and CLM; the OUTPUT statement keywords LCL, LCLM, UCL, and UCLM; the PLOT statement keywords LCL., LCLM., UCL., and UCLM.; and the PLOT statement options CONF and PRED.

ANNOTATE=*SAS-data-set*
ANNO= *SAS-data-set*

specifies an input data set containing annotate variables, as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to plots. Features provided in this data set are applied to all plots produced in the current run of PROC REG. To add features to individual plots, use the ANNOTATE= option in the PLOT statement. This option cannot be used if the LINEPRINTER option is specified.

CORR

displays the correlation matrix for all variables listed in the MODEL or VAR statement.

COVOUT

outputs the covariance matrices for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is also specified. See the "OUTEST= Data Set" section on page 3863.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC REG. The data set can be an ordinary SAS data set or a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set. If one of these special TYPE= data sets is used, the OUTPUT, PAINT, PLOT, and REWEIGHT statements and some options in the MODEL and PRINT statements are not available. See Appendix A, "Special SAS Data Sets," for more information on TYPE= data sets. If the DATA= option is not specified, PROC REG uses the most recently created SAS data set.

EDF

outputs the number of regressors in the model excluding and including the intercept, the error degrees of freedom, and the model R^2 to the OUTEST= data set.

GOUT=*graphics-catalog*

specifies the graphics catalog in which graphics output is saved. The default *graphics-catalog* is WORK.GSEG. The GOUT= option cannot be used if the LINEPRINTER option is specified.

LINEPRINTER | LP

creates plots requested as line printer plots. If you do not specify this option, requested plots are created on a high resolution graphics device. This option is required if plots are requested and you do not have SAS/GRAPH software.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

OUTEST=*SAS-data-set*

requests that parameter estimates and optional model fit summary statistics be output to this data set. See the "OUTEST= Data Set" section on page 3863 for details. If you want to create a permanent SAS data set, you must specify a two-level name (refer to the section "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent SAS data sets).

OUTSEB

outputs the standard errors of the parameter estimates to the OUTEST= data set. The value SEB for the variable _TYPE_ identifies the standard errors. If the RIDGE= or PCOMIT= option is specified, additional observations are included and identified by the values RIDGESEB and IPCSEB, respectively, for the variable _TYPE_. The standard errors for ridge regression estimates and IPC estimates are limited in their usefulness because these estimates are biased. This option is available for all model selection methods except RSQUARE, ADJRSQ, and CP.

OUTSSCP=*SAS-data-set*

requests that the sums of squares and crossproducts matrix be output to this TYPE=SSCP data set. See the "OUTSSCP= Data Sets" section on page 3868 for details. If you want to create a permanent SAS data set, you must specify a two-level name (refer to the section "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent SAS data sets).

OUTSTB

outputs the standardized parameter estimates as well as the usual estimates to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The values RIDGESTB and IPCSTB for the variable _TYPE_ identify ridge regression estimates and IPC estimates, respectively.

OUTVIF

outputs the variance inflation factors (VIF) to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The factors are the diagonal elements of the inverse of the correlation matrix of regressors as adjusted by ridge regression or IPC analysis. These observations are identified in the output data set by the values RIDGEVIF and IPCVIF for the variable _TYPE_.

PCOMIT=*list*

requests an incomplete principal components (IPC) analysis for each value *m* in the list. The procedure computes parameter estimates using all but the last *m* principal components. Each value of *m* produces a set of IPC estimates, which are output to the OUTEST= data set. The values of *m* are saved by the variable _PCOMIT_, and the value of the variable _TYPE_ is set to IPC to identify the estimates. Only nonnegative integers can be specified with the PCOMIT= option.

If you specify the PCOMIT= option, RESTRICT statements are ignored.

PRESS

outputs the PRESS statistic to the OUTEST= data set. The values of this statistic are saved in the variable _PRESS_. This option is available for all model selection methods except RSQUARE, ADJRSQ, and CP.

RIDGE=*list*

requests a ridge regression analysis and specifies the values of the ridge constant *k* (see the "Computations for Ridge Regression and IPC Analysis" section on page 3916). Each value of *k* produces a set of ridge regression estimates that are placed in the OUTEST= data set. The values of *k* are saved by the variable _RIDGE_, and the value of the variable _TYPE_ is set to RIDGE to identify the estimates.

Only nonnegative numbers can be specified with the RIDGE= option. Example 61.10 on page 3956 illustrates this option.

If ODS graphics are in effect (see the "ODS Graphics" section on page 3922), then ridge regression plots are automatically produced. These plots consist of panels containing ridge traces for the regressors, with at most eight ridge traces per panel.

If you specify the RIDGE= option, RESTRICT statements are ignored.

RSQUARE

has the same effect as the EDF option.

SIMPLE

displays the sum, mean, variance, standard deviation, and uncorrected sum of squares for each variable used in PROC REG.

SINGULAR=*n*

tunes the mechanism used to check for singularities. The default value is machine dependent but is approximately 1E$-$7 on most machines. This option is rarely needed. Singularity checking is described in the "Computational Methods" section on page 3917.

TABLEOUT

outputs the standard errors and $100(1-\alpha)\%$ confidence limits for the parameter estimates, the t statistics for testing if the estimates are zero, and the associated p-values to the OUTEST= data set. The _TYPE_ variable values STDERR, LnB, UnB, T, and PVALUE, where $n = 100(1-\alpha)$, identify these rows in the OUTEST= data set. The α-level can be set with the ALPHA= option in the PROC REG or MODEL statement. The OUTEST= option must be specified in the PROC REG statement for this option to take effect.

USSCP

displays the uncorrected sums-of-squares and crossproducts matrix for all variables used in the procedure.

ADD Statement

ADD *variables* **;**

The ADD statement adds independent variables to the regression model. Only variables used in the VAR statement or used in MODEL statements before the first RUN statement can be added to the model. You can use the ADD statement interactively to add variables to the model or to include a variable that was previously deleted with a DELETE statement. Each use of the ADD statement modifies the MODEL label. See the "Interactive Analysis" section on page 3869 for an example.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC REG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives.

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the REG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

When a BY statement is used with PROC REG, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure. A BY statement that appears after the first RUN statement is ignored.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Contents*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

DELETE Statement

DELETE *variables* **;**

The DELETE statement deletes independent The DELETE statement performs the opposite function of the ADD statement and is used in a similar manner. Each use of the DELETE statement modifies the MODEL label. For an example of how the ADD statement is used (and how the DELETE statement can be used), see the "Interactive Analysis" section on page 3869.

FREQ Statement

FREQ *variable* **;**

When a FREQ statement appears, each observation in the input data set is assumed to represent n observations, where n is the value of the FREQ variable. The analysis produced using a FREQ statement is the same as an analysis produced using a data set that contains n observations in place of each observation in the input data set. When the procedure determines degrees of freedom for significance tests, the total number of observations is considered to be equal to the sum of the values of the FREQ variable.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

The FREQ statement must appear before the first RUN statement, or it is ignored.

ID Statement

ID *variables* **;**

When one of the MODEL statement options CLI, CLM, P, R, or INFLUENCE is requested, the variables listed in the ID statement are displayed beside each observation. These variables can be used to identify each observation. If the ID statement is omitted, the observation number is used to identify the observations.

Although there are no restrictions on the length of ID variables, PROC REG may truncate ID values to 16 characters for display purposes.

MODEL Statement

< *label:* > **MODEL** *dependents=<regressors>* < */ options* > **;**

After the keyword MODEL, the dependent (response) variables are specified, followed by an equal sign and the regressor variables. Variables specified in the MODEL statement must be numeric variables in the data set being analyzed. For example, if you want to specify a quadratic term for variable *X1* in the model, you cannot use X1*X1 in the MODEL statement but must create a new variable (for example, X1SQUARE=X1*X1) in a DATA step and use this new variable in the MODEL statement. The label in the MODEL statement is optional.

Table 61.2 lists the options available in the MODEL statement. Equations for the statistics available are given in the "Model Fit and Diagnostic Statistics" section on page 3896.

Table 61.2. MODEL Statement Options

Option	Description
Model Selection and Details of Selection	
SELECTION=	specifies model selection method
BEST=	specifies maximum number of subset models displayed or output to the OUTEST= data set
DETAILS	produces summary statistics at each step
DETAILS=	specifies the display details for forward, backward, and stepwise methods
GROUPNAMES=	provides names for groups of variables
INCLUDE=	includes first n variables in the model
MAXSTEP=	specifies maximum number of steps that may be performed
NOINT	fits a model without the intercept term
PCOMIT=	performs incomplete principal component analysis and outputs estimates to the OUTEST= data set
SLE=	sets criterion for entry into model
RIDGE=	performs ridge regression analysis and outputs estimates to the OUTEST= data set
SLS=	sets criterion for staying in model
START=	specifies number of variables in model to begin the comparing and switching process
STOP=	stops selection criterion

Table 61.2. (continued)

Option	Description
Fit Statistics	
ADJRSQ	computes adjusted R^2
AIC	computes Akaike's information criterion
B	computes parameter estimates for each model
BIC	computes Sawa's Bayesian information criterion
CP	computes Mallows' C_p statistic
GMSEP	computes estimated MSE of prediction assuming multivariate normality
JP	computes J_p, the final prediction error
MSE	computes MSE for each model
PC	computes Amemiya's prediction criterion
RMSE	displays root MSE for each model
SBC	computes the SBC statistic
SP	computes S_p statistic for each model
SSE	computes error sum of squares for each model
Data Set Options	
EDF	outputs the number of regressors, the error degrees of freedom, and the model R^2 to the OUTEST= data set
OUTSEB	outputs standard errors of the parameter estimates to the OUTEST= data set
OUTSTB	outputs standardized parameter estimates to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
OUTVIF	outputs the variance inflation factors to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
PRESS	outputs the PRESS statistic to the OUTEST= data set
RSQUARE	has same effect as the EDF option
Regression Calculations	
I	displays inverse of sums of squares and crossproducts
XPX	displays sums-of-squares and crossproducts matrix
Details on Estimates	
ACOV	displays asymptotic covariance matrix of estimates assuming heteroscedasticity
COLLIN	produces collinearity analysis
COLLINOINT	produces collinearity analysis with intercept adjusted out
CORRB	displays correlation matrix of estimates
COVB	displays covariance matrix of estimates
PARTIALR2	displays squared semi-partial correlation coefficients using Type I sums of squares
PCORR1	displays squared partial correlation coefficients using Type I sums of squares
PCORR2	displays squared partial correlation coefficients using Type II sums of squares
SCORR1	displays squared semi-partial correlation coefficients using Type I sums of squares

Table 61.2. (continued)

Option	Description
SCORR2	displays squared semi-partial correlation coefficients using Type II sums of squares
SEQB	displays a sequence of parameter estimates during selection process
SPEC	tests that first and second moments of model are correctly specified
SS1	displays the sequential sums of squares
SS2	displays the partial sums of squares
STB	displays standardized parameter estimates
TOL	displays tolerance values for parameter estimates
VIF	computes variance-inflation factors
Predicted and Residual Values	
CLB	computes $100(1-\alpha)$% confidence limits for the parameter estimates
CLI	computes $100(1-\alpha)$% confidence limits for an individual predicted value
CLM	computes $100(1-\alpha)$% confidence limits for the expected value of the dependent variable
DW	computes a Durbin-Watson statistic
DWPROB	computes a Durbin-Watson statistic and p-value
INFLUENCE	computes influence statistics
P	computes predicted values
PARTIAL	displays partial regression plots for each regressor
R	produces analysis of residuals
Display Options and Other Options	
ALL	requests the following options: ACOV, CLB, CLI, CLM, CORRB, COVB, I, P, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, XPX
ALPHA=	sets significance value for confidence and prediction intervals and tests
NOPRINT	suppresses display of results
SIGMA=	specifies the true standard deviation of error term for computing CP and BIC
SINGULAR=	sets criterion for checking for singularity

You can specify the following options in the MODEL statement after a slash (/).

ACOV

displays the estimated asymptotic covariance matrix of the estimates under the hypothesis of heteroscedasticity. See the section "Testing for Heteroscedasticity" on page 3910 for more information.

ADJRSQ

computes R^2 adjusted for degrees of freedom for each model selected (Darlington 1968; Judge et al. 1980).

AIC

outputs Akaike's information criterion for each model selected (Akaike 1969; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the AIC statistic is also added to the SubsetSelSummary table.

ALL

requests all these options: ACOV, CLB, CLI, CLM, CORRB, COVB, I, P, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, and XPX.

ALPHA=*number*

sets the significance level used for the construction of confidence intervals for the current MODEL statement. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals. This option affects the MODEL options CLB, CLI, and CLM; the OUTPUT statement keywords LCL, LCLM, UCL, and UCLM; the PLOT statement keywords LCL., LCLM., UCL., and UCLM.; and the PLOT statement options CONF and PRED. Specifying this option in the MODEL statement takes precedence over the ALPHA= option in the PROC REG statement.

B

is used with the RSQUARE, ADJRSQ, and CP model-selection methods to compute estimated regression coefficients for each model selected.

BEST=*n*

is used with the RSQUARE, ADJRSQ, and CP model-selection methods. If SELECTION=CP or SELECTION=ADJRSQ is specified, the BEST= option specifies the maximum number of subset models to be displayed or output to the OUTEST= data set. For SELECTION=RSQUARE, the BEST= option requests the maximum number of subset models for each size.

If the BEST= option is used without the B option (displaying estimated regression coefficients), the variables in each MODEL are listed in order of inclusion instead of the order in which they appear in the MODEL statement.

If the BEST= option is omitted and the number of regressors is less than 11, all possible subsets are evaluated. If the BEST= option is omitted and the number of regressors is greater than 10, the number of subsets selected is, at most, equal to the number of regressors. A small value of the BEST= option greatly reduces the CPU time required for large problems.

BIC

outputs Sawa's Bayesian information criterion for each model selected (Sawa 1978; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the BIC statistic is also added to the SubsetSelSummary table.

CLB

requests the $100(1-\alpha)$% upper- and lower-confidence limits for the parameter estimates. By default, the 95% limits are computed; the ALPHA= option in the PROC REG or MODEL statement can be used to change the α-level.

CLI

requests the $100(1-\alpha)$% upper- and lower-confidence limits for an individual predicted value. By default, the 95% limits are computed; the ALPHA= option in the PROC REG or MODEL statement can be used to change the α-level. The confidence limits reflect variation in the error, as well as variation in the parameter estimates. See the "Predicted and Residual Values" section on page 3879 and Chapter 2, "Introduction to Regression Procedures," for more information.

CLM

displays the $100(1-\alpha)$% upper- and lower-confidence limits for the expected value of the dependent variable (mean) for each observation. By default, the 95% limits are computed; the ALPHA= in the PROC REG or MODEL statement can be used to change the α-level. This is not a prediction interval (see the CLI option) because it takes into account only the variation in the parameter estimates, not the variation in the error term. See the section "Predicted and Residual Values" on page 3879 and Chapter 2 for more information.

COLLIN

requests a detailed analysis of collinearity among the regressors. This includes eigenvalues, condition indices, and decomposition of the variances of the estimates with respect to each eigenvalue. See the "Collinearity Diagnostics" section on page 3895.

COLLINOINT

requests the same analysis as the COLLIN option with the intercept variable adjusted out rather than included in the diagnostics. See the "Collinearity Diagnostics" section on page 3895.

CORRB

displays the correlation matrix of the estimates. This is the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix scaled to unit diagonals.

COVB

displays the estimated covariance matrix of the estimates. This matrix is $(\mathbf{X}'\mathbf{X})^{-1}s^2$, where s^2 is the estimated mean squared error.

CP

outputs Mallows' C_p statistic for each model selected (Mallows 1973; Hocking 1976). See the "Criteria Used in Model-Selection Methods" section on page 3876 for a discussion of the use of C_p. to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the CP statistic is also added to the SubsetSelSummary table.

DETAILS
DETAILS=*name*

specifies the level of detail produced when the BACKWARD, FORWARD or STEPWISE methods are used, where *name* can be ALL, STEPS or SUMMARY. The

DETAILS or DETAILS=ALL option produces entry and removal statistics for each variable in the model building process, ANOVA and parameter estimates at each step, and a selection summary table. The option DETAILS=STEPS provides the step information and summary table. The option DETAILS=SUMMARY produces only the summary table. The default if the DETAILS option is omitted is DETAILS=STEPS.

DW

calculates a Durbin-Watson statistic to test whether or not the errors have first-order autocorrelation. (This test is appropriate only for time series data.) The sample autocorrelation of the residuals is also produced. See the section "Autocorrelation in Time Series Data" on page 3915.

DWPROB

calculates a Durbin-Watson statistic and a p-value to test whether or not the errors have first-order autocorrelation. Note that it is not necessary to specify the DW option if the DWPROB option is specified. (This test is appropriate only for time series data.) The sample autocorrelation of the residuals is also produced. See the section "Autocorrelation in Time Series Data" on page 3915.

EDF

outputs the number of regressors in the model excluding and including the intercept, the error degrees of freedom, and the model R^2 to the OUTEST= data set.

GMSEP

outputs the estimated mean square error of prediction assuming that both independent and dependent variables are multivariate normal (Stein 1960; Darlington 1968). Note that Hocking's formula (1976, eq. 4.20) contains a misprint: "$n - 1$" should read "$n - 2$.") to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the GMSEP statistic is also added to the SubsetSelSummary table.

GROUPNAMES=*'name1' 'name2' . . .*

provides names for variable groups. This option is available only in the BACKWARD, FORWARD, and STEPWISE methods. The group name can be up to 32 characters. Subsets of independent variables listed in the MODEL statement can be designated as variable groups. This is done by enclosing the appropriate variables in braces. Variables in the same group are entered into or removed from the regression model at the same time. However, if the tolerance of any variable (see the TOL option on page 3832) in a group is less than the setting of the SINGULAR= option, then the variable is not entered into the model with the rest of its group. If the GROUPNAMES= option is not used, then the names GROUP1, GROUP2, . . ., GROUP*n* are assigned to groups encountered in the MODEL statement. Variables not enclosed by braces are used as groups of a single variable.

For example,

```
model y={x1 x2} x3 / selection=stepwise
  groupnames='x1 x2' 'x3';
```

As another example,

```
model y={ht wgt age} bodyfat / selection=forward
  groupnames='htwgtage' 'bodyfat';
```

I

displays the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix. The inverse of the crossproducts matrix is bordered by the parameter estimates and SSE matrices.

INCLUDE=*n*

forces the first *n* independent variables listed in the MODEL statement to be included in all models. The selection methods are performed on the other variables in the MODEL statement. The INCLUDE= option is not available with SELECTION=NONE.

INFLUENCE

requests a detailed analysis of the influence of each observation on the estimates and the predicted values. See the "Influence Diagnostics" section on page 3898 for details.

JP

outputs J_p, the estimated mean square error of prediction for each model selected assuming that the values of the regressors are fixed and that the model is correct to the OUTEST= data set. The J_p statistic is also called the final prediction error (FPE) by Akaike (Nicholson 1948; Lord 1950; Mallows 1967; Darlington 1968; Rothman 1968; Akaike 1969; Hocking 1976; Judge et al. 1980). If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the J_p statistic is also added to the SubsetSelSummary table.

MSE

computes the mean square error for each model selected (Darlington 1968).

MAXSTEP=*n*

specifies the maximum number of steps that are done when SELECTION=FORWARD, SELECTION=BACKWARD or SELECTION=STEPWISE is used. The default value is the number of independent variables in the model for the forward and backward methods and three times this number for the stepwise method.

NOINT

suppresses the intercept term that is otherwise included in the model.

NOPRINT

suppresses the normal display of regression results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

OUTSEB

outputs the standard errors of the parameter estimates to the OUTEST= data set. The value SEB for the variable _TYPE_ identifies the standard errors. If the RIDGE= or PCOMIT= option is specified, additional observations are included and identified by the values RIDGESEB and IPCSEB, respectively, for the variable _TYPE_. The standard errors for ridge regression estimates and incomplete principal components (IPC) estimates are limited in their usefulness because these estimates are biased. This option is available for all model-selection methods except RSQUARE, ADJRSQ, and CP.

OUTSTB

outputs the standardized parameter estimates as well as the usual estimates to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The values RIDGESTB and IPCSTB for the variable _TYPE_ identify ridge regression estimates and IPC estimates, respectively.

OUTVIF

outputs the variance inflation factors (VIF) to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The factors are the diagonal elements of the inverse of the correlation matrix of regressors as adjusted by ridge regression or IPC analysis. These observations are identified in the output data set by the values RIDGEVIF and IPCVIF for the variable _TYPE_.

P

calculates predicted values from the input data and the estimated model. The display includes the observation number, the ID variable (if one is specified), the actual and predicted values, and the residual. If the CLI, CLM, or R option is specified, the P option is unnecessary. See the section "Predicted and Residual Values" on page 3879 for more information.

PARTIAL

requests partial regression leverage plots for each regressor. If ODS Graphics are in effect (see the "ODS Graphics" section on page 3922), then these partial plots are produced in panels with up to six plots per panel. See the "Influence Diagnostics" section on page 3898 for more information.

PARTIALR2 < **(** < **TESTS** > < **SEQTESTS** > **)** >

See the SCORR1 option.

PC

outputs Amemiya's prediction criterion for each model selected (Amemiya 1976; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the PC statistic is also added to the SubsetSelSummary table.

PCOMIT=*list*

requests an IPC analysis for each value *m* in the list. The procedure computes parameter estimates using all but the last *m* principal components. Each value of *m* produces a set of IPC estimates, which is output to the OUTEST= data set. The values of *m* are saved by the variable _PCOMIT_, and the value of the variable _TYPE_ is set to IPC to identify the estimates. Only nonnegative integers can be specified with the PCOMIT= option.

If you specify the PCOMIT= option, RESTRICT statements are ignored. The PCOMIT= option is ignored if you use the SELECTION= option in the MODEL statement.

PCORR1

displays the squared partial correlation coefficients using Type I Sum of Squares (SS). This is calculated as SS/(SS+SSE), where SSE is the error Sum of Squares.

PCORR2

displays the squared partial correlation coefficients using Type II sums of squares. These are calculated the same way as with the PCORR1 option, except that Type II SS are used instead of Type I SS.

PRESS

outputs the PRESS statistic to the OUTEST= data set. The values of this statistic are saved in the variable _PRESS_. This option is available for all model-selection methods except RSQUARE, ADJRSQ, and CP.

R

requests an analysis of the residuals. The results include everything requested by the P option plus the standard errors of the mean predicted and residual values, the studentized residual, and Cook's D statistic to measure the influence of each observation on the parameter estimates. See the section "Predicted and Residual Values" on page 3879 for more information.

RIDGE=*list*

requests a ridge regression analysis and specifies the values of the ridge constant k (see the "Computations for Ridge Regression and IPC Analysis" section on page 3916). Each value of k produces a set of ridge regression estimates that are placed in the OUTEST= data set. The values of k are saved by the variable _RIDGE_, and the value of the variable _TYPE_ is set to RIDGE to identify the estimates.

Only nonnegative numbers can be specified with the RIDGE= option. Example 61.10 on page 3956 illustrates this option.

If you specify the RIDGE= option, RESTRICT statements are ignored. The RIDGE= option is ignored if you use the SELECTION= option in the MODEL statement.

RMSE

displays the root mean square error for each model selected.

RSQUARE

has the same effect as the EDF option.

SBC

outputs the SBC statistic for each model selected (Schwarz 1978; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the SBC statistic is also added to the SubsetSelSummary table.

SCORR1 $<$ **(** $<$ **TESTS** $>$ $<$ **SEQTESTS** $>$ **)** $>$

displays the squared semi-partial correlation coefficients using Type I sums of squares. This is calculated as SS/SST, where SST is the corrected total SS. If the NOINT option is used, the uncorrected total SS is used in the denominator. The optional arguments TESTS and SEQTESTS request F-tests, p-values, and cumulative R-Square values as variables are sequentially added to a model. The F-test values are computed as the Type I sum of squares for the variable in question divided by a mean square error. If you specify the TESTS option, the denominator MSE is the residual mean square for the full model specified in the MODEL statement. If you specify the SEQTESTS option, the denominator MSE is the residual mean square for the model containing all the independent variables that have been added to the model up to and including the variable in question. The TESTS and SEQTESTS options are not supported if you specify model selection methods, or the RIDGE or PCOMIT options. Note that the PARTIALR2 option is a synonym for the SCORR1 option.

SCORR2 $<$ **(TESTS)** $>$

displays the squared semi-partial correlation coefficients using Type II sums of squares. These are calculated the same way as with the SCORR1 option, except that Type II SS are used instead of Type I SS. The optional TEST argument requests F-tests and p-values as variables are sequentially added to a model. The F-test values are computed as the Type II sum of squares for the variable in question divided by the residual mean square for the full model specified in the MODEL statement. The TESTS option is not supported if you specify model selection methods, or the RIDGE or PCOMIT options.

SELECTION=*name*

specifies the method used to select the model, where *name* can be FORWARD (or F), BACKWARD (or B), STEPWISE, MAXR, MINR, RSQUARE, ADJRSQ, CP, or NONE (use the full model). The default method is NONE. See the "Model-Selection Methods" section on page 3873 for a description of each method.

SEQB

produces a sequence of parameter estimates as each variable is entered into the model. This is displayed as a matrix where each row is a set of parameter estimates.

SIGMA=*n*

specifies the true standard deviation of the error term to be used in computing the CP and BIC statistics. If the SIGMA= option is not specified, an estimate from the full model is used. This option is available in the RSQUARE, ADJRSQ, and CP model-selection methods only.

SINGULAR=*n*

tunes the mechanism used to check for singularities. Specifying this option in the MODEL statement takes precedence over the SINGULAR= option in the PROC REG statement. The default value is machine dependent but is approximately 1E$-$7 on most machines. This option is rarely needed. Singularity checking is described in the "Computational Methods" section on page 3917.

SLENTRY=*value*
SLE=*value*

specifies the significance level for entry into the model used in the FORWARD and STEPWISE methods. The defaults are 0.50 for FORWARD and 0.15 for STEPWISE.

SLSTAY=*value*
SLS=*value*

specifies the significance level for staying in the model for the BACKWARD and STEPWISE methods. The defaults are 0.10 for BACKWARD and 0.15 for STEPWISE.

SP

outputs the S_p statistic for each model selected (Hocking 1976) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the SP statistic is also added to the SubsetSelSummary table.

SPEC

performs a test that the first and second moments of the model are correctly specified. See the section "Testing for Heteroscedasticity" on page 3910 for more information.

SS1

displays the sequential sums of squares (Type I SS) along with the parameter estimates for each term in the model. See Chapter 11, "The Four Types of Estimable Functions," for more information on the different types of sums of squares.

SS2

displays the partial sums of squares (Type II SS) along with the parameter estimates for each term in the model. See the SS1 option also.

SSE

computes the error sum of squares for each model selected.

START=*s*

is used to begin the comparing-and-switching process in the MAXR, MINR, and STEPWISE methods for a model containing the first s independent variables in the MODEL statement, where s is the START value. For these methods, the default is START=0.

For the RSQUARE, ADJRSQ, and CP methods, START=s specifies the smallest number of regressors to be reported in a subset model. For these methods, the default is START=1.

The START= option cannot be used with model-selection methods other than the six described here.

STB

produces standardized regression coefficients. A standardized regression coefficient is computed by dividing a parameter estimate by the ratio of the sample standard deviation of the dependent variable to the sample standard deviation of the regressor.

STOP=*s*

causes PROC REG to stop when it has found the “best” s-variable model, where s is the STOP value. For the RSQUARE, ADJRSQ, and CP methods, STOP=s specifies the largest number of regressors to be reported in a subset model. For the MAXR and MINR methods, STOP=s specifies the largest number of regressors to be included in the model.

The default setting for the STOP= option is the number of variables in the MODEL statement. This option can be used only with the MAXR, MINR, RSQUARE, ADJRSQ and CP methods.

TOL

produces tolerance values for the estimates. Tolerance for a variable is defined as $1-R^2$, where R^2 is obtained from the regression of the variable on all other regressors in the model. See the section “Collinearity Diagnostics” on page 3895 for more detail.

VIF

produces variance inflation factors with the parameter estimates. Variance inflation is the reciprocal of tolerance. See the section “Collinearity Diagnostics” on page 3895 for more detail.

XPX

displays the $\mathbf{X}'\mathbf{X}$ crossproducts matrix for the model. The crossproducts matrix is bordered by the $\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}'\mathbf{Y}$ matrices.

MTEST Statement

< *label:* > **MTEST** < *equation* < *, . . . , equation* > > < */ options* > **;**

where each *equation* is a linear function composed of coefficients and variable names. The *label* is optional.

The MTEST statement is used to test hypotheses in multivariate regression models where there are several dependent variables fit to the same regressors. If no equations or options are specified, the MTEST statement tests the hypothesis that all estimated parameters except the intercept are zero.

The hypotheses that can be tested with the MTEST statement are of the form

$$(\mathbf{L}\beta - \mathbf{cj})\mathbf{M} = 0$$

where $\mathbf{L}$ is a linear function on the regressor side, β is a matrix of parameters, $\mathbf{c}$ is a column vector of constants, $\mathbf{j}$ is a row vector of ones, and $\mathbf{M}$ is a linear function on the dependent side. The special case where the constants are zero is

$$\mathbf{L}\beta\mathbf{M} = 0$$

See the section “Multivariate Tests” on page 3910 for more details.

Each linear function extends across either the regressor variables or the dependent variables. If the equation is across the dependent variables, then the constant term, if specified, must be zero. The equations for the regressor variables form the $\mathbf{L}$ matrix and $\mathbf{c}$ vector in the preceding formula; the equations for dependent variables form the $\mathbf{M}$ matrix. If no equations for the dependent variables are given, PROC REG uses an identity matrix for $\mathbf{M}$, testing the same hypothesis across all dependent variables. If no equations for the regressor variables are given, PROC REG forms a linear function corresponding to a test that all the nonintercept parameters are zero.

As an example, consider the following statements:

```
model y1 y2 y3=x1 x2 x3;
mtest x1,x2;
mtest y1-y2, y2 -y3, x1;
mtest y1-y2;
```

The first MTEST statement tests the hypothesis that the $X1$ and $X2$ parameters are zero for $Y1$, $Y2$ and $Y3$. In addition, the second MTEST statement tests the hypothesis that the $X1$ parameter is the same for all three dependent variables. For the same model, the third MTEST statement tests the hypothesis that all parameters except the intercept are the same for dependent variables $Y1$ and $Y2$.

You can specify the following options in the MTEST statement.

CANPRINT
displays the canonical correlations for the hypothesis combinations and the dependent variable combinations. If you specify

```
mtest / canprint;
```

the canonical correlations between the regressors and the dependent variables are displayed.

DETAILS
displays the $\mathbf{M}$ matrix and various intermediate calculations.

PRINT
displays the $\mathbf{H}$ and $\mathbf{E}$ matrices.

OUTPUT Statement

OUTPUT $<$ **OUT=***SAS-data-set* $>$ *keyword=names*
$< \ldots$ *keyword=names* $>$ **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated after fitting the model. The OUTPUT statement refers to the most recent MODEL statement. At least one *keyword=names* specification is required.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set. If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-set-name*). For more information on permanent SAS data sets, refer to the section “SAS Files” in *SAS Language Reference: Concepts*.

The OUTPUT statement cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set for PROC REG. See the “Input Data Sets” section on page 3860 for more details.

The statistics created in the OUTPUT statement are described in this section. More details are contained in the “Predicted and Residual Values” section on page 3879 and the “Influence Diagnostics” section on page 3898. Also see Chapter 2, “Introduction to Regression Procedures,” for definitions of the statistics available from the REG procedure.

You can specify the following options in the OUTPUT statement.

OUT=*SAS data set*

gives the name of the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword=names

specifies the statistics to include in the output data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable or variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains that statistic for the first dependent variable listed in the MODEL statement; the second variable contains the statistic for the second dependent variable in the MODEL statement, and so on. The list of variables following the equal sign can be shorter than the list of dependent variables in the MODEL statement. In this case, the procedure creates the new names in order of the dependent variables in the MODEL statement.

For example, the SAS statements

```
proc reg data=a;
   model y z=x1 x2;
   output out=b
      p=yhat zhat
      r=yresid zresid;
run;
```

create an output data set named b. In addition to the variables in the input data set, b contains the following variables:

- yhat, with values that are predicted values of the dependent variable y
- zhat, with values that are predicted values of the dependent variable z
- yresid, with values that are the residual values of y
- zresid, with values that are the residual values of z

You can specify the following keywords in the OUTPUT statement. See the "Model Fit and Diagnostic Statistics" section on page 3896 for computational formulas.

Table 61.3. Keywords for OUTPUT Statement

Keyword	**Description**
COOKD=*names*	Cook's D influence statistic
COVRATIO=*names*	standard influence of observation on covariance of betas, as discussed in the "Influence Diagnostics" section on page 3898
DFFITS=*names*	standard influence of observation on predicted value
H=*names*	leverage, $x_i(\mathbf{X}'\mathbf{X})^{-1}x_i'$
LCL=*names*	lower bound of a $100(1-\alpha)$% confidence interval for an individual prediction. This includes the variance of the error, as well as the variance of the parameter estimates.
LCLM=*names*	lower bound of a $100(1-\alpha)$% confidence interval for the expected value (mean) of the dependent variable
PREDICTED | P=*names*	predicted values
PRESS=*names*	ith residual divided by $(1-h)$, where h is the leverage, and where the model has been refit without the ith observation
RESIDUAL | R=*names*	residuals, calculated as ACTUAL minus PREDICTED
RSTUDENT=*names*	a studentized residual with the current observation deleted
STDI=*names*	standard error of the individual predicted value
STDP=*names*	standard error of the mean predicted value
STDR=*names*	standard error of the residual
STUDENT=*names*	studentized residuals, which are the residuals divided by their standard errors
UCL=*names*	upper bound of a $100(1-\alpha)$% confidence interval for an individual prediction
UCLM=*names*	upper bound of a $100(1-\alpha)$% confidence interval for the expected value (mean) of the dependent variable

PAINT Statement

PAINT < *condition* | **ALLOBS** > < / *options* > **;**

PAINT < **STATUS | UNDO** > **;**

The PAINT statement selects observations to be *painted* or highlighted in a scatter plot on line printer output; the PAINT statement is ignored if the LINEPRINTER option is not specified in the PROC REG statement.

All observations that satisfy *condition* are painted using some specific symbol. The PAINT statement does not generate a scatter plot and must be followed by a PLOT statement, which does generate a scatter plot. Several PAINT statements can be used before a PLOT statement, and all prior PAINT statement requests are applied to all later PLOT statements.

The PAINT statement lists the observation numbers of the observations selected, the total number of observations selected, and the plotting symbol used to paint the points.

On a plot, paint symbols take precedence over all other symbols. If any position contains more than one painted point, the paint symbol for the observation plotted last is used.

The PAINT statement cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set for PROC REG. Also, the PAINT statement cannot be used for models with more than one dependent variable. Note that the syntax for the PAINT statement is the same as the syntax for the REWEIGHT statement.

For detailed examples of painting scatter plots, see the section "Painting Scatter Plots" on page 3889.

Specifying Condition

Condition is used to select observations to be painted. The syntax of *condition* is

variable compare value

or

variable compare value logical variable compare value

where

variable is one of the following:

- a variable name in the input data set
- OBS**.**, which is the observation number
- *keyword***.**, where *keyword* is a keyword for a statistic requested in the OUTPUT statement

compare is an operator that compares *variable* to *value*. *Compare* can be any one of the following: <, <=, >, >=, =, ˆ =. The operators LT, LE, GT, GE, EQ, and NE can be used instead of the preceding symbols. Refer to the "Expressions" section in *SAS Language Reference: Concepts* for more information on comparison operators.

value gives an unformatted value of *variable*. Observations are selected to be painted if they satisfy the condition created by *variable compare value*. *Value* can be a number or a character string. If *value* is a character string, it must be eight characters or less and must be enclosed in quotes. In addition, *value* is case-sensitive. In other words, the statements

```
paint name='henry';
```

and

```
      paint name='Henry';
```

are not the same.

logical is one of two logical operators. Either AND or OR can be used. To specify AND, use AND or the symbol &. To specify OR, use OR or the symbol |.

Examples of the *variable compare value* form are

```
paint name='Henry';
paint residual.>=20;
paint obs.=99;
```

Examples of the *variable compare value* *logical* *variable compare value* form are

```
paint name='Henry'|name='Mary';
paint residual.>=20 or residual.<=10;
paint obs.>=11 and residual.<=20;
```

Using ALLOBS

Instead of specifying *condition*, the ALLOBS option can be used to select all observations. This is most useful when you want to unpaint all observations. For example,

```
paint allobs / reset;
```

resets the symbols for all observations.

Options in the PAINT Statement

The following options can be used when either a condition is specified, the ALLOBS option is specified, or when nothing is specified before the slash. If only an option is listed, the option applies to the observations selected in the previous PAINT statement, *not* to the observations selected by reapplying the condition from the previous PAINT statement. For example, in the statements

```
paint r.>0 / symbol='a';
reweight r.>0;
refit;
paint / symbol='b';
```

the second PAINT statement paints only those observations selected in the first PAINT statement. No additional observations are painted even if, after refitting the model, there are new observations that meet the condition in the first PAINT statement.

Note: Options are not available when either the UNDO or STATUS option is used.

You can specify the following options after a slash (/).

NOLIST

suppresses the display of the selected observation numbers. If the NOLIST option is not specified, a list of observations selected is written to the log. The list includes the observation numbers and painting symbol used to paint the points. The total number of observations selected to be painted is also shown.

RESET

changes the painting symbol to the current default symbol, effectively unpainting the observations selected. If you set the default symbol by using the SYMBOL= option in the PLOT statement, the RESET option in the PAINT statement changes the painting symbol to the symbol you specified. Otherwise, the default symbol of ’1’ is used.

SYMBOL = ’character’

specifies a painting symbol. If the SYMBOL= option is omitted, the painting symbol is either the one used in the most recent PAINT statement or, if there are no previous PAINT statements, the symbol ’@’. For example,

```
paint / symbol=’#’;
```

changes the painting symbol for the observations selected by the most recent PAINT statement to ’#’. As another example,

```
paint temp lt 22 / symbol=’c’;
```

changes the painting symbol to ’c’ for all observations with TEMP<22. In general, the numbers 1, 2, . . . , 9 and the asterisk are not recommended as painting symbols. These symbols are used as default symbols in the PLOT statement, where they represent the number of replicates at a point. If SYMBOL=” is used, no painting is done in the current plot. If SYMBOL=’ ’ is used, observations are painted with a blank and are no longer seen on the plot.

STATUS and UNDO

Instead of specifying *condition* or the ALLOBS option, you can use the STATUS or UNDO option as follows:

STATUS

lists (on the log) the observation number and plotting symbol of all currently painted observations.

UNDO

undoes changes made by the most recent PAINT statement. Observations may be, but are not necessarily, unpainted. For example,

```
paint obs. <=10 / symbol=’a’;
 ...other interactive statements
paint obs.=1 / symbol=’b’;
 ...other interactive statements
paint undo;
```

The last PAINT statement changes the plotting symbol used for observation 1 back to ’a’. If the statement

```
paint / reset;
```

is used instead, observation 1 is unpainted.

PLOT Statement

PLOT < *yvariable*******xvariable* >< *=symbol* >
< . . . *yvariable*******xvariable* > < *=symbol* > < */ options* >**;**

The PLOT statement in PROC REG displays scatter plots with *yvariable* on the vertical axis and *xvariable* on the horizontal axis. Line printer plots are generated if the LINEPRINTER option is specified in the PROC REG statement; otherwise, the traditional high-resolution graphics plots are created. Points in line printer plots can be marked with *symbols*, while global graphics statements such as GOPTIONS and SYMBOL are used to enhance the high-resolution graphics plots. Note that the plots you request using the PLOT statement are independent of the experimental ODS graphics (see the “ODS Graphics” section on page 3922) that are now available in PROC REG.

As with most other interactive statements, the PLOT statement implicitly refits the model. For example, if a PLOT statement is preceded by a REWEIGHT statement, the model is recomputed, and the plot reflects the new model.

If there are multiple MODEL statements preceding a PLOT statement, then the PLOT statement refers to the latest MODEL statement.

The PLOT statement cannot be used when TYPE=CORR, TYPE=COV, or TYPE=SSCP data sets are used as input to PROC REG.

You can specify several PLOT statements for each MODEL statement, and you can specify more than one plot in each PLOT statement. For detailed examples of using the PLOT statement and its options, see the section “Producing Scatter Plots” on page 3882.

Specifying Yvariables, Xvariables, and Symbol

More than one *yvariable*∗*xvariable* pair can be specified to request multiple plots. The *yvariables* and *xvariables* can be

- any variables specified in the VAR or MODEL statement before the first RUN statement
- *keyword***.**, where *keyword* is a regression diagnostic statistic available in the OUTPUT statement (see Table 61.4 on page 3842). For example,

  ```
  plot predicted.*residual.;
  ```

generates one plot of the predicted values by the residuals for each dependent variable in the MODEL statement. These statistics can also be plotted against any of the variables in the VAR or MODEL statements.

- the keyword OBS. (the observation number), which can be plotted against any of the preceding variables
- the keyword NPP. or NQQ., which can be used with any of the preceding variables to construct normal P-P or Q-Q plots, respectively (see the section "Construction of Q-Q and P-P Plots" on page 3917 and Example 61.8 on page 3953 for more information)
- keywords for model fit summary statistics available in the OUTEST= data set with _TYPE_= PARMS (see Table 61.4 on page 3842). A SELECTION= method (other than NONE) must be requested in the MODEL statement for these variables to be plotted. If one member of a *yvariable*∗*xvariable* pair is from the OUTEST= data set, the other member must also be from the OUTEST= data set.

The OUTPUT statement and the OUTEST= option are not required when their keywords are specified in the PLOT statement.

The *yvariable* and *xvariable* specifications can be replaced by a set of variables and statistics enclosed in parentheses. When this occurs, all possible combinations of *yvariable* and *xvariable* are generated. For example, the following two statements are equivalent.

```
    plot (y1 y2)*(x1 x2);
    plot y1*x1 y1*x2 y2*x1 y2*x2;
```

The statement

```
  plot;
```

is equivalent to respecifying the most recent PLOT statement without any options. However, the line printer options COLLECT, HPLOTS=, SYMBOL=, and VPLOTS=, described in the "Line Printer Plots" section on page 3848, apply across PLOT statements and remain in effect if they have been previously specified.

Options used for the traditional high-resolution graphics plots are described in the following section; see for more information.

Traditional High-Resolution Graphics Plots

The display of high-resolution graphics plots is described in the following paragraphs, the options are summarized in Table 61.4 and described in the section "Dictionary of PLOT Statement Options" on page 3844, and the "Examples" section on page 3924 contains several examples of the graphics output.

Several line printer statements and options are not supported for high-resolution graphics. In particular the PAINT statement is disabled, as are the PLOT statement options CLEAR, COLLECT, HPLOTS=, NOCOLLECT, SYMBOL=, and

VPLOTS=. To display more than one plot per page or to collect plots from multiple PLOT statements, use the PROC GREPLAY statement (refer to *SAS/GRAPH Software: Reference*). Also note that high-resolution graphics options are not recognized for line printer plots.

The fitted model equation and a label are displayed in the top margin of the plot; this display can be suppressed with the NOMODEL option. If the label is requested but cannot fit on one line, it is not displayed. The equation and label are displayed on one line when possible; if more lines are required, the label is displayed in the first line with the model equation in successive lines. If displaying the entire equation causes the plot to be unacceptably small, the equation is truncated. Table 61.5 on page 3843 lists options to control the display of the equation. The "Examples" section on page 3924 illustrates the display of the model equation.

Four statistics are displayed by default in the right margin: the number of observations, R^2, the adjusted R^2, and the root mean square error. (See Output 61.4.1 on page 3949.) The display of these statistics can be suppressed with the NOSTAT option. You can specify other options to request the display of various statistics in the right margin; see Table 61.5 on page 3843.

A default reference line at zero is displayed if residuals are plotted; see Output 61.7.1 on page 3952. If the dependent variable is plotted against the independent variable in a simple linear regression model, the fitted regression line is displayed by default. (See Output 61.4.1 on page 3949.) Default reference lines can be suppressed with the NOLINE option; the lines are not displayed if the OVERLAY option is specified.

Specialized plots are requested with special options. For each coefficient, the RIDGEPLOT option plots the ridge estimates against the ridge values *k*; see the description of the RIDGEPLOT option in the section "Dictionary of PLOT Statement Options" beginning on page 3844 and Example 61.10 on page 3956 for more details. The CONF option plots $100(1-\alpha)\%$ confidence intervals for the mean while the PRED option plots $100(1-\alpha)\%$ prediction intervals; see the description of these options in the section "Dictionary of PLOT Statement Options" beginning on page 3844 and in Example 61.9 on page 3955 for more details.

If a SELECTION= method is requested, the fitted model equation and the statistics displayed in the margin correspond to the selected model. For the ADJRSQ and CP methods, the selected model is treated as a submodel of the full model. If a CP.*NP. plot is requested, the CHOCKING= and CMALLOWS= options display model selection reference lines; see the descriptions of these options in the section "Dictionary of PLOT Statement Options" beginning on page 3844 and Example 61.5 on page 3949 for more details.

PLOT Statement *variable* Keywords

The following table lists the keywords available as PLOT statement *xvariables* and *yvariables*. All keywords have a trailing dot; for example, "*COOKD.*" requests Cook's D statistic. Neither the OUTPUT statement nor the OUTEST= option needs to be specified.

Table 61.4. Keywords for PLOT Statement *xvariables* and *yvariables*

Keyword	**Description**
Diagnostic Statistics	
COOKD.	Cook's D influence statistics
COVRATIO.	standard influence of observation on covariance of betas
DFFITS.	standard influence of observation on predicted value
H.	leverage
LCL.	lower bound of $100(1-\alpha)\%$ confidence interval for individual prediction
LCLM.	lower bound of $100(1-\alpha)\%$ confidence interval for the mean of the dependent variable
PREDICTED. | PRED. | P.	predicted values
PRESS.	residuals from refitting the model with current observation deleted
RESIDUAL. | R.	residuals
RSTUDENT.	studentized residuals with the current observation deleted
STDI.	standard error of the individual predicted value
STDP.	standard error of the mean predicted value
STDR.	standard error of the residual
STUDENT.	residuals divided by their standard errors
UCL.	upper bound of $100(1-\alpha)\%$ confidence interval for individual prediction
UCLM.	upper bound of $100(1-\alpha)\%$ confidence interval for the mean of the dependent variables
Other Keywords Used with Diagnostic Statistics	
NPP.	normal probability-probability plot
NQQ.	normal quantile-quantile plot
OBS.	observation number (cannot plot against OUTEST= statistics)
Model Fit Summary Statistics	
ADJRSQ.	adjusted R-square
AIC.	Akaike's information criterion
BIC.	Sawa's Bayesian information criterion
CP.	Mallows' C_p statistic
EDF.	error degrees of freedom
GMSEP.	estimated MSE of prediction, assuming multivariate normality
IN.	number of regressors in the model not including the intercept
JP.	final prediction error
MSE.	mean squared error
NP.	number of parameters in the model (including the intercept)
PC.	Amemiya's prediction criterion
RMSE.	root MSE
RSQ.	R-square
SBC.	SBC statistic
SP.	SP statistic
SSE.	error sum of squares

Summary of PLOT Statement Graphics Options

The following table lists the PLOT statement *options* by function. These *options* are available unless the LINEPRINTER option is specified in the PROC REG statement. For complete descriptions, see the section "Dictionary of PLOT Statement Options" beginning on page 3844.

Table 61.5. High-Resolution Graphics Options

Option	**Description**
General Graphics Options	
ANNOTATE= *SAS-data-set*	specifies the annotate data set
CHOCKING=*color*	requests a reference line for C_p model selection criteria
CMALLOWS=*color*	requests a reference line for the C_p model selection criterion
CONF	requests plots of $100(1-\alpha)$% confidence intervals for the mean
DESCRIPTION= '*string*'	specifies a description for graphics catalog member
NAME='*string*'	names the plot in graphics catalog
OVERLAY	overlays plots from the same model
PRED	requests plots of $100(1-\alpha)$% prediction intervals for individual responses
RIDGEPLOT	requests the ridge trace for ridge regression
Axis and Legend Options	
LEGEND=*LEGENDn*	specifies LEGEND statement to be used
HAXIS=*values*	specifies tick mark values for horizontal axis
VAXIS=*values*	specifies tick mark values for vertical axis
Reference Line Options	
HREF=*values*	specifies reference lines perpendicular to horizontal axis
LHREF=*linetype*	specifies line style for HREF= lines
LLINE=*linetype*	specifies line style for lines displayed by default
LVREF=*linetype*	specifies line style for VREF= lines
NOLINE	suppresses display of any default reference line
VREF=*values*	specifies reference lines perpendicular to vertical axis
Color Options	
CAXIS=*color*	specifies color for axis line and tick marks
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CLINE=*color*	specifies color for lines displayed by default
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
Options for Displaying the Fitted Model Equation	
MODELFONT=*font*	specifies font of model equation and model label
MODELHT=*value*	specifies text height of model equation and model label
MODELLAB=*'label'*	specifies model label
NOMODEL	suppresses display of the fitted model and the label
Options for Displaying Statistics in the Plot Margin	
AIC	displays Akaike's information criterion

Table 61.5. (continued)

Option	Description
BIC	displays Sawa's Bayesian information criterion
CP	displays Mallows' C_p statistic
EDF	displays the error degrees of freedom
GMSEP	displays the estimated MSE of prediction assuming multivariate normality
IN	displays the number of regressors in the model not including the intercept
JP	displays the J_p statistic
MSE	displays the mean squared error
NOSTAT	suppresses display of the default statistics: the number of observations, R-square, adjusted R-square, and the root mean square error
NP	displays the number of parameters in the model including the intercept, if any
PC	displays the PC statistic
SBC	displays the SBC statistic
SP	displays the S(p) statistic
SSE	displays the error sum of squares
STATFONT=*font*	specifies font of text displayed in the margin
STATHT=*value*	specifies height of text displayed in the margin

Dictionary of PLOT Statement Options

The following entries describe the PLOT statement *options* in detail. Note that these *options* are available unless you specify the LINEPRINTER option in the PROC REG statement.

AIC

displays Akaike's information criterion in the plot margin.

ANNOTATE=*SAS-data-set*
ANNO=*SAS-data-set*

specifies an input data set that contains appropriate variables for annotation. This applies only to displays created with the current PLOT statement. Refer to *SAS/GRAPH Software: Reference* for more information.

BIC

displays Sawa's Bayesian information criterion in the plot margin.

CAXIS=*color*
CAXES=*color*
CA=*color*

specifies the color for the axes, frame, and tick marks.

CFRAME=*color*
CFR=*color*

specifies the color for filling the area enclosed by the axes and the frame.

CHOCKING=*color*

requests reference lines corresponding to the equations $C_p = p$ and $C_p = 2p - p_{full}$, where p_{full} is the number of parameters in the full model (excluding the intercept) and p is the number of parameters in the subset model (including the intercept). The *color* must be specified; the $C_p = p$ line is solid and the $C_p = 2p - p_{full}$ line is dashed. Only PLOT statements of the form PLOT CP.*NP. produce these lines.

For the purpose of parameter estimation, Hocking (1976) suggests selecting a model where $C_p \leq 2p - p_{full}$. For the purpose of prediction, Hocking suggests the criterion $C_p \leq p$. You can request the single reference line $C_p = p$ with the CMALLOWS= option. If, for example, you specify both CHOCKING=RED and CMALLOWS=BLUE, then the $C_p = 2p - p_{full}$ line is red and the $C_p = p$ line is blue (see Example 61.5 on page 3949).

CHREF=*color*
CH=*color*

specifies the color for lines requested with the HREF= option.

CLINE=*color*
CL=*color*

specifies the color for lines displayed by default. See the NOLINE option later in this section for details.

CMALLOWS=*color*

requests a $C_p = p$ reference line, where p is the number of parameters (including the intercept) in the subset model. The *color* must be specified; the line is solid. Only PLOT statements of the form PLOT CP.*NP. produce this line.

Mallows (1973) suggests that all subset models with C_p small and near p be considered for further study. See the CHOCKING= option for related model selection criteria.

CONF

is a keyword used as a shorthand option to request plots that include $(100 - \alpha)\%$ confidence intervals for the mean response (see Example 61.9 on page 3955). The ALPHA= option in the PROC REG or MODEL statement selects the significance level α, which is 0.05 by default. The CONF option is valid for simple regression models only, and is ignored for plots where confidence intervals are inappropriate. The CONF option replaces the CONF95 option; however, the CONF95 option is still supported when the ALPHA= option is not specified. The OVERLAY option is ignored when the CONF option is specified.

CP

displays Mallows' C_p statistic in the plot margin.

CTEXT=*color*
CT=*color*

specifies the color for text including tick mark labels, axis labels, the fitted model label and equation, the statistics displayed in the margin, and legends. (See Example 61.6 on page 3950.)

CVREF=*color*
CV=*color*
specifies the color for lines requested with the VREF= option.

DESCRIPTION='*string* **'**
DESC='*string* **'**
specifies a descriptive string, up to 40 characters, that appears in the description field of the PROC GREPLAY master menu.

EDF
displays the error degrees of freedom in the plot margin.

GMSEP
displays the estimated mean square error of prediction in the plot margin. Note that the estimate is calculated under the assumption that both independent and dependent variables have a multivariate normal distribution.

HAXIS=*values*
HA=*values*
specifies tick mark values for the horizontal axis.

HREF=*values*
specifies where reference lines perpendicular to the horizontal axis are to appear.

IN
displays the number of regressors in the model (not including the intercept) in the plot margin.

JP
displays the J_p statistic in the plot margin.

LEGEND=LEGEND*n*
specifies the LEGENDn statement to be used. The LEGENDn statement is a global graphics statement; refer to *SAS/GRAPH Software: Reference* for more information.

LHREF=*linetype*
LH=*linetype*
specifies the line style for lines requested with the HREF= option. The default *linetype* is 2. Note that LHREF=1 requests a solid line. Refer to *SAS/GRAPH Software: Reference* for a table of available line types.

LLINE=*linetype*
LL=*linetype*
specifies the line style for reference lines displayed by default; see the NOLINE option for details. The default *linetype* is 2. Note that LLINE=1 requests a solid line.

LVREF=*linetype*
LV=*linetype*
specifies the line style for lines requested with the VREF= option. The default *linetype* is 2. Note that LVREF=1 requests a solid line.

MODELFONT=*font*
specifies the font used for displaying the fitted model label and the fitted model equa-

tion. Refer to *SAS/GRAPH Software: Reference* for tables of software fonts.

MODELHT=*height*

specifies the text height for the fitted model label and the fitted model equation.

MODELLAB='*label* **'**

specifies the label to be displayed with the fitted model equation. By default, no label is displayed. If the label does not fit on one line, it is not displayed. See the explanation in the section "Traditional High-Resolution Graphics Plots" beginning on page 3840 for more information.

MSE

displays the mean squared error in the plot margin.

NAME='*string* **'**

specifies a descriptive string, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default *string* is REG.

NOLINE

suppresses the display of default reference lines. A default reference line at zero is displayed if residuals are plotted. If the dependent variable is plotted against the independent variable in a simple regression model, then the fitted regression line is displayed by default. Default reference lines are not displayed if the OVERLAY option is specified.

NOMODEL

suppresses the display of the fitted model equation.

NOSTAT

suppresses the display of statistics in the plot margin. By default, the number of observations, R-square, adjusted R-square, and the root MSE are displayed.

NP

displays the number of regressors in the model including the intercept, if any, in the plot margin.

OVERLAY

overlays all plots specified in the PLOT statement from the same model on one set of axes. The variables for the first plot label the axes. The procedure automatically scales the axes to fit all of the variables unless the HAXIS= or VAXIS= option is used. Default reference lines are not displayed. A default legend is produced; the LEGEND= option can be used to customize the legend. See Example 61.11 on page 3958.

PC

displays the PC statistic in the plot margin.

PRED

is a keyword used as a shorthand option to request plots that include $(100 - \alpha)\%$ prediction intervals for individual responses (see Example 61.9 on page 3955). The ALPHA= option in the PROC REG or MODEL statement selects the significance level α, which is 0.05 by default. The PRED option is valid for simple regression models only, and is ignored for plots where prediction intervals are inappropriate.

The PRED option replaces the PRED95 option; however, the PRED95 option is still supported when the ALPHA= option is not specified. The OVERLAY option is ignored when the PRED option is specified.

RIDGEPLOT

creates overlaid plots of ridge estimates against ridge values for each coefficient. The points corresponding to the estimates of each coefficient in the plot are connected by lines. For ridge estimates to be computed and plotted, the OUTEST= option must be specified in the PROC REG statement, and the RIDGE= list must be specified in either the PROC REG or the MODEL statement. See Example 61.10 on page 3956.

SBC

displays the SBC statistic in the plot margin.

SP

displays the S_p statistic in the plot margin.

SSE

displays the error sum of squares in the plot margin.

STATFONT=*font*

specifies the font used for displaying the statistics that appear in the plot margin. Refer to *SAS/GRAPH Software: Reference* for tables of software fonts.

STATHT=*height*

specifies the text height of the statistics that appear in the plot margin.

USEALL

specifies that predicted values at data points with missing dependent variable(s) are included on appropriate plots. By default, only points used in constructing the SSCP matrix appear on plots.

VAXIS=*values*
VA=*values*

specifies tick mark values for the vertical axis.

VREF=*values*

specifies where reference lines perpendicular to the vertical axis are to appear.

Line Printer Plots

Line printer plots are requested with the LINEPRINTER option in the PROC REG statement. Points in line printer plots can be marked with *symbols*, which can be specified as a single character enclosed in quotes or the name of any variable in the input data set.

If a character variable is used for the symbol, the first (left-most) nonblank character in the formatted value of the variable is used as the plotting symbol. If a character in quotes is specified, that character becomes the plotting symbol. If a character is used as the plotting symbol, and if there are different plotting symbols needed at the same point, the symbol '?' is used at that point.

If an unformatted numeric variable is used for the symbol, the symbols '1', '2', . . . , '9' are used for variable values 1, 2, . . . , 9. For noninteger values, only the integer portion is used as the plotting symbol. For values of 10 or greater, the symbol '*' is used. For negative values, a '?' is used. If a numeric variable is used, and if there is more than one plotting symbol needed at the same point, the sum of the variable values is used at that point. If the sum exceeds 9, the symbol '*' is used.

If a symbol is not specified, the number of replicates at the point is displayed. The symbol '*' is used if there are ten or more replicates.

If the LINEPRINTER option is used, you can specify the following options in the PLOT statement after a slash (/):

CLEAR

clears any collected scatter plots before plotting begins but does not turn off the COLLECT option. Use this option when you want to begin a new collection with the plots in the current PLOT statement. For more information on collecting plots, see the COLLECT and NOCOLLECT options in this section.

COLLECT

specifies that plots begin to be collected from one PLOT statement to the next and that subsequent plots show an overlay of all collected plots. This option enables you to overlay plots before and after changes to the model or to the data used to fit the model. Plots collected before changes are unaffected by the changes and can be overlaid on later plots. You can request more than one plot with this option, and you do not need to request the same number of plots in subsequent PLOT statements. If you specify an unequal number of plots, plots in corresponding positions are overlaid. For example, the statements

```
plot residual.*predicted. y*x / collect;
run;
```

produce two plots. If these statements are then followed by

```
plot residual.*x;
run;
```

two plots are again produced. The first plot shows residual against X values overlaid on residual against predicted values. The second plot is the same as that produced by the first PLOT statement.

Axes are scaled for the first plot or plots collected. The axes are not rescaled as more plots are collected.

Once specified, the COLLECT option remains in effect until the NOCOLLECT option is specified.

HPLOTS=*number*

sets the number of scatter plots that can be displayed across the page. The procedure begins with one plot per page. The value of the HPLOTS= option remains in effect until you change it in a later PLOT statement. See the VPLOTS= option for an example.

NOCOLLECT

specifies that the collection of scatter plots ends after adding the plots in the current PLOT statement. PROC REG starts with the NOCOLLECT option in effect. After you specify the NOCOLLECT option, any following PLOT statement produces a new plot that contains only the plots requested by that PLOT statement.

For more information, see the COLLECT option.

OVERLAY

allows requested scatter plots to be superimposed. The axes are scaled so that points on all plots are shown. If the HPLOTS= or VPLOTS= option is set to more than one, the overlaid plot occupies the first position on the page. The OVERLAY option is similar to the COLLECT option in that both options produce superimposed plots. However, OVERLAY superimposes only the plots in the associated PLOT statement; COLLECT superimposes plots across PLOT statements. The OVERLAY option can be used when the COLLECT option is in effect.

SYMBOL='*character*'

changes the default plotting symbol used for all scatter plots produced in the current and in subsequent PLOT statements. Both SYMBOL=” and SYMBOL=’ ’ are allowed.

If the SYMBOL= option has not been specified, the default symbol is ’1’ for positions with one observation, ’2’ for positions with two observations, and so on. For positions with more than 9 observations, ’*’ is used. The SYMBOL= option (or a plotting symbol) is needed to avoid any confusion caused by this default convention. Specifying a particular symbol is especially important when either the OVERLAY or COLLECT option is being used.

If you specify the SYMBOL= option and use a number for *character*, that number is used for all points in the plot. For example, the statement

```
plot y*x / symbol='1';
```

produces a plot with the symbol ’1’ used for all points.

If you specify a plotting symbol and the SYMBOL= option, the plotting symbol overrides the SYMBOL= option. For example, in the statements

```
plot y*x y*v='.' / symbol='*';
```

the symbol used for the plot of Y against X is ’*’, and a ’.’ is used for the plot of Y against V.

If a paint symbol is defined with a PAINT statement, the paint symbol takes precedence over both the SYMBOL= option and the default plotting symbol for the PLOT statement.

VPLOTS=*number*

sets the number of scatter plots that can be displayed down the page. The procedure begins with one plot per page. The value of the VPLOTS= option remains in effect until you change it in a later PLOT statement.

For example, to specify a total of six plots per page, with two rows of three plots, use the HPLOTS= and VPLOTS= options as follows:

```
plot y1*x1 y1*x2 y1*x3 y2*x1 y2*x2 y2*x3 /
     hplots=3 vplots=2;
run;
```

PRINT Statement

PRINT < *options* > < **ANOVA** > < **MODELDATA** > **;**

The PRINT statement enables you to interactively display the results of MODEL statement options, produce an ANOVA table, display the data for variables used in the current model, or redisplay the options specified in a MODEL or a previous PRINT statement. In addition, like most other interactive statements in PROC REG, the PRINT statement implicitly refits the model; thus, effects of REWEIGHT statements are seen in the resulting tables. If the experimental ODS graphics are in effect (see the "ODS Graphics" section on page 3922), the PRINT statement also requests the display of the ODS graphics associated with the current model.

The following specifications can appear in the PRINT statement:

options interactively displays the results of MODEL statement options, where *options* is one or more of the following: ACOV, ALL, CLI, CLM, COLLIN, COLLINOINT, CORRB, COVB, DW, I, INFLUENCE, P, PARTIAL, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, or XPX. See the "MODEL Statement" section on page 3821 for a description of these options.

ANOVA produces the ANOVA table associated with the current model. This is either the model specified in the last MODEL statement or the model that incorporates changes made by ADD, DELETE or REWEIGHT statements after the last MODEL statement.

MODELDATA displays the data for variables used in the current model.

Use the statement

```
print;
```

to reprint options in the most recently specified PRINT or MODEL statement.

Options that require original data values, such as R or INFLUENCE, cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set to PROC REG. See the "Input Data Sets" section on page 3860 for more detail.

REFIT Statement

REFIT;

The REFIT statement causes the current model and corresponding statistics to be recomputed immediately. No output is generated by this statement. The REFIT statement is needed after one or more REWEIGHT statements to cause them to take effect before subsequent PAINT or REWEIGHT statements. This is sometimes necessary when you are using statistical conditions in REWEIGHT statements. For example, with these statements

```
paint student.>2;
plot student.*p.;
reweight student.>2;
refit;
paint student.>2;
plot student.*p.;
```

the second PAINT statement paints any additional observations that meet the condition after deleting observations and refitting the model. The REFIT statement is used because the REWEIGHT statement does not cause the model to be recomputed. In this particular example, the same effect could be achieved by replacing the REFIT statement with a PLOT statement.

Most interactive statements can be used to implicitly refit the model; any plots or statistics produced by these statements reflect changes made to the model and changes made to the data used to compute the model. The two exceptions are the PAINT and REWEIGHT statements, which do not cause the model to be recomputed.

RESTRICT Statement

RESTRICT *equation* < , . . . , *equation* > **;**

A RESTRICT statement is used to place restrictions on the parameter estimates in the MODEL preceding it. More than one RESTRICT statement can follow each MODEL statement. Each RESTRICT statement replaces any previous RESTRICT statement. To lift all restrictions on a model, submit a new MODEL statement. If there are several restrictions, separate them with commas. The statement

```
restrict equation1=equation2=equation3;
```

is equivalent to imposing the two restrictions

```
restrict equation1=equation2;
restrict equation2=equation3;
```

Each restriction is written as a linear equation and can be written as

equation

or

equation = equation

The form of each *equation* is

$$c_1 \times variable_1 \pm c_2 \times variable_2 \pm \cdots \pm c_n \times variable_n$$

where the c_j's are constants and the *variable*$_j$'s are any regressor variables.

When no equal sign appears, the linear combination is set equal to zero. Each variable name mentioned must be a variable in the MODEL statement to which the RESTRICT statement refers. The keyword INTERCEPT can also be used as a variable name, and it refers to the intercept parameter in the regression model.

Note that the parameters associated with the variables are restricted, not the variables themselves. Restrictions should be consistent and not redundant.

Examples of valid RESTRICT statements include the following:

```
restrict x1;
restrict a+b=l;
restrict a=b=c;
restrict a=b, b=c;
restrict 2*f=g+h, intercept+f=0;
restrict f=g=h=intercept;
```

The third and fourth statements in this list produce identical restrictions. You cannot specify

```
restrict f-g=0,
         f-intercept=0,
         g-intercept=1;
```

because the three restrictions are not consistent. If these restrictions are included in a RESTRICT statement, one of the restrict parameters is set to zero and has zero degrees of freedom, indicating that PROC REG is unable to apply a restriction.

The restrictions usually operate even if the model is not of full rank. Check to ensure that DF$= -1$ for each restriction. In addition, the Model DF should decrease by 1 for each restriction.

The parameter estimates are those that minimize the quadratic criterion (SSE) subject to the restrictions. If a restriction cannot be applied, its parameter value and degrees of freedom are listed as zero.

The method used for restricting the parameter estimates is to introduce a Lagrangian parameter for each restriction (Pringle and Raynor 1971). The estimates of these parameters are displayed with test statistics. Note that the t statistic reported for the Lagrangian parameters does not follow a Student's t distribution, but its square follows a beta distribution (LaMotte 1994). The p-value for these parameters is computed using the beta distribution.

The Lagrangian parameter γ measures the sensitivity of the SSE to the restriction constant. If the restriction constant is changed by a small amount ϵ, the SSE is changed by $2\gamma\epsilon$. The t ratio tests the significance of the restrictions. If γ is zero, the restricted estimates are the same as the unrestricted estimates, and a change in the restriction constant in either direction increases the SSE.

RESTRICT statements are ignored if the PCOMIT= or RIDGE= option is specified in the PROC REG statement.

REWEIGHT Statement

REWEIGHT < *condition* | **ALLOBS** > < / *options* > **;**

REWEIGHT < **STATUS** | **UNDO** > **;**

The REWEIGHT statement interactively changes the weights of observations that are used in computing the regression equation. The REWEIGHT statement can change observation weights, or set them to zero, which causes selected observations to be excluded from the analysis. When a REWEIGHT statement sets observation weights to zero, the observations are not deleted from the data set. More than one REWEIGHT statement can be used. The requests from all REWEIGHT statements are applied to the subsequent statements. Each use of the REWEIGHT statement modifies the MODEL label.

The model and corresponding statistics are not recomputed after a REWEIGHT statement. For example, with the following statements

```
reweight r.>0;
reweight r.>0;
```

the second REWEIGHT statement does not exclude any additional observations since the model is not recomputed after the first REWEIGHT statement. Use either a REFIT statement to explicitly refit the model, or implicitly refit the model by following the REWEIGHT statement with any other interactive statement except a PAINT statement or another REWEIGHT statement.

The REWEIGHT statement cannot be used if a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as an input data set to PROC REG. Note that the syntax used in the REWEIGHT statement is the same as the syntax in the PAINT statement.

The syntax of the REWEIGHT statement is described in the following sections. For detailed examples of using this statement see the section "Reweighting Observations in an Analysis" on page 3903.

Specifying Condition

Condition is used to find observations to be reweighted. The syntax of *condition* is

variable compare value

or

variable compare value *logical* *variable compare value*

where

variable is one of the following:

- a variable name in the input data set
- OBS**.** which is the observation number
- *keyword* **.**, where *keyword* is a keyword for a statistic requested in the OUTPUT statement. The keyword specification is applied to all dependent variables in the model.

compare is an operator that compares *variable* to *value*. *Compare* can be any one of the following: <, <=, >, >=, =, ˆ =. The operators LT, LE, GT, GE, EQ, and NE can be used instead of the preceding symbols. Refer to the “Expressions” chapter in *SAS Language Reference: Concepts* for more information on comparison operators.

value gives an unformatted value of *variable*. Observations are selected to be reweighted if they satisfy the condition created by *variable compare value*. *Value* can be a number or a character string. If *value* is a character string, it must be eight characters or less and must be enclosed in quotes. In addition, *value* is case-sensitive. In other words, the following two statements are not the same:

```
reweight name='steve';

reweight name='Steve';
```

logical is one of two logical operators. Either AND or OR can be used. To specify AND, use AND or the symbol &. To specify OR, use OR or the symbol |.

Examples of the *variable compare value* form are

```
reweight obs. le 10;
reweight temp=55;
reweight type='new';
```

Examples of the *variable compare value* *logical* *variable compare value* form are

```
reweight obs.<=10 and residual.<2;
reweight student.<-2 or student.>2;
reweight name='Mary' | name='Susan';
```

Using ALLOBS

Instead of specifying *condition*, you can use the ALLOBS option to select all observations. This is most useful when you want to restore the original weights of all observations. For example,

```
reweight allobs / reset;
```

resets weights for all observations and uses all observations in the subsequent analysis. Note that

```
reweight allobs;
```

specifies that all observations be excluded from analysis. Consequently, using ALLOBS is useful only if you also use one of the options discussed in the following section.

Options in the REWEIGHT Statement

The following options can be used when either a condition, ALLOBS, or nothing is specified before the slash. If only an option is listed, the option applies to the observations selected in the previous REWEIGHT statement, not to the observations selected by reapplying the condition from the previous REWEIGHT statement. For example, with the statements

```
reweight r.>0 / weight=0.1;
refit;
reweight;
```

the second REWEIGHT statement excludes from the analysis only those observations selected in the first REWEIGHT statement. No additional observations are excluded even if there are new observations that meet the condition in the first REWEIGHT statement.

Note: Options are not available when either the UNDO or STATUS option is used.

NOLIST
suppresses the display of the selected observation numbers. If you omit the NOLIST option, a list of observations selected is written to the log.

RESET
resets the observation weights to their original values as defined by the WEIGHT statement or to WEIGHT=1 if no WEIGHT statement is specified. For example,

```
reweight / reset;
```

resets observation weights to the original weights in the data set. If previous REWEIGHT statements have been submitted, this REWEIGHT statement applies only to the observations selected by the previous REWEIGHT statement. Note that, although the RESET option does reset observation weights to their original values, it does not cause the model and corresponding statistics to be recomputed.

WEIGHT=*value*

changes observation weights to the specified nonnegative real number. If you omit the WEIGHT= option, the observation weights are set to zero, and observations are excluded from the analysis. For example,

```
reweight name='Alan';
 ...other interactive statements
reweight / weight=0.5;
```

The first REWEIGHT statement changes weights to zero for all observations with name='Alan', effectively deleting these observations. The subsequent analysis does not include these observations. The second REWEIGHT statement applies only to those observations selected by the previous REWEIGHT statement, and it changes the weights to 0.5 for all the observations with NAME='Alan'. Thus, the next analysis includes all original observations; however, those observations with NAME='Alan' have their weights set to 0.5.

STATUS and UNDO

If you omit *condition* and the ALLOBS options, you can specify one of the following options.

STATUS

writes to the log the observation's number and the weight of all reweighted observations. If an observation's weight has been set to zero, it is reported as deleted. However, the observation is not deleted from the data set, only from the analysis.

UNDO

undoes the changes made by the most recent REWEIGHT statement. Weights may be, but are not necessarily, reset. For example, in these statements

```
reweight student.>2 / weight=0.1;
reweight;
reweight undo;
```

the first REWEIGHT statement sets the weights of observations that satisfy the condition to 0.1. The second REWEIGHT statement sets the weights of the same observations to zero. The third REWEIGHT statement undoes the second, changing the weights back to 0.1.

TEST Statement

< *label:* > **TEST** *equation* < *, . . . , equation* > < */ options* > **;**

The TEST statement tests hypotheses about the parameters estimated in the preceding MODEL statement. It has the same syntax as the RESTRICT statement except that it allows an option. Each equation specifies a linear hypothesis to be tested. The rows of the hypothesis are separated by commas.

Variable names must correspond to regressors, and each variable name represents the coefficient of the corresponding variable in the model. An optional label is useful to identify each test with a name. The keyword INTERCEPT can be used instead of a variable name to refer to the model's intercept.

The REG procedure performs an F test for the joint hypotheses specified in a single TEST statement. More than one TEST statement can accompany a MODEL statement. The numerator is the usual quadratic form of the estimates; the denominator is the mean squared error. If hypotheses can be represented by

$$\mathbf{L}\beta = \mathbf{c}$$

then the numerator of the F test is

$$\mathbf{Q} = (\mathbf{Lb} - \mathbf{c})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb} - \mathbf{c})$$

divided by degrees of freedom, where $\mathbf{b}$ is the estimate of β. For example,

```
model y=a1 a2 b1 b2;
aplus: test a1+a2=1;
b1:    test b1=0, b2=0;
b2:    test b1, b2;
```

The last two statements are equivalent; since no constant is specified, zero is assumed.

Note that, when the ACOV option is specified in the MODEL statement, tests are recomputed using the heteroscedasticity consistent covariance matrix (see the section "Testing for Heteroscedasticity" on page 3910).

One option can be specified in the TEST statement after a slash (/):

PRINT

displays intermediate calculations. This includes $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$ bordered by $\mathbf{Lb} - \mathbf{c}$, and $(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}$ bordered by $(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb} - \mathbf{c})$.

VAR Statement

VAR *variables* **;**

The VAR statement is used to include numeric variables in the crossproducts matrix that are not specified in the first MODEL statement.

Variables not listed in MODEL statements before the first RUN statement must be listed in the VAR statement if you want the ability to add them interactively to the model with an ADD statement, to include them in a new MODEL statement, or to plot them in a scatter plot with the PLOT statement.

In addition, if you want to use options in the PROC REG statement and do not want to fit a model to the data (with a MODEL statement), you must use a VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set with values that are relative weights for a weighted least-squares fit. If the weight value is proportional to the reciprocal of the variance for each observation, then the weighted estimates are the best linear unbiased estimates (BLUE).

Values of the weight variable must be nonnegative. If an observation's weight is zero, the observation is deleted from the analysis. If a weight is negative or missing, it is set to zero, and the observation is excluded from the analysis. A more complete description of the WEIGHT statement can be found in Chapter 32, "The GLM Procedure."

Observation weights can be changed interactively with the REWEIGHT statement; see the section "REWEIGHT Statement" beginning on page 3854.

Details

Missing Values

PROC REG constructs only one crossproducts matrix for the variables in all regressions. If any variable needed for any regression is missing, the observation is excluded from all estimates. If you include variables with missing values in the VAR statement, the corresponding observations are excluded from all analyses, even if you never include the variables in a model. PROC REG assumes that you may want to include these variables after the first RUN statement and deletes observations with missing values.

Input Data Sets

PROC REG does not compute new regressors. For example, if you want a quadratic term in your model, you should create a new variable when you prepare the input data. For example, the statement

```
model y=x1 x1*x1;
```

is not valid. Note that this MODEL statement is valid in the GLM procedure.

The input data set for most applications of PROC REG contains standard rectangular data, but special TYPE=CORR, TYPE=COV, or TYPE=SSCP data sets can also be used. TYPE=CORR and TYPE=COV data sets created by the CORR procedure contain means and standard deviations. In addition, TYPE=CORR data sets contain correlations and TYPE=COV data sets contain covariances. TYPE=SSCP data sets created in previous runs of PROC REG that used the OUTSSCP= option contain the sums of squares and crossproducts of the variables. See Appendix A, "Special SAS Data Sets," and the "SAS Files" section in *SAS Language Reference: Concepts* for more information on special SAS data sets.

These summary files save CPU time. It takes nk^2 operations (where n=number of observations and k=number of variables) to calculate crossproducts; the regressions are of the order k^3. When n is in the thousands and k is less than 10, you can save 99 percent of the CPU time by reusing the SSCP matrix rather than recomputing it.

When you want to use a special SAS data set as input, PROC REG must determine the TYPE for the data set. PROC CORR and PROC REG automatically set the type for their output data sets. However, if you create the data set by some other means (such as a DATA step) you must specify its type with the TYPE= data set option. If the TYPE for the data set is not specified when the data set is created, you can specify TYPE= as a data set option in the DATA= option in the PROC REG statement. For example,

```
proc reg data=a(type=corr);
```

When TYPE=CORR, TYPE=COV, or TYPE=SSCP data sets are used with PROC REG, statements and options that require the original data values have no effect. The OUTPUT, PAINT, PLOT, and REWEIGHT statements and the MODEL and PRINT statement options P, R, CLM, CLI, DW, INFLUENCE, and PARTIAL are disabled since the original observations needed to calculate predicted and residual values are not present.

Example Using TYPE=CORR Data Set

This example uses PROC CORR to produce an input data set for PROC REG. The fitness data for this analysis can be found in Example 61.1 on page 3924.

```
proc corr data=fitness outp=r noprint;
   var Oxygen RunTime Age Weight RunPulse MaxPulse RestPulse;
```

```
proc print data=r;
proc reg data=r;
   model Oxygen=RunTime Age Weight;
run;
```

Since the OUTP= data set from PROC CORR is automatically set to TYPE=CORR, the TYPE= data set option is not required in this example. The data set containing the correlation matrix is displayed by the PRINT procedure as shown in Figure 61.12. Figure 61.13 shows results from the regression using the TYPE=CORR data as an input data set.

```
                                                                                  Rest
 Obs   _TYPE_   _NAME_        Oxygen    RunTime        Age     Weight  RunPulse  MaxPulse     Pulse

   1    MEAN                 47.3758    10.5861    47.6774    77.4445   169.645   173.774   53.4516
   2    STD                   5.3272     1.3874     5.2114     8.3286    10.252     9.164    7.6194
   3    N                    31.0000    31.0000    31.0000    31.0000    31.000    31.000   31.0000
   4    CORR    Oxygen        1.0000    -0.8622    -0.3046    -0.1628    -0.398    -0.237   -0.3994
   5    CORR    RunTime      -0.8622     1.0000     0.1887     0.1435     0.314     0.226    0.4504
   6    CORR    Age          -0.3046     0.1887     1.0000    -0.2335    -0.338    -0.433   -0.1641
   7    CORR    Weight       -0.1628     0.1435    -0.2335     1.0000     0.182     0.249    0.0440
   8    CORR    RunPulse     -0.3980     0.3136    -0.3379     0.1815     1.000     0.930    0.3525
   9    CORR    MaxPulse     -0.2367     0.2261    -0.4329     0.2494     0.930     1.000    0.3051
  10    CORR    RestPulse    -0.3994     0.4504    -0.1641     0.0440     0.352     0.305    1.0000
```

Figure 61.12. TYPE=CORR Data Set Created by PROC CORR

```
                            The REG Procedure
                              Model: MODEL1
                       Dependent Variable: Oxygen

                          Analysis of Variance

                                         Sum of           Mean
Source                     DF           Squares         Square    F Value    Pr > F

Model                       3         656.27095      218.75698      30.27    <.0001
Error                      27         195.11060        7.22632
Corrected Total            30         851.38154


             Root MSE              2.68818    R-Square     0.7708
             Dependent Mean       47.37581    Adj R-Sq     0.7454
             Coeff Var             5.67416


                          Parameter Estimates

                            Parameter       Standard
     Variable      DF        Estimate          Error    t Value    Pr > |t|

     Intercept      1        93.12615        7.55916      12.32      <.0001
     RunTime        1        -3.14039        0.36738      -8.55      <.0001
     Age            1        -0.17388        0.09955      -1.75      0.0921
     Weight         1        -0.05444        0.06181      -0.88      0.3862
```

Figure 61.13. Regression on TYPE=CORR Data Set

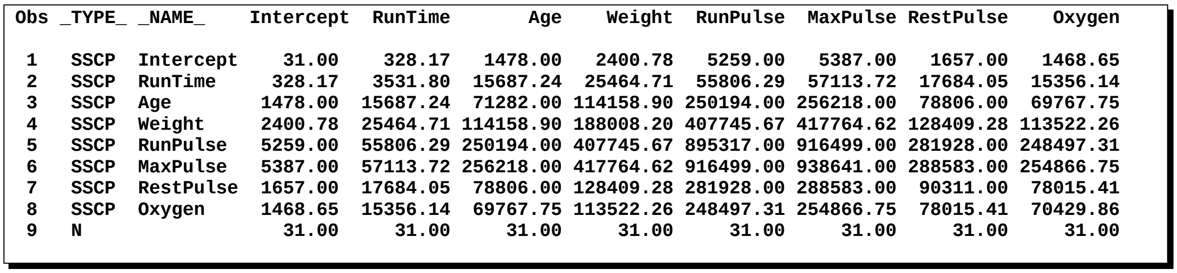
Example Using TYPE=SSCP Data Set

The following example uses the saved crossproducts matrix:

```
proc reg data=fitness outsscp=sscp noprint;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse;
proc print data=sscp;
proc reg data=sscp;
   model Oxygen=RunTime Age Weight;
run;
```

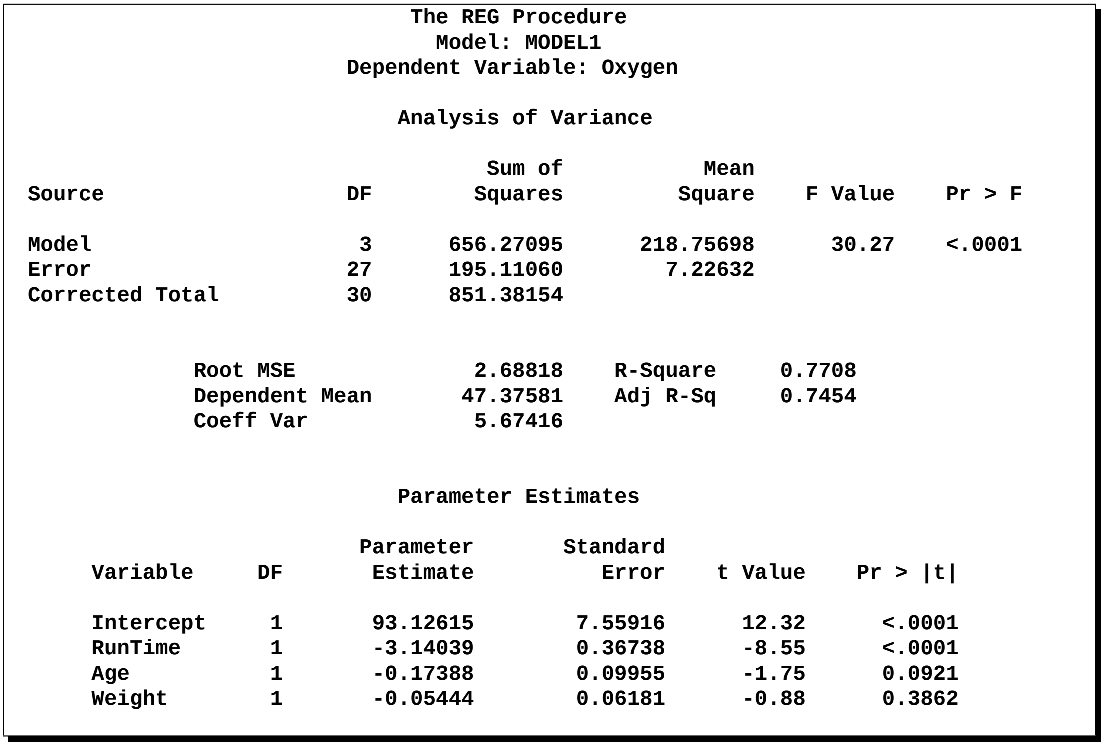
First, all variables are used to fit the data and create the SSCP data set. Figure 61.14 shows the PROC PRINT display of the SSCP data set. The SSCP data set is then used as the input data set for PROC REG, and a reduced model is fit to the data. Figure 61.15 also shows the PROC REG results for the reduced model. (For the PROC REG results for the full model, see Figure 61.27 on page 3877.)

In the preceding example, the TYPE= data set option is not required since PROC REG sets the OUTSSCP= data set to TYPE=SSCP.



```
Obs _TYPE_ _NAME_    Intercept  RunTime       Age    Weight  RunPulse  MaxPulse RestPulse    Oxygen

 1   SSCP  Intercept    31.00    328.17   1478.00   2400.78   5259.00   5387.00   1657.00   1468.65
 2   SSCP  RunTime     328.17   3531.80  15687.24  25464.71  55806.29  57113.72  17684.05  15356.14
 3   SSCP  Age        1478.00  15687.24  71282.00 114158.90 250194.00 256218.00  78806.00  69767.75
 4   SSCP  Weight     2400.78  25464.71 114158.90 188008.20 407745.67 417764.62 128409.28 113522.26
 5   SSCP  RunPulse   5259.00  55806.29 250194.00 407745.67 895317.00 916499.00 281928.00 248497.31
 6   SSCP  MaxPulse   5387.00  57113.72 256218.00 417764.62 916499.00 938641.00 288583.00 254866.75
 7   SSCP  RestPulse  1657.00  17684.05  78806.00 128409.28 281928.00 288583.00  90311.00  78015.41
 8   SSCP  Oxygen     1468.65  15356.14  69767.75 113522.26 248497.31 254866.75  78015.41  70429.86
 9   N                  31.00     31.00     31.00     31.00     31.00     31.00     31.00     31.00
```

Figure 61.14. TYPE=SSCP Data Set Created by PROC CORR



```
                          The REG Procedure
                            Model: MODEL1
                     Dependent Variable: Oxygen

                         Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                     3      656.27095      218.75698      30.27    <.0001
Error                    27      195.11060        7.22632
Corrected Total          30      851.38154


             Root MSE              2.68818    R-Square     0.7708
             Dependent Mean       47.37581    Adj R-Sq     0.7454
             Coeff Var             5.67416


                         Parameter Estimates

                           Parameter       Standard
     Variable     DF        Estimate          Error    t Value    Pr > |t|

     Intercept     1        93.12615        7.55916      12.32      <.0001
     RunTime       1        -3.14039        0.36738      -8.55      <.0001
     Age           1        -0.17388        0.09955      -1.75      0.0921
     Weight        1        -0.05444        0.06181      -0.88      0.3862
```

Figure 61.15. Regression on TYPE=SSCP Data Set

Output Data Sets

OUTEST= Data Set

The OUTEST= specification produces a TYPE=EST output SAS data set containing estimates and optional statistics from the regression models. For each BY group on each dependent variable occurring in each MODEL statement, PROC REG outputs an observation to the OUTEST= data set. The variables output to the data set are as follows:

- the BY variables, if any
- _MODEL_, a character variable containing the label of the corresponding MODEL statement, or MODELn if no label is specified, where n is 1 for the first MODEL statement, 2 for the second model statement, and so on
- _TYPE_, a character variable with the value 'PARMS' for every observation
- _DEPVAR_, the name of the dependent variable
- _RMSE_, the root mean squared error or the estimate of the standard deviation of the error term
- Intercept, the estimated intercept, unless the NOINT option is specified
- all the variables listed in any MODEL or VAR statement. Values of these variables are the estimated regression coefficients for the model. A variable

that does not appear in the model corresponding to a given observation has a missing value in that observation. The dependent variable in each model is given a value of -1.

If you specify the COVOUT option, the covariance matrix of the estimates is output after the estimates; the _TYPE_ variable is set to the value ’COV’ and the names of the rows are identified by the 8-byte character variable, _NAME_.

If you specify the TABLEOUT option, the following statistics listed by _TYPE_ are added after the estimates:

- STDERR, the standard error of the estimate
- T, the t statistic for testing if the estimate is zero
- PVALUE, the associated p-value
- LnB, the $100(1-\alpha)$ lower confidence for the estimate, where n is the nearest integer to $100(1-\alpha)$ and α defaults to 0.05 or is set using the ALPHA= option in the PROC REG or MODEL statement
- UnB, the $100(1-\alpha)$ upper confidence for the estimate

Specifying the option ADJRSQ, AIC, BIC, CP, EDF, GMSEP, JP, MSE, PC, RSQUARE, SBC, SP, or SSE in the PROC REG or MODEL statement automatically outputs these statistics and the model R^2 for each model selected, regardless of the model selection method. Additional variables, in order of occurrence, are as follows.

- _IN_, the number of regressors in the model not including the intercept
- _P_, the number of parameters in the model including the intercept, if any
- _EDF_, the error degrees of freedom
- _SSE_, the error sum of squares, if the SSE option is specified
- _MSE_, the mean squared error, if the MSE option is specified
- _RSQ_, the R^2 statistic
- _ADJRSQ_, the adjusted R^2, if the ADJRSQ option is specified
- _CP_, the C_p statistic, if the CP option is specified
- _SP_, the S_p statistic, if the SP option is specified
- _JP_, the J_p statistic, if the JP option is specified
- _PC_, the PC statistic, if the PC option is specified
- _GMSEP_, the GMSEP statistic, if the GMSEP option is specified
- _AIC_, the AIC statistic, if the AIC option is specified
- _BIC_, the BIC statistic, if the BIC option is specified
- _SBC_, the SBC statistic, if the SBC option is specified

The following is an example with a display of the OUTEST= data set. This example uses the population data given in the section "Polynomial Regression" beginning on page 3804. Figure 61.16 on page 3865 through Figure 61.18 on page 3866 show the regression equations and the resulting OUTEST= data set.

```
proc reg data=USPopulation outest=est;
   m1: model Population=Year;
   m2: model Population=Year YearSq;
proc print data=est;
run;
```

```
                                 The REG Procedure
                                     Model: m1
                          Dependent Variable: Population

                               Analysis of Variance

                                          Sum of           Mean
Source                        DF         Squares         Square    F Value    Pr > F

Model                          1          146869         146869     228.92    <.0001
Error                         20           12832      641.58160
Corrected Total               21          159700


             Root MSE              25.32946    R-Square     0.9197
             Dependent Mean        94.64800    Adj R-Sq     0.9156
             Coeff Var             26.76175


                               Parameter Estimates

                               Parameter        Standard
    Variable       DF           Estimate           Error    t Value    Pr > |t|

    Intercept       1        -2345.85498       161.39279     -14.54      <.0001
    Year            1            1.28786         0.08512      15.13      <.0001
```

Figure 61.16. Regression Output for Model M1

```
                          The REG Procedure
                              Model: m2
                   Dependent Variable: Population

                         Analysis of Variance

                                      Sum of           Mean
Source                   DF          Squares         Square    F Value    Pr > F

Model                     2           159529          79765    8864.19    <.0001
Error                    19        170.97193        8.99852
Corrected Total          21           159700


             Root MSE              2.99975    R-Square     0.9989
             Dependent Mean       94.64800    Adj R-Sq     0.9988
             Coeff Var             3.16938


                         Parameter Estimates

                             Parameter       Standard
    Variable      DF          Estimate          Error    t Value    Pr > |t|

    Intercept      1             21631      639.50181      33.82      <.0001
    Year           1         -24.04581        0.67547     -35.60      <.0001
    YearSq         1           0.00668     0.00017820      37.51      <.0001
```

Figure 61.17. Regression Output for Model M2

```
Obs _MODEL_ _TYPE_  _DEPVAR_   _RMSE_ Intercept   Year   Population     YearSq

 1    m1    PARMS  Population 25.3295  -2345.85   1.2879     -1       .
 2    m2    PARMS  Population  2.9998  21630.89 -24.0458     -1      .006684346
```

Figure 61.18. OUTEST= Data Set

The following modification of the previous example uses the TABLEOUT and ALPHA= options to obtain additional information in the OUTEST= data set:

```
proc reg data=USPopulation outest=est tableout alpha=0.1;
   m1: model Population=Year/noprint;
   m2: model Population=Year YearSq/noprint;
proc print data=est;
run;
```

Notice that the TABLEOUT option causes standard errors, t statistics, p-values, and confidence limits for the estimates to be added to the OUTEST= data set. Also note that the ALPHA= option is used to set the confidence level at 90%. The OUTEST= data set follows.

```
Obs _MODEL_ _TYPE_   _DEPVAR_    _RMSE_ Intercept      Year Population  YearSq

  1   m1    PARMS  Population 25.3295  -2345.85   1.2879      -1        .
  2   m1    STDERR Population 25.3295    161.39   0.0851       .        .
  3   m1    T      Population 25.3295    -14.54  15.1300       .        .
  4   m1    PVALUE Population 25.3295      0.00   0.0000       .        .
  5   m1    L90B   Population 25.3295  -2624.21   1.1411       .        .
  6   m1    U90B   Population 25.3295  -2067.50   1.4347       .        .
  7   m2    PARMS  Population  2.9998  21630.89 -24.0458      -1      0.0067
  8   m2    STDERR Population  2.9998    639.50   0.6755       .      0.0002
  9   m2    T      Population  2.9998     33.82 -35.5988       .     37.5096
 10   m2    PVALUE Population  2.9998      0.00   0.0000       .      0.0000
 11   m2    L90B   Population  2.9998  20525.11 -25.2138       .      0.0064
 12   m2    U90B   Population  2.9998  22736.68 -22.8778       .      0.0070
```

Figure 61.19. The OUTEST= Data Set When TABLEOUT is Specified

A slightly different OUTEST= data set is created when you use the RSQUARE selection method. This example requests only the “best” model for each subset size but asks for a variety of model selection statistics, as well as the estimated regression coefficients. An OUTEST= data set is created and displayed. See Figure 61.20 and Figure 61.21 for results.

```
proc reg data=fitness outest=est;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare mse jp gmsep cp aic bic sbc b best=1;
proc print data=est;
run;
```

```
                                   The REG Procedure
                                     Model: MODEL1
                               Dependent Variable: Oxygen

                               R-Square Selection Method

Number in                                                     Estimated MSE
  Model     R-Square       C(p)          AIC          BIC   of Prediction         J(p)           MSE            SBC

       1      0.7434    13.6988      64.5341      65.4673          8.0546       8.0199       7.53384       67.40210
------------------------------------------------------------------------------------------------------------------
       2      0.7642    12.3894      63.9050      64.8212          7.9478       7.8621       7.16842       68.20695
------------------------------------------------------------------------------------------------------------------
       3      0.8111     6.9596      59.0373      61.3127          6.8583       6.7253       5.95669       64.77326
------------------------------------------------------------------------------------------------------------------
       4      0.8368     4.8800      56.4995      60.3996          6.3984       6.2053       5.34346       63.66941
------------------------------------------------------------------------------------------------------------------
       5      0.8480     5.1063      56.2986      61.5667          6.4565       6.1782       5.17634       64.90250
------------------------------------------------------------------------------------------------------------------
       6      0.8487     7.0000      58.1616      64.0748          6.9870       6.5804       5.36825       68.19952

Number in             --------------------------------------Parameter Estimates-------------------------------------
  Model     R-Square     Intercept          Age         Weight        RunTime       RunPulse      RestPulse      MaxPulse

       1      0.7434      82.42177            .              .       -3.31056              .              .             .
---------------------------------------------------------------------------------------------------------------------
       2      0.7642      88.46229     -0.15037              .       -3.20395              .              .             .
---------------------------------------------------------------------------------------------------------------------
       3      0.8111     111.71806     -0.25640              .       -2.82538       -0.13091              .             .
---------------------------------------------------------------------------------------------------------------------
       4      0.8368      98.14789     -0.19773              .       -2.76758       -0.34811              .       0.27051
---------------------------------------------------------------------------------------------------------------------
       5      0.8480     102.20428     -0.21962       -0.07230       -2.68252       -0.37340              .       0.30491
---------------------------------------------------------------------------------------------------------------------
       6      0.8487     102.93448     -0.22697       -0.07418       -2.62865       -0.36963       -0.02153       0.30322
```

Figure 61.20. PROC REG Output for Physical Fitness Data: Best Models

```
                                                                                                          Max
   Obs  _MODEL_  _TYPE_  _DEPVAR_   _RMSE_  Intercept      Age      Weight    RunTime  RunPulse  RestPulse   Pulse

    1   MODEL1   PARMS    Oxygen   2.74478    82.422      .          .       -3.31056    .          .          .
    2   MODEL1   PARMS    Oxygen   2.67739    88.462  -0.15037       .       -3.20395    .          .          .
    3   MODEL1   PARMS    Oxygen   2.44063   111.718  -0.25640       .       -2.82538  -0.13091    .          .
    4   MODEL1   PARMS    Oxygen   2.31159    98.148  -0.19773       .       -2.76758  -0.34811    .        0.27051
    5   MODEL1   PARMS    Oxygen   2.27516   102.204  -0.21962  -0.072302  -2.68252  -0.37340    .        0.30491
    6   MODEL1   PARMS    Oxygen   2.31695   102.934  -0.22697  -0.074177  -2.62865  -0.36963  -0.021534  0.30322


   Obs  Oxygen   _IN_   _P_   _EDF_    _MSE_     _RSQ_      _CP_      _JP_    _GMSEP_    _AIC_     _BIC_     _SBC_

    1     -1       1     2      29    7.53384   0.74338   13.6988   8.01990   8.05462   64.5341   65.4673   67.4021
    2     -1       2     3      28    7.16842   0.76425   12.3894   7.86214   7.94778   63.9050   64.8212   68.2069
    3     -1       3     4      27    5.95669   0.81109    6.9596   6.72530   6.85833   59.0373   61.3127   64.7733
    4     -1       4     5      26    5.34346   0.83682    4.8800   6.20531   6.39837   56.4995   60.3996   63.6694
    5     -1       5     6      25    5.17634   0.84800    5.1063   6.17821   6.45651   56.2986   61.5667   64.9025
    6     -1       6     7      24    5.36825   0.84867    7.0000   6.58043   6.98700   58.1616   64.0748   68.1995
```

Figure 61.21. PROC PRINT Output for Physical Fitness Data: OUTEST= Data Set

OUTSSCP= Data Sets

The OUTSSCP= option produces a TYPE=SSCP output SAS data set containing sums of squares and crossproducts. A special row (observation) and column (variable) of the matrix called Intercept contain the number of observations and sums.

Observations are identified by the character variable _NAME_. The data set contains all variables used in MODEL statements. You can specify additional variables that you want included in the crossproducts matrix with a VAR statement.

The SSCP data set is used when a large number of observations are explored in many different runs. The SSCP data set can be saved and used for subsequent runs, which are much less expensive since PROC REG never reads the original data again. If you run PROC REG once to create only a SSCP data set, you should list all the variables that you may need in a VAR statement or include all the variables that you may need in a MODEL statement.

The following example uses the fitness data from Example 61.1 on page 3924 to produce an output data set with the OUTSSCP= option. The resulting output is shown in Figure 61.22.

```
proc reg data=fitness outsscp=sscp;
   var Oxygen RunTime Age Weight RestPulse RunPulse MaxPulse;
proc print data=sscp;
run;
```

Since a model is not fit to the data and since the only request is to create the SSCP data set, a MODEL statement is not required in this example. However, since the MODEL statement is not used, the VAR statement is required.

```
Obs   _TYPE_   _NAME_      Intercept      Oxygen    RunTime        Age      Weight   RestPulse    RunPulse    MaxPulse

 1     SSCP    Intercept       31.00     1468.65     328.17    1478.00     2400.78     1657.00     5259.00     5387.00
 2     SSCP    Oxygen        1468.65    70429.86   15356.14   69767.75   113522.26    78015.41   248497.31   254866.75
 3     SSCP    RunTime        328.17    15356.14    3531.80   15687.24    25464.71    17684.05    55806.29    57113.72
 4     SSCP    Age           1478.00    69767.75   15687.24   71282.00   114158.90    78806.00   250194.00   256218.00
 5     SSCP    Weight        2400.78   113522.26   25464.71  114158.90   188008.20   128409.28   407745.67   417764.62
 6     SSCP    RestPulse     1657.00    78015.41   17684.05   78806.00   128409.28    90311.00   281928.00   288583.00
 7     SSCP    RunPulse      5259.00   248497.31   55806.29  250194.00   407745.67   281928.00   895317.00   916499.00
 8     SSCP    MaxPulse      5387.00   254866.75   57113.72  256218.00   417764.62   288583.00   916499.00   938641.00
 9     N                       31.00       31.00      31.00      31.00       31.00       31.00       31.00       31.00
```

Figure 61.22. SSCP Data Set Created with OUTSSCP= Option: REG Procedure

Interactive Analysis

PROC REG enables you to change interactively both the model and the data used to compute the model, and to produce and highlight scatter plots. See the section "Using PROC REG Interactively" on page 3812 for an overview of interactive analysis using PROC REG. The following statements can be used interactively (without reinvoking PROC REG): ADD, DELETE, MODEL, MTEST, OUTPUT, PAINT, PLOT, PRINT, REFIT, RESTRICT, REWEIGHT, and TEST. All interactive features are disabled if there is a BY statement.

The ADD, DELETE and REWEIGHT statements can be used to modify the current MODEL. Every use of an ADD, DELETE or REWEIGHT statement causes the model label to be modified by attaching an additional number to it. This number is the cumulative total of the number of ADD, DELETE or REWEIGHT statements following the current MODEL statement.

A more detailed explanation of changing the data used to compute the model is given in the section “Reweighting Observations in an Analysis” on page 3903. Extra features for line printer scatter plots are discussed in the section “Line Printer Scatter Plot Features” on page 3882.

The following example illustrates the usefulness of the interactive features. First, the full regression model is fit to the class data (see the “Getting Started” section on page 3800), and Figure 61.23 is produced.

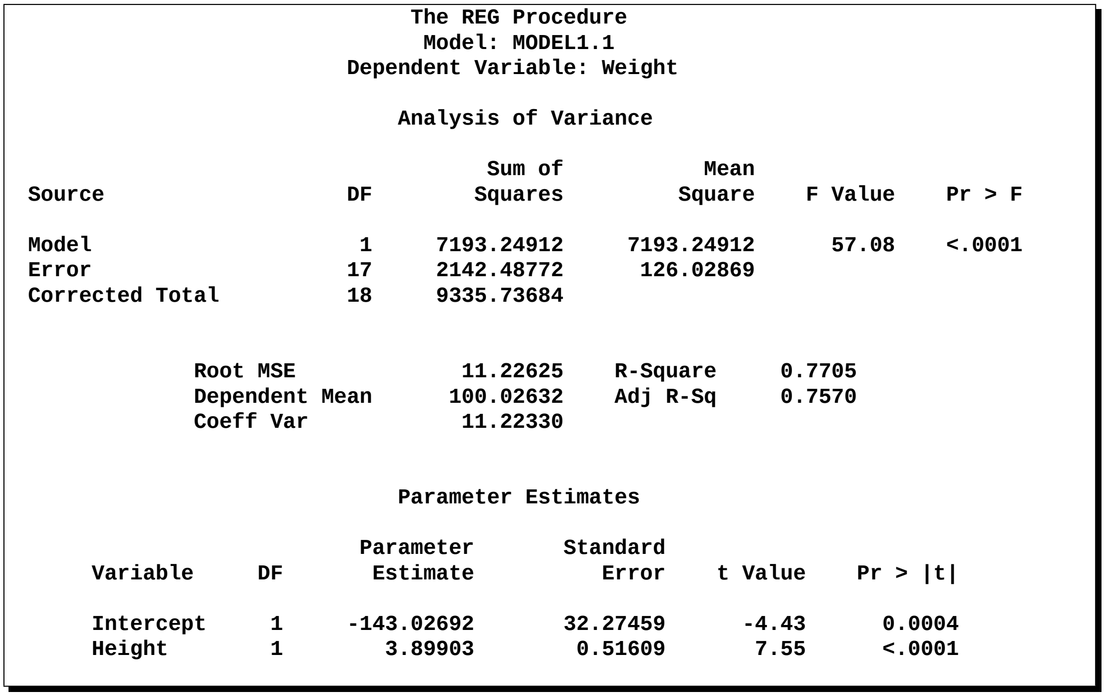
```
proc reg data=Class;
   model Weight=Age Height;
run;
```

```
                              The REG Procedure
                                Model: MODEL1
                          Dependent Variable: Weight

                             Analysis of Variance

                                      Sum of           Mean
Source                    DF         Squares         Square    F Value    Pr > F

Model                      2      7215.63710     3607.81855      27.23    <.0001
Error                     16      2120.09974      132.50623
Corrected Total           18      9335.73684


             Root MSE             11.51114    R-Square     0.7729
             Dependent Mean      100.02632    Adj R-Sq     0.7445
             Coeff Var            11.50811


                             Parameter Estimates

                            Parameter       Standard
     Variable      DF        Estimate          Error    t Value    Pr > |t|

     Intercept      1      -141.22376       33.38309      -4.23      0.0006
     Age            1         1.27839        3.11010       0.41      0.6865
     Height         1         3.59703        0.90546       3.97      0.0011
```

Figure 61.23. Interactive Analysis: Full Model

Next, the regression model is reduced by the following statements, and Figure 61.24 is produced.

```
delete age;
print;
run;
```



```
                          The REG Procedure
                           Model: MODEL1.1
                      Dependent Variable: Weight

                         Analysis of Variance

                                     Sum of          Mean
Source                   DF         Squares        Square    F Value    Pr > F

Model                     1      7193.24912    7193.24912      57.08    <.0001
Error                    17      2142.48772     126.02869
Corrected Total          18      9335.73684


             Root MSE             11.22625    R-Square     0.7705
             Dependent Mean      100.02632    Adj R-Sq     0.7570
             Coeff Var            11.22330


                         Parameter Estimates

                        Parameter       Standard
     Variable      DF    Estimate          Error    t Value    Pr > |t|

     Intercept      1  -143.02692       32.27459      -4.43      0.0004
     Height         1     3.89903        0.51609       7.55      <.0001
```

Figure 61.24. Interactive Analysis: Reduced Model

Note that the MODEL label has been changed from MODEL1 to MODEL1.1, as the original MODEL has been changed by the delete statement.

The following statements generate a scatter plot of the residuals against the predicted values from the full model. Figure 61.25 is produced, and the scatter plot shows a possible outlier.

```
add age;
plot r.*p. / cframe=ligr;
run;
```

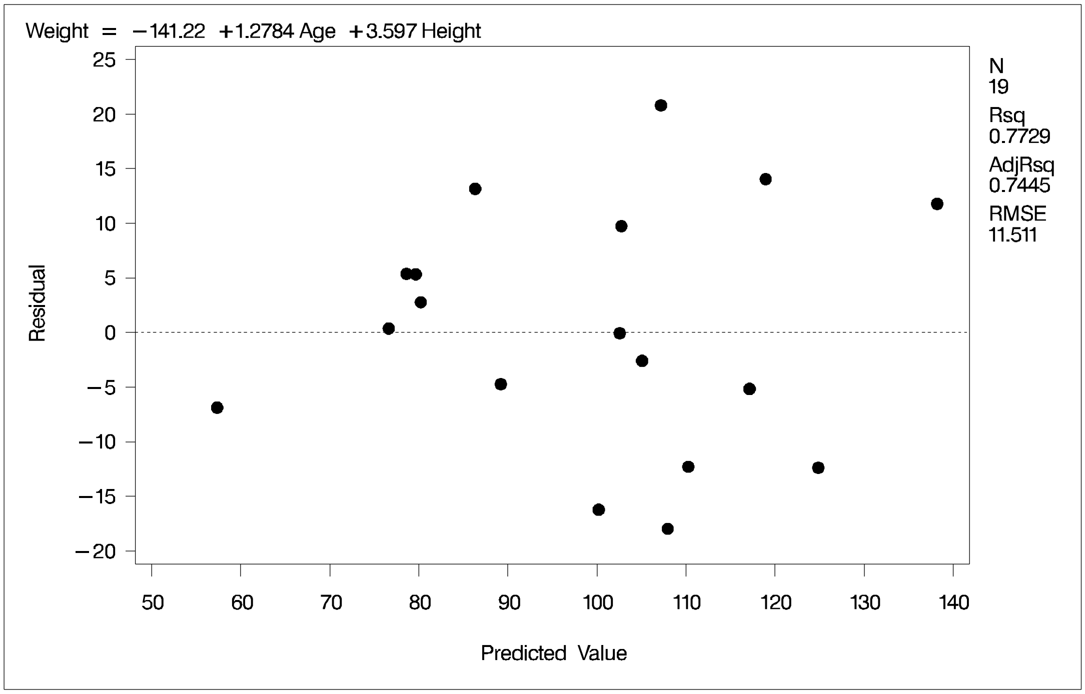


Figure 61.25. Interactive Analysis: Scatter Plot

The following statements delete the observation with the largest residual, refit the regression model, and produce a scatter plot of residuals against predicted values for the refitted model. Figure 61.26 shows the new scatter plot.

```
reweight r.>20;
plot / cframe=ligr;
run;
```

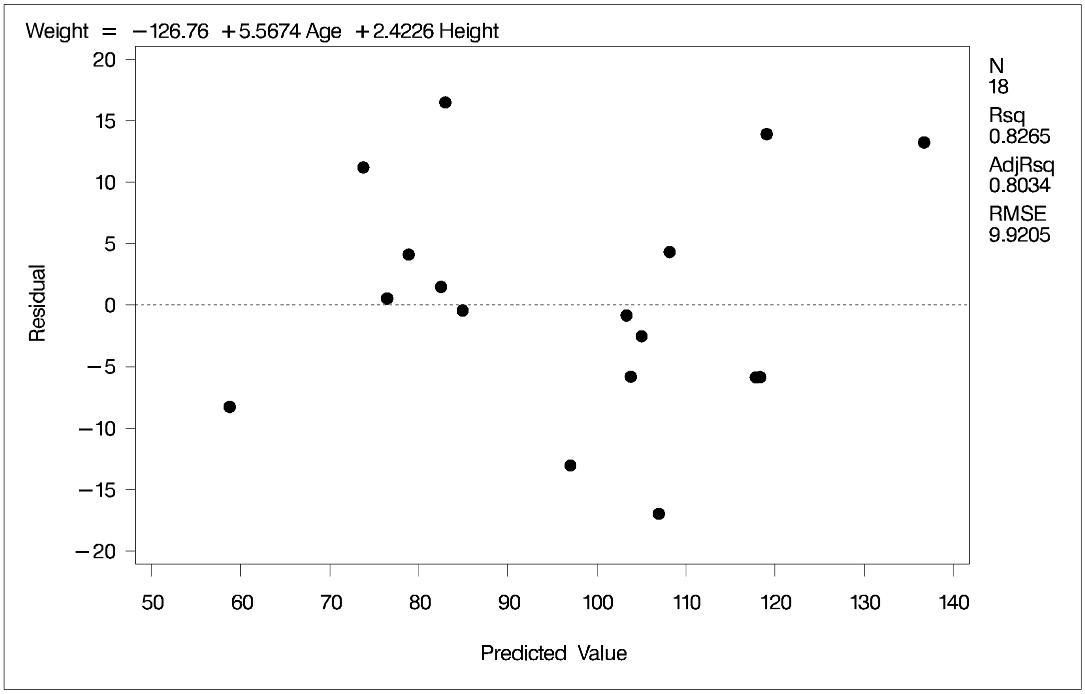


Figure 61.26. Interactive Analysis: Scatter Plot for Refitted Model

Model-Selection Methods

The nine methods of model selection implemented in PROC REG are specified with the SELECTION= option in the MODEL statement. Each method is discussed in this section.

Full Model Fitted (NONE)

This method is the default and provides no model selection capability. The complete model specified in the MODEL statement is used to fit the model. For many regression analyses, this may be the only method you need.

Forward Selection (FORWARD)

The forward-selection technique begins with no variables in the model. For each of the independent variables, the FORWARD method calculates F statistics that reflect the variable's contribution to the model if it is included. The p-values for these F statistics are compared to the SLENTRY= value that is specified in the MODEL statement (or to 0.50 if the SLENTRY= option is omitted). If no F statistic has a significance level greater than the SLENTRY= value, the FORWARD selection stops. Otherwise, the FORWARD method adds the variable that has the largest F statistic to the model. The FORWARD method then calculates F statistics again for the variables still remaining outside the model, and the evaluation process is repeated. Thus, variables are added one by one to the model until no remaining variable produces a significant F statistic. Once a variable is in the model, it stays.

Backward Elimination (BACKWARD)

The backward elimination technique begins by calculating F statistics for a model, including all of the independent variables. Then the variables are deleted from the model one by one until all the variables remaining in the model produce F statistics significant at the SLSTAY= level specified in the MODEL statement (or at the 0.10 level if the SLSTAY= option is omitted). At each step, the variable showing the smallest contribution to the model is deleted.

Stepwise (STEPWISE)

The stepwise method is a modification of the forward-selection technique and differs in that variables already in the model do not necessarily stay there. As in the forward-selection method, variables are added one by one to the model, and the F statistic for a variable to be added must be significant at the SLENTRY= level. After a variable is added, however, the stepwise method looks at all the variables already included in the model and deletes any variable that does not produce an F statistic significant at the SLSTAY= level. Only after this check is made and the necessary deletions accomplished can another variable be added to the model. The stepwise process ends when none of the variables outside the model has an F statistic significant at the SLENTRY= level and every variable in the model is significant at the SLSTAY= level, or when the variable to be added to the model is the one just deleted from it.

Maximum R^2 Improvement (MAXR)

The maximum R^2 improvement technique does not settle on a single model. Instead, it tries to find the “best” one-variable model, the “best” two-variable model, and so forth, although it is not guaranteed to find the model with the largest R^2 for each size.

The MAXR method begins by finding the one-variable model producing the highest R^2. Then another variable, the one that yields the greatest increase in R^2, is added. Once the two-variable model is obtained, each of the variables in the model is compared to each variable not in the model. For each comparison, the MAXR method determines if removing one variable and replacing it with the other variable increases R^2. After comparing all possible switches, the MAXR method makes the switch that produces the largest increase in R^2. Comparisons begin again, and the process continues until the MAXR method finds that no switch could increase R^2. Thus, the two-variable model achieved is considered the “best” two-variable model the technique can find. Another variable is then added to the model, and the comparing-and-switching process is repeated to find the “best” three-variable model, and so forth.

The difference between the STEPWISE method and the MAXR method is that all switches are evaluated before any switch is made in the MAXR method . In the STEPWISE method, the “worst” variable may be removed without considering what adding the “best” remaining variable might accomplish. The MAXR method may require much more computer time than the STEPWISE method.

Minimum R^2 (MINR) Improvement

The MINR method closely resembles the MAXR method, but the switch chosen is the one that produces the smallest increase in R^2. For a given number of variables in the model, the MAXR and MINR methods usually produce the same "best" model, but the MINR method considers more models of each size.

R^2 Selection (RSQUARE)

The RSQUARE method finds subsets of independent variables that best predict a dependent variable by linear regression in the given sample. You can specify the largest and smallest number of independent variables to appear in a subset and the number of subsets of each size to be selected. The RSQUARE method can efficiently perform all possible subset regressions and display the models in decreasing order of R^2 magnitude within each subset size. Other statistics are available for comparing subsets of different sizes. These statistics, as well as estimated regression coefficients, can be displayed or output to a SAS data set.

The subset models selected by the RSQUARE method are optimal in terms of R^2 for the given sample, but they are not necessarily optimal for the population from which the sample is drawn or for any other sample for which you may want to make predictions. If a subset model is selected on the basis of a large R^2 value or any other criterion commonly used for model selection, then all regression statistics computed for that model under the assumption that the model is given a priori, including all statistics computed by PROC REG, are biased.

While the RSQUARE method is a useful tool for exploratory model building, no statistical method can be relied on to identify the "true" model. Effective model building requires substantive theory to suggest relevant predictors and plausible functional forms for the model.

The RSQUARE method differs from the other selection methods in that RSQUARE always identifies the model with the largest R^2 for each number of variables considered. The other selection methods are not guaranteed to find the model with the largest R^2. The RSQUARE method requires much more computer time than the other selection methods, so a different selection method such as the STEPWISE method is a good choice when there are many independent variables to consider.

Adjusted R^2 Selection (ADJRSQ)

This method is similar to the RSQUARE method, except that the adjusted R^2 statistic is used as the criterion for selecting models, and the method finds the models with the highest adjusted R^2 within the range of sizes.

Mallows' C_p Selection (CP)

This method is similar to the ADJRSQ method, except that Mallows' C_p statistic is used as the criterion for model selection. Models are listed in ascending order of C_p.

Additional Information on Model-Selection Methods

If the RSQUARE or STEPWISE procedure (as documented in *SAS User's Guide: Statistics, Version 5 Edition*) is requested, PROC REG with the appropriate model-selection method is actually used.

Reviews of model-selection methods by Hocking (1976) and Judge et al. (1980) describe these and other variable-selection methods.

Criteria Used in Model-Selection Methods

When many significance tests are performed, each at a level of, for example, 5 percent, the overall probability of rejecting at least one true null hypothesis is much larger than 5 percent. If you want to guard against including any variables that do not contribute to the predictive power of the model in the population, you should specify a very small SLE= significance level for the FORWARD and STEPWISE methods and a very small SLS= significance level for the BACKWARD and STEPWISE methods.

In most applications, many of the variables considered have some predictive power, however small. If you want to choose the model that provides the best prediction using the sample estimates, you need only to guard against estimating more parameters than can be reliably estimated with the given sample size, so you should use a moderate significance level, perhaps in the range of 10 percent to 25 percent.

In addition to R^2, the C_p statistic is displayed for each model generated in the model-selection methods. The C_p statistic is proposed by Mallows (1973) as a criterion for selecting a model. It is a measure of total squared error defined as

$$C_p = \frac{SSE_p}{s^2} - (N - 2p)$$

where s^2 is the MSE for the full model, and SSE_p is the sum-of-squares error for a model with p parameters including the intercept, if any. If C_p is plotted against p, Mallows recommends the model where C_p first approaches p. When the right model is chosen, the parameter estimates are unbiased, and this is reflected in C_p near p. For further discussion, refer to Daniel and Wood (1980).

The Adjusted R^2 statistic is an alternative to R^2 that is adjusted for the number of parameters in the model. The adjusted R^2 statistic is calculated as

$$\text{ADJRSQ} = 1 - \frac{(n - i)(1 - R^2)}{n - p}$$

where n is the number of observations used in fitting the model, and i is an indicator variable that is 1 if the model includes an intercept, and 0 otherwise.

Limitations in Model-Selection Methods

The use of model-selection methods can be time-consuming in some cases because there is no built-in limit on the number of independent variables, and the calculations for a large number of independent variables can be lengthy. The recommended limit on the number of independent variables for the MINR method is $20 + i$, where i is the value of the INCLUDE= option.

For the RSQUARE, ADJRSQ, or CP methods, with a large value of the BEST= option, adding one more variable to the list from which regressors are selected may significantly increase the CPU time. Also, the time required for the analysis is highly dependent on the data and on the values of the BEST=, START=, and STOP= options.

Parameter Estimates and Associated Statistics

The following example uses the fitness data from Example 61.1 on page 3924. Figure 61.28 shows the parameter estimates and the tables from the SS1, SS2, STB, CLB, COVB, and CORRB options:

```
proc reg data=fitness;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse
         / ss1 ss2 stb clb covb corrb;
run;
```

The procedure first displays an Analysis of Variance table (Figure 61.27). The F statistic for the overall model is significant, indicating that the model explains a significant portion of the variation in the data.

```
                              The REG Procedure
                                Model: MODEL1
                          Dependent Variable: Oxygen

                             Analysis of Variance

                                       Sum of          Mean
Source                    DF          Squares        Square    F Value    Pr > F

Model                      6        722.54361     120.42393      22.43    <.0001
Error                     24        128.83794       5.36825
Corrected Total           30        851.38154


             Root MSE              2.31695    R-Square     0.8487
             Dependent Mean       47.37581    Adj R-Sq     0.8108
             Coeff Var             4.89057
```

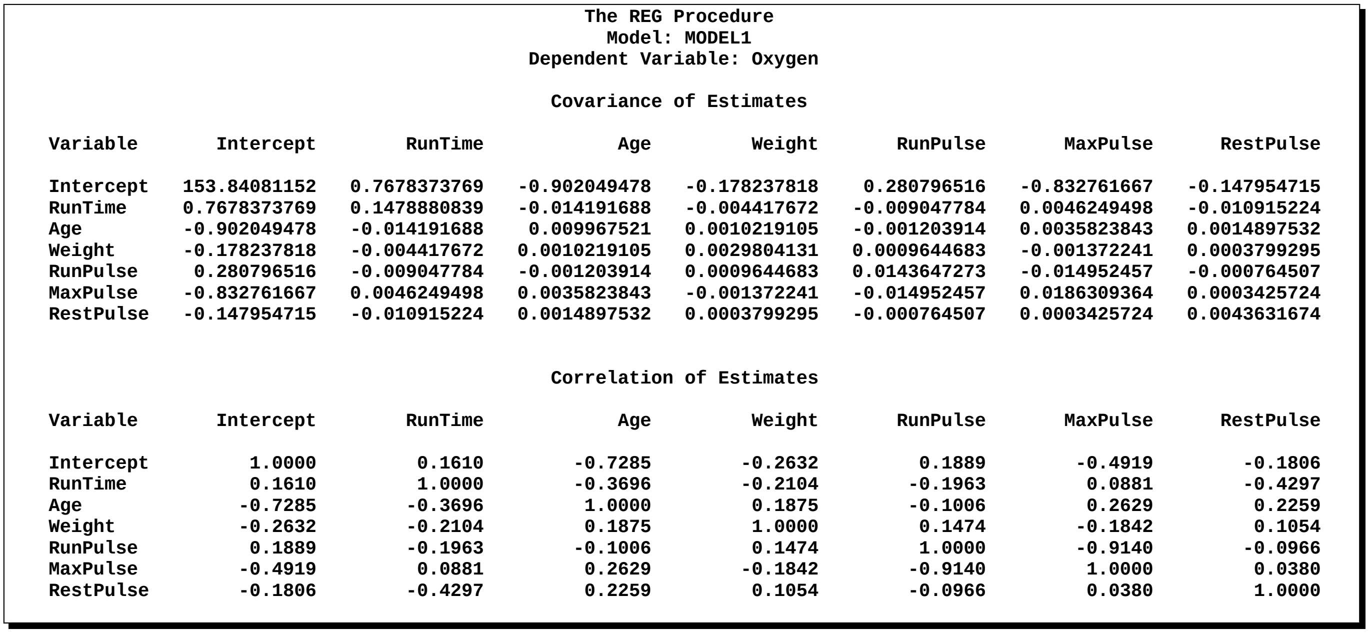
Figure 61.27. ANOVA Table

The procedure next displays Parameter Estimates and some associated statistics (Figure 61.28). First, the estimates are shown, followed by their Standard Errors. The next two columns of the table contain the t statistics and the corresponding probabilities for testing the null hypothesis that the parameter is not significantly different from zero. These probabilities are usually referred to as p-values. For example, the Intercept term in the model is estimated to be 102.9 and is significantly different from zero. The next two columns of the table are the result of requesting the SS1 and SS2 options, and they show sequential and partial Sums of Squares (SS) associated with each variable. The Standardized Estimates (produced by the STB option) are the parameter estimates that result when all variables are standardized to a mean of 0 and a variance of 1. These estimates are computed by multiplying the original estimates by the standard deviation of the regressor (independent) variable and then dividing by the standard deviation of the dependent variable. The CLB option adds the upper and lower 95% confidence limits for the parameter estimates; the α level can be changed by specifying the ALPHA= option in the PROC REG or MODEL statement.

```
                                                The REG Procedure
                                                  Model: MODEL1
                                           Dependent Variable: Oxygen

                                               Parameter Estimates

                 Parameter     Standard                                                Standardized
Variable   DF     Estimate        Error  t Value  Pr > |t|     Type I SS   Type II SS      Estimate     95% Confidence Limits

Intercept   1    102.93448     12.40326     8.30    <.0001         69578    369.72831             0     77.33541    128.53355
RunTime     1     -2.62865      0.38456    -6.84    <.0001     632.90010    250.82210      -0.68460     -3.42235     -1.83496
Age         1     -0.22697      0.09984    -2.27    0.0322      17.76563     27.74577      -0.22204     -0.43303     -0.02092
Weight      1     -0.07418      0.05459    -1.36    0.1869       5.60522      9.91059      -0.11597     -0.18685      0.03850
RunPulse    1     -0.36963      0.11985    -3.08    0.0051      38.87574     51.05806      -0.71133     -0.61699     -0.12226
MaxPulse    1      0.30322      0.13650     2.22    0.0360      26.82640     26.49142       0.52161      0.02150      0.58493
RestPulse   1     -0.02153      0.06605    -0.33    0.7473       0.57051      0.57051      -0.03080     -0.15786      0.11480
```

Figure 61.28. SS1, SS2, STB, CLB, COVB, and CORRB Options: Parameter Estimates

The final two tables are produced as a result of requesting the COVB and CORRB options (Figure 61.29). These tables show the estimated covariance matrix of the parameter estimates, and the estimated correlation matrix of the estimates.



```
                                        The REG Procedure
                                          Model: MODEL1
                                   Dependent Variable: Oxygen

                                    Covariance of Estimates

Variable        Intercept         RunTime             Age          Weight        RunPulse        MaxPulse       RestPulse

Intercept    153.84081152    0.7678373769    -0.902049478    -0.178237818     0.280796516    -0.832761667    -0.147954715
RunTime      0.7678373769    0.1478880839    -0.014191688    -0.004417672    -0.009047784    0.0046249498    -0.010915224
Age          -0.902049478    -0.014191688     0.009967521    0.0010219105    -0.001203914    0.0035823843    0.0014897532
Weight       -0.178237818    -0.004417672    0.0010219105    0.0029804131    0.0009644683    -0.001372241    0.0003799295
RunPulse      0.280796516    -0.009047784    -0.001203914    0.0009644683    0.0143647273    -0.014952457    -0.000764507
MaxPulse     -0.832761667    0.0046249498    0.0035823843    -0.001372241    -0.014952457    0.0186309364    0.0003425724
RestPulse    -0.147954715    -0.010915224    0.0014897532    0.0003799295    -0.000764507    0.0003425724    0.0043631674


                                    Correlation of Estimates

Variable        Intercept         RunTime             Age          Weight        RunPulse        MaxPulse       RestPulse

Intercept          1.0000          0.1610         -0.7285         -0.2632          0.1889         -0.4919         -0.1806
RunTime            0.1610          1.0000         -0.3696         -0.2104         -0.1963          0.0881         -0.4297
Age               -0.7285         -0.3696          1.0000          0.1875         -0.1006          0.2629          0.2259
Weight            -0.2632         -0.2104          0.1875          1.0000          0.1474         -0.1842          0.1054
RunPulse           0.1889         -0.1963         -0.1006          0.1474          1.0000         -0.9140         -0.0966
MaxPulse          -0.4919          0.0881          0.2629         -0.1842         -0.9140          1.0000          0.0380
RestPulse         -0.1806         -0.4297          0.2259          0.1054         -0.0966          0.0380          1.0000
```

Figure 61.29. SS1, SS2, STB, CLB, COVB, and CORRB Options: Covariances and Correlations

For further discussion of the parameters and statistics, see the "Displayed Output" section on page 3918, and Chapter 2, "Introduction to Regression Procedures."

Predicted and Residual Values

The display of the predicted values and residuals is controlled by the P, R, CLM, and CLI options in the MODEL statement. The P option causes PROC REG to display the observation number, the ID value (if an ID statement is used), the actual value, the predicted value, and the residual. The R, CLI, and CLM options also produce the items under the P option. Thus, P is unnecessary if you use one of the other options.

The R option requests more detail, especially about the residuals. The standard errors of the mean predicted value and the residual are displayed. The studentized residual, which is the residual divided by its standard error, is both displayed and plotted. A measure of influence, Cook's D, is displayed. Cook's D measures the change to the estimates that results from deleting each observation (Cook 1977, 1979). This statistic is very similar to DFFITS.

The CLM option requests that PROC REG display the $100(1-\alpha)\%$ lower and upper confidence limits for the mean predicted values. This accounts for the variation due to estimating the parameters only. If you want a $100(1-\alpha)\%$ confidence interval for observed values, then you can use the CLI option, which adds in the variability of the error term. The α level can be specified with the ALPHA= option in the PROC REG or MODEL statement.

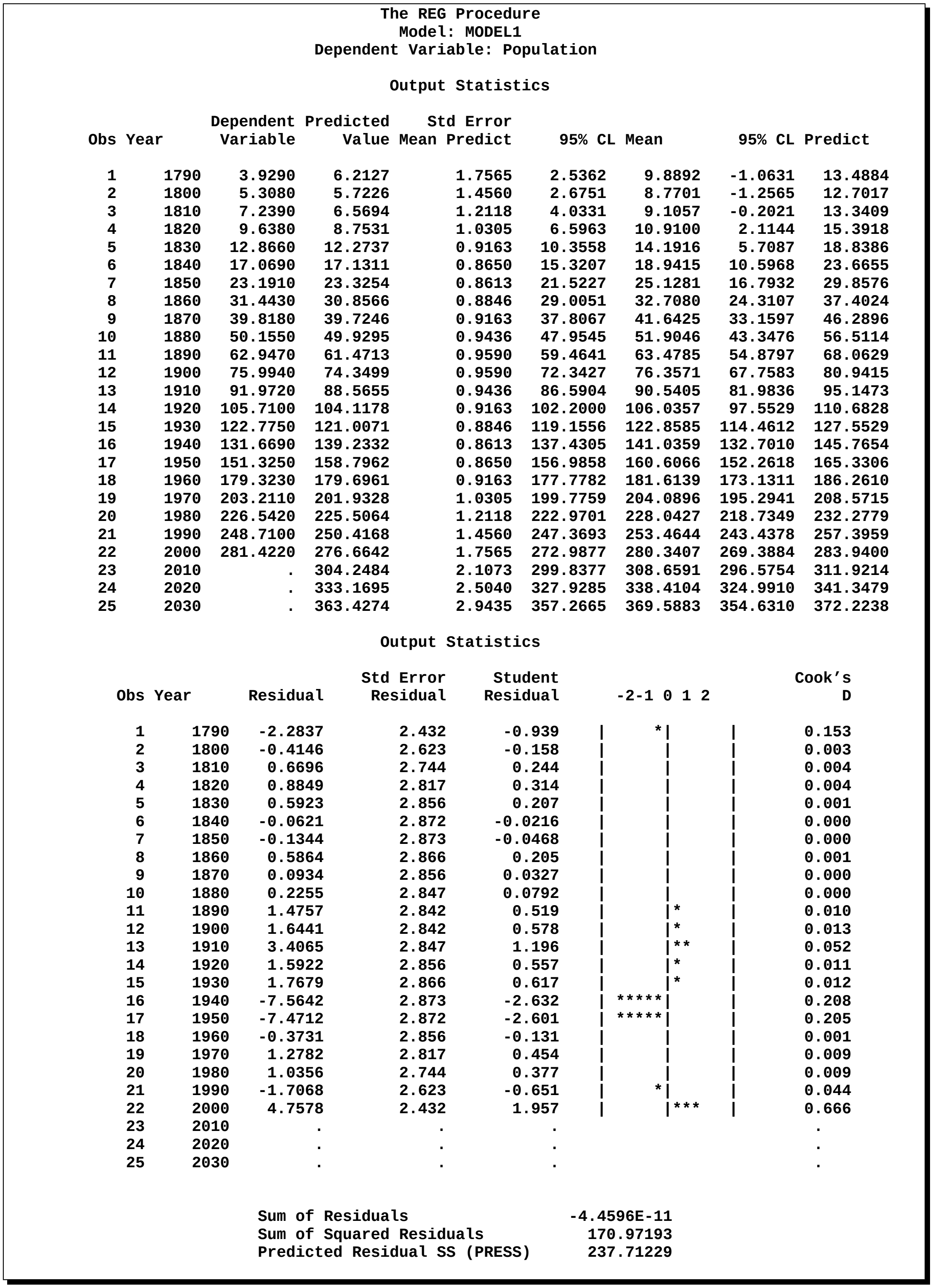
You can use these statistics in PLOT and PAINT statements. This is useful in performing a variety of regression diagnostics. For definitions of the statistics produced by these options, see Chapter 2, “Introduction to Regression Procedures.”

The following example uses the US population data found on the section “Polynomial Regression” beginning on page 3804.

```
data USPop2;
   input Year @@;
   YearSq=Year*Year;
   datalines;
2010 2020 2030
;
data USPop2;
   set USPopulation USPop2;

proc reg data=USPop2;
   id Year;
   model Population=Year YearSq / r cli clm;
run;
```

```
                                The REG Procedure
                                  Model: MODEL1
                         Dependent Variable: Population

                              Analysis of Variance

                                        Sum of           Mean
Source                   DF            Squares         Square    F Value    Pr > F

Model                     2             159529          79765    8864.19    <.0001
Error                    19          170.97193        8.99852
Corrected Total          21             159700


             Root MSE              2.99975    R-Square     0.9989
             Dependent Mean       94.64800    Adj R-Sq     0.9988
             Coeff Var             3.16938


                              Parameter Estimates

                             Parameter       Standard
    Variable      DF          Estimate          Error    t Value    Pr > |t|

    Intercept      1             21631      639.50181      33.82      <.0001
    Year           1         -24.04581        0.67547     -35.60      <.0001
    YearSq         1           0.00668     0.00017820      37.51      <.0001
```

Figure 61.30. Regression Using the R, CLI, and CLM Options



```
                                  The REG Procedure
                                    Model: MODEL1
                             Dependent Variable: Population

                                   Output Statistics

                Dependent Predicted    Std Error
     Obs Year    Variable     Value Mean Predict     95% CL Mean        95% CL Predict

       1   1790    3.9290    6.2127       1.7565   2.5362    9.8892   -1.0631   13.4884
       2   1800    5.3080    5.7226       1.4560   2.6751    8.7701   -1.2565   12.7017
       3   1810    7.2390    6.5694       1.2118   4.0331    9.1057   -0.2021   13.3409
       4   1820    9.6380    8.7531       1.0305   6.5963   10.9100    2.1144   15.3918
       5   1830   12.8660   12.2737       0.9163  10.3558   14.1916    5.7087   18.8386
       6   1840   17.0690   17.1311       0.8650  15.3207   18.9415   10.5968   23.6655
       7   1850   23.1910   23.3254       0.8613  21.5227   25.1281   16.7932   29.8576
       8   1860   31.4430   30.8566       0.8846  29.0051   32.7080   24.3107   37.4024
       9   1870   39.8180   39.7246       0.9163  37.8067   41.6425   33.1597   46.2896
      10   1880   50.1550   49.9295       0.9436  47.9545   51.9046   43.3476   56.5114
      11   1890   62.9470   61.4713       0.9590  59.4641   63.4785   54.8797   68.0629
      12   1900   75.9940   74.3499       0.9590  72.3427   76.3571   67.7583   80.9415
      13   1910   91.9720   88.5655       0.9436  86.5904   90.5405   81.9836   95.1473
      14   1920  105.7100  104.1178       0.9163 102.2000  106.0357   97.5529  110.6828
      15   1930  122.7750  121.0071       0.8846 119.1556  122.8585  114.4612  127.5529
      16   1940  131.6690  139.2332       0.8613 137.4305  141.0359  132.7010  145.7654
      17   1950  151.3250  158.7962       0.8650 156.9858  160.6066  152.2618  165.3306
      18   1960  179.3230  179.6961       0.9163 177.7782  181.6139  173.1311  186.2610
      19   1970  203.2110  201.9328       1.0305 199.7759  204.0896  195.2941  208.5715
      20   1980  226.5420  225.5064       1.2118 222.9701  228.0427  218.7349  232.2779
      21   1990  248.7100  250.4168       1.4560 247.3693  253.4644  243.4378  257.3959
      22   2000  281.4220  276.6642       1.7565 272.9877  280.3407  269.3884  283.9400
      23   2010         .  304.2484       2.1073 299.8377  308.6591  296.5754  311.9214
      24   2020         .  333.1695       2.5040 327.9285  338.4104  324.9910  341.3479
      25   2030         .  363.4274       2.9435 357.2665  369.5883  354.6310  372.2238

                                   Output Statistics

                            Std Error    Student                           Cook's
        Obs Year   Residual  Residual   Residual      -2-1 0 1 2              D

          1 1790    -2.2837     2.432     -0.939    |      *|      |      0.153
          2 1800    -0.4146     2.623     -0.158    |       |      |      0.003
          3 1810     0.6696     2.744      0.244    |       |      |      0.004
          4 1820     0.8849     2.817      0.314    |       |      |      0.004
          5 1830     0.5923     2.856      0.207    |       |      |      0.001
          6 1840    -0.0621     2.872    -0.0216    |       |      |      0.000
          7 1850    -0.1344     2.873    -0.0468    |       |      |      0.000
          8 1860     0.5864     2.866      0.205    |       |      |      0.001
          9 1870     0.0934     2.856     0.0327    |       |      |      0.000
         10 1880     0.2255     2.847     0.0792    |       |      |      0.000
         11 1890     1.4757     2.842      0.519    |       |*     |      0.010
         12 1900     1.6441     2.842      0.578    |       |*     |      0.013
         13 1910     3.4065     2.847      1.196    |       |**    |      0.052
         14 1920     1.5922     2.856      0.557    |       |*     |      0.011
         15 1930     1.7679     2.866      0.617    |       |*     |      0.012
         16 1940    -7.5642     2.873     -2.632    | *****|      |      0.208
         17 1950    -7.4712     2.872     -2.601    | *****|      |      0.205
         18 1960    -0.3731     2.856     -0.131    |       |      |      0.001
         19 1970     1.2782     2.817      0.454    |       |      |      0.009
         20 1980     1.0356     2.744      0.377    |       |      |      0.009
         21 1990    -1.7068     2.623     -0.651    |      *|      |      0.044
         22 2000     4.7578     2.432      1.957    |       |***   |      0.666
         23 2010          .         .          .                            .
         24 2020          .         .          .                            .
         25 2030          .         .          .                            .


                  Sum of Residuals                  -4.4596E-11
                  Sum of Squared Residuals            170.97193
                  Predicted Residual SS (PRESS)       237.71229
```

Figure 61.31. Regression Using the R, CLI, and CLM Options

After producing the usual Analysis of Variance and Parameter Estimates tables

(Figure 61.30), the procedure displays the results of requesting the options for predicted and residual values (Figure 61.31). For each observation, the requested information is shown. Note that the ID variable is used to identify each observation. Also note that, for observations with missing dependent variables, the predicted value, standard error of the predicted value, and confidence intervals for the predicted value are still available.

The plot of studentized residuals and Cook's D statistics are displayed as a result of requesting the R option. In the plot of studentized residuals, a large number of observations with absolute values greater than two indicates an inadequate model. A version of the studentized residual plot can be created on a high-resolution graphics device; see Example 61.7 on page 3952 for a similar example.

Line Printer Scatter Plot Features

This section discusses the special options available with line printer scatter plots. Detailed examples of high resolution graphics plots and options are given in Example 61.6 on page 3950.

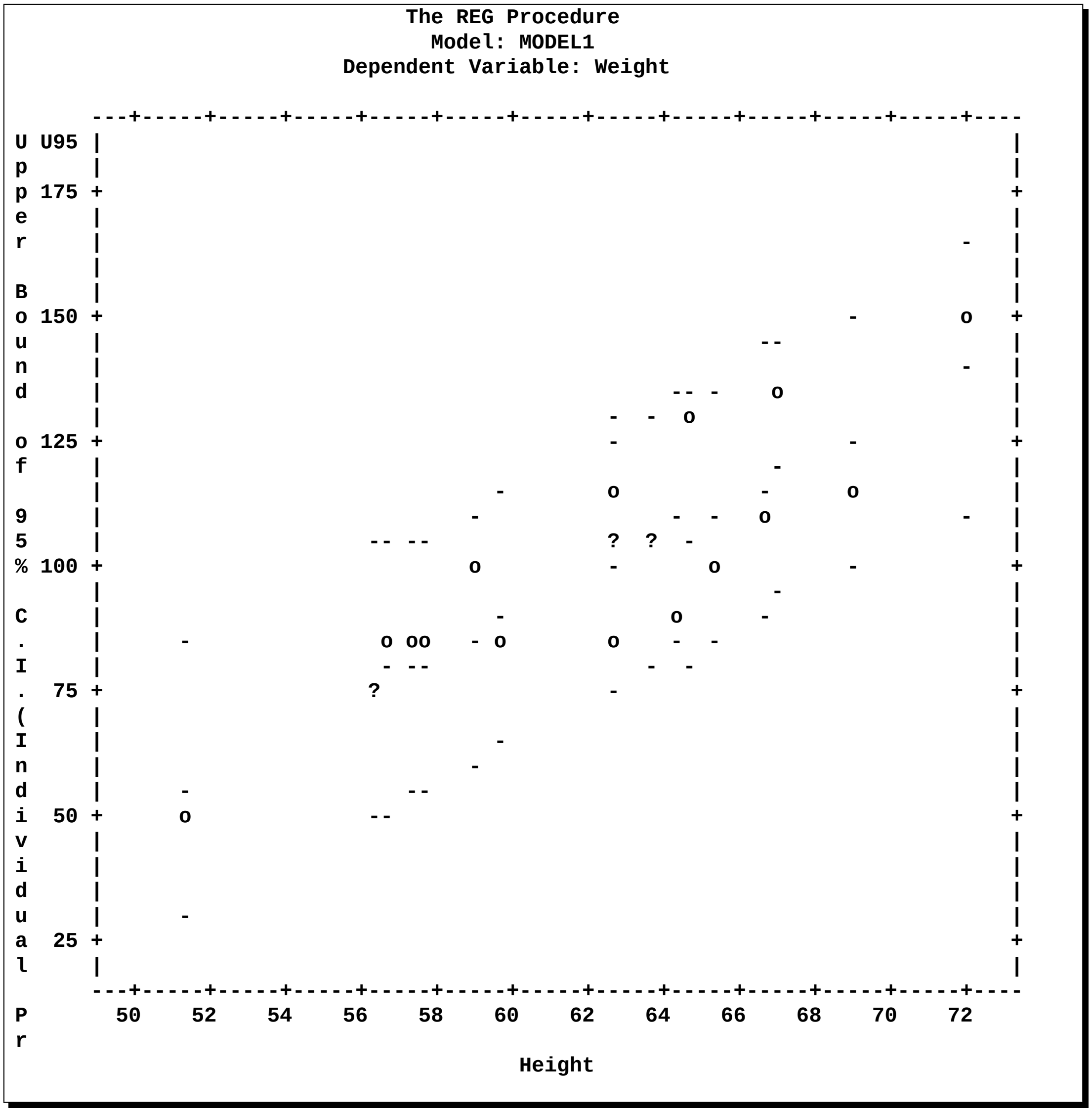
Producing Scatter Plots

The interactive PLOT statement available in PROC REG enables you to look at scatter plots of data and diagnostic statistics. These plots can help you to evaluate the model and detect outliers in your data. Several options enable you to place multiple plots on a single page, superimpose plots, and collect plots to be overlaid by later plots. The PAINT statement can be used to highlight points on a plot. See the section "Painting Scatter Plots" on page 3889 for more information on painting.

The Class data set introduced in is used in the following examples.

You can superimpose several plots with the OVERLAY option. With the following statements, a plot of Weight against Height is overlaid with plots of the predicted values and the 95% prediction intervals. The model on which the statistics are based is the full model including Height and Age. These statements produce Figure 61.32:

```
proc reg data=Class lineprinter;
   model Weight=Height Age / noprint;
   plot (ucl. lcl. p.)*Height='-' Weight*Height
        / overlay symbol='o';
run;
```



```
                               The REG Procedure
                                 Model: MODEL1
                          Dependent Variable: Weight

      ---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
U U95 |                                                                        |
p     |                                                                        |
p 175 +                                                                        +
e     |                                                                        |
r     |                                                                    -   |
      |                                                                        |
B     |                                                                        |
o 150 +                                                           -        o   +
u     |                                                    --                  |
n     |                                                                    -   |
d     |                                             -- -    o                  |
      |                                        -  -  o                         |
o 125 +                                        -                  -            +
f     |                                                     -                  |
      |                               -        o           -      o            |
9     |                             -               -  -   o               -   |
5     |                     -- --              ?  ?  -                         |
% 100 +                             o          -       o          -            +
      |                                                     -                  |
C     |                               -             o      -                   |
.     |      -               o oo   - o        o    -  -                       |
I     |                      - --                 -  -                         |
.  75 +                     ?                  -                               +
(     |                                                                        |
I     |                               -                                        |
n     |                             -                                          |
d     |      -                 --                                              |
i  50 +      o              --                                                 +
v     |                                                                        |
i     |                                                                        |
d     |                                                                        |
u     |      -                                                                 |
a  25 +                                                                        +
l     |                                                                        |
      ---+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+----
P       50    52    54    56    58    60    62    64    66    68    70    72
r
                                        Height
```

Figure 61.32. Scatter Plot Showing Data, Predicted Values, and Confidence Limits

In this plot, the data values are marked with the symbol ’o’ and the predicted values and prediction interval limits are labeled with the symbol ’-’. The plot is scaled to accommodate the points from all plots. This is an important difference from the COLLECT option, which does not rescale plots after the first plot or plots are collected. You could separate the overlaid plots by using the following statements:

```
plot;
run;
```

This places each of the four plots on a separate page, while the statements

```
plot / overlay;
run;
```

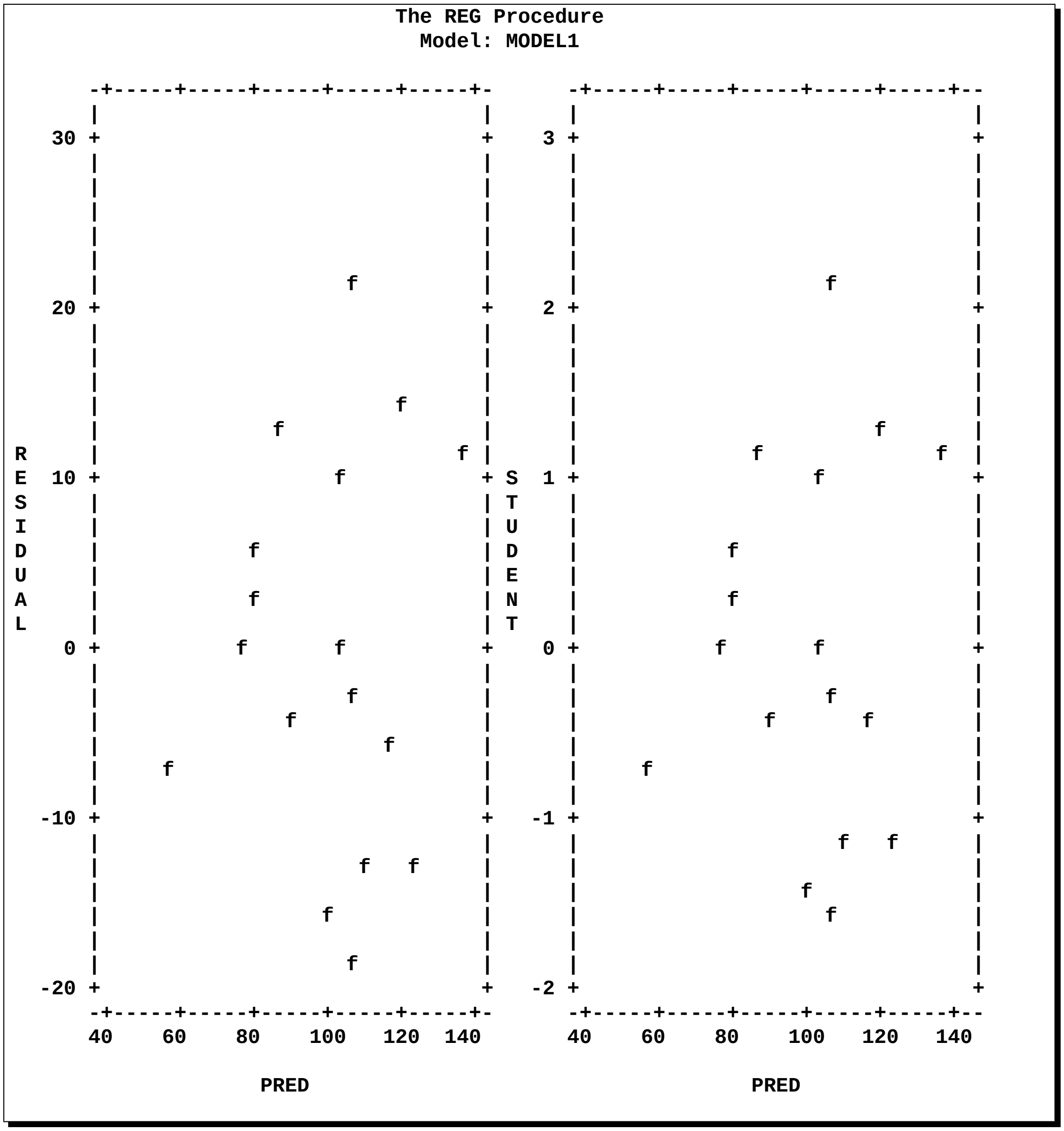
repeat the previous overlaid plot. In general, the statement

```
plot;
```

is equivalent to respecifying the most recent PLOT statement without any options. However, the COLLECT, HPLOTS=, SYMBOL=, and VPLOTS= options apply across PLOT statements and remain in effect.

The next example shows how you can overlay plots of statistics before and after a change in the model. For the full model involving Height and Age, the ordinary residuals and the studentized residuals are plotted against the predicted values. The COLLECT option causes these plots to be collected or retained for re-display later. The option HPLOTS=2 allows the two plots to appear side by side on one page. The symbol ’f’ is used on these plots to identify them as resulting from the full model. These statements produce Figure 61.33:

```
plot r.*p. student.*p. / collect hplots=2 symbol=’f’;
run;
```



```
                               The REG Procedure
                                 Model: MODEL1

      -+-----+-----+-----+-----+-----+-      -+-----+-----+-----+-----+-----+--
      |                               |      |                                |
   30 +                               +    3 +                                +
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                    f          |      |                    f           |
   20 +                               +    2 +                                +
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                        f      |      |                                |
      |              f                |      |                        f       |
R     |                             f |      |              f              f  |
E  10 +                   f           + S  1 +                   f            +
S     |                               | T    |                                |
I     |                               | U    |                                |
D     |            f                  | D    |            f                   |
U     |                               | E    |                                |
A     |            f                  | N    |            f                   |
L     |                               | T    |                                |
    0 +           f       f           +    0 +           f       f            +
      |                               |      |                                |
      |                    f          |      |                    f           |
      |               f               |      |               f       f        |
      |                       f       |      |                                |
      |     f                         |      |     f                          |
      |                               |      |                                |
  -10 +                               +   -1 +                                +
      |                               |      |                     f   f      |
      |                     f   f     |      |                                |
      |                               |      |                  f             |
      |                  f            |      |                    f           |
      |                               |      |                                |
      |                    f          |      |                                |
  -20 +                               +   -2 +                                +
      -+-----+-----+-----+-----+-----+-      -+-----+-----+-----+-----+-----+--
      40    60    80    100   120  140       40    60    80    100   120   140

                    PRED                                    PRED
```

Figure 61.33. Collecting Residual Plots for the Full Model

Note that these plots are not overlaid. The COLLECT option does not overlay the plots in one PLOT statement but retains them so that they can be overlaid by later plots. When the COLLECT option appears in a PLOT statement, the plots in that statement become the first plots in the collection.

Next, the model is reduced by deleting the Age variable. The PLOT statement requests the same plots as before but labels the points with the symbol ’r’ denoting the reduced model. The following statements produce Figure 61.34:

```
delete Age;
plot r.*p. student.*p. / symbol=’r’;
run;
```

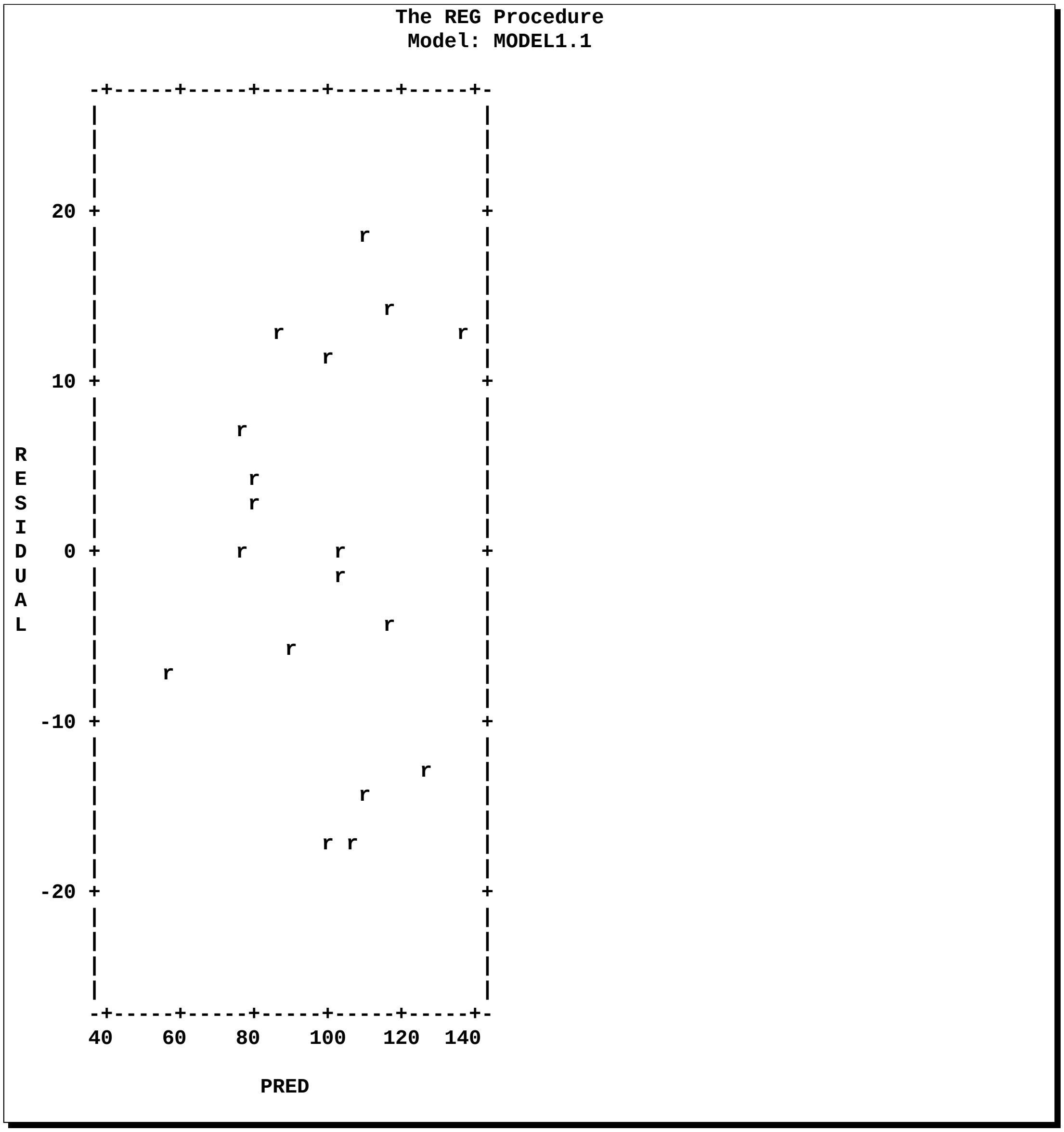
```
                               The REG Procedure
                                Model: MODEL1.1

      -+-----+-----+-----+-----+-----+-      -+-----+-----+-----+-----+-----+--
      |                               |      |                                |
  30 +                               +    3 +                                +
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                               |      |                                |
      |                    f          |      |                    f           |
  20 +                               +    2 +                                +
      |                     r         |      |                                |
      |                               |      |                     r          |
      |                               |      |                                |
      |                       rf      |      |                       r        |
      |              ?              r |      |                        f    r  |
R     |                  r          f |      |              ?   r          f  |
E  10 +                   f           + S  1 +                   f            +
S     |                               | T    |                                |
I     |           r                   | U    |                                |
D     |            f                  | D    |           rf                   |
U     |            r                  | E    |            r                   |
A     |            ?                  | N    |            ?                   |
L     |                               | T    |                                |
   0 +           ?       ?           +    0 +           ?       ?            +
      |                   r           |      |                   r            |
      |                    f          |      |                    f           |
      |               f       r       |      |               f       ?        |
      |               r       f       |      |               r                |
      |     ?                         |      |     ?                          |
      |                               |      |                                |
  -10 +                               +   -1 +                                +
      |                               |      |                     f   f      |
      |                     f   fr    |      |                     r    r     |
      |                     r         |      |                  f             |
      |                  f            |      |                  r ?           |
      |                  r r          |      |                                |
      |                    f          |      |                                |
  -20 +                               +   -2 +                                +
      -+-----+-----+-----+-----+-----+-      -+-----+-----+-----+-----+-----+--
      40    60    80    100   120  140       40    60    80    100   120   140

                    PRED                                    PRED
```

Figure 61.34. Overlaid Residual Plots for Full and Reduced Models

Notice that the COLLECT option causes the corresponding plots to be overlaid. Also notice that the DELETE statement causes the model label to be changed from MODEL1 to MODEL1.1. The points labeled ’f’ are from the full model, and points labeled ’r’ are from the reduced model. Positions labeled ’?’ contain at least one point from each model. In this example, the OVERLAY option cannot be used because all of the plots to be overlaid cannot be specified in one PLOT statement. With the COLLECT option, any changes to the model or the data used to fit the model do not affect plots collected before the changes. Collected plots are always reproduced exactly as they first appear. (Similarly, a PAINT statement does not affect plots collected before the PAINT statement is issued.)

The previous example overlays the residual plots for two different models. You may prefer to see them side by side on the same page. This can also be done with the COLLECT option by using a blank plot. Continuing from the last example, the COLLECT, HPLOTS=2, and SYMBOL='r' options are still in effect. In the following PLOT statement, the CLEAR option deletes the collected plots and allows the specified plot to begin a new collection. The plot created is the residual plot for the reduced model. These statements produce Figure 61.35:

```
plot r.*p. / clear;
run;
```

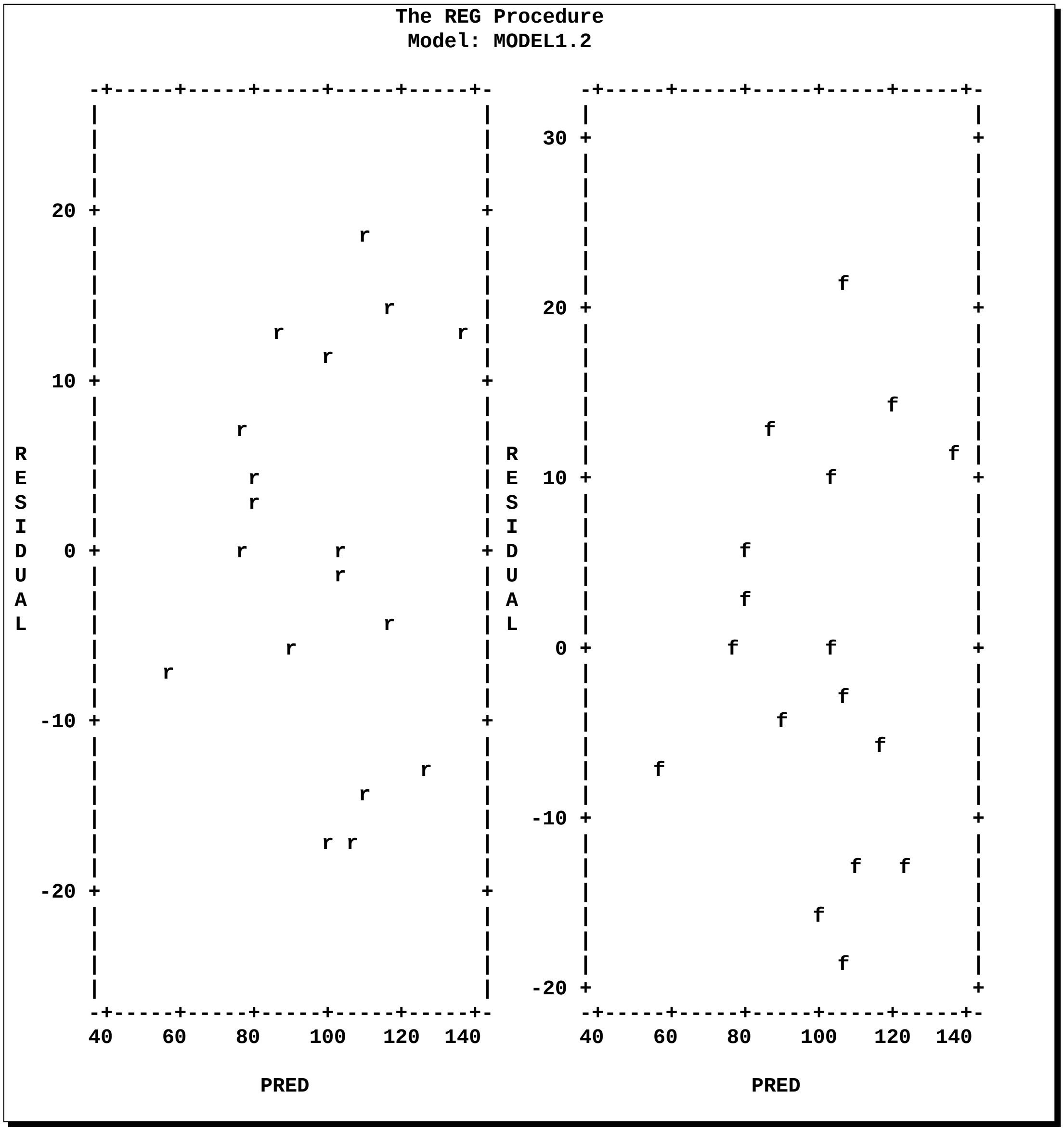


```
                               The REG Procedure
                                Model: MODEL1.1

      -+-----+-----+-----+-----+-----+-
      |                               |
      |                               |
      |                               |
      |                               |
   20 +                               +
      |                     r         |
      |                               |
      |                               |
      |                       r       |
      |              r              r |
      |                  r            |
   10 +                               +
      |                               |
      |           r                   |
R     |                               |
E     |            r                  |
S     |            r                  |
I     |                               |
D   0 +           r       r           +
U     |                   r           |
A     |                               |
L     |                       r       |
      |               r               |
      |     r                         |
      |                               |
  -10 +                               +
      |                               |
      |                          r    |
      |                     r         |
      |                               |
      |                  r r          |
      |                               |
  -20 +                               +
      |                               |
      |                               |
      |                               |
      |                               |
      -+-----+-----+-----+-----+-----+-
      40    60    80    100   120  140

                    PRED
```

Figure 61.35. Residual Plot for Reduced Model Only

The next statements add the variable AGE to the model and place the residual plot for the full model next to the plot for the reduced model. Notice that a blank plot is created in the first plot request by placing nothing between the quotes. Since the COLLECT option is in effect, this plot is superimposed on the residual plot for the reduced model. The residual plot for the full model is created by the second request. The result is the desired side-by-side plots. The NOCOLLECT option turns off the collection process after the specified plots are added and displayed. Any PLOT statements that follow show only the newly specified plots. These statements produce Figure 61.36:

```
add Age;
plot r.*p.=’’ r.*p.=’f’ / nocollect;
run;
```



```
                               The REG Procedure
                                Model: MODEL1.2

      -+-----+-----+-----+-----+-----+-       -+-----+-----+-----+-----+-----+-
      |                               |       |                               |
      |                               |    30 +                               +
      |                               |       |                               |
      |                               |       |                               |
   20 +                               +       |                               |
      |                     r         |       |                               |
      |                               |       |                               |
      |                               |       |                    f          |
      |                       r       |    20 +                               +
      |              r              r |       |                               |
      |                  r            |       |                               |
   10 +                               +       |                               |
      |                               |       |                        f      |
      |           r                   |       |              f                |
R     |                               | R     |                             f |
E     |            r                  | E  10 +                   f           +
S     |            r                  | S     |                               |
I     |                               | I     |                               |
D   0 +           r       r           + D     |            f                  |
U     |                   r           | U     |                               |
A     |                               | A     |            f                  |
L     |                       r       | L     |                               |
      |               r               |     0 +           f       f           +
      |     r                         |       |                               |
      |                               |       |                    f          |
  -10 +                               +       |               f               |
      |                               |       |                       f       |
      |                          r    |       |     f                         |
      |                     r         |       |                               |
      |                               |   -10 +                               +
      |                  r r          |       |                               |
      |                               |       |                     f   f     |
  -20 +                               +       |                               |
      |                               |       |                  f            |
      |                               |       |                               |
      |                               |       |                    f          |
      |                               |   -20 +                               +
      -+-----+-----+-----+-----+-----+-       -+-----+-----+-----+-----+-----+-
      40    60    80    100   120  140        40    60    80    100   120  140

                    PRED                                    PRED
```

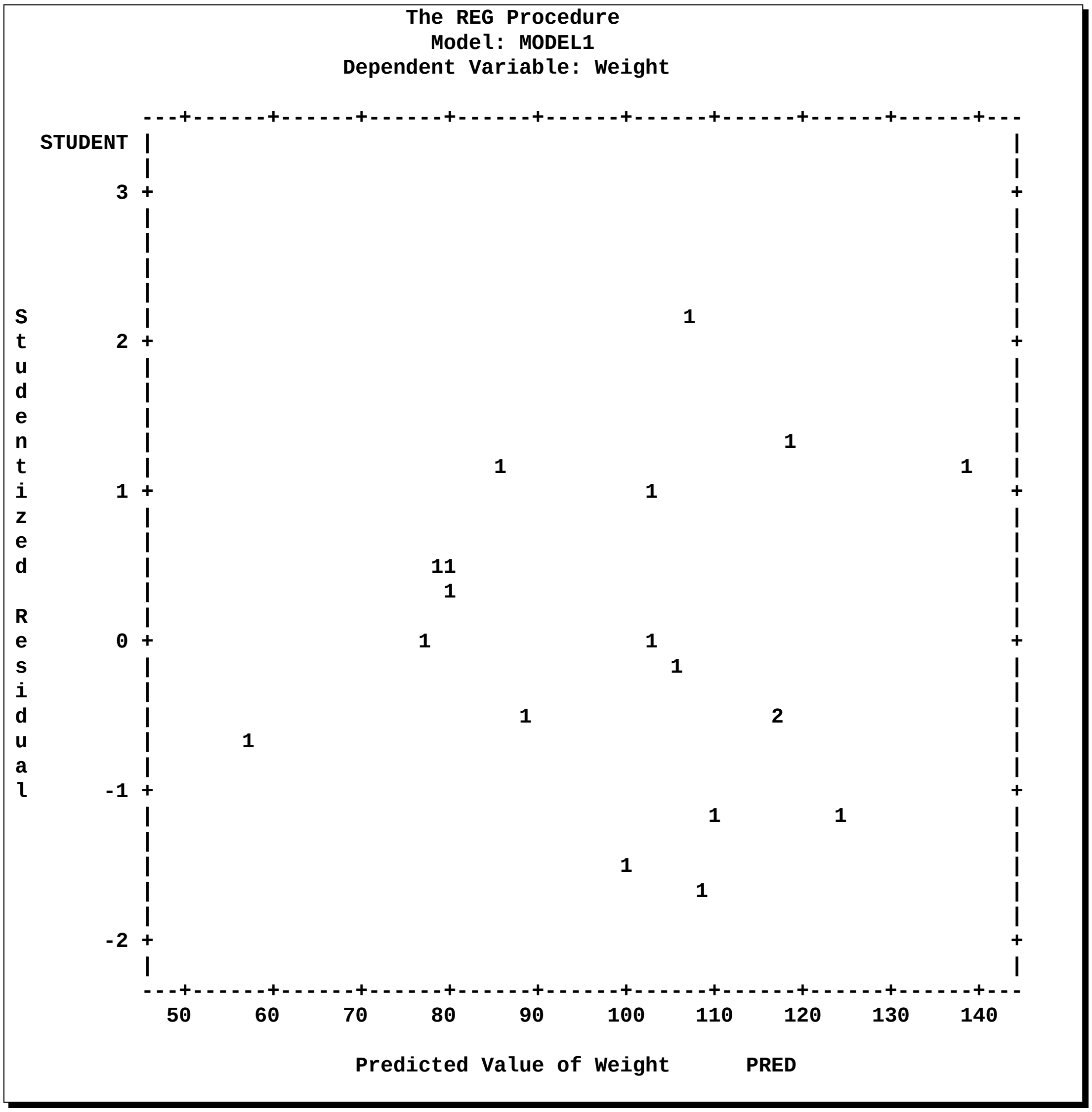
Figure 61.36. Side-by-Side Residual Plots for the Full and Reduced Models

Frequently, when the COLLECT option is in effect, you want the current and following PLOT statements to show only the specified plots. To do this, use both the CLEAR and NOCOLLECT options in the current PLOT statement.

Painting Scatter Plots

Painting scatter plots is a useful interactive tool that enables you to mark points of interest in scatter plots. Painting can be used to identify extreme points in scatter plots or to reveal the relationship between two scatter plots. The CLASS data (from the "Simple Linear Regression" section on page 3800) is used to illustrate some of these applications. First, a scatter plot of the studentized residuals against the predicted values is generated. This plot is shown in Figure 61.37.

```
proc reg data=Class lineprinter;
   model Weight=Age Height / noprint;
   plot student.*p.;
run;
```

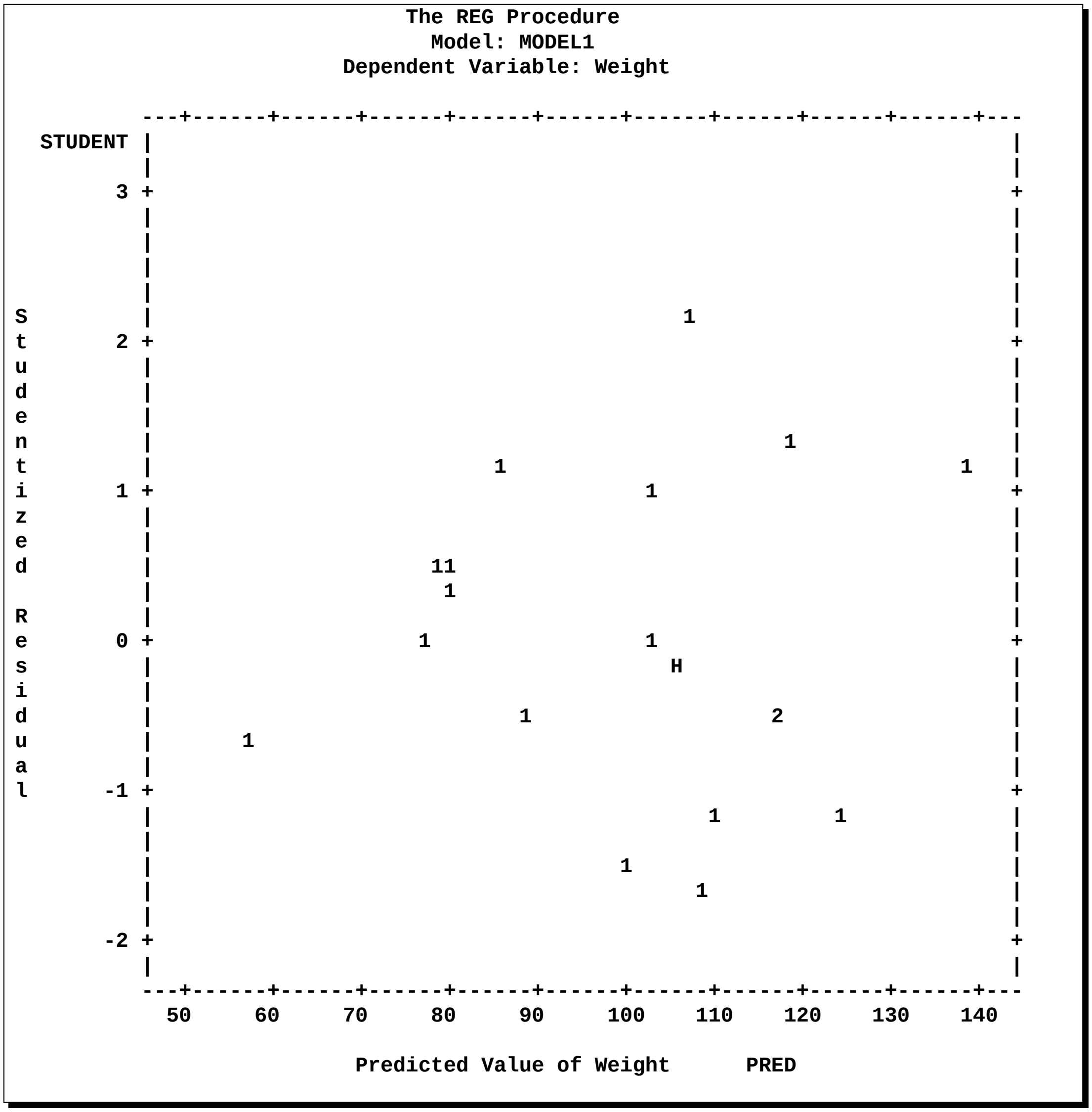


```
                               The REG Procedure
                                 Model: MODEL1
                          Dependent Variable: Weight

          ---+------+------+------+------+------+------+------+------+------+---
  STUDENT |                                                                    |
          |                                                                    |
        3 +                                                                    +
          |                                                                    |
          |                                                                    |
          |                                                                    |
          |                                                                    |
S         |                                          1                         |
t       2 +                                                                    +
u         |                                                                    |
d         |                                                                    |
e         |                                                                    |
n         |                                                  1                 |
t         |                           1                                    1   |
i       1 +                                       1                            +
z         |                                                                    |
e         |                                                                    |
d         |                      11                                            |
          |                       1                                            |
R         |                                                                    |
e       0 +                     1                 1                            +
s         |                                         1                          |
i         |                                                                    |
d         |                             1                   2                  |
u         |       1                                                            |
a         |                                                                    |
l      -1 +                                                                    +
          |                                            1         1             |
          |                                                                    |
          |                                     1                              |
          |                                           1                        |
          |                                                                    |
       -2 +                                                                    +
          |                                                                    |
          ---+------+------+------+------+------+------+------+------+------+---
            50     60     70     80     90     100    110    120    130    140

                           Predicted Value of Weight      PRED
```

Figure 61.37. Plotting Studentized Residuals Against Predicted Values

Then, the following statements identify the observation ’Henry’ in the scatter plot and produce Figure 61.38:

```
paint Name=’Henry’ / symbol = ’H’;
plot;
run;
```



```
                               The REG Procedure
                                 Model: MODEL1
                          Dependent Variable: Weight

          ---+------+------+------+------+------+------+------+------+------+---
  STUDENT |                                                                    |
          |                                                                    |
        3 +                                                                    +
          |                                                                    |
          |                                                                    |
          |                                                                    |
          |                                                                    |
S         |                                          1                         |
t       2 +                                                                    +
u         |                                                                    |
d         |                                                                    |
e         |                                                                    |
n         |                                                  1                 |
t         |                           1                                    1   |
i       1 +                                       1                            +
z         |                                                                    |
e         |                                                                    |
d         |                      11                                            |
          |                       1                                            |
R         |                                                                    |
e       0 +                     1                 1                            +
s         |                                         H                          |
i         |                                                                    |
d         |                             1                   2                  |
u         |       1                                                            |
a         |                                                                    |
l      -1 +                                                                    +
          |                                            1         1             |
          |                                                                    |
          |                                     1                              |
          |                                           1                        |
          |                                                                    |
       -2 +                                                                    +
          |                                                                    |
          ---+------+------+------+------+------+------+------+------+------+---
            50     60     70     80     90    100    110    120    130    140

                           Predicted Value of Weight      PRED
```

Figure 61.38. Painting One Observation

Next, the following statements identify observations with large absolute residuals:

```
paint student.>=2 or student.<=-2 / symbol='s';
plot;
run;
```

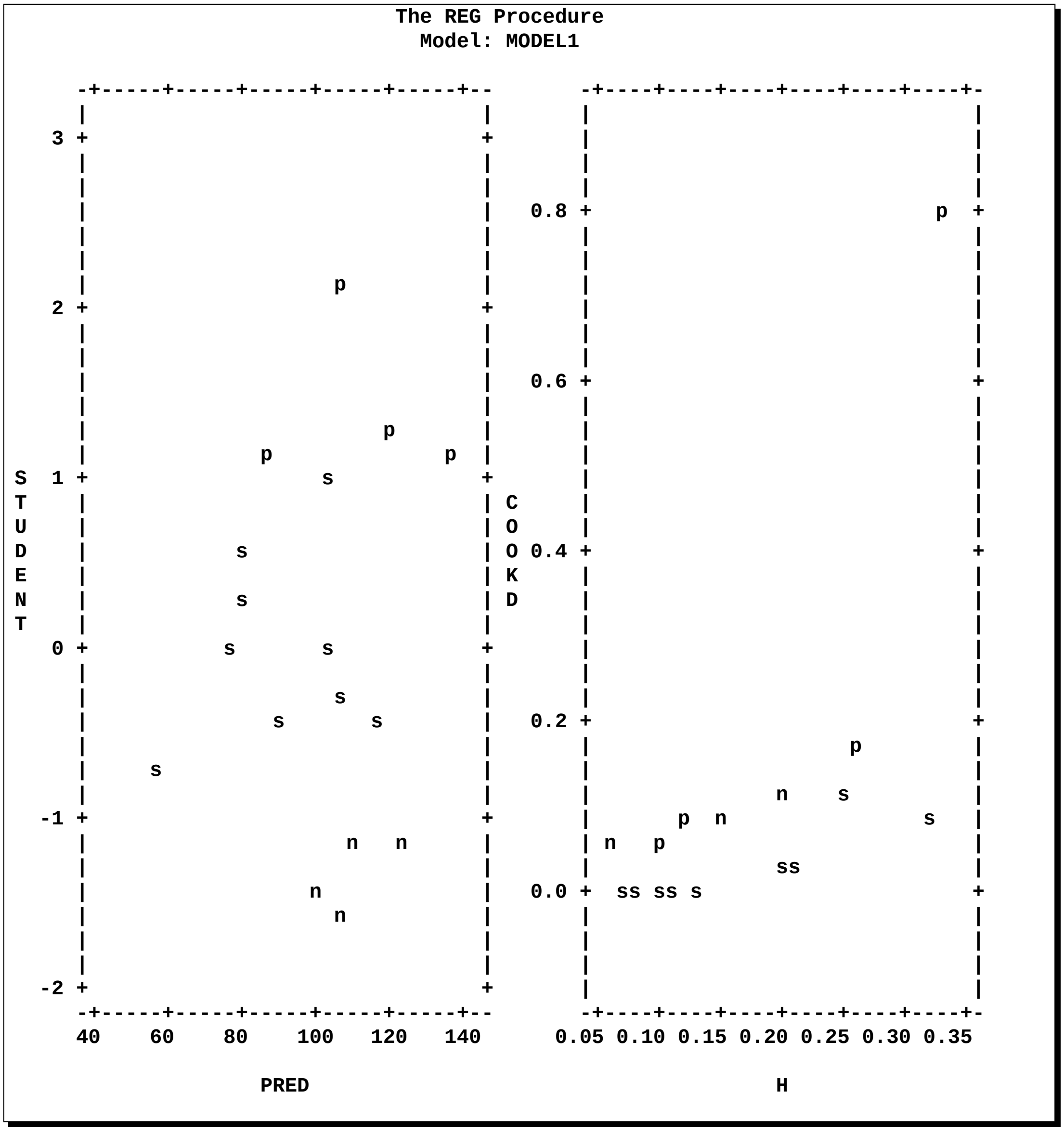
The log shows the observation numbers found with these conditions and gives the painting symbol and the number of observations found. Note that the previous PAINT statement is also used in the PLOT statement. Figure 61.39 shows the scatter plot produced by the preceding statements.

```
                               The REG Procedure
                                 Model: MODEL1
                          Dependent Variable: Weight

          ---+------+------+------+------+------+------+------+------+------+---
  STUDENT |                                                                    |
          |                                                                    |
        3 +                                                                    +
          |                                                                    |
          |                                                                    |
          |                                                                    |
          |                                                                    |
S         |                                          s                         |
t       2 +                                                                    +
u         |                                                                    |
d         |                                                                    |
e         |                                                                    |
n         |                                                  1                 |
t         |                           1                                    1   |
i       1 +                                       1                            +
z         |                                                                    |
e         |                                                                    |
d         |                      11                                            |
          |                       1                                            |
R         |                                                                    |
e       0 +                     1                 1                            +
s         |                                         H                          |
i         |                                                                    |
d         |                             1                   2                  |
u         |       1                                                            |
a         |                                                                    |
l      -1 +                                                                    +
          |                                            1         1             |
          |                                                                    |
          |                                     1                              |
          |                                           1                        |
          |                                                                    |
       -2 +                                                                    +
          |                                                                    |
          ---+------+------+------+------+------+------+------+------+------+---
            50     60     70     80     90    100    110    120    130    140

                           Predicted Value of Weight      PRED
```

Figure 61.39. Painting Several Observations

The following statements relate two different scatter plots. These statements produce Figure 61.40.

```
paint student.>=1 / symbol=’p’;
paint student.<1 and student.>-1 / symbol=’s’;
paint student.<=-1 / symbol=’n’;
plot student. * p. cookd. * h. / hplots=2;
run;
```



```
                              The REG Procedure
                                Model: MODEL1

     -+-----+-----+-----+-----+-----+--       -+----+----+----+----+----+----+-
     |                                |       |                               |
   3 +                                +       |                               |
     |                                |       |                               |
     |                                |       |                               |
     |                                |   0.8 +                            p  +
     |                                |       |                               |
     |                                |       |                               |
     |                    p           |       |                               |
   2 +                                +       |                               |
     |                                |       |                               |
     |                                |       |                               |
     |                                |   0.6 +                               +
     |                                |       |                               |
     |                        p       |       |                               |
     |              p              p  |       |                               |
S  1 +                   s            +       |                               |
T    |                                | C     |                               |
U    |                                | O     |                               |
D    |            s                   | O 0.4 +                               +
E    |                                | K     |                               |
N    |            s                   | D     |                               |
T    |                                |       |                               |
   0 +           s       s            +       |                               |
     |                                |       |                               |
     |                    s           |       |                               |
     |               s       s        |   0.2 +                               +
     |                                |       |                     p         |
     |     s                          |       |                               |
     |                                |       |               n    s          |
  -1 +                                +       |       p  n                s   |
     |                     n   n      |       | n   p                         |
     |                                |       |               ss              |
     |                  n             |   0.0 +  ss ss s                      +
     |                    n           |       |                               |
     |                                |       |                               |
     |                                |       |                               |
  -2 +                                +       |                               |
     -+-----+-----+-----+-----+-----+--       -+----+----+----+----+----+----+-
     40    60    80    100   120   140      0.05 0.10 0.15 0.20 0.25 0.30 0.35

                    PRED                                      H
```

Figure 61.40. Painting Observations on More than One Plot

Models of Less Than Full Rank

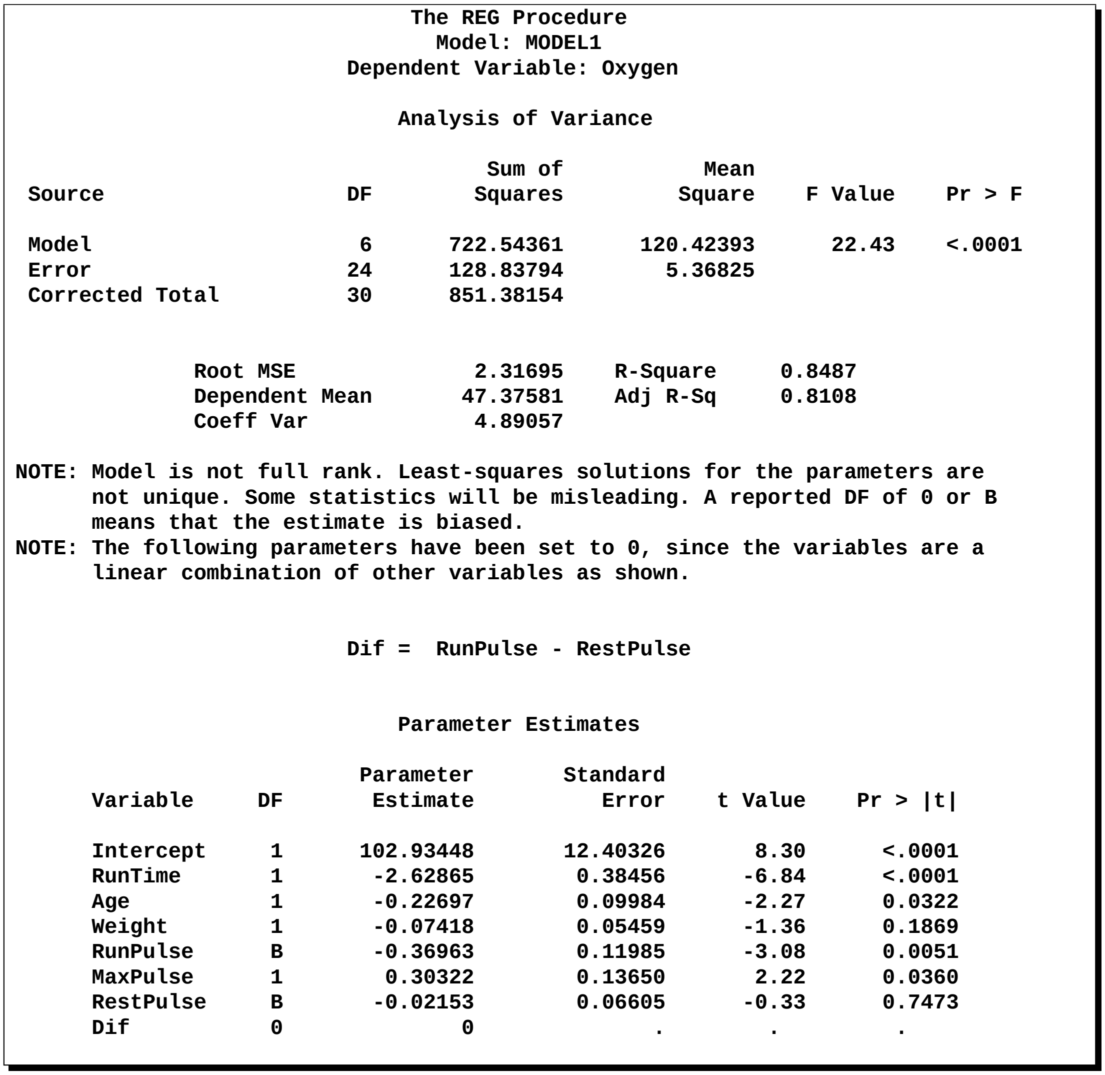
If the model is not full rank, there are an infinite number of least-squares solutions for the estimates. PROC REG chooses a nonzero solution for all variables that are linearly independent of previous variables and a zero solution for other variables. This solution corresponds to using a generalized inverse in the normal equations, and the expected values of the estimates are the Hermite normal form of $\mathbf{X}$ multiplied by the true parameters:

$$E(\mathbf{b}) = (\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X})\beta$$

Degrees of freedom for the zeroed estimates are reported as zero. The hypotheses that are not testable have t tests reported as missing. The message that the model is not full rank includes a display of the relations that exist in the matrix.

The next example uses the fitness data from Example 61.1 on page 3924. The variable Dif=RunPulse−RestPulse is created. When this variable is included in the model along with RunPulse and RestPulse, there is a linear dependency (or exact collinearity) between the independent variables. Figure 61.41 shows how this problem is diagnosed.

```
data fit2;
   set fitness; Dif=RunPulse-RestPulse;
proc reg data=fit2;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse Dif;
run;
```



```
                             The REG Procedure
                               Model: MODEL1
                        Dependent Variable: Oxygen

                           Analysis of Variance

                                      Sum of           Mean
 Source                   DF         Squares         Square    F Value    Pr > F

 Model                     6       722.54361      120.42393      22.43    <.0001
 Error                    24       128.83794        5.36825
 Corrected Total          30       851.38154


             Root MSE              2.31695    R-Square     0.8487
             Dependent Mean       47.37581    Adj R-Sq     0.8108
             Coeff Var             4.89057

NOTE: Model is not full rank. Least-squares solutions for the parameters are
      not unique. Some statistics will be misleading. A reported DF of 0 or B
      means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a
      linear combination of other variables as shown.


                        Dif =  RunPulse - RestPulse


                           Parameter Estimates

                             Parameter       Standard
      Variable      DF        Estimate          Error    t Value    Pr > |t|

      Intercept      1       102.93448       12.40326       8.30      <.0001
      RunTime        1        -2.62865        0.38456      -6.84      <.0001
      Age            1        -0.22697        0.09984      -2.27      0.0322
      Weight         1        -0.07418        0.05459      -1.36      0.1869
      RunPulse       B        -0.36963        0.11985      -3.08      0.0051
      MaxPulse       1         0.30322        0.13650       2.22      0.0360
      RestPulse      B        -0.02153        0.06605      -0.33      0.7473
      Dif            0               0              .          .           .
```

Figure 61.41. Model That Is Not Full Rank: REG Procedure

PROC REG produces a message informing you that the model is less than full rank. Parameters with DF=0 are not estimated, and parameters with DF=B are biased. In addition, the form of the linear dependency among the regressors is displayed.

Collinearity Diagnostics

When a regressor is nearly a linear combination of other regressors in the model, the affected estimates are unstable and have high standard errors. This problem is called *collinearity* or *multicollinearity*. It is a good idea to find out which variables are nearly collinear with which other variables. The approach in PROC REG follows that of Belsley, Kuh, and Welsch (1980). PROC REG provides several methods for detecting collinearity with the COLLIN, COLLINOINT, TOL, and VIF options.

The COLLIN option in the MODEL statement requests that a collinearity analysis be performed. First, $\mathbf{X}'\mathbf{X}$ is scaled to have 1s on the diagonal. If you specify the COLLINOINT option, the intercept variable is adjusted out first. Then the eigenvalues and eigenvectors are extracted. The analysis in PROC REG is reported with eigenvalues of $\mathbf{X}'\mathbf{X}$ rather than singular values of $\mathbf{X}$. The eigenvalues of $\mathbf{X}'\mathbf{X}$ are the squares of the singular values of $\mathbf{X}$.

The condition indices are the square roots of the ratio of the largest eigenvalue to each individual eigenvalue. The largest condition index is the condition number of the scaled $\mathbf{X}$ matrix. Belsey, Kuh, and Welsch (1980) suggest that, when this number is around 10, weak dependencies may be starting to affect the regression estimates. When this number is larger than 100, the estimates may have a fair amount of numerical error (although the statistical standard error almost always is much greater than the numerical error).

For each variable, PROC REG produces the proportion of the variance of the estimate accounted for by each principal component. A collinearity problem occurs when a component associated with a high condition index contributes strongly (variance proportion greater than about 0.5) to the variance of two or more variables.

The VIF option in the MODEL statement provides the Variance Inflation Factors (VIF). These factors measure the inflation in the variances of the parameter estimates due to collinearities that exist among the regressor (independent) variables. There are no formal criteria for deciding if a VIF is large enough to affect the predicted values.

The TOL option requests the tolerance values for the parameter estimates. The tolerance is defined as $1/VIF$.

For a complete discussion of the preceding methods, refer to Belsley, Kuh, and Welsch (1980). For a more detailed explanation of using the methods with PROC REG, refer to Freund and Littell (1986).

This example uses the COLLIN option on the fitness data found in Example 61.1 on page 3924. The following statements produce Figure 61.42.

```
proc reg data=fitness;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse
         / tol vif collin;
run;
```

```
                                        The REG Procedure
                                          Model: MODEL1
                                   Dependent Variable: Oxygen

                                      Analysis of Variance

                                                   Sum of           Mean
                      Source                DF        Squares         Square    F Value    Pr > F

                      Model                  6      722.54361      120.42393      22.43    <.0001
                      Error                 24      128.83794        5.36825
                      Corrected Total       30      851.38154


                                   Root MSE              2.31695    R-Square     0.8487
                                   Dependent Mean       47.37581    Adj R-Sq     0.8108
                                   Coeff Var             4.89057


                                      Parameter Estimates

                                 Parameter       Standard                                          Variance
            Variable     DF       Estimate          Error    t Value    Pr > |t|      Tolerance      Inflation

            Intercept     1      102.93448       12.40326       8.30      <.0001              .              0
            RunTime       1       -2.62865        0.38456      -6.84      <.0001        0.62859        1.59087
            Age           1       -0.22697        0.09984      -2.27      0.0322        0.66101        1.51284
            Weight        1       -0.07418        0.05459      -1.36      0.1869        0.86555        1.15533
            RunPulse      1       -0.36963        0.11985      -3.08      0.0051        0.11852        8.43727
            MaxPulse      1        0.30322        0.13650       2.22      0.0360        0.11437        8.74385
            RestPulse     1       -0.02153        0.06605      -0.33      0.7473        0.70642        1.41559


                                     Collinearity Diagnostics

                        Condition  ---------------------------------Proportion of Variation---------------------------------
Number    Eigenvalue        Index    Intercept       RunTime           Age        Weight      RunPulse      MaxPulse     RestPulse

     1       6.94991      1.00000   0.00002326    0.00021086    0.00015451    0.00019651    0.00000862    0.00000634    0.00027850
     2       0.01868     19.29087      0.00218       0.02522       0.14632       0.01042    0.00000244    0.00000743       0.39064
     3       0.01503     21.50072   0.00061541       0.12858       0.15013       0.23571       0.00119       0.00125       0.02809
     4       0.00911     27.62115      0.00638       0.60897       0.03186       0.18313       0.00149       0.00123       0.19030
     5       0.00607     33.82918      0.00133       0.12501       0.11284       0.44442       0.01506       0.00833       0.36475
     6       0.00102     82.63757      0.79966       0.09746       0.49660       0.10330       0.06948       0.00561       0.02026
     7    0.00017947    196.78560      0.18981       0.01455       0.06210       0.02283       0.91277       0.98357       0.00568
```

Figure 61.42. Regression Using the TOL, VIF, and COLLIN Options

Model Fit and Diagnostic Statistics

This section gathers the formulas for the statistics available in the MODEL, PLOT, and OUTPUT statements. The model to be fit is $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, and the parameter estimate is denoted by $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$. The subscript i denotes values for the ith observation, the parenthetical subscript (i) means that the statistic is computed using all observations except the ith observation, and the subscript jj indicates the jth diagonal matrix entry. The ALPHA= option in the PROC REG or MODEL statement is used to set the α value for the t statistics.

Table 61.6 contains the summary statistics for assessing the fit of the model.

Table 61.6. Formulas and Definitions for Model Fit Summary Statistics

MODEL Option or Statistic	**Definition or Formula**
n	the number of observations
p	the number of parameters including the intercept
i	1 if there is an intercept, 0 otherwise
$\hat{\sigma}^2$	the estimate of pure error variance from the SIGMA= option or from fitting the full model
SST_0	the uncorrected total sum of squares for the dependent variable
SST_1	the total sum of squares corrected for the mean for the dependent variable
SSE	the error sum of squares
MSE	$\frac{\text{SSE}}{n-p}$
R^2	$1 - \frac{\text{SSE}}{\text{SST}_i}$
ADJRSQ	$1 - \frac{(n-i)(1-R^2)}{n-p}$
AIC	$n \ln \left(\frac{\text{SSE}}{n} \right) + 2p$
BIC	$n \ln \left(\frac{\text{SSE}}{n} \right) + 2(p+2)q - 2q^2 \text{ where } q = \frac{n\hat{\sigma}^2}{\text{SSE}}$
CP (C_p)	$\frac{\text{SSE}}{\hat{\sigma}^2} + 2p - n$
GMSEP	$\frac{\text{MSE}(n+1)(n-2)}{n(n-p-1)} = \frac{1}{n} S_p (n+1)(n-2)$
JP (J_p)	$\frac{n+p}{n}\text{MSE}$
PC	$\frac{n+p}{n-p}(1-R^2) = J_p \left(\frac{n}{\text{SST}_i} \right)$
PRESS	the sum of squares of $predr_i$ (see Table 61.7)
RMSE	$\sqrt{\text{MSE}}$
SBC	$n \ln \left(\frac{\text{SSE}}{n} \right) + p \ln(n)$
SP (S_p)	$\frac{\text{MSE}}{n-p-1}$

Table 61.7 contains the diagnostic statistics and their formulas; these formulas and further information can be found in Chapter 2, “Introduction to Regression Procedures,” and in the “Influence Diagnostics” section on page 3898. Each statistic is computed for each observation.

Table 61.7. Formulas and Definitions for Diagnostic Statistics

MODEL Option or Statistic	Formula
PRED ($\widehat{Y}_i$)	$\mathbf{X}_i\mathbf{b}$
RES (r_i)	$\mathbf{Y}_i - \widehat{Y}_i$
H (h_i)	$x_i(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i'$
STDP	$\sqrt{h_i\widehat{\sigma}^2}$
STDI	$\sqrt{(1+h_i)\widehat{\sigma}^2}$
STDR	$\sqrt{(1-h_i)\widehat{\sigma}^2}$
LCL	$\widehat{Y}_i - t_{\frac{\alpha}{2}}\mathrm{STDI}$
LCLM	$\widehat{Y}_i - t_{\frac{\alpha}{2}}\mathrm{STDP}$
UCL	$\widehat{Y}_i + t_{\frac{\alpha}{2}}\mathrm{STDI}$
UCLM	$\widehat{Y}_i + t_{\frac{\alpha}{2}}\mathrm{STDP}$
STUDENT	$\frac{r_i}{\mathrm{STDR}_i}$
RSTUDENT	$\frac{r_i}{\hat{\sigma}_{(i)}\sqrt{1-h_i}}$
COOKD	$\frac{1}{p}\mathrm{STUDENT}^2(\frac{\mathrm{STDP}}{\mathrm{STDR}^2})$
COVRATIO	$\frac{\det(\hat{\sigma}^2_{(i)}(\mathbf{X}'_{(i)}\mathbf{x}_{(i)})^{-1})}{\det(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1})}$
DFFITS	$\frac{(\widehat{Y}_i - \widehat{Y}_{(i)})}{(\hat{\sigma}_{(i)}\sqrt{h_i})}$
DFBETAS$_j$	$\frac{\mathbf{b}_j - \mathbf{b}_{(i)j}}{\hat{\sigma}_{(i)}\sqrt{(\mathbf{X}'\mathbf{X})_{jj}}}$
PRESS($predr_i$)	$\frac{r_i}{1-h_i}$

Influence Diagnostics

This section discusses the INFLUENCE option, which produces several influence statistics, and the PARTIAL option, which produces partial regression leverage plots.

The INFLUENCE Option

The INFLUENCE option (in the MODEL statement) requests the statistics proposed by Belsley, Kuh, and Welsch (1980) to measure the influence of each observation on the estimates. Influential observations are those that, according to various criteria, appear to have a large influence on the parameter estimates.

Let $\mathbf{b}(i)$ be the parameter estimates after deleting the ith observation; let $s(i)^2$ be the variance estimate after deleting the ith observation; let $\mathbf{X}(i)$ be the $\mathbf{X}$ matrix without the ith observation; let $\hat{y}(i)$ be the ith value predicted without using the ith observation; let $r_i = y_i - \hat{y}_i$ be the ith residual; and let h_i be the ith diagonal of the projection matrix for the predictor space, also called the *hat matrix*:

$$h_i = \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i'$$

Belsley, Kuh, and Welsch propose a cutoff of $2p/n$, where n is the number of observations used to fit the model and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated.

For each observation, PROC REG first displays the residual, the studentized residual (RSTUDENT), and the h_i. The studentized residual RSTUDENT differs slightly from STUDENT since the error variance is estimated by $s^2_{(i)}$ without the ith observation, not by s^2. For example,

$$\text{RSTUDENT} = \frac{r_i}{s_{(i)}\sqrt{(1-h_i)}}$$

Observations with RSTUDENT larger than 2 in absolute value may need some attention.

The COVRATIO statistic measures the change in the determinant of the covariance matrix of the estimates by deleting the ith observation:

$$\text{COVRATIO} = \frac{\det\left(s^2(i)(\mathbf{X}_{(i)}'\mathbf{X}_{(i)})^{-1}\right)}{\det\left(s^2(\mathbf{X}'\mathbf{X})^{-1}\right)}$$

Belsley, Kuh, and Welsch suggest that observations with

$$|\text{COVRATIO} - 1| \geq \frac{3p}{n}$$

where p is the number of parameters in the model and n is the number of observations used to fit the model, are worth investigation.

The DFFITS statistic is a scaled measure of the change in the predicted value for the ith observation and is calculated by deleting the ith observation. A large value indicates that the observation is very influential in its neighborhood of the $\mathbf{X}$ space.

$$\text{DFFITS} = \frac{\hat{y}_i - \hat{y}_{(i)}}{s_{(i)}\sqrt{h_{(i)}}}$$

Large values of DFFITS indicate influential observations. A general cutoff to consider is 2; a size-adjusted cutoff recommended by Belsley, Kuh, and Welsch is $2\sqrt{p/n}$, where n and p are as defined previously.

The DFFITS statistic is very similar to Cook's D, defined in the section "Predicted and Residual Values" on page 3879.

The DFBETAS statistics are the scaled measures of the change in each parameter estimate and are calculated by deleting the ith observation:

$$\text{DFBETAS}_j = \frac{b_j - b_{(i)j}}{s_{(i)}\sqrt{(\mathbf{X}'\mathbf{X})_{jj}}}$$

where $(\mathbf{X}'\mathbf{X})_{jj}$ is the (j,j)th element of $(\mathbf{X}'\mathbf{X})^{-1}$.

In general, large values of DFBETAS indicate observations that are influential in estimating a given parameter. Belsley, Kuh, and Welsch recommend 2 as a general cutoff value to indicate influential observations and $2/\sqrt{n}$ as a size-adjusted cutoff.

Figure 61.43 shows the tables produced by the INFLUENCE option for the population example (the section "Polynomial Regression" beginning on page 3804). See Figure 61.30 for the fitted regression equation.

```
proc reg data=USPopulation;
   model Population=Year YearSq / influence;
run;
```

```
                                 The REG Procedure
                                  Model: MODEL1
                          Dependent Variable: Population

                                 Output Statistics

                              Hat Diag       Cov              -----------DFBETAS-----------
Obs   Residual   RStudent        H         Ratio     DFFITS   Intercept      Year     YearSq

  1    -2.2837    -0.9361     0.3429     1.5519    -0.6762     -0.4924    0.4862    -0.4802
  2    -0.4146    -0.1540     0.2356     1.5325    -0.0855     -0.0540    0.0531    -0.0523
  3     0.6696     0.2379     0.1632     1.3923     0.1050      0.0517   -0.0505     0.0494
  4     0.8849     0.3065     0.1180     1.3128     0.1121      0.0335   -0.0322     0.0310
  5     0.5923     0.2021     0.0933     1.2883     0.0648      0.0040   -0.0032     0.0025
  6    -0.0621    -0.0210     0.0831     1.2827    -0.0063      0.0012   -0.0012     0.0013
  7    -0.1344    -0.0455     0.0824     1.2813    -0.0136      0.0054   -0.0055     0.0056
  8     0.5864     0.1994     0.0870     1.2796     0.0615     -0.0339    0.0343    -0.0347
  9     0.0934     0.0318     0.0933     1.2969     0.0102     -0.0067    0.0067    -0.0068
 10     0.2255     0.0771     0.0990     1.3040     0.0255     -0.0182    0.0183    -0.0183
 11     1.4757     0.5090     0.1022     1.2550     0.1717     -0.1272    0.1275    -0.1276
 12     1.6441     0.5680     0.1022     1.2420     0.1916     -0.1426    0.1426    -0.1424
 13     3.4065     1.2109     0.0990     1.0320     0.4013     -0.2895    0.2889    -0.2880
 14     1.5922     0.5470     0.0933     1.2345     0.1755     -0.1173    0.1167    -0.1160
 15     1.7679     0.6064     0.0870     1.2123     0.1871     -0.1076    0.1067    -0.1056
 16    -7.5642    -3.2147     0.0824     0.3286    -0.9636      0.4130   -0.4063     0.3987
 17    -7.4712    -3.1550     0.0831     0.3425    -0.9501      0.2131   -0.2048     0.1957
 18    -0.3731    -0.1272     0.0933     1.2936    -0.0408     -0.0007    0.0012    -0.0016
 19     1.2782     0.4440     0.1180     1.2906     0.1624      0.0415   -0.0432     0.0449
 20     1.0356     0.3687     0.1632     1.3741     0.1628      0.0732   -0.0749     0.0766
 21    -1.7068    -0.6406     0.2356     1.4380    -0.3557     -0.2107    0.2141    -0.2176
 22     4.7578     2.1312     0.3429     0.9113     1.5395      1.0656   -1.0793     1.0933


                          Sum of Residuals                   -4.4596E-11
                          Sum of Squared Residuals             170.97193
                          Predicted Residual SS (PRESS)        237.71229
```

Figure 61.43. Regression Using the INFLUENCE Option

In Figure 61.43, observations 16, 17, and 19 exceed the cutoff value of 2 for RSTUDENT. None of the observations exceeds the general cutoff of 2 for DFFITS or the DFBETAS, but observations 16, 17, and 19 exceed at least one of the size-adjusted cutoffs for these statistics. Observations 1 and 19 exceed the cutoff for the hat diagonals, and observations 1, 2, 16, 17, and 18 exceed the cutoffs for COVRATIO. Taken together, these statistics indicate that you should look first at observations 16, 17, and 19 and then perhaps investigate the other observations that exceeded a cutoff.

The PARTIAL Option

The PARTIAL option in the MODEL statement produces partial regression leverage plots. If the experimental ODS graphics are not in effect, this option requires the use of the LINEPRINTER option in the PROC REG statement. One plot is created for each regressor in the current full model. For example, plots are produced for regressors included by using ADD statements; plots are not produced for interim models in the various model-selection methods but only for the full model. If you use a model-selection method and the final model contains only a subset of the original regressors, the PARTIAL option still produces plots for all regressors in the full model. If the experimental ODS graphics are in effect, these plots are produced as high-resolution graphics, in panels with a maximum of six partial regression leverage plots plots per panel. Multiple panels are displayed for models with more than six regressors.

For a given regressor, the partial regression leverage plot is the plot of the dependent variable and the regressor after they have been made orthogonal to the other regressors in the model. These can be obtained by plotting the residuals for the dependent variable against the residuals for the selected regressor, where the residuals for the dependent variable are calculated with the selected regressor omitted, and the residuals for the selected regressor are calculated from a model where the selected regressor is regressed on the remaining regressors. A line fit to the points has a slope equal to the parameter estimate in the full model.

When the experimental ODS graphics are not in effect, points in the plot are marked by the number of replicates appearing at one position. The symbol '*' is used if there are ten or more replicates. If an ID statement is specified, the left-most nonblank character in the value of the ID variable is used as the plotting symbol.

The following statements use the fitness data in Example 61.1 on page 3924 with the PARTIAL option and the ODS GRAPHICS statement to produce the partial regression leverage plots. The plots are shown in Figure 61.44. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the REG procedure, see the “ODS Graphics” section on page 3922.

```
ods html;
ods graphics on;

proc reg data=fitness;
   model Oxygen=RunTime Weight Age / partial;
run;

ods graphics off;
ods html close;
```

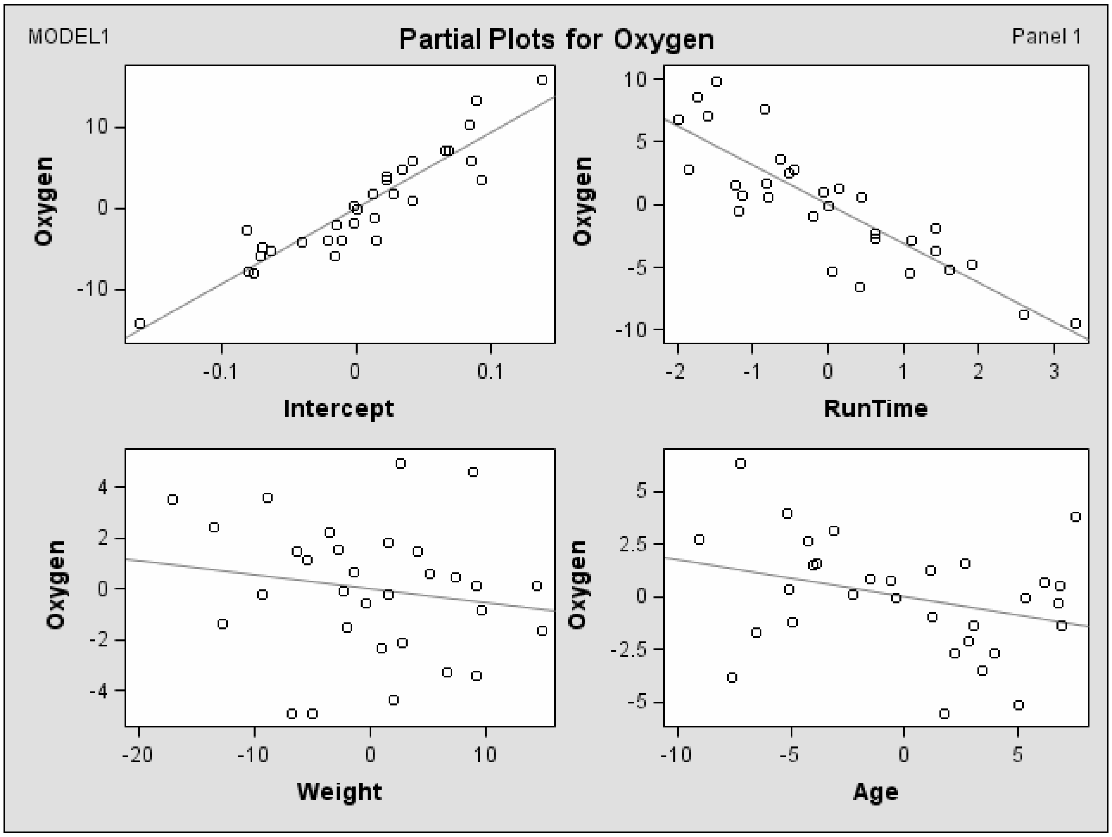


Figure 61.44. Partial Regression Leverage Plots (Experimental)

The following statements create a similar panel of partial regression plots using the OUTPUT dataset and the GPLOT procedure. Four plots (created by regressing Oxygen and one of the variables on the remaining variables) are displayed in Figure 61.45. Notice that the Int variable is explicitly added to be used as the intercept term.

```
data fitness2;
   set fitness;
   Int=1;
proc reg data=fitness2 noprint;
   model Oxygen Int = RunTime Weight Age / noint;
   output out=temp r=ry rx;
symbol1 c=blue;
proc gplot data=temp;
   plot ry*rx / cframe=ligr;
   label ry='Oxygen'
         rx='Intercept';
run;
```

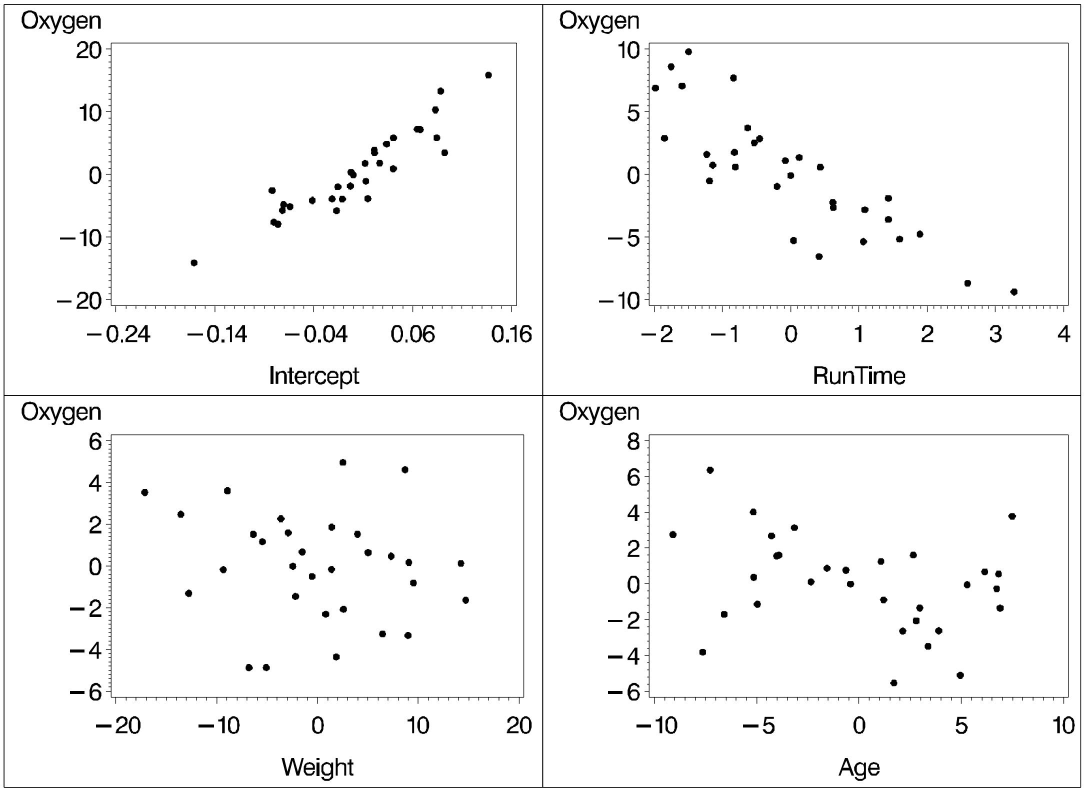


Figure 61.45. Partial Regression Leverage Plots

Reweighting Observations in an Analysis

Reweighting observations is an interactive feature of PROC REG that enables you to change the weights of observations used in computing the regression equation. Observations can also be deleted from the analysis (not from the data set) by changing their weights to zero. The Class data (in the "Getting Started" section on page 3800) are used to illustrate some of the features of the REWEIGHT statement. First, the full model is fit, and the residuals are displayed in Figure 61.46.

```
proc reg data=Class;
   model Weight=Age Height / p;
   id Name;
run;
```

```
                         The REG Procedure
                           Model: MODEL1
                    Dependent Variable: Weight

                         Output Statistics

                            Dependent    Predicted
      Obs    Name            Variable        Value     Residual

        1    Alfred          112.5000     124.8686     -12.3686
        2    Alice            84.0000      78.6273       5.3727
        3    Barbara          98.0000     110.2812     -12.2812
        4    Carol           102.5000     102.5670      -0.0670
        5    Henry           102.5000     105.0849      -2.5849
        6    James            83.0000      80.2266       2.7734
        7    Jane             84.5000      89.2191      -4.7191
        8    Janet           112.5000     102.7663       9.7337
        9    Jeffrey          84.0000     100.2095     -16.2095
       10    John             99.5000      86.3415      13.1585
       11    Joyce            50.5000      57.3660      -6.8660
       12    Judy             90.0000     107.9625     -17.9625
       13    Louise           77.0000      76.6295       0.3705
       14    Mary            112.0000     117.1544      -5.1544
       15    Philip          150.0000     138.2164      11.7836
       16    Robert          128.0000     107.2043      20.7957
       17    Ronald          133.0000     118.9529      14.0471
       18    Thomas           85.0000      79.6676       5.3324
       19    William         112.0000     117.1544      -5.1544


         Sum of Residuals                              0
         Sum of Squared Residuals             2120.09974
         Predicted Residual SS (PRESS)        3272.72186
```

Figure 61.46. Full Model for CLASS Data, Residuals Shown

Upon examining the data and residuals, you realize that observation 17 (Ronald) was mistakenly included in the analysis. Also, you would like to examine the effect of reweighting to 0.5 those observations with residuals that have absolute values greater than or equal to 17.

```
reweight obs.=17;
reweight r. le -17 or r. ge 17 / weight=0.5;
print p;
run;
```

At this point, a message (on the log) appears that tells you which observations have been reweighted and what the new weights are. Figure 61.47 is produced.

```
                           The REG Procedure
                             Model: MODEL1.2
                       Dependent Variable: Weight

                            Output Statistics

                           Weight    Dependent    Predicted
         Obs    Name      Variable    Variable        Value     Residual

           1    Alfred      1.0000    112.5000     121.6250      -9.1250
           2    Alice       1.0000     84.0000      79.9296       4.0704
           3    Barbara     1.0000     98.0000     107.5484      -9.5484
           4    Carol       1.0000    102.5000     102.1663       0.3337
           5    Henry       1.0000    102.5000     104.3632      -1.8632
           6    James       1.0000     83.0000      79.9762       3.0238
           7    Jane        1.0000     84.5000      87.8225      -3.3225
           8    Janet       1.0000    112.5000     103.6889       8.8111
           9    Jeffrey     1.0000     84.0000      98.7606     -14.7606
          10    John        1.0000     99.5000      85.3117      14.1883
          11    Joyce       1.0000     50.5000      58.6811      -8.1811
          12    Judy        0.5000     90.0000     106.8740     -16.8740
          13    Louise      1.0000     77.0000      76.8377       0.1623
          14    Mary        1.0000    112.0000     116.2429      -4.2429
          15    Philip      1.0000    150.0000     135.9688      14.0312
          16    Robert      0.5000    128.0000     103.5150      24.4850
          17    Ronald           0    133.0000     117.8121      15.1879
          18    Thomas      1.0000     85.0000      78.1398       6.8602
          19    William     1.0000    112.0000     116.2429      -4.2429


                  Sum of Residuals                           0
                  Sum of Squared Residuals          1500.61194
                  Predicted Residual SS (PRESS)     2287.57621

NOTE: The above statistics use observation weights or frequencies.
```

Figure 61.47. Model with Reweighted Observations

The first REWEIGHT statement excludes observation 17, and the second REWEIGHT statement reweights observations 12 and 16 to 0.5. An important feature to note from this example is that the model is not refit until after the PRINT statement. REWEIGHT statements do not cause the model to be refit. This is so that multiple REWEIGHT statements can be applied to a subsequent model.

In this example, since the intent is to reweight observations with large residuals, the observation that was mistakenly included in the analysis should be deleted; then, the model should be fit for those remaining observations, and the observations with large residuals should be reweighted. To accomplish this, use the REFIT statement. Note that the model label has been changed from MODEL1 to MODEL1.2 as two REWEIGHT statements have been used. These statements produce Figure 61.48:

```
reweight allobs / weight=1.0;
reweight obs.=17;
refit;
reweight r. le -17 or r. ge 17 / weight=.5;
print;
run;
```

```
                              The REG Procedure
                               Model: MODEL1.5
                         Dependent Variable: Weight

                              Output Statistics

                             Weight     Dependent     Predicted
       Obs    Name          Variable     Variable        Value      Residual

         1    Alfred          1.0000     112.5000     120.9716       -8.4716
         2    Alice           1.0000      84.0000      79.5342        4.4658
         3    Barbara         1.0000      98.0000     107.0746       -9.0746
         4    Carol           1.0000     102.5000     101.5681        0.9319
         5    Henry           1.0000     102.5000     103.7588       -1.2588
         6    James           1.0000      83.0000      79.7204        3.2796
         7    Jane            1.0000      84.5000      87.5443       -3.0443
         8    Janet           1.0000     112.5000     102.9467        9.5533
         9    Jeffrey         1.0000      84.0000      98.3117      -14.3117
        10    John            1.0000      99.5000      85.0407       14.4593
        11    Joyce           1.0000      50.5000      58.6253       -8.1253
        12    Judy            1.0000      90.0000     106.2625      -16.2625
        13    Louise          1.0000      77.0000      76.5908        0.4092
        14    Mary            1.0000     112.0000     115.4651       -3.4651
        15    Philip          1.0000     150.0000     134.9953       15.0047
        16    Robert          0.5000     128.0000     103.1923       24.8077
        17    Ronald               0     133.0000     117.0299       15.9701
        18    Thomas          1.0000      85.0000      78.0288        6.9712
        19    William         1.0000     112.0000     115.4651       -3.4651


                  Sum of Residuals                            0
                  Sum of Squared Residuals           1637.81879
                  Predicted Residual SS (PRESS)      2473.87984

NOTE: The above statistics use observation weights or frequencies.
```

Figure 61.48. Observations Excluded from Analysis, Model Refitted and Observations Reweighted

Notice that this results in a slightly different model than the previous set of statements: only observation 16 is reweighted to 0.5. Also note that the model label is now MODEL1.5 since five REWEIGHT statements have been used for this model.

Another important feature of the REWEIGHT statement is the ability to nullify the effect of a previous or all REWEIGHT statements. First, assume that you have several REWEIGHT statements in effect and you want to restore the original weights of all the observations. The following REWEIGHT statement accomplishes this and produces Figure 61.49:

```
reweight allobs / reset;
print;
run;
```

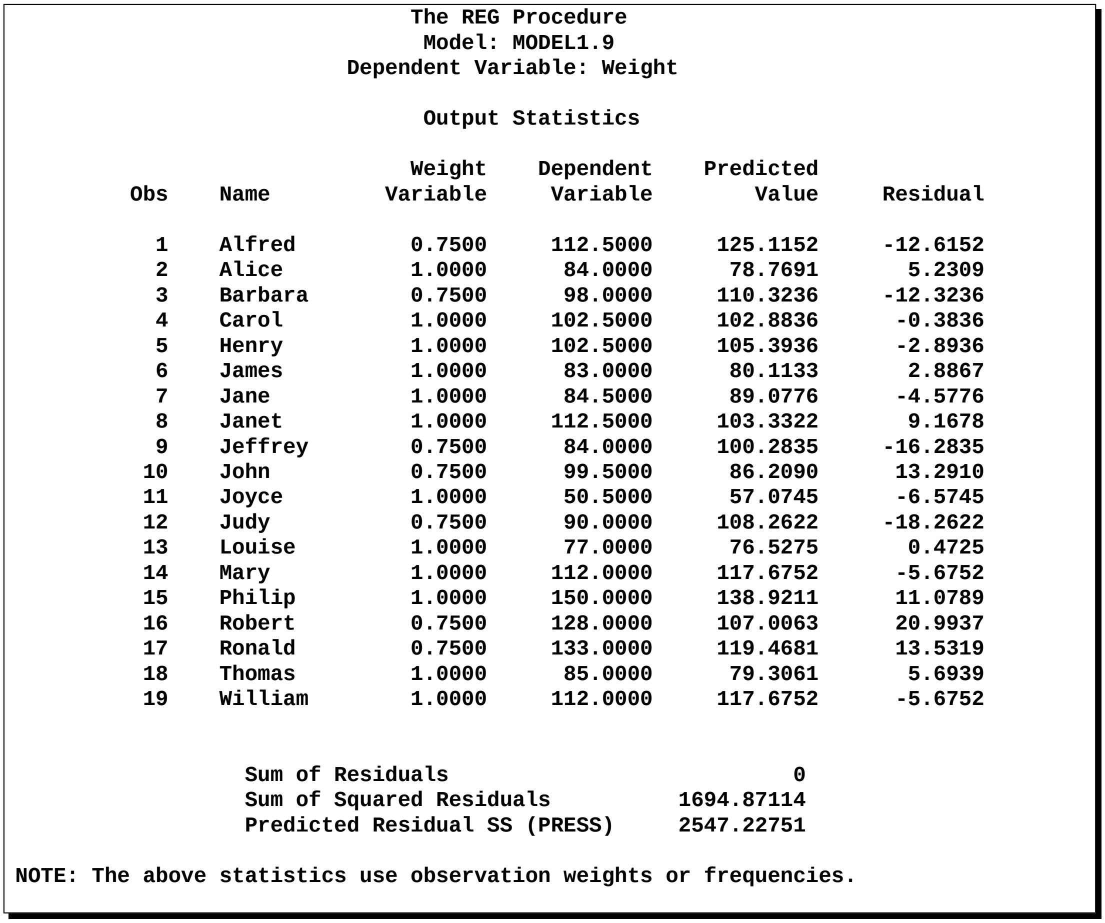
```
                        The REG Procedure
                         Model: MODEL1.6
                   Dependent Variable: Weight

                        Output Statistics

                          Dependent    Predicted
           Obs    Name     Variable        Value     Residual

             1    Alfred   112.5000     124.8686     -12.3686
             2    Alice     84.0000      78.6273       5.3727
             3    Barbara   98.0000     110.2812     -12.2812
             4    Carol    102.5000     102.5670      -0.0670
             5    Henry    102.5000     105.0849      -2.5849
             6    James     83.0000      80.2266       2.7734
             7    Jane      84.5000      89.2191      -4.7191
             8    Janet    112.5000     102.7663       9.7337
             9    Jeffrey   84.0000     100.2095     -16.2095
            10    John      99.5000      86.3415      13.1585
            11    Joyce     50.5000      57.3660      -6.8660
            12    Judy      90.0000     107.9625     -17.9625
            13    Louise    77.0000      76.6295       0.3705
            14    Mary     112.0000     117.1544      -5.1544
            15    Philip   150.0000     138.2164      11.7836
            16    Robert   128.0000     107.2043      20.7957
            17    Ronald   133.0000     118.9529      14.0471
            18    Thomas    85.0000      79.6676       5.3324
            19    William  112.0000     117.1544      -5.1544


              Sum of Residuals                          0
              Sum of Squared Residuals         2120.09974
              Predicted Residual SS (PRESS)    3272.72186
```

Figure 61.49. Restoring Weights of All Observations

The resulting model is identical to the original model specified at the beginning of this section. Notice that the model label is now MODEL1.6. Note that the Weight column does not appear, since all observations have been reweighted to have weight=1.

Now suppose you want only to undo the changes made by the most recent REWEIGHT statement. Use REWEIGHT UNDO for this. The following statements produce Figure 61.50:

```
reweight r. le -12 or r. ge 12 / weight=.75;
reweight r. le -17 or r. ge 17 / weight=.5;
reweight undo;
print;
run;
```



```
                                The REG Procedure
                                 Model: MODEL1.9
                           Dependent Variable: Weight

                                Output Statistics

                             Weight    Dependent    Predicted
       Obs    Name          Variable    Variable        Value     Residual

         1    Alfred          0.7500    112.5000     125.1152     -12.6152
         2    Alice           1.0000     84.0000      78.7691       5.2309
         3    Barbara         0.7500     98.0000     110.3236     -12.3236
         4    Carol           1.0000    102.5000     102.8836      -0.3836
         5    Henry           1.0000    102.5000     105.3936      -2.8936
         6    James           1.0000     83.0000      80.1133       2.8867
         7    Jane            1.0000     84.5000      89.0776      -4.5776
         8    Janet           1.0000    112.5000     103.3322       9.1678
         9    Jeffrey         0.7500     84.0000     100.2835     -16.2835
        10    John            0.7500     99.5000      86.2090      13.2910
        11    Joyce           1.0000     50.5000      57.0745      -6.5745
        12    Judy            0.7500     90.0000     108.2622     -18.2622
        13    Louise          1.0000     77.0000      76.5275       0.4725
        14    Mary            1.0000    112.0000     117.6752      -5.6752
        15    Philip          1.0000    150.0000     138.9211      11.0789
        16    Robert          0.7500    128.0000     107.0063      20.9937
        17    Ronald          0.7500    133.0000     119.4681      13.5319
        18    Thomas          1.0000     85.0000      79.3061       5.6939
        19    William         1.0000    112.0000     117.6752      -5.6752


                  Sum of Residuals                             0
                  Sum of Squared Residuals            1694.87114
                  Predicted Residual SS (PRESS)       2547.22751

NOTE: The above statistics use observation weights or frequencies.
```

Figure 61.50. Example of UNDO in REWEIGHT Statement

The resulting model reflects changes made only by the first REWEIGHT statement since the third REWEIGHT statement negates the effect of the second REWEIGHT statement. Observations 1, 3, 9, 10, 12, 16, and 17 have their weights changed to 0.75. Note that the label MODEL1.9 reflects the use of nine REWEIGHT statements for the current model.

Now suppose you want to reset the observations selected by the most recent REWEIGHT statement to their original weights. Use the REWEIGHT statement with the RESET option to do this. The following statements produce Figure 61.51:

```
reweight r. le -12 or r. ge 12 / weight=.75;
reweight r. le -17 or r. ge 17 / weight=.5;
reweight / reset;
print;
run;
```

```
                          The REG Procedure
                           Model: MODEL1.12
                     Dependent Variable: Weight

                           Output Statistics

                            Weight    Dependent    Predicted
         Obs    Name        Variable   Variable        Value     Residual

           1    Alfred        0.7500   112.5000     126.0076     -13.5076
           2    Alice         1.0000    84.0000      77.8727       6.1273
           3    Barbara       0.7500    98.0000     111.2805     -13.2805
           4    Carol         1.0000   102.5000     102.4703       0.0297
           5    Henry         1.0000   102.5000     105.1278      -2.6278
           6    James         1.0000    83.0000      80.2290       2.7710
           7    Jane          1.0000    84.5000      89.7199      -5.2199
           8    Janet         1.0000   112.5000     102.0122      10.4878
           9    Jeffrey       0.7500    84.0000     100.6507     -16.6507
          10    John          0.7500    99.5000      86.6828      12.8172
          11    Joyce         1.0000    50.5000      56.7703      -6.2703
          12    Judy          1.0000    90.0000     108.1649     -18.1649
          13    Louise        1.0000    77.0000      76.4327       0.5673
          14    Mary          1.0000   112.0000     117.1975      -5.1975
          15    Philip        1.0000   150.0000     138.7581      11.2419
          16    Robert        1.0000   128.0000     108.7016      19.2984
          17    Ronald        0.7500   133.0000     119.0957      13.9043
          18    Thomas        1.0000    85.0000      80.3076       4.6924
          19    William       1.0000   112.0000     117.1975      -5.1975


                  Sum of Residuals                          0
                  Sum of Squared Residuals         1879.08980
                  Predicted Residual SS (PRESS)    2959.57279

NOTE: The above statistics use observation weights or frequencies.
```

Figure 61.51. REWEIGHT Statement with RESET option

Note that observations that meet the condition of the second REWEIGHT statement (residuals with an absolute value greater than or equal to 17) now have weights reset to their original value of 1. Observations 1, 3, 9, 10, and 17 have weights of 0.75, but observations 12 and 16 (which meet the condition of the second REWEIGHT statement) have their weights reset to 1.

Notice how the last three examples show three ways to change weights back to a previous value. In the first example, ALLOBS and the RESET option are used to change weights for all observations back to their original values. In the second example, the UNDO option is used to negate the effect of a previous REWEIGHT statement, thus changing weights for observations selected in the previous REWEIGHT statement to the weights specified in still another REWEIGHT statement. In the third example, the RESET option is used to change weights for observations selected in a previous REWEIGHT statement back to their original values. Finally, note that the label MODEL1.12 indicates that twelve REWEIGHT statements have been applied to the original model.

Testing for Heteroscedasticity

The regression model is specified as $y_i = \mathbf{x}_i\beta + \epsilon_i$, where the ϵ_i's are identically and independently distributed: $E(\epsilon) = 0$ and $E(\epsilon'\epsilon) = \sigma^2\mathbf{I}$. If the ϵ_i's are not independent or their variances are not constant, the parameter estimates are unbiased, but the estimate of the covariance matrix is inconsistent. In the case of heteroscedasticity, the ACOV option provides a consistent estimate of the covariance matrix. If the regression data are from a simple random sample, the ACOV option produces the covariance matrix. This matrix is

$$(\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathrm{diag}(e_i^2)\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}$$

where

$$e_i = y_i - \mathbf{x}_i\mathbf{b}$$

The SPEC option performs a model specification test. The null hypothesis for this test maintains that the errors are homoscedastic, independent of the regressors and that several technical assumptions about the model specification are valid. For details, see theorem 2 and assumptions 1–7 of White (1980). When the model is correctly specified and the errors are independent of the regressors, the rejection of this null hypothesis is evidence of heteroscedasticity. In implementing this test, an estimator of the average covariance matrix (White 1980, p. 822) is constructed and inverted. The nonsingularity of this matrix is one of the assumptions in the null hypothesis about the model specification. When PROC REG determines this matrix to be numerically singular, a generalized inverse is used and a note to this effect is written to the log. In such cases, care should be taken in interpreting the results of this test.

When you specify the SPEC option, tests listed in the TEST statement are performed with both the usual covariance matrix and the heteroscedasticity consistent covariance matrix. Tests performed with the consistent covariance matrix are asymptotic. For more information, refer to White (1980).

Both the ACOV and SPEC options can be specified in a MODEL or PRINT statement.

Multivariate Tests

The MTEST statement described in the "MTEST Statement" section on page 3832 can test hypotheses involving several dependent variables in the form

$$(\mathbf{L}\beta - \mathbf{cj})\mathbf{M} = 0$$

where $\mathbf{L}$ is a linear function on the regressor side, β is a matrix of parameters, $\mathbf{c}$ is a column vector of constants, $\mathbf{j}$ is a row vector of ones, and $\mathbf{M}$ is a linear function on the dependent side. The special case where the constants are zero is

$$\mathbf{L}\beta\mathbf{M} = 0$$

To test this hypothesis, PROC REG constructs two matrices called $\mathbf{H}$ and $\mathbf{E}$ that correspond to the numerator and denominator of a univariate F test:

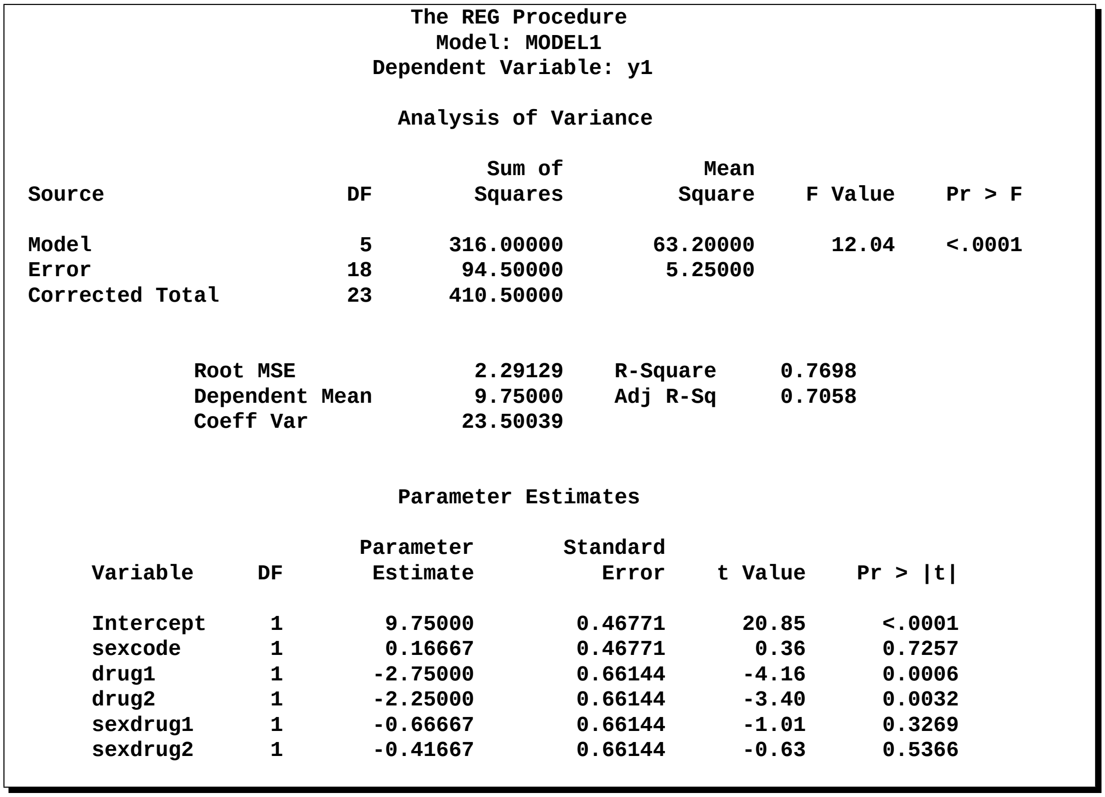
$$\begin{aligned} \mathbf{H} &= \mathbf{M}'(\mathbf{LB} - \mathbf{cj})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{LB} - \mathbf{cj})\mathbf{M} \\ \mathbf{E} &= \mathbf{M}'(\mathbf{Y}'\mathbf{Y} - \mathbf{B}'(\mathbf{X}'\mathbf{X})\mathbf{B})\mathbf{M} \end{aligned}$$

These matrices are displayed for each MTEST statement if the PRINT option is specified.

Four test statistics based on the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ or $(\mathbf{E} + \mathbf{H})^{-1}\mathbf{H}$ are formed. These are Wilks' Lambda, Pillai's Trace, the Hotelling-Lawley Trace, and Roy's maximum root. These test statistics are discussed in Chapter 2, "Introduction to Regression Procedures."

The following statements perform a multivariate analysis of variance and produce Figure 61.52 through Figure 61.56:

```
* Manova Data from Morrison (1976, 190);
  data a;
     input sex $ drug $ @;
     do rep=1 to 4;
        input y1 y2 @;
        sexcode=(sex='m')-(sex='f');
        drug1=(drug='a')-(drug='c');
        drug2=(drug='b')-(drug='c');
        sexdrug1=sexcode*drug1;
        sexdrug2=sexcode*drug2;
        output;
     end;
     datalines;
  m a  5  6  5  4  9  9  7  6
  m b  7  6  7  7  9 12  6  8
  m c 21 15 14 11 17 12 12 10
  f a  7 10  6  6  9  7  8 10
  f b 10 13  8  7  7  6  6  9
  f c 16 12 14  9 14  8 10  5
  ;
  proc reg;
     model y1 y2=sexcode drug1 drug2 sexdrug1 sexdrug2;
     y1y2drug: mtest y1=y2, drug1,drug2;
     drugshow: mtest drug1, drug2 / print canprint;
  run;
```



```
                              The REG Procedure
                                Model: MODEL1
                           Dependent Variable: y1

                            Analysis of Variance

                                     Sum of           Mean
Source                   DF         Squares         Square    F Value    Pr > F

Model                     5       316.00000       63.20000      12.04    <.0001
Error                    18        94.50000        5.25000
Corrected Total          23       410.50000


             Root MSE              2.29129    R-Square     0.7698
             Dependent Mean        9.75000    Adj R-Sq     0.7058
             Coeff Var            23.50039


                            Parameter Estimates

                            Parameter       Standard
     Variable      DF        Estimate          Error    t Value    Pr > |t|

     Intercept      1         9.75000        0.46771      20.85      <.0001
     sexcode        1         0.16667        0.46771       0.36      0.7257
     drug1          1        -2.75000        0.66144      -4.16      0.0006
     drug2          1        -2.25000        0.66144      -3.40      0.0032
     sexdrug1       1        -0.66667        0.66144      -1.01      0.3269
     sexdrug2       1        -0.41667        0.66144      -0.63      0.5366
```

Figure 61.52. Multivariate Analysis of Variance: REG Procedure

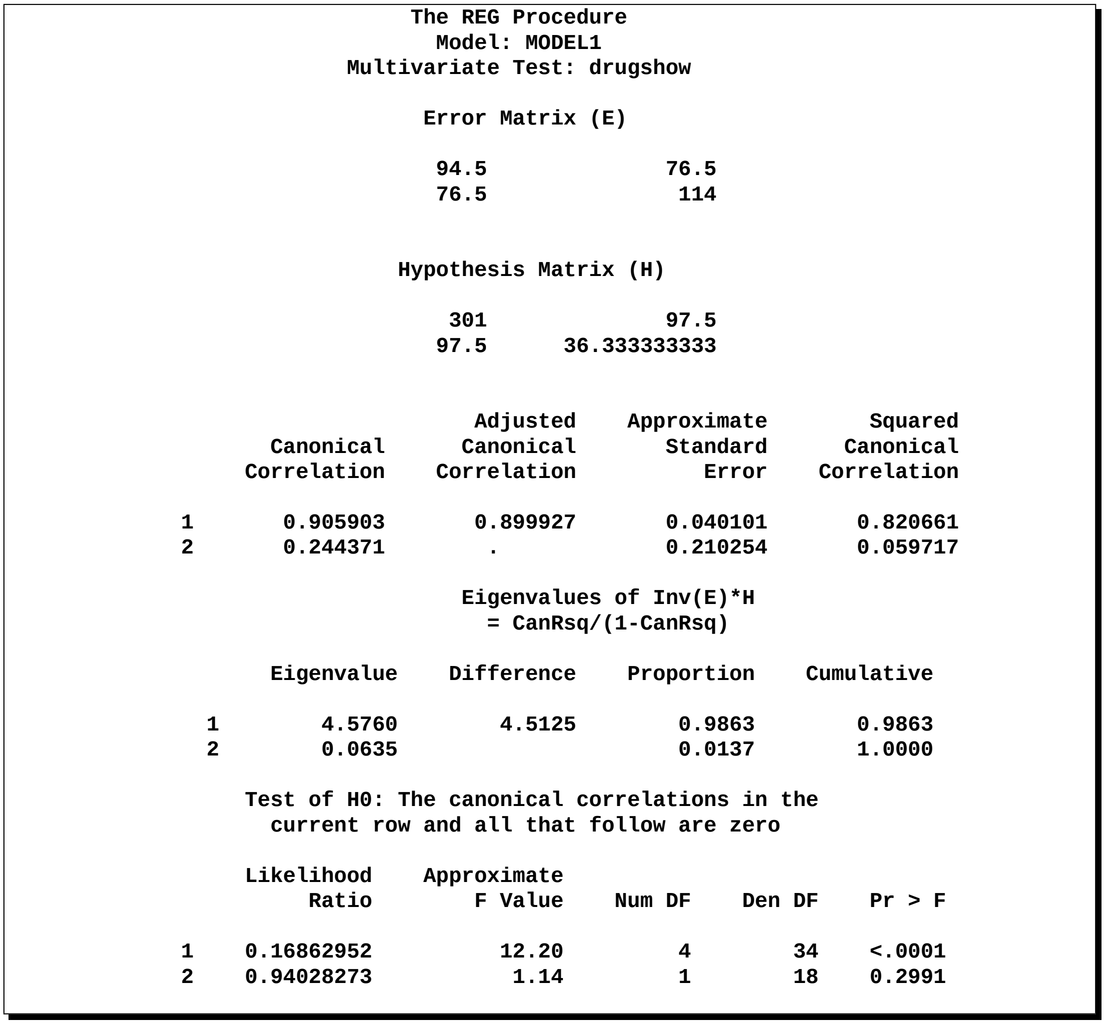
```
                                 The REG Procedure
                                   Model: MODEL1
                               Dependent Variable: y2

                                Analysis of Variance

                                        Sum of           Mean
Source                   DF            Squares         Square    F Value    Pr > F

Model                     5           69.33333       13.86667       2.19    0.1008
Error                    18          114.00000        6.33333
Corrected Total          23          183.33333


             Root MSE              2.51661    R-Square     0.3782
             Dependent Mean        8.66667    Adj R-Sq     0.2055
             Coeff Var            29.03782


                                Parameter Estimates

                                 Parameter       Standard
      Variable      DF            Estimate          Error    t Value    Pr > |t|

      Intercept      1             8.66667        0.51370      16.87      <.0001
      sexcode        1             0.16667        0.51370       0.32      0.7493
      drug1          1            -1.41667        0.72648      -1.95      0.0669
      drug2          1            -0.16667        0.72648      -0.23      0.8211
      sexdrug1       1            -1.16667        0.72648      -1.61      0.1257
      sexdrug2       1            -0.41667        0.72648      -0.57      0.5734
```

Figure 61.53. Multivariate Analysis of Variance: REG Procedure

```
                                 The REG Procedure
                                   Model: MODEL1
                             Multivariate Test: y1y2drug

                    Multivariate Statistics and Exact F Statistics

                                 S=1    M=0    N=8

Statistic                           Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda                  0.28053917      23.08         2        18    <.0001
Pillai's Trace                 0.71946083      23.08         2        18    <.0001
Hotelling-Lawley Trace         2.56456456      23.08         2        18    <.0001
Roy's Greatest Root            2.56456456      23.08         2        18    <.0001
```

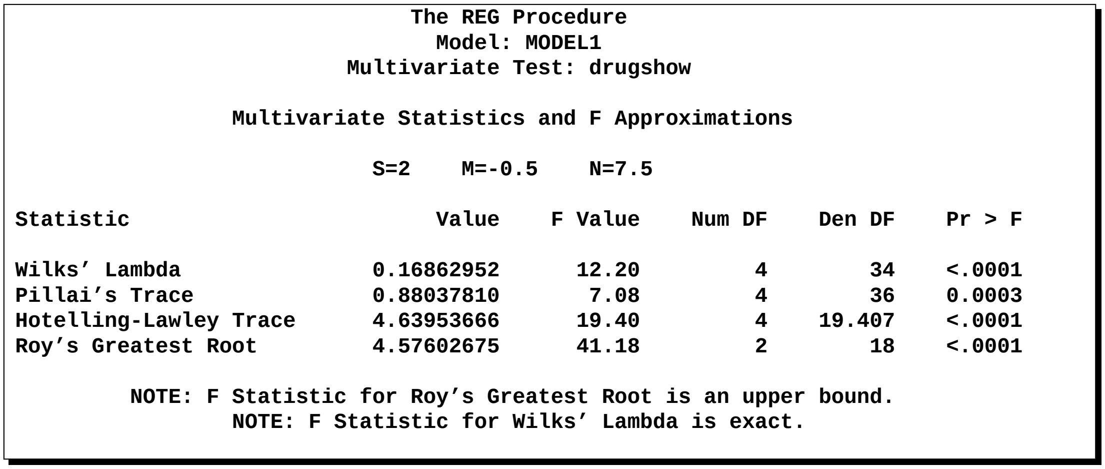
Figure 61.54. Multivariate Analysis of Variance: First Test

The four multivariate test statistics are all highly significant, giving strong evidence that the coefficients of drug1 and drug2 are not the same across dependent variables y1 and y2.



```
                                The REG Procedure
                                  Model: MODEL1
                           Multivariate Test: drugshow

                                 Error Matrix (E)

                                  94.5              76.5
                                  76.5               114


                               Hypothesis Matrix (H)

                                   301              97.5
                                  97.5      36.333333333


                                     Adjusted    Approximate       Squared
                     Canonical      Canonical       Standard     Canonical
                   Correlation    Correlation          Error    Correlation

              1       0.905903       0.899927       0.040101       0.820661
              2       0.244371        .             0.210254       0.059717

                                    Eigenvalues of Inv(E)*H
                                      = CanRsq/(1-CanRsq)

                     Eigenvalue    Difference    Proportion    Cumulative

                1        4.5760        4.5125        0.9863        0.9863
                2        0.0635                      0.0137        1.0000

                   Test of H0: The canonical correlations in the
                     current row and all that follow are zero

                   Likelihood    Approximate
                        Ratio        F Value    Num DF    Den DF    Pr > F

              1    0.16862952          12.20         4        34    <.0001
              2    0.94028273           1.14         1        18    0.2991
```

Figure 61.55. Multivariate Analysis of Variance: Second Test



```
                              The REG Procedure
                                Model: MODEL1
                         Multivariate Test: drugshow

                  Multivariate Statistics and F Approximations

                             S=2    M=-0.5    N=7.5

Statistic                         Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda                0.16862952      12.20         4        34    <.0001
Pillai's Trace               0.88037810       7.08         4        36    0.0003
Hotelling-Lawley Trace       4.63953666      19.40         4    19.407    <.0001
Roy's Greatest Root          4.57602675      41.18         2        18    <.0001

         NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                 NOTE: F Statistic for Wilks' Lambda is exact.
```

Figure 61.56. Multivariate Analysis of Variance: Second Test (continued)

The four multivariate test statistics are all highly significant, giving strong evidence that the coefficients of drug1 and drug2 are not zero for both dependent variables.

Autocorrelation in Time Series Data

When regression is performed on time series data, the errors may not be independent. Often errors are autocorrelated; that is, each error is correlated with the error immediately before it. Autocorrelation is also a symptom of systematic lack of fit. The DW option provides the Durbin-Watson d statistic to test that the autocorrelation is zero:

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

The value of d is close to 2 if the errors are uncorrelated. The distribution of d is reported by Durbin and Watson (1951). Tables of the distribution are found in most econometrics textbooks, such as Johnston (1972) and Pindyck and Rubinfeld (1981).

The sample autocorrelation estimate is displayed after the Durbin-Watson statistic. The sample is computed as

$$r = \frac{\sum_{i=2}^{n} e_i e_{i-1}}{\sum_{i=1}^{n} e_i^2}$$

This autocorrelation of the residuals may not be a very good estimate of the autocorrelation of the true errors, especially if there are few observations and the independent variables have certain patterns. If there are missing observations in the regression, these measures are computed as though the missing observations did not exist.

Positive autocorrelation of the errors generally tends to make the estimate of the error variance too small, so confidence intervals are too narrow and true null hypotheses are rejected with a higher probability than the stated significance level. Negative autocorrelation of the errors generally tends to make the estimate of the error variance too large, so confidence intervals are too wide and the power of significance tests is reduced. With either positive or negative autocorrelation, least-squares parameter estimates are usually not as efficient as generalized least-squares parameter estimates. For more details, refer to Judge et al. (1985, Chapter 8) and the *SAS/ETS User's Guide*.

The following SAS statements request the DW option for the US population data (see Figure 61.57):

```
proc reg data=USPopulation;
   model Population=Year YearSq / dw;
run;
```

```
                         The REG Procedure
                           Model: MODEL1
                   Dependent Variable: Population

                 Durbin-Watson D                1.191
                 Number of Observations            22
                 1st Order Autocorrelation      0.323
```

Figure 61.57. Regression Using DW Option

Computations for Ridge Regression and IPC Analysis

In ridge regression analysis, the crossproduct matrix for the independent variables is centered (the NOINT option is ignored if it is specified) and scaled to one on the diagonal elements. The ridge constant *k* (specified with the RIDGE= option) is then added to each diagonal element of the crossproduct matrix. The ridge regression estimates are the least-squares estimates obtained by using the new crossproduct matrix.

Let $\mathbf{X}$ be an $n \times p$ matrix of the independent variables after centering the data, and let $\mathbf{Y}$ be an $n \times 1$ vector corresponding to the dependent variable. Let $\mathbf{D}$ be a $p \times p$ diagonal matrix with diagonal elements as in $\mathbf{X}'\mathbf{X}$. The ridge regression estimate corresponding to the ridge constant k can be computed as

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{Z}'\mathbf{Z} + k\mathbf{I}_p)^{-1}\mathbf{Z}'\mathbf{Y}$$

where $\mathbf{Z} = \mathbf{X}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{I}_p$ is a $p{\times}p$ identity matrix.

For IPC analysis, the smallest *m* eigenvalues of $\mathbf{Z}'\mathbf{Z}$ (where *m* is specified with the PCOMIT= option) are omitted to form the estimates.

For information about ridge regression and IPC standardized parameter estimates, parameter estimate standard errors, and variance inflation factors, refer to Rawlings (1988), Neter, Wasserman, and Kutner (1990), and Marquardt and Snee (1975). Unlike Rawlings (1988), the REG procedure uses the mean squared errors of the submodels instead of the full model MSE to compute the standard errors of the parameter estimates.

Construction of Q-Q and P-P Plots

If a normal probability-probability or quantile-quantile plot for the variable x is requested, the n nonmissing values of x are first ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

If a Q-Q plot is requested (with a PLOT statement of the form PLOT *yvariable*$*$NQQ.), the i^{th} ordered value $x_{(i)}$ is represented by a point with *y*-coordinate $x_{(i)}$ and *x*-coordinate $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi(\cdot)$ is the standard normal distribution.

If a P-P plot is requested (with a PLOT statement of the form PLOT *yvariable*$*$NPP.), the i^{th} ordered value $x_{(i)}$ is represented by a point with *y*-coordinate $\frac{i}{n}$ and *x*-coordinate $\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right)$, where μ is the mean of the nonmissing x-values and σ is the standard deviation. If an x-value has multiplicity k (that is, $x_{(i)} = \cdots = x_{(i+k-1)}$), then only the point $\left(\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right), \frac{i+k-1}{n}\right)$ is displayed.

Computational Methods

The REG procedure first composes a crossproducts matrix. The matrix can be calculated from input data, reformed from an input correlation matrix, or read in from an SSCP data set. For each model, the procedure selects the appropriate crossproducts from the main matrix. The normal equations formed from the crossproducts are solved using a sweep algorithm (Goodnight 1979). The method is accurate for data that are reasonably scaled and not too collinear.

The mechanism that PROC REG uses to check for singularity involves the diagonal (pivot) elements of $\mathbf{X}'\mathbf{X}$ as it is being swept. If a pivot is less than SINGULAR*CSS, then a singularity is declared and the pivot is not swept (where CSS is the corrected sum of squares for the regressor and SINGULAR is machine dependent but is approximately 1E$-$7 on most machines or reset in the PROC statement).

The sweep algorithm is also used in many places in the model-selection methods. The RSQUARE method uses the leaps and bounds algorithm by Furnival and Wilson (1974).

Computer Resources in Regression Analysis

The REG procedure is efficient for ordinary regression; however, requests for optional features can greatly increase the amount of time required.

The major computational expense in the regression analysis is the collection of the crossproducts matrix. For p variables and n observations, the time required is proportional to np^2. For each model run, PROC REG needs time roughly proportional to k^3, where k is the number of regressors in the model. Add an additional nk^2 for one of the R, CLM, or CLI options and another nk^2 for the INFLUENCE option.

Most of the memory that PROC REG needs to solve large problems is used for crossproducts matrices. PROC REG requires $4p^2$ bytes for the main crossproducts matrix plus $4k^2$ bytes for the largest model. If several output data sets are requested, memory is also needed for buffers.

See the "Input Data Sets" section on page 3860 for information on how to use TYPE=SSCP data sets to reduce computing time.

Displayed Output

Many of the more specialized tables are described in detail in previous sections. Most of the formulas for the statistics are in Chapter 2, "Introduction to Regression Procedures," while other formulas can be found in the section "Model Fit and Diagnostic Statistics" on page 3896 and the "Influence Diagnostics" section on page 3898.

The analysis-of-variance table includes

- the Source of the variation, Model for the fitted regression, Error for the residual error, and C Total for the total variation after correcting for the mean. The Uncorrected Total Variation is produced when the NOINT option is used.
- the degrees of freedom (DF) associated with the source
- the Sum of Squares for the term
- the Mean Square, the sum of squares divided by the degrees of freedom
- the F Value for testing the hypothesis that all parameters are zero except for the intercept. This is formed by dividing the mean square for Model by the mean square for Error.
- the Prob>F, the probability of getting a greater F statistic than that observed if the hypothesis is true. This is the significance probability.

Other statistics displayed include the following:

- Root MSE is an estimate of the standard deviation of the error term. It is calculated as the square root of the mean square error.
- Dep Mean is the sample mean of the dependent variable.
- C.V. is the coefficient of variation, computed as 100 times Root MSE divided by Dep Mean. This expresses the variation in unitless values.
- R-Square is a measure between 0 and 1 that indicates the portion of the (corrected) total variation that is attributed to the fit rather than left to residual error. It is calculated as SS(Model) divided by SS(Total). It is also called the *coefficient of determination*. It is the square of the multiple correlation; in other words, the square of the correlation between the dependent variable and the predicted values.

- Adj R-Sq, the adjusted R^2, is a version of R^2 that has been adjusted for degrees of freedom. It is calculated as

  $$\bar{R}^2 = 1 - \frac{(n-i)(1-R^2)}{n-p}$$

  where i is equal to 1 if there is an intercept and 0 otherwise; n is the number of observations used to fit the model; and p is the number of parameters in the model.

The parameter estimates and associated statistics are then displayed, and they include the following:

- the Variable used as the regressor, including the name Intercept to represent the estimate of the intercept parameter
- the degrees of freedom (DF) for the variable. There is one degree of freedom unless the model is not full rank.
- the Parameter Estimate
- the Standard Error, the estimate of the standard deviation of the parameter estimate
- T for H0: Parameter=0, the t test that the parameter is zero. This is computed as the Parameter Estimate divided by the Standard Error.
- the Prob > |T|, the probability that a t statistic would obtain a greater absolute value than that observed given that the true parameter is zero. This is the two-tailed significance probability.

If model-selection methods other than NONE, RSQUARE, ADJRSQ, or CP are used, the analysis-of-variance table and the parameter estimates with associated statistics are produced at each step. Also displayed are

- C(p), which is Mallows' C_p statistic
- bounds on the condition number of the correlation matrix for the variables in the model (Berk 1977)

After statistics for the final model are produced, the following is displayed when the method chosen is FORWARD, BACKWARD, or STEPWISE:

- a Summary table listing Step number, Variable Entered or Removed, Partial and Model R-Square, and C(p) and F statistics

The RSQUARE method displays its results beginning with the model containing the fewest independent variables and producing the largest R^2. Results for other models with the same number of variables are then shown in order of decreasing R^2, and so on, for models with larger numbers of variables. The ADJRSQ and CP methods group models of all sizes together and display results beginning with the model having the optimal value of adjusted $R^2$and C_p, respectively.

For each model considered, the RSQUARE, ADJRSQ, and CP methods display the following:

- Number in Model or IN, the number of independent variables used in each model
- R-Square or RSQ, the squared multiple correlation coefficient

If the B option is specified, the RSQUARE, ADJRSQ, and CP methods produce the following:

- Parameter Estimates, the estimated regression coefficients

If the B option is not specified, the RSQUARE, ADJRSQ, and CP methods display the following:

- Variables in Model, the names of the independent variables included in the model

ODS Table Names

PROC REG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 61.8. ODS Tables Produced in PROC REG

ODS Table Name	Description	Statement	Option
ACovEst	Consistent covariance of estimates matrix	MODEL	ALL, ACOV
ACovTestANOVA	Test ANOVA using ACOV estimates	TEST	ACOV (MODEL statement)
ANOVA	Model ANOVA table	MODEL	default
CanCorr	Canonical correlations for hypothesis combinations	MTEST	CANPRINT
CollinDiag	Collinearity Diagnostics table	MODEL	COLLIN
CollinDiagNoInt	Collinearity Diagnostics for no intercept model	MODEL	COLLINOINT
ConditionBounds	Bounds on condition number	MODEL	(SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR) and DETAILS

Table 61.8. (continued)

ODS Table Name	Description	Statement	Option
Corr	Correlation matrix for analysis variables	PROC	ALL, CORR
CorrB	Correlation of estimates	MODEL	CORRB
CovB	Covariance of estimates	MODEL	COVB
CrossProducts	Bordered model **X'X** matrix	MODEL	ALL, XPX
DWStatistic	Durbin-Watson statistic	MODEL	ALL, DW
DependenceEquations	Linear dependence equations	MODEL	default if needed
Eigenvalues	MTest eigenvalues	MTEST	CANPRINT
Eigenvectors	MTest eigenvectors	MTEST	CANPRINT
EntryStatistics	Entry statistics for selection methods	MODEL	(SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR) and DETAILS
ErrorPlusHypothesis	MTest error plus hypothesis matrix **H**+**E**	MTEST	PRINT
ErrorSSCP	MTest error matrix **E**	MTEST	PRINT
FitStatistics	Model fit statistics	MODEL	default
HypothesisSSCP	MTest hypothesis matrix	MTEST	PRINT
InvMTestCov	Inv(**L** Ginv(**X'X**) **L**') and Inv(**Lb-c**)	MTEST	DETAILS
InvTestCov	Inv(**L** Ginv(**X'X**) **L**') and Inv(**Lb-c**)	TEST	PRINT
InvXPX	Bordered **X'X** inverse matrix	MODEL	I
MTestCov	**L** Ginv(**X'X**) **L**' and **Lb-c**	MTEST	DETAILS
MTransform	MTest matrix **M**, across dependents	MTEST	DETAILS
MultStat	Multivariate test statistics	MTEST	default
NObs	Number of observations		default
OutputStatistics	Output statistics table	MODEL	ALL, CLI, CLM, INFLUENCE, P, R
ParameterEstimates	Model parameter estimates	MODEL	default
RemovalStatistics	Removal statistics for selection methods	MODEL	(SELECTION=BACKWARD | STEPWISE | MAXR | MINR) and DETAILS
ResidualStatistics	Residual statistics and PRESS statistic	MODEL	ALL, CLI, CLM, INFLUENCE, P, R
SelParmEst	Parameter estimates for selection methods	MODEL	SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR
SelectionSummary	Selection summary for forward, backward and stepwise methods	MODEL	SELECTION=BACKWARD | FORWARD | STEPWISE
SeqParmEst	Sequential parameter estimates	MODEL	SEQB

Table 61.8. (continued)

ODS Table Name	Description	Statement	Option
SimpleStatistics	Simple statistics for analysis variables	PROC	ALL, SIMPLE
SpecTest	White's heteroscedasticity test	MODEL	ALL, SPEC
SubsetSelSummary	Selection summary for R-Square, Adj-RSq and Cp methods	MODEL	SELECTION=RSQUARE | ADJRSQ | CP
TestANOVA	Test ANOVA table	TEST	default
TestCov	**L** Ginv(**X**'**X**) **L**' and **Lb-c**	TEST	PRINT
USSCP	Uncorrected SSCP matrix for analysis variables	PROC	ALL, USSCP

ODS Graphics (Experimental)

This section describes the use of ODS for creating statistical graphs with the REG procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

When the experimental ODS graphics are in effect, the REG procedure produces a variety of plots. For models with multiple dependent variables, separate plots are produced for each dependent variable. For jobs with more than one MODEL statement, plots are produced for each model statement.

The plots available are as follows:

- With a single regressor, a scatterplot of the input data overlayed with the fitted regression line, confidence band, and prediction limits.
- A summary panel of fit diagnostics:
  - Residuals versus the predicted values
  - Studentized residuals versus the predicted values
  - Studentized residuals versus the leverage
  - Normal quantile plot of the residuals
  - Dependent variable values versus the predicted values
  - Cook's D versus observation number
  - Histogram of the residuals

  - A “Residual-Fit” (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals. This plot “shows how much variation in the data is explained by the fit and how much remains in the residuals” (Cleveland, 1993).

  If the PLOTS(UNPACKPANELS) option is specified in the PROC REG statement, then the eight plots in the fit diagnostics panel are displayed individually.

- Panels of the residuals versus the regressors in the model. Note that each panel contains at most six plots, and multiple panels are used in the case that there are more than six regressors (including the intercept) in the model.
- If the PARTIAL option is specified in a MODEL statement, panels of the partial regression plots for each regressor (see the “The PARTIAL Option” section on page 3901). Note that each panel contains at most six partial plots, and multiple panels are used in the case that there are more than six regressors in the model.
- If the RIDGE= option is specified in the model statement, panels of ridge traces versus the specified ridge parameters for each regressor in the model. At most eight ridge traces are included on a panel and multiple panels are used for models with more than eight regressors.

PLOTS (*general-plot-options*)

specifies characteristics of the graphics produced when you use the experimental ODS GRAPHICS statement. You can specify the following *general-plot-options* in parentheses after the PLOTS option:

UNPACK | UNPACKPANELS specifies that plots in the fit diagnostics panel should be displayed separately.

MAXPOINTS=*number* | NONE specifies that plots with elements that require processing more than *number* points are suppressed. The default is MAXPOINTS=5000. This cutoff is ignored if you specify MAXPOINTS=NONE.

ODS Graph Names

PROC REG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 61.9.

To request these graphs you must specify the ODS GRAPHICS statement. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

Table 61.9. ODS Graphics Produced by PROC REG

ODS Graph Name	Plot Description	PLOTS Option
ActualByPredicted	Dependent variable versus predicted values	UNPACKPANELS
CooksD	Cook's D statistic versus observation number	UNPACKPANELS
DiagnosticsPanel	Panel of fit diagnostics	
Fit	Regression line, confidence band, and prediction limits overlayed on scatterplot of data	
PartialPlotPanel*i*	Panel *i* of partial regression plots	
QQPlot	Normal quantile plot residuals	UNPACKPANELS
ResidualByPredicted	Residuals versus predicted values	UNPACKPANELS
ResidualHistogram	Histogram of fit residuals	UNPACKPANELS
ResidualPanel*i*	Panel *i* of residuals versus regressors	
RFPlot	Side-by-side plots of quantiles of centered fit and residuals	UNPACKPANELS
RidgePanel*i*	Panel *i* of ridge traces	
RStudentByLeverage	Studentized residuals versus leverage	UNPACKPANELS
RStudentByPredicted	Studentized residuals versus predicted values	UNPACKPANELS

Examples

Example 61.1. Aerobic Fitness Prediction

Aerobic fitness (measured by the ability to consume oxygen) is fit to some simple exercise tests. The goal is to develop an equation to predict fitness based on the exercise tests rather than on expensive and cumbersome oxygen consumption measurements. Three model-selection methods are used: forward selection, backward selection, and MAXR selection. The following statements produce Output 61.1.1 through Output 61.1.5. (Collinearity diagnostics for the full model are shown in Figure 61.42 on page 3896.)

```
*-------------------Data on Physical Fitness-------------------*
| These measurements were made on men involved in a physical   |
| fitness course at N.C.State Univ. The variables are Age      |
| (years), Weight (kg), Oxygen intake rate (ml per kg body     |
| weight per minute), time to run 1.5 miles (minutes), heart   |
| rate while resting, heart rate while running (same time      |
| Oxygen rate measured), and maximum heart rate recorded while |
| running.                                                     |
| ***Certain values of MaxPulse were changed for this analysis.|
*--------------------------------------------------------------*;
data fitness;
   input Age Weight Oxygen RunTime RestPulse RunPulse MaxPulse @@;
   datalines;
44 89.47 44.609 11.37 62 178 182   40 75.07 45.313 10.07 62 185 185
44 85.84 54.297  8.65 45 156 168   42 68.15 59.571  8.17 40 166 172
38 89.02 49.874  9.22 55 178 180   47 77.45 44.811 11.63 58 176 176
40 75.98 45.681 11.95 70 176 180   43 81.19 49.091 10.85 64 162 170
44 81.42 39.442 13.08 63 174 176   38 81.87 60.055  8.63 48 170 186
44 73.03 50.541 10.13 45 168 168   45 87.66 37.388 14.03 56 186 192
45 66.45 44.754 11.12 51 176 176   47 79.15 47.273 10.60 47 162 164
54 83.12 51.855 10.33 50 166 170   49 81.42 49.156  8.95 44 180 185
51 69.63 40.836 10.95 57 168 172   51 77.91 46.672 10.00 48 162 168
48 91.63 46.774 10.25 48 162 164   49 73.37 50.388 10.08 67 168 168
57 73.37 39.407 12.63 58 174 176   54 79.38 46.080 11.17 62 156 165
52 76.32 45.441  9.63 48 164 166   50 70.87 54.625  8.92 48 146 155
51 67.25 45.118 11.08 48 172 172   54 91.63 39.203 12.88 44 168 172
51 73.71 45.790 10.47 59 186 188   57 59.08 50.545  9.93 49 148 155
49 76.32 48.673  9.40 56 186 188   48 61.24 47.920 11.50 52 170 176
52 82.78 47.467 10.50 53 170 172
;
proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=forward;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=backward;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=maxr;
run;
```

The FORWARD model-selection method begins with no variables in the model and adds RunTime, then Age,...

Output 61.1.1. Forward Selection Method: PROC REG

```
                               The REG Procedure
                                 Model: MODEL1
                          Dependent Variable: Oxygen

                           Forward Selection: Step 1

         Variable RunTime Entered: R-Square = 0.7434 and C(p) = 13.6988


                              Analysis of Variance

                                     Sum of           Mean
 Source                   DF        Squares         Square    F Value    Pr > F

 Model                     1      632.90010      632.90010      84.01    <.0001
 Error                    29      218.48144        7.53384
 Corrected Total          30      851.38154


                    Parameter     Standard
       Variable      Estimate        Error   Type II SS  F Value  Pr > F

       Intercept     82.42177      3.85530   3443.36654   457.05  <.0001
       RunTime       -3.31056      0.36119    632.90010    84.01  <.0001

                        Bounds on condition number: 1, 1
--------------------------------------------------------------------------------

                           Forward Selection: Step 2


           Variable Age Entered: R-Square = 0.7642 and C(p) = 12.3894


                              Analysis of Variance

                                     Sum of           Mean
 Source                   DF        Squares         Square    F Value    Pr > F

 Model                     2      650.66573      325.33287      45.38    <.0001
 Error                    28      200.71581        7.16842
 Corrected Total          30      851.38154


                    Parameter     Standard
       Variable      Estimate        Error   Type II SS  F Value  Pr > F

       Intercept     88.46229      5.37264   1943.41071   271.11  <.0001
       Age           -0.15037      0.09551     17.76563     2.48  0.1267
       RunTime       -3.20395      0.35877    571.67751    79.75  <.0001

                   Bounds on condition number: 1.0369, 4.1478
--------------------------------------------------------------------------------
```

...then RunPulse, then MaxPulse,...

```
                           Forward Selection: Step 3


         Variable RunPulse Entered: R-Square = 0.8111 and C(p) = 6.9596


                              Analysis of Variance

                                      Sum of           Mean
Source                   DF          Squares         Square    F Value    Pr > F

Model                     3        690.55086      230.18362      38.64    <.0001
Error                    27        160.83069        5.95669
Corrected Total          30        851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    111.71806     10.23509    709.69014   119.14  <.0001
      Age           -0.25640      0.09623     42.28867     7.10  0.0129
      RunTime       -2.82538      0.35828    370.43529    62.19  <.0001
      RunPulse      -0.13091      0.05059     39.88512     6.70  0.0154

                  Bounds on condition number: 1.3548, 11.597
--------------------------------------------------------------------------------

                           Forward Selection: Step 4


         Variable MaxPulse Entered: R-Square = 0.8368 and C(p) = 4.8800


                              Analysis of Variance

                                      Sum of           Mean
Source                   DF          Squares         Square    F Value    Pr > F

Model                     4        712.45153      178.11288      33.33    <.0001
Error                    26        138.93002        5.34346
Corrected Total          30        851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept     98.14789     11.78569    370.57373    69.35  <.0001
      Age           -0.19773      0.09564     22.84231     4.27  0.0488
      RunTime       -2.76758      0.34054    352.93570    66.05  <.0001
      RunPulse      -0.34811      0.11750     46.90089     8.78  0.0064
      MaxPulse       0.27051      0.13362     21.90067     4.10  0.0533

                  Bounds on condition number: 8.4182, 76.851
--------------------------------------------------------------------------------
```

...and finally, Weight. The final variable available to add to the model, RestPulse, is not added since it does not meet the 50% (the default value of the SLE option is 0.5 for FORWARD selection) significance-level criterion for entry into the model.

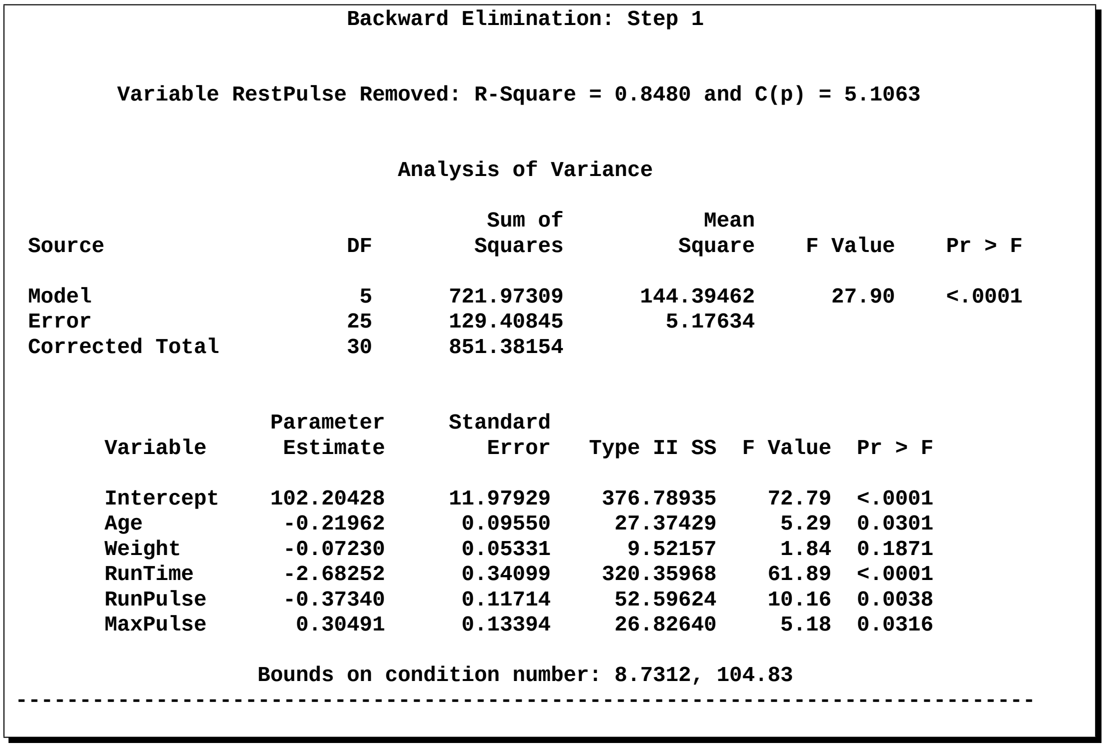
```
                            Forward Selection: Step 5


         Variable Weight Entered: R-Square = 0.8480 and C(p) = 5.1063


                               Analysis of Variance

                                      Sum of           Mean
Source                   DF          Squares         Square    F Value    Pr > F

Model                     5        721.97309      144.39462      27.90    <.0001
Error                    25        129.40845        5.17634
Corrected Total          30        851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    102.20428     11.97929    376.78935    72.79  <.0001
      Age           -0.21962      0.09550     27.37429     5.29  0.0301
      Weight        -0.07230      0.05331      9.52157     1.84  0.1871
      RunTime       -2.68252      0.34099    320.35968    61.89  <.0001
      RunPulse      -0.37340      0.11714     52.59624    10.16  0.0038
      MaxPulse       0.30491      0.13394     26.82640     5.18  0.0316

                  Bounds on condition number: 8.7312, 104.83
--------------------------------------------------------------------------------


No other variable met the 0.5000 significance level for entry into the model.



                          Summary of Forward Selection

       Variable    Number    Partial     Model
Step   Entered     Vars In   R-Square   R-Square    C(p)     F Value   Pr > F

  1    RunTime        1       0.7434     0.7434    13.6988     84.01   <.0001
  2    Age            2       0.0209     0.7642    12.3894      2.48   0.1267
  3    RunPulse       3       0.0468     0.8111     6.9596      6.70   0.0154
  4    MaxPulse       4       0.0257     0.8368     4.8800      4.10   0.0533
  5    Weight         5       0.0112     0.8480     5.1063      1.84   0.1871
```

The BACKWARD model-selection method begins with the full model.

Output 61.1.2. Backward Selection Method: PROC REG

```
                              The REG Procedure
                                Model: MODEL2
                         Dependent Variable: Oxygen

                         Backward Elimination: Step 0

          All Variables Entered: R-Square = 0.8487 and C(p) = 7.0000


                             Analysis of Variance

                                       Sum of           Mean
Source                    DF          Squares         Square    F Value    Pr > F

Model                      6        722.54361      120.42393      22.43    <.0001
Error                     24        128.83794        5.36825
Corrected Total           30        851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    102.93448     12.40326    369.72831    68.87  <.0001
      Age           -0.22697      0.09984     27.74577     5.17  0.0322
      Weight        -0.07418      0.05459      9.91059     1.85  0.1869
      RunTime       -2.62865      0.38456    250.82210    46.72  <.0001
      RunPulse      -0.36963      0.11985     51.05806     9.51  0.0051
      RestPulse     -0.02153      0.06605      0.57051     0.11  0.7473
      MaxPulse       0.30322      0.13650     26.49142     4.93  0.0360

                 Bounds on condition number: 8.7438, 137.13
--------------------------------------------------------------------------------
```

RestPulse is the first variable deleted,...



```
                          Backward Elimination: Step 1


         Variable RestPulse Removed: R-Square = 0.8480 and C(p) = 5.1063


                              Analysis of Variance

                                      Sum of           Mean
Source                    DF         Squares         Square    F Value    Pr > F

Model                      5       721.97309      144.39462      27.90    <.0001
Error                     25       129.40845        5.17634
Corrected Total           30       851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    102.20428     11.97929    376.78935    72.79  <.0001
      Age           -0.21962      0.09550     27.37429     5.29  0.0301
      Weight        -0.07230      0.05331      9.52157     1.84  0.1871
      RunTime       -2.68252      0.34099    320.35968    61.89  <.0001
      RunPulse      -0.37340      0.11714     52.59624    10.16  0.0038
      MaxPulse       0.30491      0.13394     26.82640     5.18  0.0316

                   Bounds on condition number: 8.7312, 104.83
--------------------------------------------------------------------------------
```

...followed by Weight. No other variables are deleted from the model since the variables remaining (Age,RunTime, RunPulse, and MaxPulse) are all significant at the 10% (the default value of the SLS option is 0.1 for the BACKWARD elimination method) significance level.

```
                          Backward Elimination: Step 2


          Variable Weight Removed: R-Square = 0.8368 and C(p) = 4.8800


                              Analysis of Variance

                                       Sum of          Mean
 Source                    DF         Squares        Square    F Value    Pr > F

 Model                      4       712.45153     178.11288      33.33    <.0001
 Error                     26       138.93002       5.34346
 Corrected Total           30       851.38154


                    Parameter     Standard
       Variable      Estimate        Error   Type II SS  F Value  Pr > F

       Intercept     98.14789     11.78569    370.57373    69.35  <.0001
       Age           -0.19773      0.09564     22.84231     4.27  0.0488
       RunTime       -2.76758      0.34054    352.93570    66.05  <.0001
       RunPulse      -0.34811      0.11750     46.90089     8.78  0.0064
       MaxPulse       0.27051      0.13362     21.90067     4.10  0.0533

                    Bounds on condition number: 8.4182, 76.851
--------------------------------------------------------------------------------


      All variables left in the model are significant at the 0.1000 level.



                         Summary of Backward Elimination

        Variable    Number    Partial     Model
 Step   Removed     Vars In   R-Square   R-Square    C(p)     F Value   Pr > F

   1    RestPulse       5      0.0007     0.8480     5.1063      0.11   0.7473
   2    Weight          4      0.0112     0.8368     4.8800      1.84   0.1871
```

The MAXR method tries to find the "best" one-variable model, the "best" two-variable model, and so on. For the fitness data, the one-variable model contains RunTime; the two-variable model contains RunTime and Age;

Output 61.1.3. Maximum R-Square Improvement Selection Method: PROC REG

```
                                The REG Procedure
                                 Model: MODEL3
                           Dependent Variable: Oxygen

                       Maximum R-Square Improvement: Step 1

             Variable RunTime Entered: R-Square = 0.7434 and C(p) = 13.6988


                               Analysis of Variance

                                       Sum of           Mean
 Source                   DF          Squares         Square    F Value    Pr > F

 Model                     1        632.90010      632.90010      84.01    <.0001
 Error                    29        218.48144        7.53384
 Corrected Total          30        851.38154


                   Parameter     Standard
       Variable     Estimate        Error   Type II SS  F Value  Pr > F

       Intercept    82.42177      3.85530   3443.36654   457.05  <.0001
       RunTime      -3.31056      0.36119    632.90010    84.01  <.0001

                        Bounds on condition number: 1, 1
--------------------------------------------------------------------------------


                The above model is the best  1-variable model found.

                       Maximum R-Square Improvement: Step 2


               Variable Age Entered: R-Square = 0.7642 and C(p) = 12.3894


                               Analysis of Variance

                                       Sum of           Mean
 Source                   DF          Squares         Square    F Value    Pr > F

 Model                     2        650.66573      325.33287      45.38    <.0001
 Error                    28        200.71581        7.16842
 Corrected Total          30        851.38154


                   Parameter     Standard
       Variable     Estimate        Error   Type II SS  F Value  Pr > F

       Intercept    88.46229      5.37264   1943.41071   271.11  <.0001
       Age          -0.15037      0.09551     17.76563     2.48  0.1267
       RunTime      -3.20395      0.35877    571.67751    79.75  <.0001

                   Bounds on condition number: 1.0369, 4.1478
--------------------------------------------------------------------------------


                The above model is the best  2-variable model found.
```

the three-variable model contains RunTime, Age, and RunPulse; the four-variable model contains Age, RunTime, RunPulse, and MaxPulse; the five-variable model

```
                       Maximum R-Square Improvement: Step 3


          Variable RunPulse Entered: R-Square = 0.8111 and C(p) = 6.9596


                               Analysis of Variance

                                        Sum of           Mean
Source                   DF            Squares         Square    F Value    Pr > F

Model                     3          690.55086      230.18362      38.64    <.0001
Error                    27          160.83069        5.95669
Corrected Total          30          851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    111.71806     10.23509    709.69014   119.14  <.0001
      Age           -0.25640      0.09623     42.28867     7.10  0.0129
      RunTime       -2.82538      0.35828    370.43529    62.19  <.0001
      RunPulse      -0.13091      0.05059     39.88512     6.70  0.0154

                  Bounds on condition number: 1.3548, 11.597
--------------------------------------------------------------------------------


               The above model is the best  3-variable model found.

                       Maximum R-Square Improvement: Step 4


          Variable MaxPulse Entered: R-Square = 0.8368 and C(p) = 4.8800


                               Analysis of Variance

                                        Sum of           Mean
Source                   DF            Squares         Square    F Value    Pr > F

Model                     4          712.45153      178.11288      33.33    <.0001
Error                    26          138.93002        5.34346
Corrected Total          30          851.38154


                   Parameter     Standard
      Variable      Estimate        Error   Type II SS  F Value  Pr > F

      Intercept     98.14789     11.78569    370.57373    69.35  <.0001
      Age           -0.19773      0.09564     22.84231     4.27  0.0488
      RunTime       -2.76758      0.34054    352.93570    66.05  <.0001
      RunPulse      -0.34811      0.11750     46.90089     8.78  0.0064
      MaxPulse       0.27051      0.13362     21.90067     4.10  0.0533

                  Bounds on condition number: 8.4182, 76.851
--------------------------------------------------------------------------------


               The above model is the best  4-variable model found.
```

contains Age, Weight, RunTime, RunPulse, and MaxPulse; and finally, the six-variable model contains all the variables in the MODEL statement.

```
                       Maximum R-Square Improvement: Step 5


             Variable Weight Entered: R-Square = 0.8480 and C(p) = 5.1063


                               Analysis of Variance

                                      Sum of           Mean
Source                    DF         Squares         Square    F Value    Pr > F

Model                      5       721.97309      144.39462      27.90    <.0001
Error                     25       129.40845        5.17634
Corrected Total           30       851.38154


                  Parameter     Standard
      Variable     Estimate        Error   Type II SS  F Value  Pr > F

      Intercept   102.20428     11.97929    376.78935    72.79  <.0001
      Age          -0.21962      0.09550     27.37429     5.29  0.0301
      Weight       -0.07230      0.05331      9.52157     1.84  0.1871
      RunTime      -2.68252      0.34099    320.35968    61.89  <.0001
      RunPulse     -0.37340      0.11714     52.59624    10.16  0.0038
      MaxPulse      0.30491      0.13394     26.82640     5.18  0.0316

                  Bounds on condition number: 8.7312, 104.83
--------------------------------------------------------------------------------


               The above model is the best  5-variable model found.

                       Maximum R-Square Improvement: Step 6


           Variable RestPulse Entered: R-Square = 0.8487 and C(p) = 7.0000


                               Analysis of Variance

                                      Sum of           Mean
Source                    DF         Squares         Square    F Value    Pr > F

Model                      6       722.54361      120.42393      22.43    <.0001
Error                     24       128.83794        5.36825
Corrected Total           30       851.38154


                  Parameter     Standard
      Variable     Estimate        Error   Type II SS  F Value  Pr > F

      Intercept   102.93448     12.40326    369.72831    68.87  <.0001
      Age          -0.22697      0.09984     27.74577     5.17  0.0322
      Weight       -0.07418      0.05459      9.91059     1.85  0.1869
      RunTime      -2.62865      0.38456    250.82210    46.72  <.0001
      RunPulse     -0.36963      0.11985     51.05806     9.51  0.0051
      RestPulse    -0.02153      0.06605      0.57051     0.11  0.7473
      MaxPulse      0.30322      0.13650     26.49142     4.93  0.0360

                  Bounds on condition number: 8.7438, 137.13
--------------------------------------------------------------------------------


               The above model is the best  6-variable model found.


                 No further improvement in R-Square is possible.
```

Note that for all three of these methods, RestPulse contributes least to the model. In the case of forward selection, it is not added to the model. In the case of backward selection, it is the first variable to be removed from the model. In the case of MAXR selection, RestPulse is included only for the full model.

For the STEPWISE, BACKWARDS and FORWARD selection methods, you can control the amount of detail displayed by using the DETAILS option. For example, the following statements display only the selection summary table for the FORWARD selection method.

```
proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=forward details=summary;
run;
```

Output 61.1.4. Forward Selection Summary

```
                              The REG Procedure
                                Model: MODEL1
                        Dependent Variable: Oxygen





                        Summary of Forward Selection

      Variable     Number    Partial     Model
Step  Entered      Vars In   R-Square    R-Square     C(p)     F Value   Pr > F

  1   RunTime            1    0.7434     0.7434     13.6988     84.01   <.0001
  2   Age                2    0.0209     0.7642     12.3894      2.48   0.1267
  3   RunPulse           3    0.0468     0.8111      6.9596      6.70   0.0154
  4   MaxPulse           4    0.0257     0.8368      4.8800      4.10   0.0533
  5   Weight             5    0.0112     0.8480      5.1063      1.84   0.1871
```

Next, the RSQUARE model-selection method is used to request R^2 and C_p statistics for all possible combinations of the six independent variables. The following statements produce Output 61.1.5

```
model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
      / selection=rsquare cp;
title 'Physical fitness data: all models';
run;
```

Output 61.1.5. All Models by the RSQUARE Method: PROC REG

```
                       Physical fitness data: all models

                               The REG Procedure
                                 Model: MODEL2
                          Dependent Variable: Oxygen

                            R-Square Selection Method

Number in
  Model    R-Square      C(p)  Variables in Model

       1     0.7434   13.6988  RunTime
       1     0.1595  106.3021  RestPulse
       1     0.1584  106.4769  RunPulse
       1     0.0928  116.8818  Age
       1     0.0560  122.7072  MaxPulse
       1     0.0265  127.3948  Weight
------------------------------------------------------------------------------
       2     0.7642   12.3894  Age RunTime
       2     0.7614   12.8372  RunTime RunPulse
       2     0.7452   15.4069  RunTime MaxPulse
       2     0.7449   15.4523  Weight RunTime
       2     0.7435   15.6746  RunTime RestPulse
       2     0.3760   73.9645  Age RunPulse
       2     0.3003   85.9742  Age RestPulse
       2     0.2894   87.6951  RunPulse MaxPulse
       2     0.2600   92.3638  Age MaxPulse
       2     0.2350   96.3209  RunPulse RestPulse
       2     0.1806  104.9523  Weight RestPulse
       2     0.1740  105.9939  RestPulse MaxPulse
       2     0.1669  107.1332  Weight RunPulse
       2     0.1506  109.7057  Age Weight
       2     0.0675  122.8881  Weight MaxPulse
------------------------------------------------------------------------------
```

```
       3     0.8111    6.9596  Age RunTime RunPulse
       3     0.8100    7.1350  RunTime RunPulse MaxPulse
       3     0.7817   11.6167  Age RunTime MaxPulse
       3     0.7708   13.3453  Age Weight RunTime
       3     0.7673   13.8974  Age RunTime RestPulse
       3     0.7619   14.7619  RunTime RunPulse RestPulse
       3     0.7618   14.7729  Weight RunTime RunPulse
       3     0.7462   17.2588  Weight RunTime MaxPulse
       3     0.7452   17.4060  RunTime RestPulse MaxPulse
       3     0.7451   17.4243  Weight RunTime RestPulse
       3     0.4666   61.5873  Age RunPulse RestPulse
       3     0.4223   68.6250  Age RunPulse MaxPulse
       3     0.4091   70.7102  Age Weight RunPulse
       3     0.3900   73.7424  Age RestPulse MaxPulse
       3     0.3568   79.0013  Age Weight RestPulse
       3     0.3538   79.4891  RunPulse RestPulse MaxPulse
       3     0.3208   84.7216  Weight RunPulse MaxPulse
       3     0.2902   89.5693  Age Weight MaxPulse
       3     0.2447   96.7952  Weight RunPulse RestPulse
       3     0.1882  105.7430  Weight RestPulse MaxPulse
-------------------------------------------------------------------------------
       4     0.8368    4.8800  Age RunTime RunPulse MaxPulse
       4     0.8165    8.1035  Age Weight RunTime RunPulse
       4     0.8158    8.2056  Weight RunTime RunPulse MaxPulse
       4     0.8117    8.8683  Age RunTime RunPulse RestPulse
       4     0.8104    9.0697  RunTime RunPulse RestPulse MaxPulse
       4     0.7862   12.9039  Age Weight RunTime MaxPulse
       4     0.7834   13.3468  Age RunTime RestPulse MaxPulse
       4     0.7750   14.6788  Age Weight RunTime RestPulse
       4     0.7623   16.7058  Weight RunTime RunPulse RestPulse
       4     0.7462   19.2550  Weight RunTime RestPulse MaxPulse
       4     0.5034   57.7590  Age Weight RunPulse RestPulse
       4     0.5025   57.9092  Age RunPulse RestPulse MaxPulse
       4     0.4717   62.7830  Age Weight RunPulse MaxPulse
       4     0.4256   70.0963  Age Weight RestPulse MaxPulse
       4     0.3858   76.4100  Weight RunPulse RestPulse MaxPulse
-------------------------------------------------------------------------------
       5     0.8480    5.1063  Age Weight RunTime RunPulse MaxPulse
       5     0.8370    6.8461  Age RunTime RunPulse RestPulse MaxPulse
       5     0.8176    9.9348  Age Weight RunTime RunPulse RestPulse
       5     0.8161   10.1685  Weight RunTime RunPulse RestPulse MaxPulse
       5     0.7887   14.5111  Age Weight RunTime RestPulse MaxPulse
       5     0.5541   51.7233  Age Weight RunPulse RestPulse MaxPulse
-------------------------------------------------------------------------------
       6     0.8487    7.0000  Age Weight RunTime RunPulse RestPulse MaxPulse
```

The models in Output 61.1.5 are arranged first by the number of variables in the model and second by the magnitude of R^2 for the model. Before making a final decision about which model to use, you would want to perform collinearity diagnostics. Note that, since many different models have been fit and the choice of a final model is based on R^2, the statistics are biased and the p-values for the parameter estimates are not valid.

Example 61.2. Predicting Weight by Height and Age

In this example, the weights of school children are modeled as a function of their heights and ages. Modeling is performed separately for boys and girls. The example shows the use of a BY statement with PROC REG, multiple MODEL statements, and the OUTEST= and OUTSSCP= options, which create data sets. Since the BY statement is used, interactive processing is not possible in this example; no statements can appear after the first RUN statement. The following statements produce Output 61.2.1 through Output 61.2.4:

```
*------------Data on Age, Weight, and Height of Children-------*
| Age (months), height (inches), and weight (pounds) were      |
| recorded for a group of school children.                     |
| From Lewis and Taylor (1967).                                |
*--------------------------------------------------------------*;

data htwt;
   input sex $ age :3.1 height weight @@;
   datalines;
f 143 56.3  85.0 f 155 62.3 105.0 f 153 63.3 108.0 f 161 59.0  92.0
f 191 62.5 112.5 f 171 62.5 112.0 f 185 59.0 104.0 f 142 56.5  69.0
f 160 62.0  94.5 f 140 53.8  68.5 f 139 61.5 104.0 f 178 61.5 103.5
f 157 64.5 123.5 f 149 58.3  93.0 f 143 51.3  50.5 f 145 58.8  89.0
f 191 65.3 107.0 f 150 59.5  78.5 f 147 61.3 115.0 f 180 63.3 114.0
f 141 61.8  85.0 f 140 53.5  81.0 f 164 58.0  83.5 f 176 61.3 112.0
f 185 63.3 101.0 f 166 61.5 103.5 f 175 60.8  93.5 f 180 59.0 112.0
f 210 65.5 140.0 f 146 56.3  83.5 f 170 64.3  90.0 f 162 58.0  84.0
f 149 64.3 110.5 f 139 57.5  96.0 f 186 57.8  95.0 f 197 61.5 121.0
f 169 62.3  99.5 f 177 61.8 142.5 f 185 65.3 118.0 f 182 58.3 104.5
f 173 62.8 102.5 f 166 59.3  89.5 f 168 61.5  95.0 f 169 62.0  98.5
f 150 61.3  94.0 f 184 62.3 108.0 f 139 52.8  63.5 f 147 59.8  84.5
f 144 59.5  93.5 f 177 61.3 112.0 f 178 63.5 148.5 f 197 64.8 112.0
f 146 60.0 109.0 f 145 59.0  91.5 f 147 55.8  75.0 f 145 57.8  84.0
f 155 61.3 107.0 f 167 62.3  92.5 f 183 64.3 109.5 f 143 55.5  84.0
f 183 64.5 102.5 f 185 60.0 106.0 f 148 56.3  77.0 f 147 58.3 111.5
f 154 60.0 114.0 f 156 54.5  75.0 f 144 55.8  73.5 f 154 62.8  93.5
f 152 60.5 105.0 f 191 63.3 113.5 f 190 66.8 140.0 f 140 60.0  77.0
f 148 60.5  84.5 f 189 64.3 113.5 f 143 58.3  77.5 f 178 66.5 117.5
f 164 65.3  98.0 f 157 60.5 112.0 f 147 59.5 101.0 f 148 59.0  95.0
f 177 61.3  81.0 f 171 61.5  91.0 f 172 64.8 142.0 f 190 56.8  98.5
f 183 66.5 112.0 f 143 61.5 116.5 f 179 63.0  98.5 f 186 57.0  83.5
f 182 65.5 133.0 f 182 62.0  91.5 f 142 56.0  72.5 f 165 61.3 106.5
f 165 55.5  67.0 f 154 61.0 122.5 f 150 54.5  74.0 f 155 66.0 144.5
f 163 56.5  84.0 f 141 56.0  72.5 f 147 51.5  64.0 f 210 62.0 116.0
f 171 63.0  84.0 f 167 61.0  93.5 f 182 64.0 111.5 f 144 61.0  92.0
f 193 59.8 115.0 f 141 61.3  85.0 f 164 63.3 108.0 f 186 63.5 108.0
f 169 61.5  85.0 f 175 60.3  86.0 f 180 61.3 110.5 m 165 64.8  98.0
m 157 60.5 105.0 m 144 57.3  76.5 m 150 59.5  84.0 m 150 60.8 128.0
m 139 60.5  87.0 m 189 67.0 128.0 m 183 64.8 111.0 m 147 50.5  79.0
m 146 57.5  90.0 m 160 60.5  84.0 m 156 61.8 112.0 m 173 61.3  93.0
m 151 66.3 117.0 m 141 53.3  84.0 m 150 59.0  99.5 m 164 57.8  95.0
m 153 60.0  84.0 m 206 68.3 134.0 m 250 67.5 171.5 m 176 63.8  98.5
m 176 65.0 118.5 m 140 59.5  94.5 m 185 66.0 105.0 m 180 61.8 104.0
m 146 57.3  83.0 m 183 66.0 105.5 m 140 56.5  84.0 m 151 58.3  86.0
m 151 61.0  81.0 m 144 62.8  94.0 m 160 59.3  78.5 m 178 67.3 119.5
m 193 66.3 133.0 m 162 64.5 119.0 m 164 60.5  95.0 m 186 66.0 112.0
m 143 57.5  75.0 m 175 64.0  92.0 m 175 68.0 112.0 m 175 63.5  98.5
```

```
m 173 69.0 112.5 m 170 63.8 112.5 m 174 66.0 108.0 m 164 63.5 108.0
m 144 59.5  88.0 m 156 66.3 106.0 m 149 57.0  92.0 m 144 60.0 117.5
m 147 57.0  84.0 m 188 67.3 112.0 m 169 62.0 100.0 m 172 65.0 112.0
m 150 59.5  84.0 m 193 67.8 127.5 m 157 58.0  80.5 m 168 60.0  93.5
m 140 58.5  86.5 m 156 58.3  92.5 m 156 61.5 108.5 m 158 65.0 121.0
m 184 66.5 112.0 m 156 68.5 114.0 m 144 57.0  84.0 m 176 61.5  81.0
m 168 66.5 111.5 m 149 52.5  81.0 m 142 55.0  70.0 m 188 71.0 140.0
m 203 66.5 117.0 m 142 58.8  84.0 m 189 66.3 112.0 m 188 65.8 150.5
m 200 71.0 147.0 m 152 59.5 105.0 m 174 69.8 119.5 m 166 62.5  84.0
m 145 56.5  91.0 m 143 57.5 101.0 m 163 65.3 117.5 m 166 67.3 121.0
m 182 67.0 133.0 m 173 66.0 112.0 m 155 61.8  91.5 m 162 60.0 105.0
m 177 63.0 111.0 m 177 60.5 112.0 m 175 65.5 114.0 m 166 62.0  91.0
m 150 59.0  98.0 m 150 61.8 118.0 m 188 63.3 115.5 m 163 66.0 112.0
m 171 61.8 112.0 m 162 63.0  91.0 m 141 57.5  85.0 m 174 63.0 112.0
m 142 56.0  87.5 m 148 60.5 118.0 m 140 56.8  83.5 m 160 64.0 116.0
m 144 60.0  89.0 m 206 69.5 171.5 m 159 63.3 112.0 m 149 56.3  72.0
m 193 72.0 150.0 m 194 65.3 134.5 m 152 60.8  97.0 m 146 55.0  71.5
m 139 55.0  73.5 m 186 66.5 112.0 m 161 56.8  75.0 m 153 64.8 128.0
m 196 64.5  98.0 m 164 58.0  84.0 m 159 62.8  99.0 m 178 63.8 112.0
m 153 57.8  79.5 m 155 57.3  80.5 m 178 63.5 102.5 m 142 55.0  76.0
m 164 66.5 112.0 m 189 65.0 114.0 m 164 61.5 140.0 m 167 62.0 107.5
m 151 59.3  87.0
;

title '----- Data on age, weight, and height of children ------';

proc reg outest=est1 outsscp=sscp1 rsquare;
   by sex;
   eq1: model  weight=height;
   eq2: model  weight=height age;

proc print data=sscp1;
   title2 'SSCP type data set';

proc print data=est1;
   title2 'EST type data set';
run;
```

Output 61.2.1. Height and Weight Data: Female Children

```
             ----- Data on age, weight, and height of children ------

------------------------------------ sex=f ------------------------------------

                               The REG Procedure
                                  Model: eq1
                          Dependent Variable: weight

                              Analysis of Variance

                                    Sum of           Mean
 Source                   DF       Squares         Square    F Value    Pr > F

 Model                     1         21507          21507     141.09    <.0001
 Error                   109         16615      152.42739
 Corrected Total         110         38121


              Root MSE             12.34615    R-Square     0.5642
              Dependent Mean       98.87838    Adj R-Sq     0.5602
              Coeff Var            12.48620


                              Parameter Estimates

                        Parameter       Standard
      Variable     DF    Estimate          Error    t Value    Pr > |t|

      Intercept     1    -153.12891      21.24814      -7.21      <.0001
      height        1       4.16361       0.35052      11.88      <.0001
```

```
             ----- Data on age, weight, and height of children ------

------------------------------------ sex=f ------------------------------------

                               The REG Procedure
                                  Model: eq2
                          Dependent Variable: weight

                              Analysis of Variance

                                    Sum of           Mean
 Source                   DF       Squares         Square    F Value    Pr > F

 Model                     2         22432          11216      77.21    <.0001
 Error                   108         15689      145.26700
 Corrected Total         110         38121


              Root MSE             12.05268    R-Square     0.5884
              Dependent Mean       98.87838    Adj R-Sq     0.5808
              Coeff Var            12.18939


                              Parameter Estimates

                        Parameter       Standard
      Variable     DF    Estimate          Error    t Value    Pr > |t|

      Intercept     1    -150.59698      20.76730      -7.25      <.0001
      height        1       3.60378       0.40777       8.84      <.0001
      age           1       1.90703       0.75543       2.52      0.0130
```

Output 61.2.2. Height and Weight Data: Male Children

```
            ----- Data on age, weight, and height of children ------

------------------------------------ sex=m -------------------------------------

                               The REG Procedure
                                   Model: eq1
                          Dependent Variable: weight

                              Analysis of Variance

                                     Sum of           Mean
 Source                   DF        Squares         Square    F Value    Pr > F

 Model                     1          31126          31126     206.24    <.0001
 Error                   124          18714      150.92222
 Corrected Total         125          49840


              Root MSE             12.28504    R-Square     0.6245
              Dependent Mean      103.44841    Adj R-Sq     0.6215
              Coeff Var            11.87552


                              Parameter Estimates

                           Parameter       Standard
      Variable     DF       Estimate          Error    t Value    Pr > |t|

      Intercept     1     -125.69807       15.99362      -7.86      <.0001
      height        1        3.68977        0.25693      14.36      <.0001
```

```
            ----- Data on age, weight, and height of children ------

------------------------------------ sex=m -------------------------------------

                               The REG Procedure
                                   Model: eq2
                          Dependent Variable: weight

                              Analysis of Variance

                                     Sum of           Mean
 Source                   DF        Squares         Square    F Value    Pr > F

 Model                     2          32975          16487     120.24    <.0001
 Error                   123          16866      137.11922
 Corrected Total         125          49840


              Root MSE             11.70979    R-Square     0.6616
              Dependent Mean      103.44841    Adj R-Sq     0.6561
              Coeff Var            11.31945


                              Parameter Estimates

                           Parameter       Standard
      Variable     DF       Estimate          Error    t Value    Pr > |t|

      Intercept     1     -113.71346       15.59021      -7.29      <.0001
      height        1        2.68075        0.36809       7.28      <.0001
      age           1        3.08167        0.83927       3.67      0.0004
```

For both females and males, the overall F statistics for both models are significant, indicating that the model explains a significant portion of the variation in the data. For females, the full model is

$$\text{weight} = -150.57 + 3.60 \times \text{height} + 1.91 \times \text{age}$$

and, for males, the full model is

$$\text{weight} = -113.71 + 2.68 \times \text{height} + 3.08 \times \text{age}$$

Output 61.2.3. SSCP Matrix

```
               ----- Data on age, weight, and height of children ------
                                     SSCP type data set

Obs    sex    _TYPE_    _NAME_      Intercept       height        weight          age

  1     f      SSCP     Intercept       111.0      6718.40      10975.50      1824.90
  2     f      SSCP     height         6718.4    407879.32     669469.85    110818.32
  3     f      SSCP     weight        10975.5    669469.85    1123360.75    182444.95
  4     f      SSCP     age            1824.9    110818.32     182444.95     30363.81
  5     f      N                        111.0       111.00        111.00       111.00
  6     m      SSCP     Intercept       126.0      7825.00      13034.50      2072.10
  7     m      SSCP     height         7825.0    488243.60     817919.60    129432.57
  8     m      SSCP     weight        13034.5    817919.60    1398238.75    217717.45
  9     m      SSCP     age            2072.1    129432.57     217717.45     34515.95
 10     m      N                        126.0       126.00        126.00       126.00
```

The OUTSSCP= data set is shown in Output 61.2.3. Note how the BY groups are separated. Observations with _TYPE_=‘N’ contain the number of observations in the associated BY group. Observations with _TYPE_=‘SSCP’ contain the rows of the uncorrected sums of squares and crossproducts matrix. The observations with _NAME_=‘Intercept’ contain crossproducts for the intercept.

Output 61.2.4. OUTEST Data Set

```
                     ----- Data on age, weight, and height of children ------
                                        EST type data set

Obs sex _MODEL_ _TYPE_ _DEPVAR_  _RMSE_ Intercept  height weight   age    _IN_ _P_ _EDF_  _RSQ_

 1   f    eq1    PARMS   weight  12.3461  -153.129 4.16361   -1      .        1   2    109  0.56416
 2   f    eq2    PARMS   weight  12.0527  -150.597 3.60378   -1   1.90703     2   3    108  0.58845
 3   m    eq1    PARMS   weight  12.2850  -125.698 3.68977   -1      .        1   2    124  0.62451
 4   m    eq2    PARMS   weight  11.7098  -113.713 2.68075   -1   3.08167     2   3    123  0.66161
```

The OUTEST= data set is displayed in Output 61.2.4; again, the BY groups are separated. The _MODEL_ column contains the labels for models from the MODEL statements. If no labels are specified, the defaults MODEL1 and MODEL2 would appear as values for _MODEL_. Note that _TYPE_=‘PARMS’ for all observations, indicating that all observations contain parameter estimates. The _DEPVAR_ column displays the dependent variable, and the _RMSE_ column gives the Root Mean Square Error for the associated model. The Intercept column gives the estimate for the intercept for the associated model, and variables with the same name as variables in the original data set (height, age) give parameter estimates for those variables. The dependent variable, weight, is shown with a value of -1. The _IN_ column contains the number of regressors in the model not including the intercept; _P_ contains the number of parameters in the model; _EDF_ contains the error degrees of freedom; and _RSQ_ contains the R^2 statistic. Finally, note that the _IN_, _P_, _EDF_ and _RSQ_ columns appear in the OUTEST= data set since the RSQUARE option is specified in the PROC REG statement.

Example 61.3. Regression with Quantitative and Qualitative Variables

At times it is desirable to have independent variables in the model that are qualitative rather than quantitative. This is easily handled in a regression framework. Regression uses qualitative variables to distinguish between populations. There are two main advantages of fitting both populations in one model. You gain the ability to test for different slopes or intercepts in the populations, and more degrees of freedom are available for the analysis.

Regression with qualitative variables is different from analysis of variance and analysis of covariance. Analysis of variance uses qualitative independent variables only. Analysis of covariance uses quantitative variables in addition to the qualitative variables in order to account for correlation in the data and reduce MSE; however, the quantitative variables are not of primary interest and merely improve the precision of the analysis.

Consider the case where Y_i is the dependent variable, $X1_i$ is a quantitative variable, $X2_i$ is a qualitative variable taking on values 0 or 1, and $X1_iX2_i$ is the interaction. The variable $X2_i$ is called a dummy, binary, or indicator variable. With values 0 or 1, it distinguishes between two populations. The model is of the form

$$Y_i = \beta_0 + \beta_1 X1_i + \beta_2 X2_i + \beta_3 X1_i X2_i + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$. The parameters to be estimated are β_0, β_1, β_2, and β_3. The number of dummy variables used is one less than the number of qualitative levels. This yields a nonsingular $X'X$ matrix. See Chapter 10 of Neter, Wasserman, and Kutner (1990) for more details.

An example from Neter, Wasserman, and Kutner (1990) follows. An economist is investigating the relationship between the size of an insurance firm and the speed at which they implement new insurance innovations. He believes that the type of firm may affect this relationship and suspects that there may be some interaction between the size and type of firm. The dummy variable in the model allows the two firms to have different intercepts. The interaction term allows the firms to have different slopes as well.

In this study, Y_i is the number of months from the time the first firm implemented the innovation to the time it was implemented by the ith firm. The variable $X1_i$ is the size of the firm, measured in total assets of the firm. The variable $X2_i$ denotes the firm type and is 0 if the firm is a mutual fund company and 1 if the firm is a stock company. The dummy variable allows each firm type to have a different intercept and slope.

The previous model can be broken down into a model for each firm type by plugging in the values for $X2_i$. If $X2_i = 0$, the model is

$$Y_i = \beta_0 + \beta_1 X1_i + \epsilon_i$$

This is the model for a mutual company. If $X2_i = 1$, the model for a stock firm is

$$Y_i = (\beta_0 + \beta_2) + (\beta_1 + \beta_3) X1_i + \epsilon_i$$

This model has intercept $\beta_0 + \beta_2$ and slope $\beta_1 + \beta_3$.

The data* follow. Note that the interaction term is created in the DATA step since polynomial effects such as size*type are not allowed in the MODEL statement in the REG procedure.

```
title 'Regression With Quantitative and Qualitative Variables';
data insurance;
   input time size type @@;
   sizetype=size*type;
   datalines;
17 151 0   26  92 0   21 175 0   30  31 0   22 104 0
 0 277 0   12 210 0   19 120 0    4 290 0   16 238 0
28 164 1   15 272 1   11 295 1   38  68 1   31  85 1
21 224 1   20 166 1   13 305 1   30 124 1   14 246 1
;
run;
```

The following statements begin the analysis:

```
proc reg data=insurance;
   model time = size type sizetype;
run;
```

The ANOVA table is displayed in Output 61.3.1.

*From Neter, J. et al., *Applied Linear Statistical Models*, Third Edition, Copyright (c) 1990, Richard D. Irwin. Reprinted with permission of The McGraw-Hill Companies.

Output 61.3.1. ANOVA Table and Parameter Estimates

```
               Regression With Quantitative and Qualitative Variables

                                  The REG Procedure
                                    Model: MODEL1
                               Dependent Variable: time

                                Analysis of Variance

                                          Sum of           Mean
Source                     DF            Squares         Square    F Value    Pr > F

Model                       3         1504.41904      501.47301      45.49    <.0001
Error                      16          176.38096       11.02381
Corrected Total            19         1680.80000


             Root MSE              3.32021    R-Square     0.8951
             Dependent Mean       19.40000    Adj R-Sq     0.8754
             Coeff Var            17.11450


                                Parameter Estimates

                                Parameter       Standard
     Variable     DF             Estimate          Error    t Value    Pr > |t|

     Intercept     1             33.83837        2.44065      13.86      <.0001
     size          1             -0.10153        0.01305      -7.78      <.0001
     type          1              8.13125        3.65405       2.23      0.0408
     sizetype      1          -0.00041714        0.01833      -0.02      0.9821
```

The overall F statistic is significant (F=45.490, p<0.0001). The interaction term is not significant (t=−0.023, p=0.9821). Hence, this term should be removed and the model re-fitted, as shown in the following statements.

```
      delete sizetype;
      print;
   run;
```

The DELETE statement removes the interaction term (sizetype) from the model. The new ANOVA table is shown in Output 61.3.2.

Output 61.3.2. ANOVA Table and Parameter Estimates

```
              Regression With Quantitative and Qualitative Variables

                                  The REG Procedure
                                   Model: MODEL1.1
                              Dependent Variable: time

                                Analysis of Variance

                                       Sum of           Mean
Source                   DF           Squares         Square    F Value    Pr > F

Model                     2        1504.41333      752.20667      72.50    <.0001
Error                    17         176.38667       10.37569
Corrected Total          19        1680.80000


             Root MSE              3.22113    R-Square     0.8951
             Dependent Mean       19.40000    Adj R-Sq     0.8827
             Coeff Var            16.60377


                                Parameter Estimates

                             Parameter       Standard
     Variable      DF         Estimate          Error    t Value    Pr > |t|

     Intercept      1         33.87407        1.81386      18.68      <.0001
     size           1         -0.10174        0.00889     -11.44      <.0001
     type           1          8.05547        1.45911       5.52      <.0001
```

The overall F statistic is still significant (F=72.497, p<0.0001). The intercept and the coefficients associated with size and type are significantly different from zero (t=18.675, p<0.0001; t=$-$11.443, p<0.0001; t=5.521, p<0.0001, respectively). Notice that the R^2 did not change with the omission of the interaction term.

The fitted model is

$$\textsf{time} = 33.87 - 0.102 \times \textsf{size} + 8.055 \times \textsf{type}$$

The fitted model for a mutual fund company ($X2_i = 0$) is

$$\textsf{time} = 33.87 - 0.102 \times \textsf{size}$$

and the fitted model for a stock company ($X2_i = 1$) is

$$\textsf{time} = (33.87 + 8.055) - 0.102 \times \textsf{size}$$

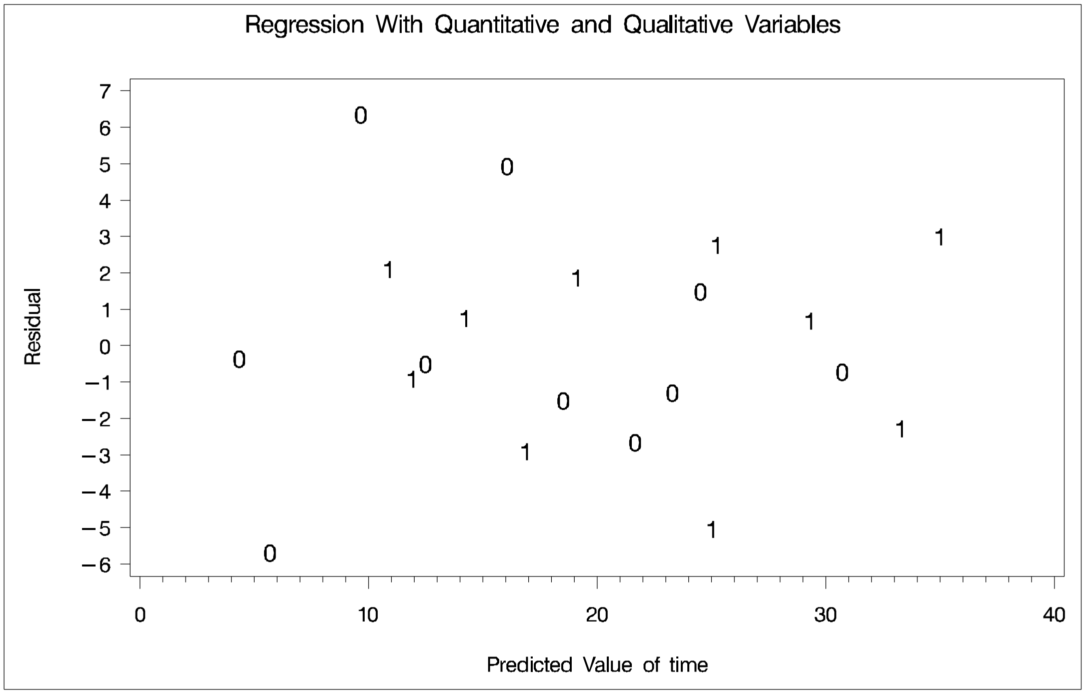
So the two models have different intercepts but the same slope.

Now plot the residual versus predicted values using the firm type as the plot symbol (PLOT=TYPE); this can be useful in determining if the firm types have different residual patterns. PROC REG does not support the `plot y*x=type` syntax for high-resolution graphics, so use PROC GPLOT to create Output 61.3.3. First, the

OUTPUT statement saves the residuals and predicted values from the new model in the OUT= data set.

```
      output out=out r=r p=p;
   run;
   symbol1 v='0' c=blue   f=swissb;
   symbol2 v='1' c=yellow f=swissb;
   axis1 label=(angle=90);
   proc gplot data=out;
      plot r*p=type    / nolegend vaxis=axis1 cframe=ligr;
      plot p*size=type / nolegend vaxis=axis1 cframe=ligr;
   run;
```

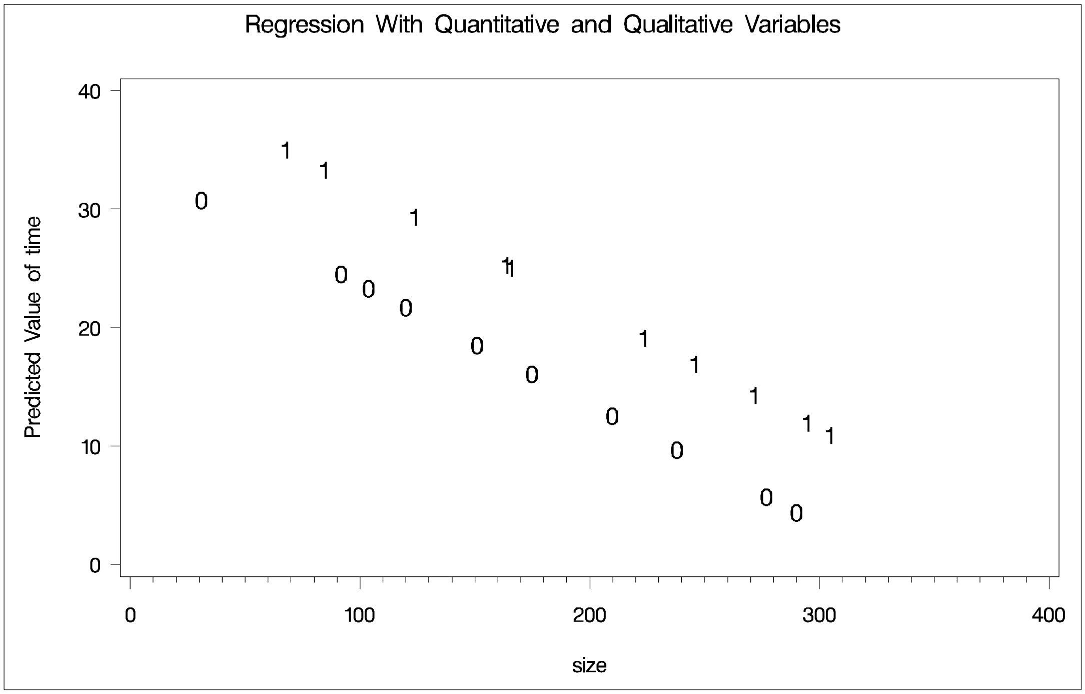
Output 61.3.3. Plot of Residual vs. Predicted Values



The residuals show no major trend. Neither firm type by itself shows a trend either. This indicates that the model is satisfactory.

A plot of the predicted values versus size appears in Output 61.3.4, where the firm type is again used as the plotting symbol.

Output 61.3.4. Plot of Predicted vs. Size



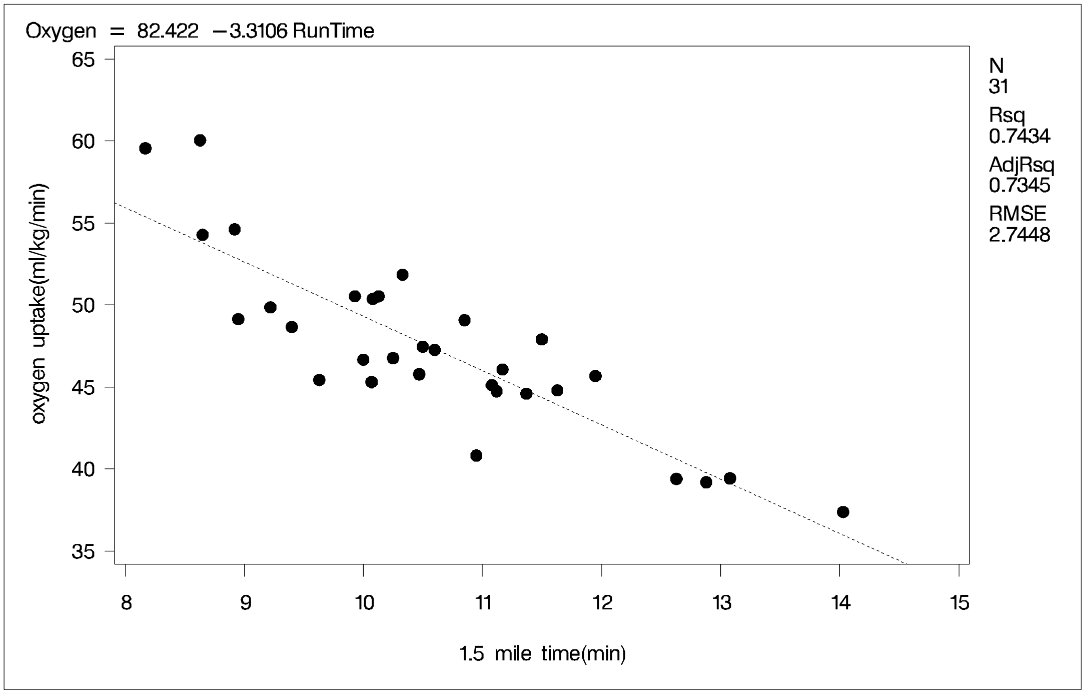
The different intercepts are very evident in this plot.

Example 61.4. Displaying Plots for Simple Linear Regression

This example introduces the basic PROC REG graphics syntax used to produce a standard plot of data from the aerobic fitness data set (Example 61.1 on page 3924). A simple linear regression of Oxygen on RunTime is performed, and a plot of Oxygen∗RunTime is requested. The fitted model, the regression line, and the four default statistics are also displayed in Output 61.4.1.

```
data fitness;
   set fitness;
   label Age      =’age(years)’
         Weight   =’weight(kg)’
         Oxygen   =’oxygen uptake(ml/kg/min)’
         RunTime  =’1.5 mile time(min)’
         RestPulse=’rest pulse’
         RunPulse =’running pulse’
         MaxPulse =’maximum running pulse’;
proc reg data=fitness;
   model Oxygen=RunTime;
   plot Oxygen*RunTime / cframe=ligr;
run;
```

Output 61.4.1. Simple Linear Regression

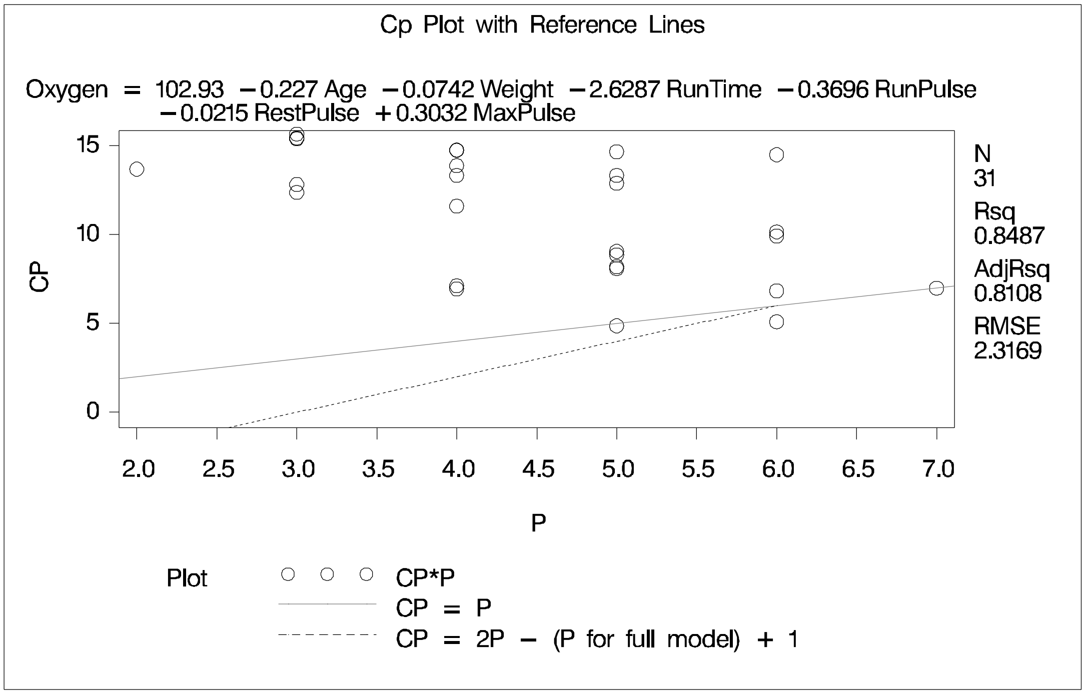


Example 61.5. Creating a C_p Plot

The C_p statistics for model selection are plotted against the number of parameters in the model, and the CHOCKING= and CMALLOWS= options draw useful reference lines. Note the four default statistics in the plot margin, the default model equation, and the default legend in Output 61.5.1.

```
title ’Cp Plot with Reference Lines’;
symbol1 c=green;
proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare noprint;
   plot cp.*np.
        / chocking=red  cmallows=blue
          vaxis=0 to 15 by 5 cframe=ligr;
run;
```

Output 61.5.1. C_p Plot



Using the criteria suggested by Hocking (1976) (see the section "Dictionary of PLOT Statement Options" beginning on page 3844), Output 61.5.1 indicates that a 6-variable model is a reasonable choice for doing parameter estimation, while a 5-variable model may be suitable for doing prediction.

Example 61.6. Controlling Plot Appearance with Graphic Options

This example uses model fit summary statistics from the OUTEST= data set to create a plot for a model selection analysis. Global graphics statements and PLOT statement options are used to control the appearance of the plot.

```
goptions ctitle=black   htitle=3.5pct ftitle=swiss
         ctext =magenta htext =3.0pct ftext =swiss
         cback =ligr    border;
symbol1 v=circle c=red h=1 w=2;
title1 'Selection=Rsquare';
title2 'plot Rsquare versus the number of parameters P in '
       'each model';

proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare noprint;
   plot rsq.*np.
        / aic bic edf gmsep jp np pc sbc sp
          haxis=2 to 7 by 1
```

```
                 caxis=red cframe=white ctext=blue
                 modellab='Full Model' modelht=2.4
                 statht=2.4;
run;
```

In the GOPTIONS statement,

BORDER frames the entire display

CBACK= specifies the background color

CTEXT= selects the default color for the border and all text, including titles, footnotes, and notes

CTITLE= specifies the title, footnote, note, and border color

HTEXT= specifies the height for all text in the display

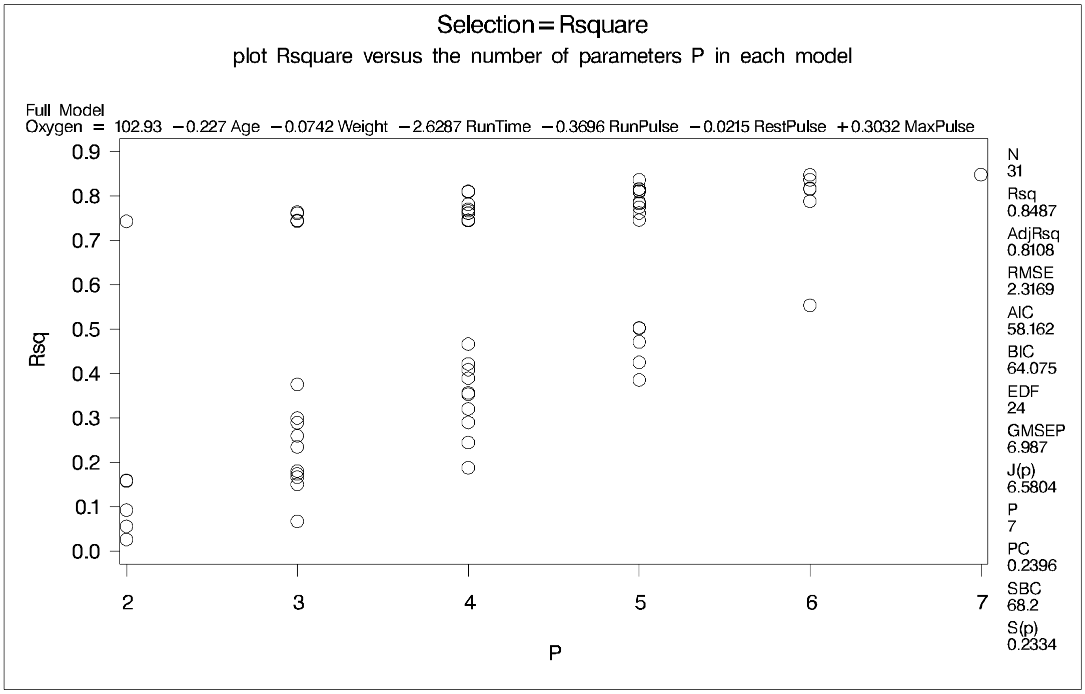
HTITLE= specifies the height for the first title line

FTEXT= selects the default font for all text, including titles, footnotes, notes, the model label and equation, the statistics, the axis labels, the tick values, and the legend

FTITLE= specifies the first title font

For more information on the GOPTIONS statement and other global graphics statements, refer to *SAS/GRAPH Software: Reference*.

Output 61.6.1. Controlling Plot Appearance and Plotting OUTEST= Statistics



In Output 61.6.1, note the following:

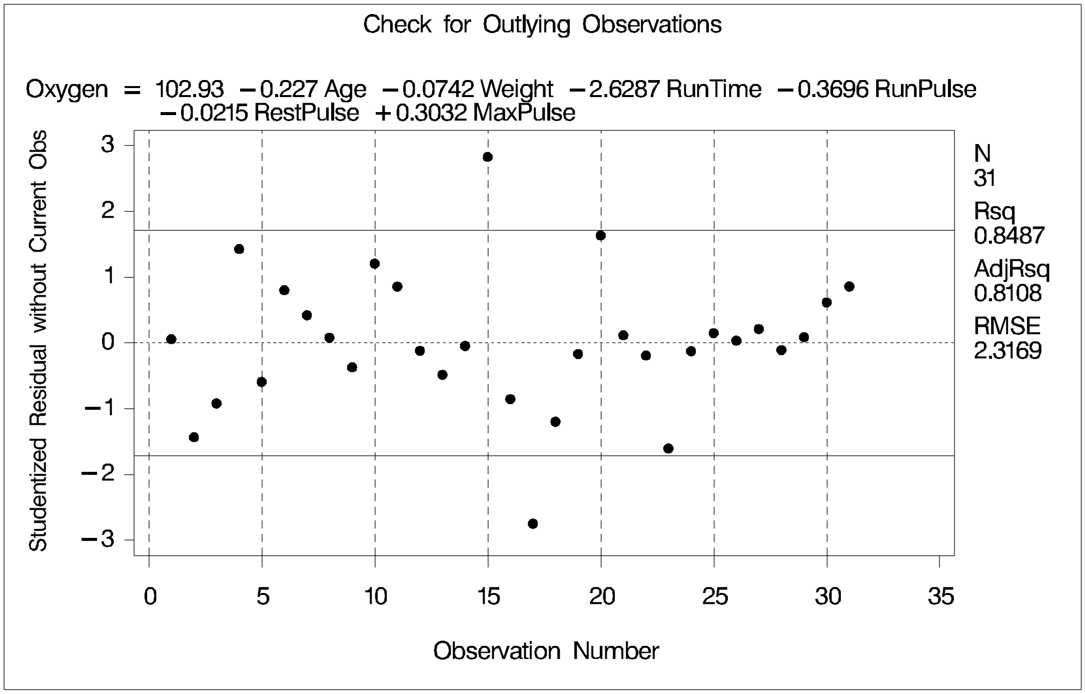
- The PLOT statement option CTEXT= affects all text not controlled by the CTITLE= option in the GOPTIONS statement. Hence, the GOPTIONS statement option CTEXT=MAGENTA has no effect. Therefore, the color of the title is black and all other text is blue.
- The area enclosed by the axes and the frame has a white background, while the background outside the plot area is gray.
- The MODELHT= option allows the entire model equation to fit on one line.
- The STATHT= option allows the statistics in the margin to fit in one column.
- The displayed statistics and the fitted model equation refer to the selected model. See the "Traditional High-Resolution Graphics Plots" section beginning on page 3840 for more information.

Example 61.7. Plotting Model Diagnostic Statistics

This example illustrates how you can display diagnostics for checking the adequacy of a regression model. The following statements plot the studentized deleted residuals against the observation number for the full model. Vertical reference lines at $\pm\text{tinv}(.95, n-p-1) = \pm 1.714$ are added to identify possible outlying Oxygen values. A vertical reference line is displayed at zero by default when the RSTUDENT option is specified. The graph is shown in Output 61.7.1. Observations 15 and 17 are indicated as possible outliers.

```
title 'Check for Outlying Observations';
symbol v=dot h=1 c=green;

proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse;
   plot rstudent.*obs.
        / vref= -1.714 1.714 cvref=blue lvref=1
          href= 0 to 30 by 5 chref=red  cframe=ligr;
run;
```

Output 61.7.1. Plotting Model Diagnostic Statistics

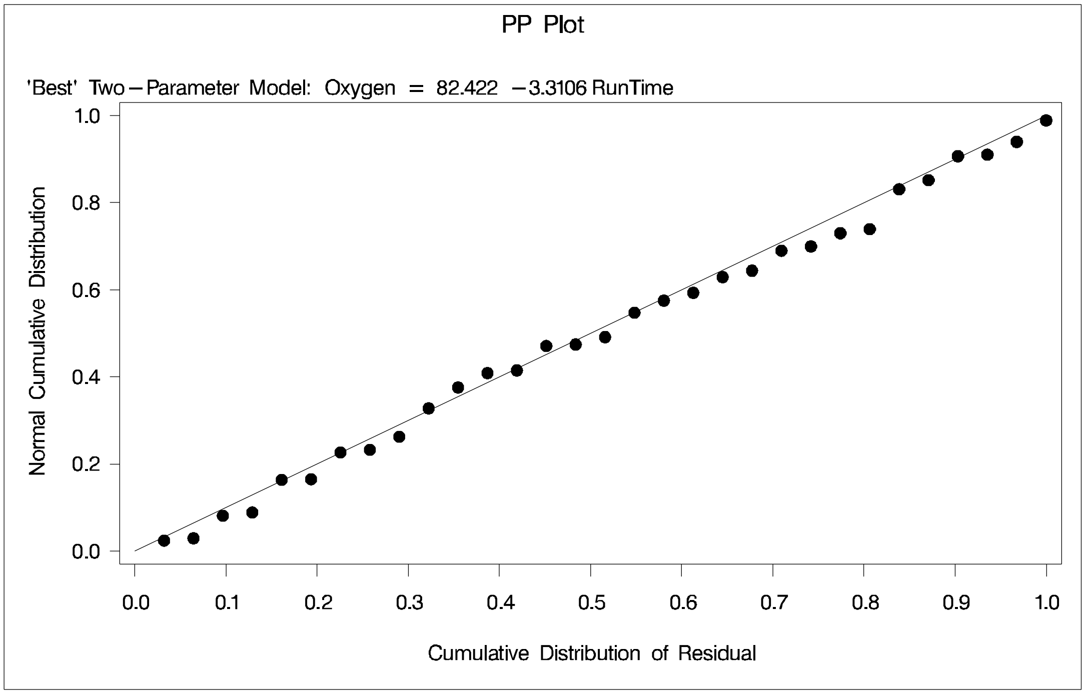


Example 61.8. Creating PP and QQ Plots

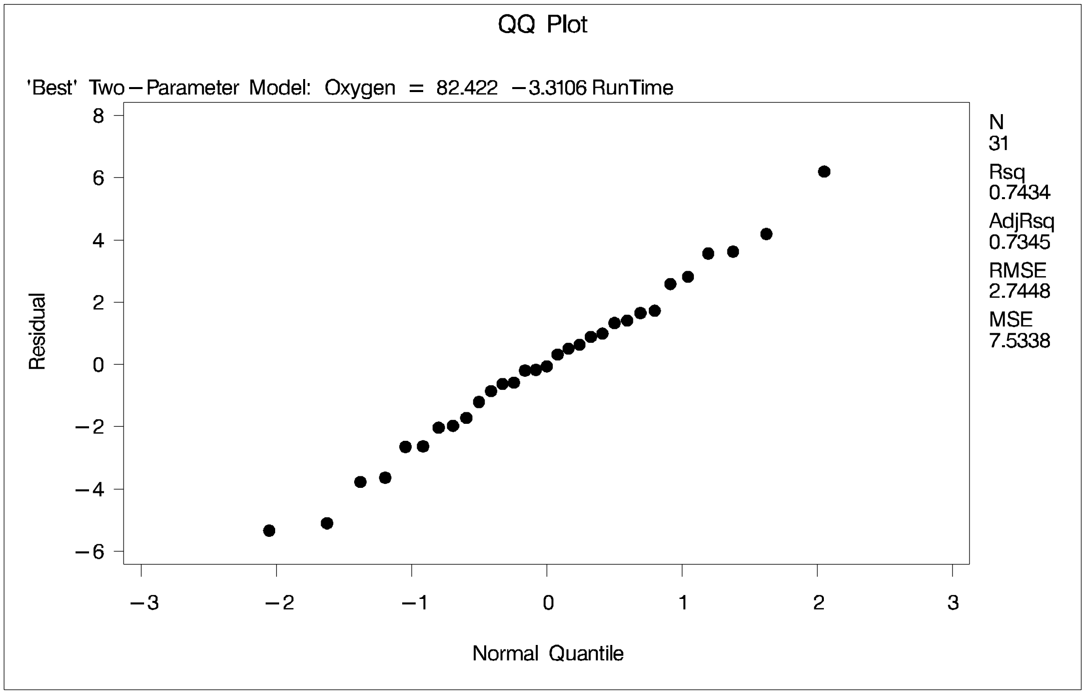
The following program creates probability-probability plots and quantile-quantile plots of the residuals (Output 61.8.1 and Output 61.8.2, respectively). An annotation data set is created to produce the (0,0)−(1,1) reference line for the PP-plot. Note that the NOSTAT option for the PP-plot suppresses the statistics that would be displayed in the margin.

```
data annote1;
   length function color $8;
   retain ysys xsys ’2’ color ’black’;
   function=’move’;
      x=0;
      y=0;
      output;
   function=’draw’;
      x=1;
      y=1;
      output;
run;

symbol1 c=blue;
proc reg data=fitness;
   title ’PP Plot’;
   model Oxygen=RunTime / noprint;
   plot npp.*r.
        / annotate=annote1 nostat cframe=ligr
          modellab="’Best’ Two-Parameter Model:";
run;
   title ’QQ Plot’;
   plot r.*nqq.
        / noline mse cframe=ligr
          modellab="’Best’ Two-Parameter Model:";
run;
```

Output 61.8.1. Normal Probability-Probability Plot for the Residuals



Output 61.8.2. Normal Quantile-Quantile Plot for the Residuals

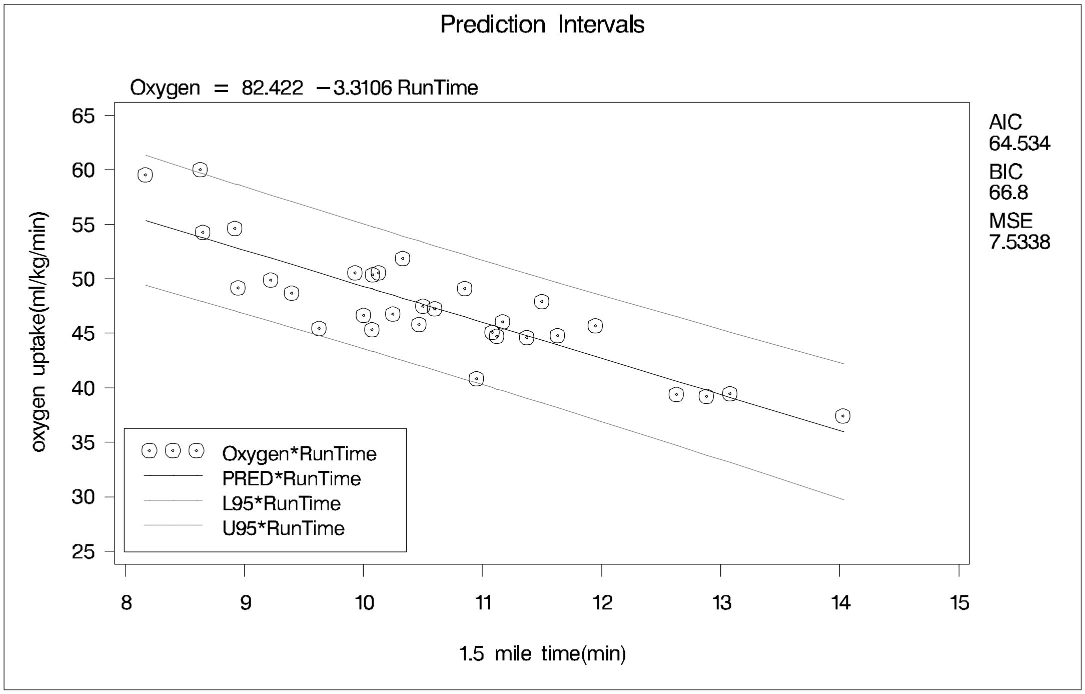


Example 61.9. Displaying Confidence and Prediction Intervals

This example illustrates how you can use shorthand commands to plot the dependent variable, the predicted value, and the 95% confidence or prediction intervals against a regressor. The following statements use the PRED option to create a plot with prediction intervals; the CONF option works similarly. Results are displayed in Output 61.9.1. Note that the statistics displayed by default in the margin are suppressed while three other statistics are exhibited.

```
legend1 position=(bottom left inside)
        across=1 cborder=red offset=(0,0)
        shape=symbol(3,1) label=none
        value=(height=.8);
title ’Prediction Intervals’;
symbol1 c=yellow v=- h=1;
symbol2 c=red;
symbol3 c=blue;
symbol4 c=blue;

proc reg data=fitness;
   model Oxygen=RunTime / noprint;
   plot Oxygen*RunTime / pred nostat mse aic bic
        caxis=red ctext=blue cframe=ligr
        legend=legend1 modellab=’         ’;
run;
```

Output 61.9.1. Prediction Intervals



Plots can be produced with both confidence and prediction intervals using the following statement.

```
plot Oxygen*RunTime / conf pred;
```

Example 61.10. Displaying the Ridge Trace for Acetylene Data

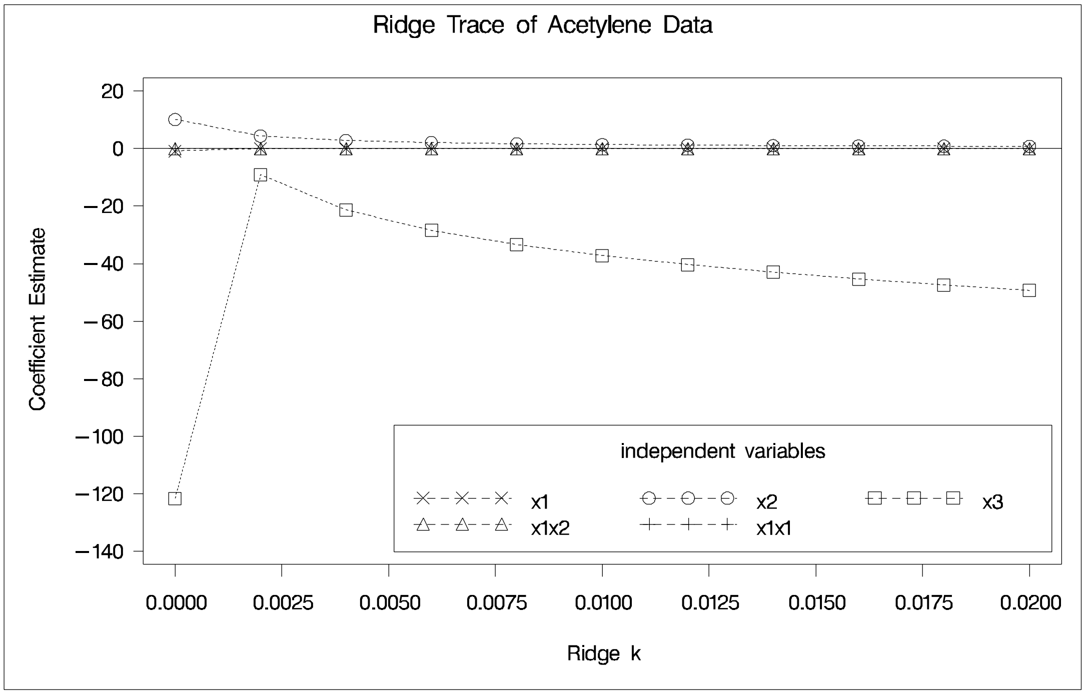
This example and Example 61.11 use the acetylene data in Marquardt and Snee (1975) to illustrate the RIDGEPLOT and OUTVIF options.

```
data acetyl;
   input x1-x4 @@;
   x1x2 = x1 * x2;
   x1x1 = x1 * x1;
   label x1  = 'reactor temperature(celsius)'
         x2  = 'h2 to n-heptone ratio'
         x3  = 'contact time(sec)'
         x4  = 'conversion percentage'
         x1x2= 'temperature-ratio interaction'
         x1x1= 'squared temperature';
   datalines;
1300  7.5 .012 49   1300  9   .012  50.2 1300 11 .0115 50.5
1300 13.5 .013 48.5 1300 17   .0135 47.5 1300 23 .012  44.5
1200  5.3 .04  28   1200  7.5 .038  31.5 1200 11 .032  34.5
1200 13.5 .026 35   1200 17   .034  38   1200 23 .041  38.5
1100  5.3 .084 15   1100  7.5 .098  17   1100 11 .092  20.5
1100 17   .086 29.5
```

```
;

title 'Ridge Trace of Acetylene Data';
symbol1 v=x        c=blue;
symbol2 v=circle   c=yellow;
symbol3 v=square   c=cyan;
symbol4 v=triangle c=green;
symbol5 v=plus     c=orange;
legend2 position=(bottom right inside)
        across=3 cborder=black offset=(0,0)
        label=(color=blue position=(top center)
               'independent variables') cframe=white;

proc reg data=acetyl outvif
         outest=b ridge=0 to 0.02 by .002;
   model x4=x1 x2 x3 x1x2 x1x1/noprint;
   plot / ridgeplot nomodel legend=legend2 nostat
          vref=0 lvref=1 cvref=blue cframe=ligr;
run;
```

The results produced by the RIDGEPLOT option are shown in Output 61.10.1. The OUTVIF option outputs the variance inflation factors to the OUTEST= data set, which is used in Example 61.11.

Output 61.10.1. Using the RIDEGPLOT Option for Ridge Regression

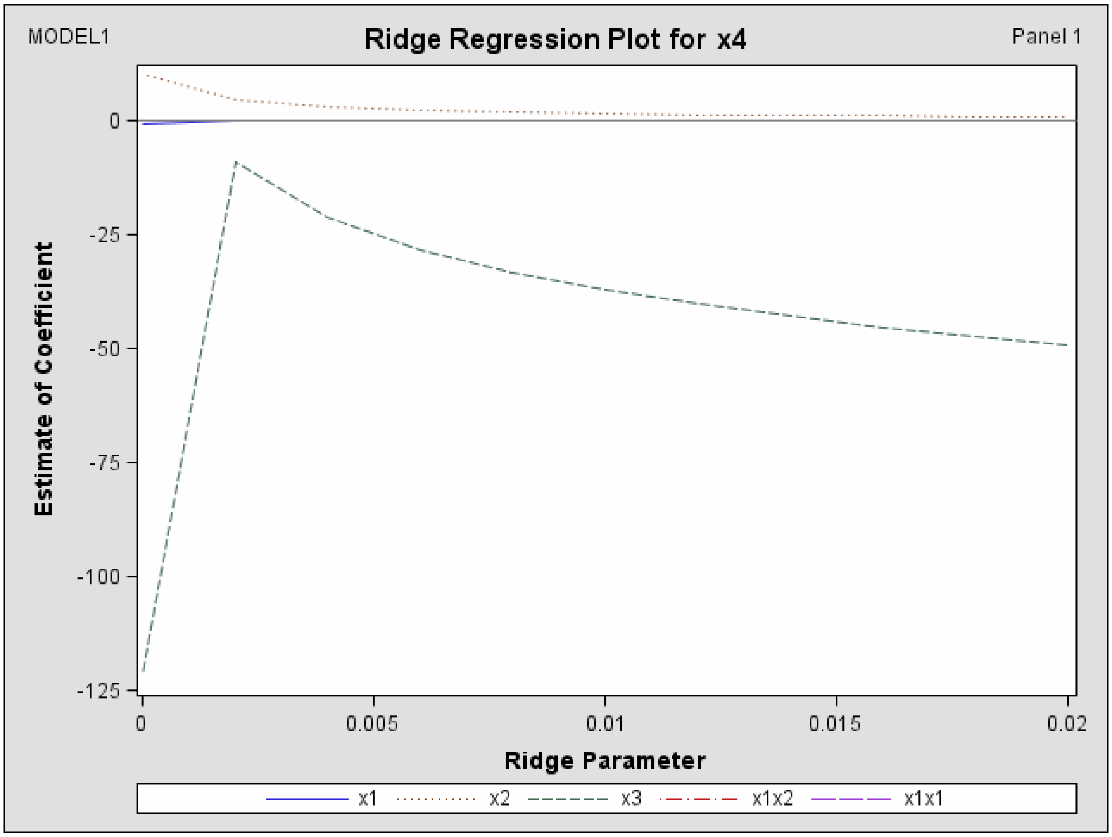


If you specify the experimental ODS GRAPHICS statement (see the "ODS Graphics" section on page 3922), a plot of ridge traces is produced, without the need to specify the RIDGEPLOT option in the PLOT statement. For general information about

ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” The following statements provide an example:

```
ods html;
ods graphics on;

proc reg data=acetyl outest=b ridge=0 to 0.02 by .002;
   model x4=x1 x2 x3 x1x2 x1x1/noprint;
run;

ods graphics off;
ods html close;
```

Output 61.10.2. Ridge Traces Produced with ODS Graphics (Experimental)



Example 61.11. Plotting Variance Inflation Factors

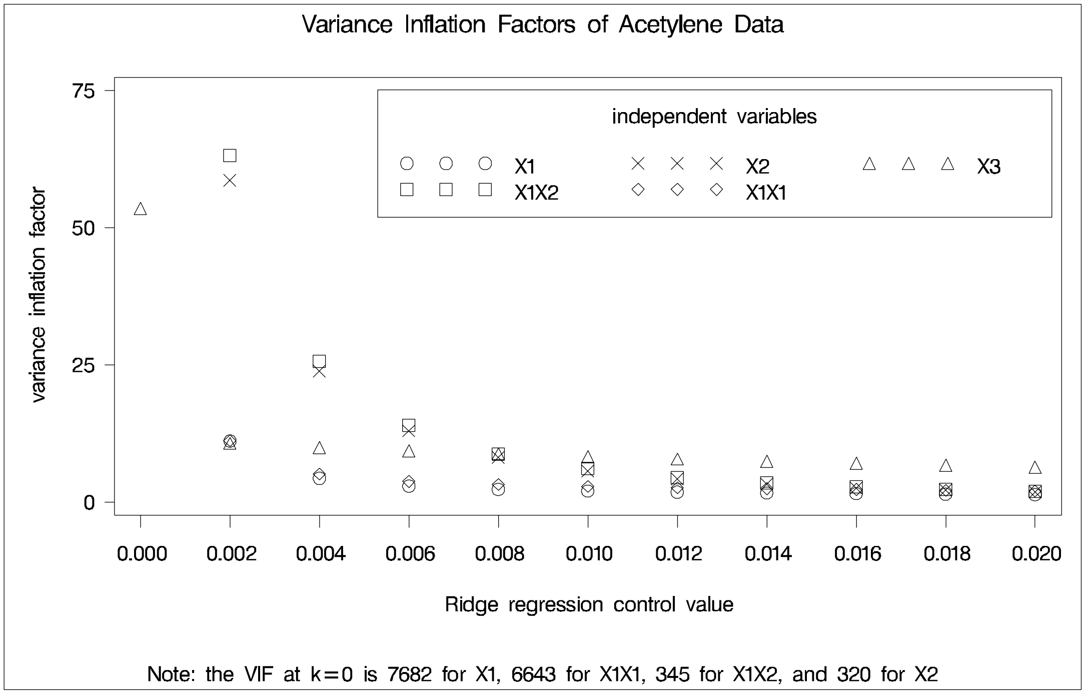
This example uses the REG procedure to create plots from a data set. The variance inflation factors (output by the OUTVIF option in the previous example) are plotted against the ridge regression control values *k*. The following statements create Output 61.11.1:

```
data b (keep=_RIDGE_ x1-x3 x1x2 x1x1);
   set b;
   if _TYPE_='RIDGEVIF';
   label x1='variance inflation factor';
run;

legend3 position=(top right inside) across=3
        cborder=black cframe=white
        label=(color=blue position=(top center)
               'independent variables')
        value=('X1' 'X2' 'X3' 'X1X2' 'X1X1');
symbol1 c=blue   /*v=circle  */;
symbol2 c=yellow /*v=x       */;
symbol3 c=cyan   /*v=triangle*/;
symbol4 c=green  /*v=square  */;
symbol5 c=orange /*v=diamond */;
title 'Variance Inflation Factors of Acetylene Data';

proc reg data=b;
   var _RIDGE_ x3 x1x2 x1x1;
   model x1=x2 / noprint;
   plot (x1 x2 x3 x1x2 x1x1)*_RIDGE_
        / nomodel nostat legend=legend3 overlay
          vaxis = 0 to 75 by 25 cframe=ligr
          haxis = 0 to .02 by .002;
   footnote "Note: the VIF at k=0 is 7682 for X1, "
            "6643 for X1X1, 345 for X1X2, and 320 for X2";
run;
```

The GPLOT procedure can create the same plot with the following statements. The resulting display is not shown in this report.

```
axis1 label=(a=90 r=0 'variance inflation factor')
      order=(0 to 75 by 25) minor=none;
proc gplot data=b;
   plot (x1 x2 x3 x1x2 x1x1)*_RIDGE_
        / legend=legend3 overlay frame
          vaxis = axis1
          haxis = 0 to .02 by .002 hminor=0;
   footnote "Note: the VIF at k=0 is 7682 for X1, "
            "6643 for X1X1, 345 for X1X2, and 320 for X2";
run;
```

Output 61.11.1. Using PROC REG to Plot the VIFs



Example 61.12. ODS Graphics

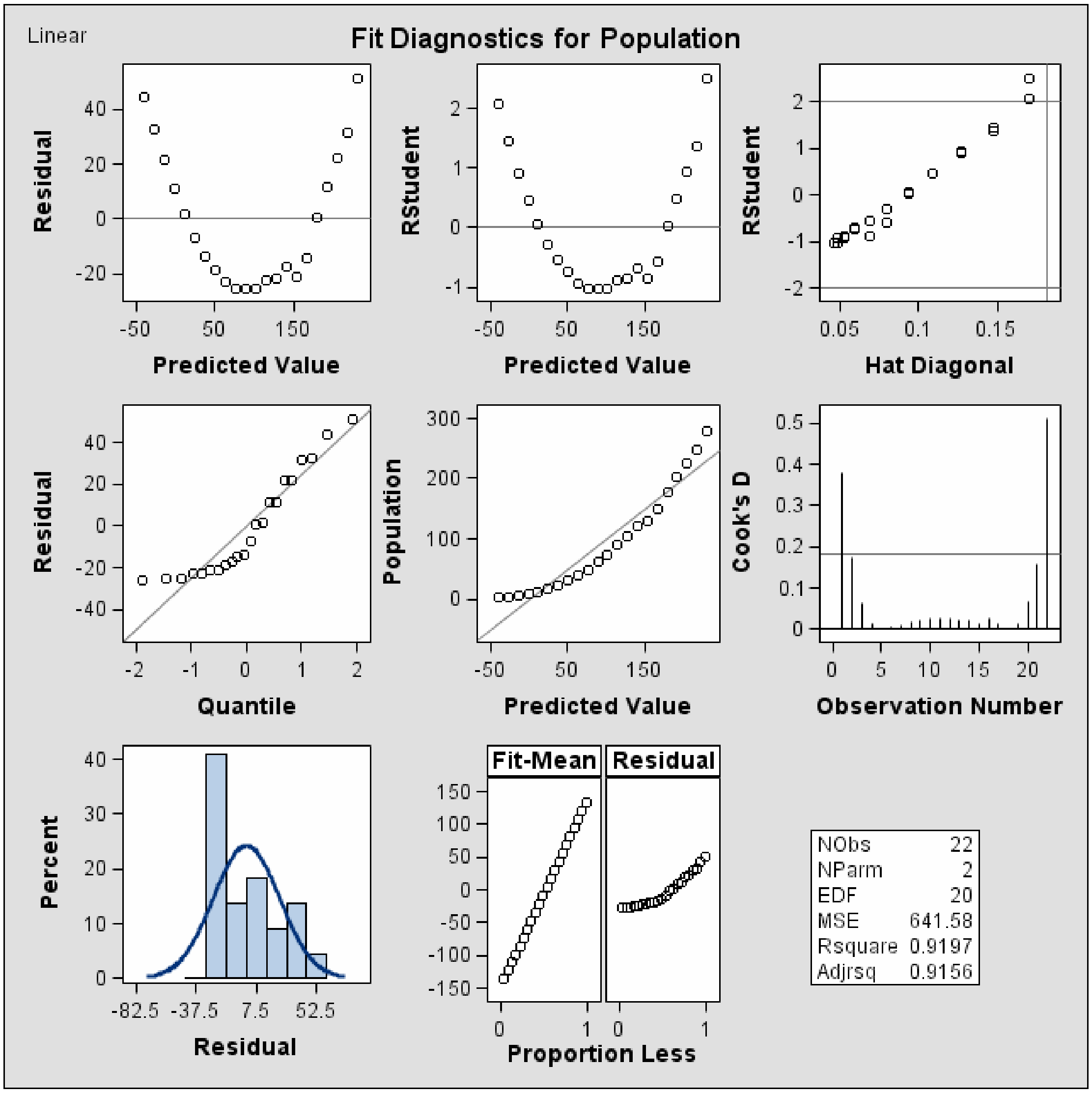
This example highlights the use of ODS for creating statistical graphs with the REG procedure. The USPopulation example is revisited, showing how these graphics can be used to enrich the analysis. Note that the ODS graphics available with PROC REG can be obtained in addition to the graphics you can request with the PLOT statement.

To request the ODS plots you need to specify the experimental ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 15, “Statistical Graphics Using ODS.” For specific information about the graphics available in the REG procedure, see the “ODS Graphics” section on page 3922.

The following statements produce the default plots:

```
ods html;
ods graphics on;

proc reg data=USPopulation;
   Linear: model Population=Year;
   Quadratic:model Population=Year YearSq;
run;quit;

ods graphics off;
ods html close;
```

Output 61.12.1. Fit Diagnostics For the Model Linear in Year (Experimental)



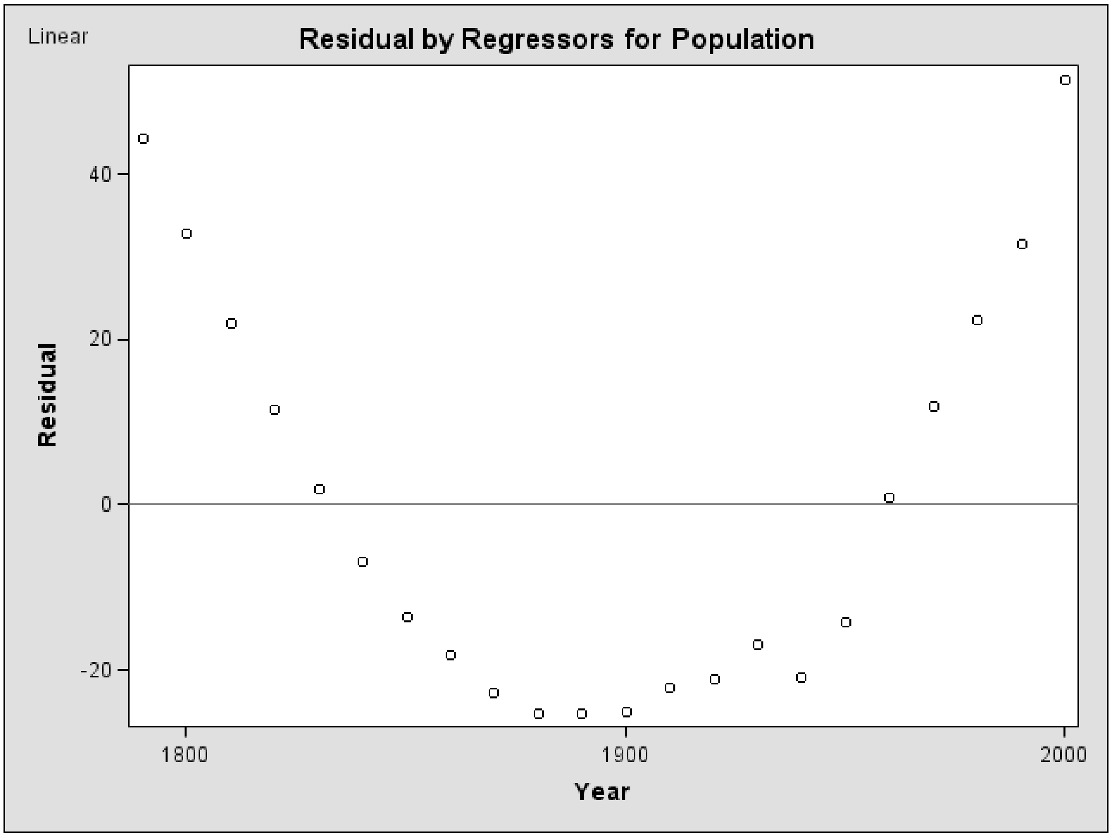
When the experimental ODS graphics are in effect, the fit diagnostic panel Output 61.12.1 is produced by default. These diagnostic plots suggest that while the linear model captures the increasing trend in the data, the model could be significantly improved by adding a term which is quadratic in the variable year:

- The plots of residual and studentized residual versus predicted value show a clear quadratic pattern.
- The plot of studentized residual versus leverage seems to indicate that there are two outlying data points. However, the plot of Cook's D distance versus observation number reveals that these two points are just the data points for the endpoint years 1790 and 2000. These points show up as apparent outliers be-

cause the departure of the linear model from the underlying quadratic behavior in the data shows up most strongly at these endpoints.

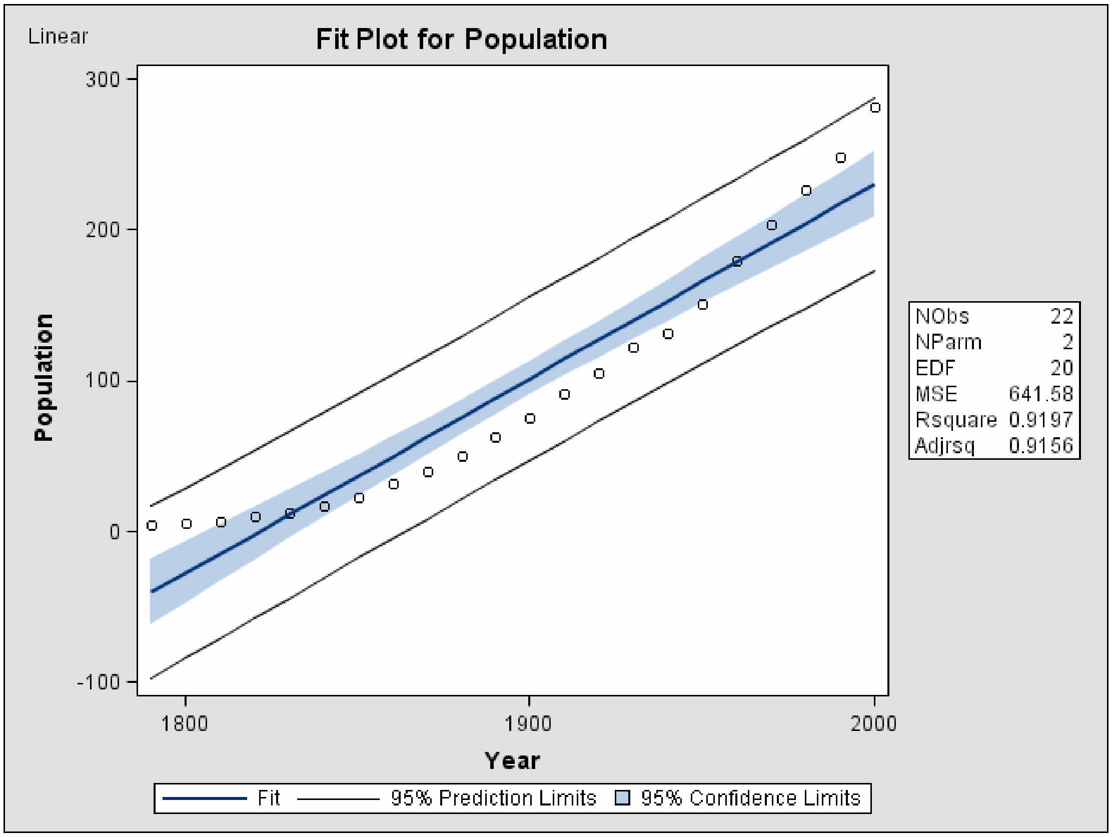
- The normal quantile plot of the residuals and the residual histogram are not consistent with the assumption of gaussian errors. This occurs as the residuals themselves still contain the quadratic behavior that is not captured by the linear model.
- The plot of the dependent variable versus the predicted value exhibits a quadratic form around the 45 degree line which represents a perfect fit.
- The “Residual-Fit” (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals shows that the spread in the residuals is no greater than the spread in the centered fit. For inappropriate models, the spread of the residuals in such a plot is often greater than the spread of the centered fit. In this case, the RF plot shows that the linear model does indeed capture the increasing trend in the data, and hence accounts for much of the variation in the response.

Output 61.12.2. Residual By Year For the Model Linear in Year (Experimental)



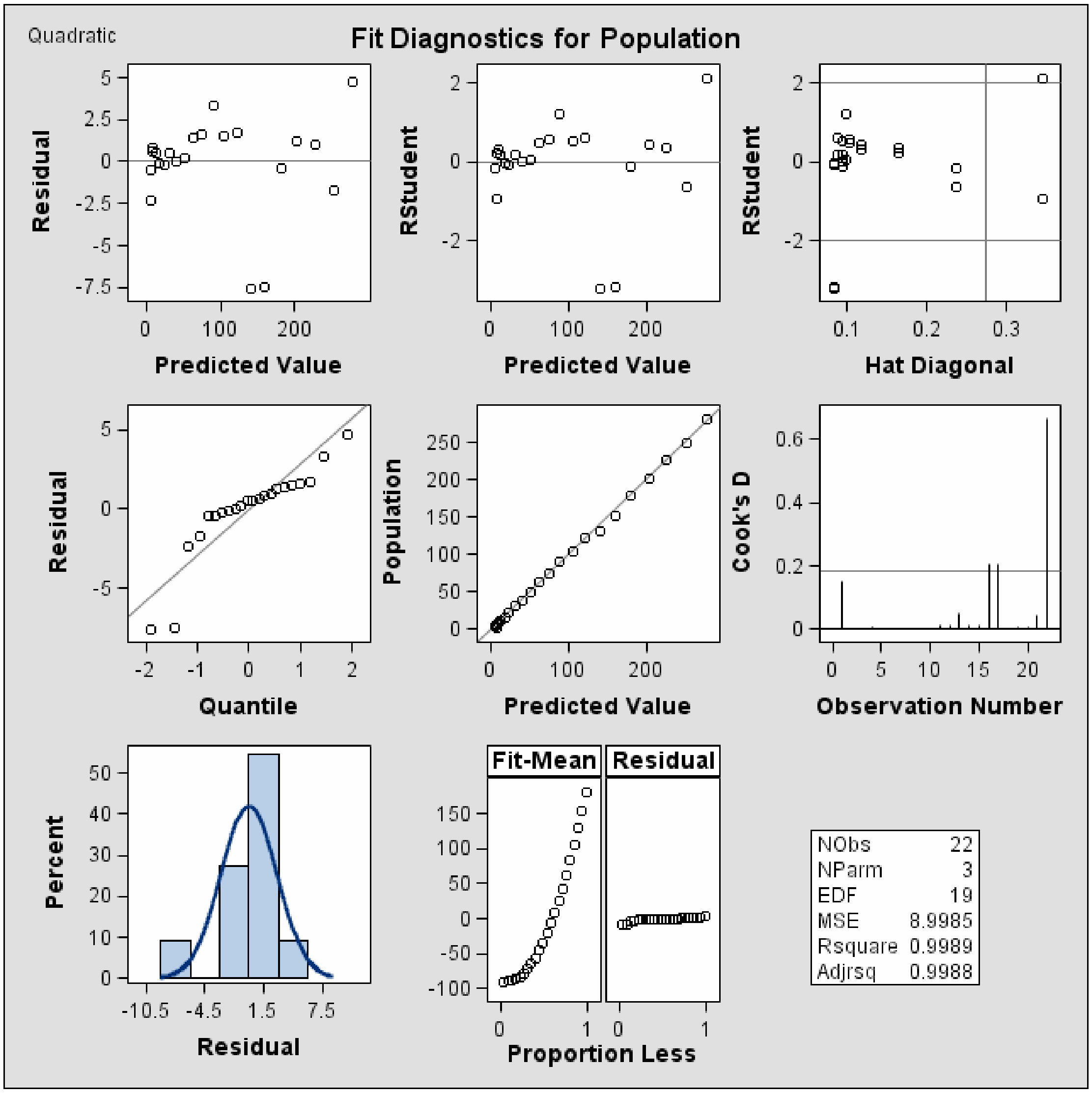
The plot of residual versus the regressor shown in Output 61.12.2 also indicates the need for a quadratic term in the model. There is a strong quadratic trend still remaining in the residuals. Note for models with multiple regressors the plots of residual versus each of the regressors are displayed in panels with up to six plots per panel.

Output 61.12.3. Fit Plot with Confidence Band and Prediction Limits (Experimental)



Output 61.12.3 shows as scatterplot of the data overlayed with the regression line, and 95% confidence band and prediction limits. Note that this plot also indicates that the model fails to capture the quadratic nature of the data. This plot is produced for models containing a single regressor. You can use the ALPHA= option in the model statement to change the significance level of the confidence band and prediction limits.

Output 61.12.4. Fit Diagnostics For the Model Linear in Year (Experimental)



NObs	22
NParm	3
EDF	19
MSE	8.9985
Rsquare	0.9989
Adjrsq	0.9988

By contrast, Output 61.12.4 shows the fit diagnostics panel for the model that includes a quadratic term for year. These diagnostics indicate that this model is significantly more successful than the corresponding linear model:

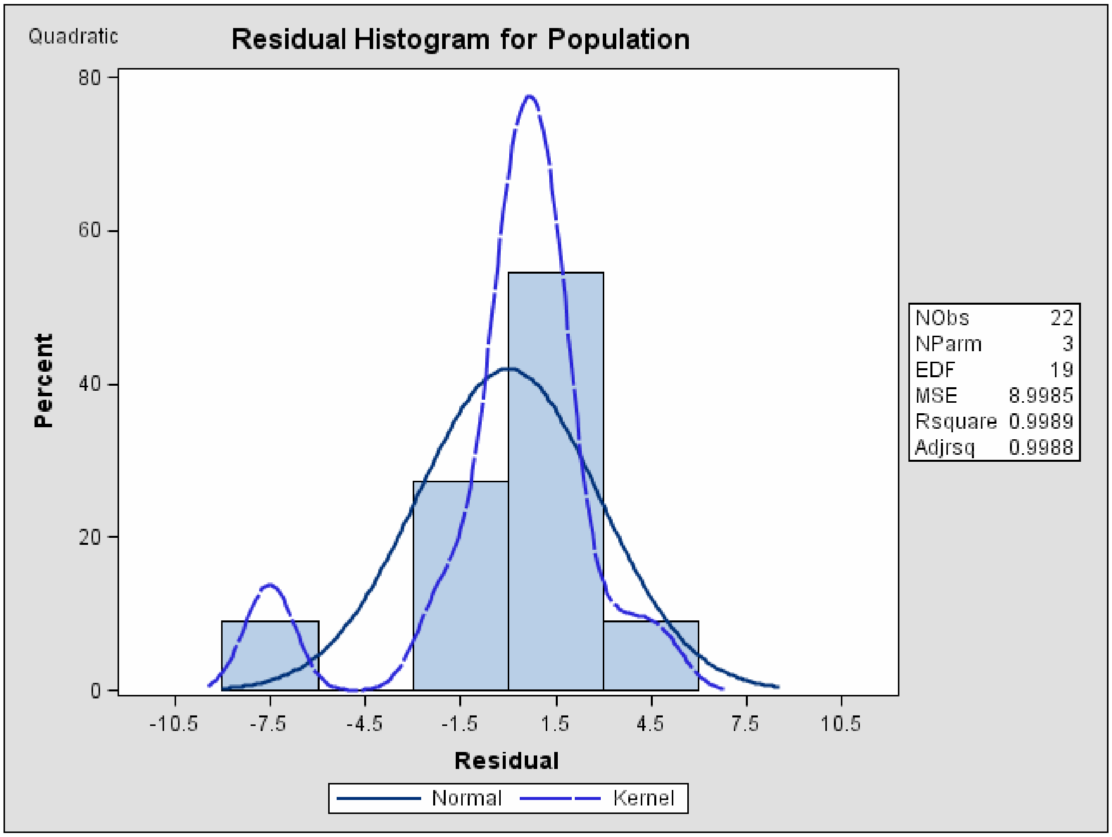
- The plots of residuals and studentized residuals versus predicted values exhibit no obvious patterns.
- The points on the plot of the dependent variable versus the predicted values lie along a 45 degree line, indicating that the model successfully predicts the behavior of the dependent variable.
- The plot of studentized residual versus leverage shows that the years 1790 and 2000 are leverage points with 2000 showing up as an outlier. This is confirmed by the plot of Cook's D distance versus observation number. This suggests that

> while the quadratic model fits the current data well, the model may not be quite so successful over a wider range of data. You might want to investigate whether the population trend over the last couple of decades is growing slightly faster than quadratically.

If you want to obtain the plots in the Diagnostics Panel as individual plots, you can do so by specifying the PLOTS(UNPACKPANELS) option in the PROC REG statement. The following statements provide an example:

```
ods html;
ods graphics on;

proc reg data=USPopulation plots(unpackpanels);
   Quadratic:model Population=Year YearSq;
run;quit;

ods graphics off;
ods html close;
```

Output 61.12.5. Residual Histogram (Experimental)



The residual histogram is shown in Output 61.12.5.

References

Akaike, H. (1969), “Fitting Autoregressive Models for Prediction,” *Annals of the Institute of Statistical Mathematics*, 21, 243–247.

Allen, D.M. (1971), “Mean Square Error of Prediction as a Criterion for Selecting Variables,” *Technometrics*, 13, 469–475.

Allen, D.M. and Cady, F.B. (1982), *Analyzing Experimental Data by Regression*, Belmont, CA: Lifetime Learning Publications.

Amemiya, T. (1976), “Selection of Regressors,” Technical Report No. 225, Stanford, CA: Stanford University.

Belsley, D.A., Kuh, E., and Welsch, R.E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons, Inc.

Berk, K.N. (1977), “Tolerance and Condition in Regression Computations,” *Journal of the American Statistical Association*, 72, 863–866.

Bock, R.D. (1975), *Multivariate Statistical Methods in Behavioral Research*, New York: McGraw-Hill Book Co.

Box, G.E.P. (1966), “The Use and Abuse of Regression,” *Technometrics*, 8, 625–629.

Cleveland, W.S. (1993), *Visualizing Data*, Summit, NJ: Hobart Press.

Cook, R.D. (1977), “Detection of Influential Observations in Linear Regression,” *Technometrics*, 19, 15–18.

Cook, R.D. (1979), “Influential Observations in Linear Regression,” *Journal of the American Statistical Association*, 74, 169–174.

Daniel, C. and Wood, F. (1980), *Fitting Equations to Data*, Revised Edition, New York: John Wiley & Sons, Inc.

Darlington, R.B. (1968), “Multiple Regression in Psychological Research and Practice,” *Psychological Bulletin*, 69, 161–182.

Draper, N. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Durbin, J. and Watson, G.S. (1951), “Testing for Serial Correlation in Least Squares Regression,” *Biometrika*, 37, 409–428.

Freund, R.J. and Littell, R.C. (1986), *SAS System for Regression, 1986 Edition*, Cary, NC: SAS Institute Inc.

Furnival, G.M. and Wilson, R.W. (1974), “Regression by Leaps and Bounds,” *Technometrics*, 16, 499–511.

Gauss, K.F. (1809), *Werke*, 4, 1–93.

Goodnight, J.H. (1979), “A Tutorial on the SWEEP Operator,” *The American Statistician*, 33, 149–158. (Also available as *The Sweep Operator: Its Importance in Statistical Computing*, SAS Technical Report R-106.)

Grunfeld, Y. (1958), “The Determinants of Corporate Investment,” unpublished thesis, Chicago, discussed in Boot, J.C.G. (1960), “Investment Demand: An Empirical Contribution to the Aggregation Problem,” *International Economic Review*, 1, 3–30.

Hocking, R.R. (1976), “The Analysis and Selection of Variables in Linear Regression,” *Biometrics*, 32, 1–50.

Johnston, J. (1972), *Econometric Methods*, New York: McGraw-Hill Book Co.

Judge, G.G., Griffiths, W.E., Hill, R.C., and Lee, T. (1980), *The Theory and Practice of Econometrics*, New York: John Wiley & Sons, Inc.

Judge, G.G., Griffiths, W.E., Hill, R.C., Lutkepohl, H., and Lee, T.C. (1985), “The Theory and Practice of Econometrics,” Second Edition, New York: John Wiley & Sons, Inc.

Kennedy, W.J. and Gentle, J.E. (1980), *Statistical Computing*, New York: Marcel Dekker, Inc.

Lewis, T. and Taylor, L.R. (1967), *Introduction to Experimental Ecology*, New York: Academic Press, Inc.

LaMotte, L.R. (1994), “A Note on the Role of Independence in t Statistics Constructed From Linear Statistics in Regression Models,” *The American Statistician*, 48, 238–240.

Lord, F.M. (1950), “Efficiency of Prediction when a Progression Equation from One Sample is Used in a New Sample,” Research Bulletin No. 50-40, Princeton, NJ: Educational Testing Service.

Mallows, C.L. (1967), “Choosing a Subset Regression,” unpublished report, Bell Telephone Laboratories.

Mallows, C.L. (1973), “Some Comments on C_p,” *Technometrics*, 15, 661–675.

Mardia, K.V., Kent, J.T., and Bibby, J.M. (1979), *Multivariate Analysis*, London: Academic Press, Inc.

Markov, A.A. (1900), *Wahrscheinlichkeitsrechnung*, Tebrer, Leipzig.

Marquardt, D.W. and Snee, R.D. (1975), “Ridge Regression in Practice,” *American Statistician*, 29 (1), 3–20.

Morrison, D.F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill, Inc.

Mosteller, F. and Tukey, J.W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley Publishing Co., Inc.

Neter, J., Wasserman, W., and Kutner, M.H. (1990), *Applied Linear Statistical Models*, Homewood, Illinois: Richard D. Irwin, Inc.

Neter, J., Wasserman, W., and Kutner, M. H. (1990), *Applied Linear Statistical Models*, Third Edition, Homewood, IL: Irwin.

Nicholson, G.E., Jr. (1948), “The Application of a Regression Equation to a New Sample,” unpublished Ph.D. dissertation, University of North Carolina at Chapel Hill.

Pillai, K.C.S. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of the Philippines.

Pindyck, R.S. and Rubinfeld, D.L. (1981), *Econometric Models and Econometric Forecasts*, Second Edition, New York: McGraw-Hill Book Co.

Pringle, R.M. and Raynor, A.A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Company.

Rao, C.R. (1973), *Linear Statistical Inference and Its Applications*, Second Edition, New York: John Wiley & Sons, Inc.

Rawlings, J.O. (1988), *Applied Regression Analysis: A Research Tool*, Belmont, California: Wadsworth, Inc.

Rothman, D. (1968), Letter to the editor, *Technometrics*, 10, 432.

Sall, J.P. (1981), *SAS Regression Applications*, Revised Edition, SAS Technical Report A-102, Cary, NC: SAS Institute Inc.

Sawa, T. (1978), “Information Criteria for Discriminating Among Alternative Regression Models,” *Econometrica*, 46, 1273–1282.

Schwarz, G. (1978), “Estimating the Dimension of a Model,” *Annals of Statistics*, 6, 461–464.

Stein, C. (1960), “Multiple Regression,” in *Contributions to Probability and Statistics*, eds. I. Olkin et al., Stanford, CA: Stanford University Press.

Timm, N.H. (1975), *Multivariate Analysis with Applications in Education and Psychology*, Monterey, CA: Brooks-Cole Publishing Co.

Weisberg, S. (1980), *Applied Linear Regression*, New York: John Wiley & Sons, Inc.

White, H. (1980), “A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity,” *Econometrics*, 48, 817–838.

Chapter 62
The ROBUSTREG Procedure

Chapter Contents

Chapter 62
The ROBUSTREG Procedure

Overview

The main purpose of robust regression is to detect outliers and provide resistant (stable) results in the presence of outliers. In order to achieve this stability, robust regression limits the influence of outliers. Historically, three classes of problems have been addressed with robust regression techniques:

- problems with outliers in the y-direction (response direction)
- problems with multivariate outliers in the x-space (i.e., outliers in the covariate space, which are also referred to as leverage points)
- problems with outliers in both the y-direction and the x-space

Many methods have been developed in response to these problems. However, in statistical applications of outlier detection and robust regression, the methods most commonly used today are Huber M estimation, high breakdown value estimation, and combinations of these two methods. The new ROBUSTREG procedure in this version provides four such methods: M estimation, LTS estimation, S estimation, and MM estimation.

1. M estimation was introduced by Huber (1973), and it is the simplest approach both computationally and theoretically. Although it is not robust with respect to leverage points, it is still used extensively in analyzing data for which it can be assumed that the contamination is mainly in the response direction.
2. Least Trimmed Squares (LTS) estimation is a high breakdown value method introduced by Rousseeuw (1984). The breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness. The performance of this method was improved by the FAST-LTS algorithm of Rousseeuw and Van Driessen (2000).
3. S estimation is a high breakdown value method introduced by Rousseeuw and Yohai (1984). With the same breakdown value, it has a higher statistical efficiency than LTS estimation.
4. MM estimation, introduced by Yohai (1987), combines high breakdown value estimation and M estimation. It has both the high breakdown property and a higher statistical efficiency than S estimation.

Experimental graphics are now available with the ROBUSTREG procedure. For more information, see the section "ODS Graphics" on page 4013.

Getting Started

The following examples demonstrate how you can use the ROBUSTREG procedure to fit a linear regression model and conduct outlier and leverage point diagnostics.

M Estimation

This example shows how you can use the ROBUSTREG procedure to do M estimation, which is a commonly used method for outlier detection and robust regression when contamination is mainly in the response direction.

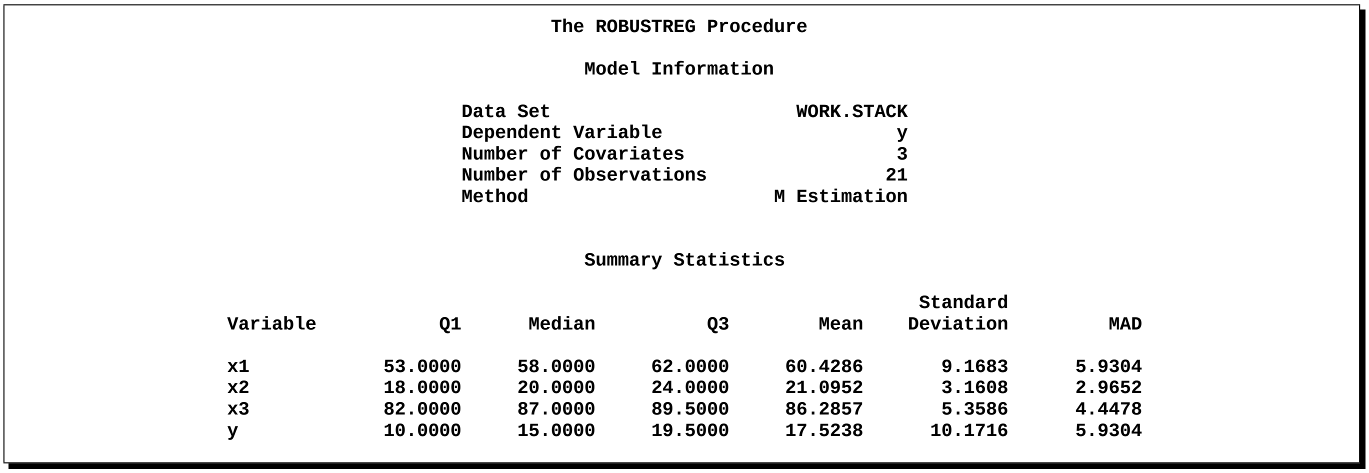
```
data stack;
   input  x1 x2 x3 y;
   datalines;
80  27   89  42
80  27   88  37
75  25   90  37
62  24   87  28
62  22   87  18
62  23   87  18
62  24   93  19
62  24   93  20
58  23   87  15
58  18   80  14
58  18   89  14
58  17   88  13
58  18   82  11
58  19   93  12
50  18   89   8
50  18   86   7
50  19   72   8
50  19   79   8
50  20   80   9
56  20   82  15
70  20   91  15
;
```

The data set stack is the well-known stackloss data set presented by Brownlee (1965). The data describe the operation of a plant for the oxidation of ammonia to nitric acid and consist of 21 four-dimensional observations. The explanatory variables for the response stackloss (y) are the rate of operation (x1), the cooling water inlet temperature (x2), and the acid concentration (x3).

The following ROBUSTREG statements analyze the data:

```
proc robustreg data=stack;
   model y = x1 x2 x3 / diagnostics leverage;
   id    x1;
   test  x3;
run;
```

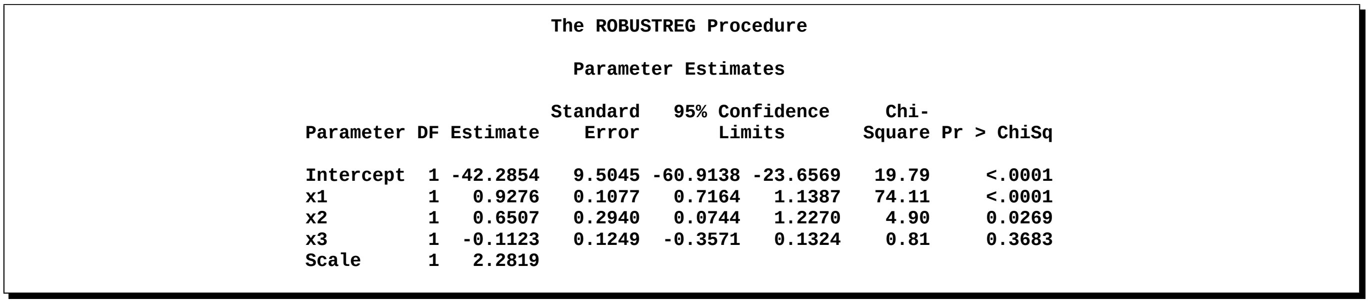
By default, the procedure does M estimation with the bisquare weight function, and it uses the median method for estimating the scale parameter. The MODEL statement specifies the covariate effects. The DIAGNOSTICS option requests a table for outlier diagnostics, and the LEVERAGE option adds leverage point diagnostic results to this table for continuous covariate effects. The ID statement specifies that variable x1 is used to identify each observation in this table. If the ID statement is omitted, the observation number is used to identify the observations. The TEST statement requests a test of significance for the covariate effects specified. The results of this analysis are displayed in the following figures.



```
                                The ROBUSTREG Procedure

                                   Model Information

                      Data Set                          WORK.STACK
                      Dependent Variable                         y
                      Number of Covariates                       3
                      Number of Observations                    21
                      Method                          M Estimation


                                   Summary Statistics

                                                                   Standard
      Variable            Q1      Median          Q3        Mean  Deviation          MAD

      x1             53.0000     58.0000     62.0000     60.4286     9.1683       5.9304
      x2             18.0000     20.0000     24.0000     21.0952     3.1608       2.9652
      x3             82.0000     87.0000     89.5000     86.2857     5.3586       4.4478
      y              10.0000     15.0000     19.5000     17.5238    10.1716       5.9304
```

Figure 62.1. Model Fitting Information and Summary Statistics

Figure 62.1 displays the model fitting information and summary statistics for the response variable and the continuous covariates. The columns labeled Q1, Median, and Q3 provide the lower quantile, median, and upper quantile. The column labeled MAD provides a robust estimate of the univariate scale, which is computed as the corrected median absolute deviation (MAD). The columns labeled Mean and Standard Deviation provide the usual mean and standard deviation. Large difference between the standard deviation and the MAD for a variable indicates some big jumps for this variable. In the stackloss data, the stackloss (reponse variable y) has the biggest difference between the standard deviation and the dispersion.



```
                                The ROBUSTREG Procedure

                                  Parameter Estimates

                                Standard   95% Confidence     Chi-
             Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

             Intercept  1 -42.2854   9.5045 -60.9138 -23.6569   19.79     <.0001
             x1         1   0.9276   0.1077   0.7164   1.1387   74.11     <.0001
             x2         1   0.6507   0.2940   0.0744   1.2270    4.90     0.0269
             x3         1  -0.1123   0.1249  -0.3571   0.1324    0.81     0.3683
             Scale      1   2.2819
```

Figure 62.2. Model Parameter Estimates

Figure 62.2 displays the table of robust parameter estimates, standard errors, and confidence limits. The row labeled Scale provides a point estimate of the scale parameter in the linear regression model, which is obtained by the median method. See the section "M Estimation" on page 3993 for more information about scale estimation methods. For the stackloss data, M estimation yields the fitted linear model:

$$\hat{y} = -42.2845 + 0.9276x1 + 0.6507x2 - 0.1123x3$$

```
                                 The ROBUSTREG Procedure

                                       Diagnostics

                                      Robust                         Standardized
                         Mahalanobis     MCD                              Robust
Obs          x1             Distance  Distance      Leverage            Residual       Outlier

  1     80.000000             2.2536    5.5284         *                  1.0995
  2     80.000000             2.3247    5.6374         *                 -1.1409
  3     75.000000             1.5937    4.1972         *                  1.5604
  4     62.000000             1.2719    1.5887                            3.0381          *
 21     70.000000             2.1768    3.6573         *                 -4.5733          *


                                   Diagnostics Summary

                            Observation
                            Type            Proportion      Cutoff

                            Outlier             0.0952      3.0000
                            Leverage            0.1905      3.0575
```

Figure 62.3. Diagnostics

Figure 62.3 displays outlier and leverage point diagnostics. Standardized robust residuals are computed based on the estimated parameters. Both the Mahalanobis distance and the robust MCD distance are displayed. Outliers and leverage points, identified with asterisks, are defined by the standardized robust residuals and robust MCD distances which exceed the corresponding cutoff values displayed in the diagnostics profile. Observations 4 and 21 are outliers because their standardized robust residuals exceed the cutoff value in absolute value. The procedure detects 4 observations with high leverage, which is contributed mainly by x1, especially for the first three observations. This can be verified by the definition of the robust MCD distance in the section "Robust Multivariate Location and Scale Estimates" on page 4009. Leverage points (points with high leverage) with smaller standardized robust residuals than the cutoff value in absolute value are called good leverage points; otherwise, called bad leverage points. Observations 21 is a bad leverage point.

Two particularly useful plots for revealing outliers and leverage points are a scatter plot of the standardized robust residuals against the robust distances (RDPLOT) and a scatter plot of the robust distances against the classical Mahalanobis distances (DDPLOT).

For the stackloss data, the following statements produce the RDPLOT in Figure 62.4 and the DDPLOT in Figure 62.5. The histogram and the normal quantile-quantile plots for the standardized robust residuals are also created with the

RESHISTOGRAM and RESQQPLOT options in the PROC statement. See Figure 62.6 and Figure 62.7.

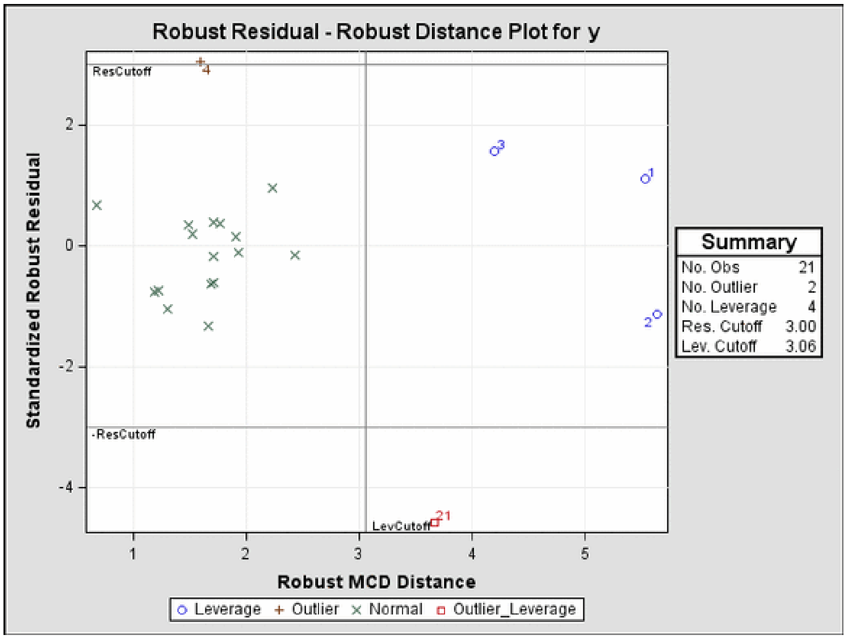


Figure 62.4. RDPLOT for Stackloss Data (Experimental)

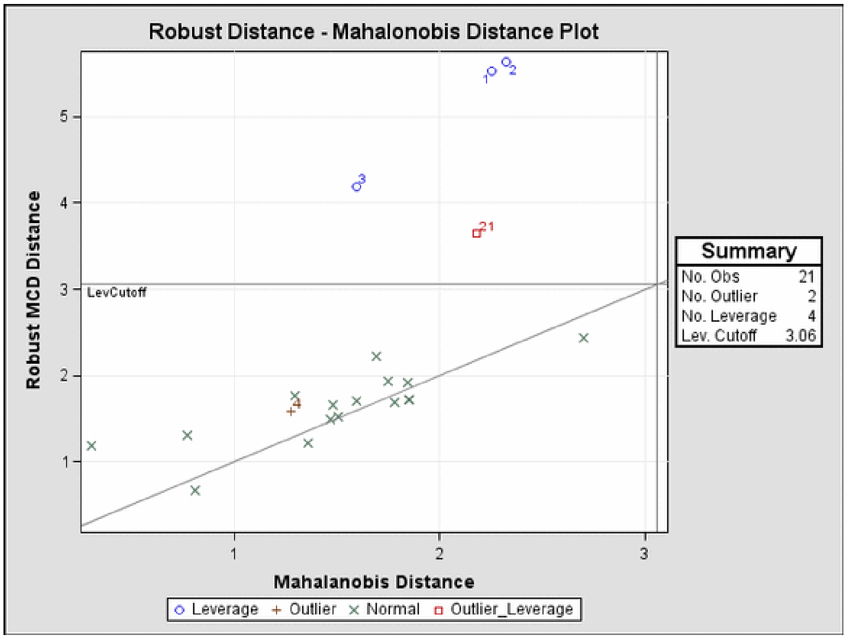


Figure 62.5. DDPLOT for Stackloss Data (Experimental)

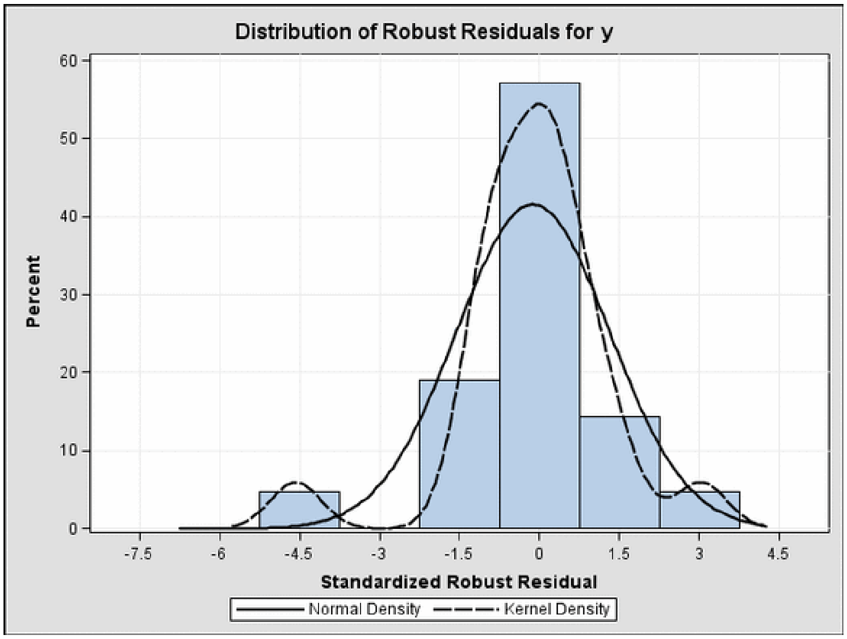


Figure 62.6. Histogram (Experimental)

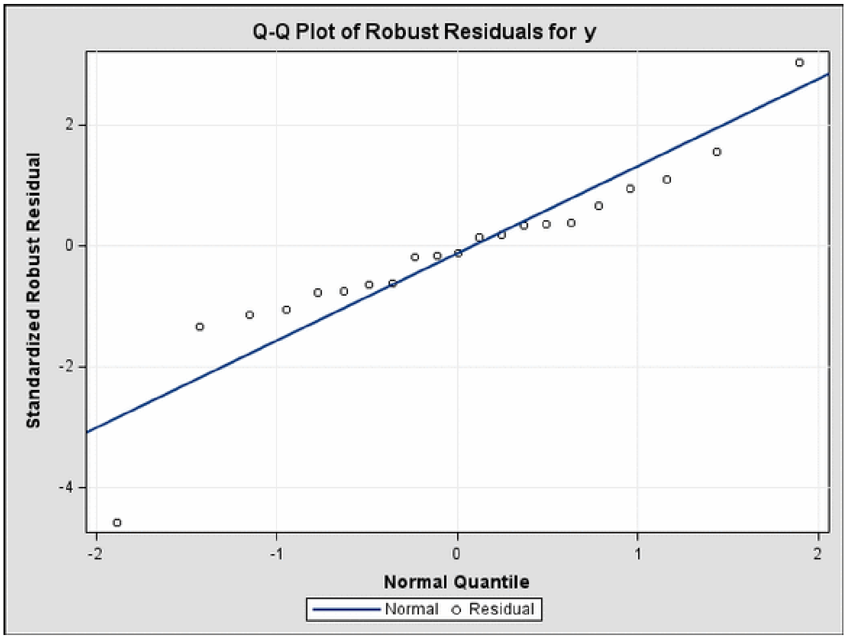


Figure 62.7. Q-Q PLOT (Experimental)

```
ods html;
ods graphics on;

proc robustreg data=stack
        plots=(rdplot ddplot reshistogram resqqplot);
   model y = x1 x2 x3;
run;

ods graphics off;
ods html close;
```

These plots are helpful in identifying outliers, good, and bad high leverage points.

These graphical displays are requested by specifying the experimental ODS GRAPHICS statement and the experimental PLOT | PLOTS= option in the PROC statement. For general information about ODS graphics, see Chapter 15, "Statistical Graphics Using ODS." For specific information about the graphics available in the ROBUSTREG procedure, see the section "ODS Graphics" on page 4013.

```
                         The ROBUSTREG Procedure

                             Goodness-of-Fit

                          Statistic        Value

                          R-Square        0.6659
                          AICR           29.5231
                          BICR           36.3361
                          Deviance      125.7905
```

Figure 62.8. Goodness-of-Fit

Figure 62.8 displays robust versions of goodness-of-fit statistics for the model. You can use the robust information criteria, AICR and BICR, for model selection and comparison. For both AICR and BICR, the lower the value the more describable the model.

```
                         The ROBUSTREG Procedure

                           Robust Linear Tests

                                  Test

                     Test                      Chi-
Test            Statistic    Lambda  DF    Square  Pr > ChiSq

Rho                0.9378    0.7977   1      1.18      0.2782
Rn2                0.8092             1      0.81      0.3683
```

Figure 62.9. Test of Significance

Figure 62.9 displays the test results requested by the TEST statement. The ROBUSTREG procedure conducts two robust linear tests, the ρ-test and the R_n^2-test. See the section "Linear Tests" on page 3998 for information on how the procedure computes test statistics and the correction factor *lambda*. You can conclude that the effect x3 is not significant.

For the bisquare weight function, the default constant c is 4.685 such that the asymptotic efficiency of the M estimates is 95% with the Gaussian distribution. See the section “M Estimation” on page 3993 for details. The smaller the constant c, the lower the asymptotic efficiency but the sharper the M estimate as an outlier detector. For the stackloss data set, you could consider using a sharper outlier detector.

In the following invocation of the ROBUSTREG procedure, a smaller constant, e.g. $c = 3.5$, is used.

```
proc robustreg method=m(wf=bisquare(c=3.5)) data=stack;
   model y = x1 x2 x3 / diagnostics leverage;
   id    x1;
   test  x3;
run;
```

```
                                 The ROBUSTREG Procedure

                                   Parameter Estimates

                                     Standard   95% Confidence     Chi-
                   Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

                   Intercept  1 -37.1076   5.4731 -47.8346 -26.3805   45.97     <.0001
                   x1         1   0.8191   0.0620   0.6975   0.9407  174.28     <.0001
                   x2         1   0.5173   0.1693   0.1855   0.8492    9.33     0.0022
                   x3         1  -0.0728   0.0719  -0.2138   0.0681    1.03     0.3111
                   Scale      1   1.4265
```

Figure 62.10. Model Parameter Estimates

Figure 62.10 displays the table of robust parameter estimates, standard errors, and confidence limits with the constant $c = 3.5$. The refitted linear model is:

$$\hat{y} = -37.1076 + 0.8191x1 + 0.5173x2 - 0.0728x3$$

```
                                 The ROBUSTREG Procedure

                                       Diagnostics

                                         Robust                      Standardized
                        Mahalanobis         MCD                            Robust
   Obs         x1          Distance    Distance      Leverage           Residual       Outlier

     1     80.000000         2.2536      5.5284          *                4.2719          *
     2     80.000000         2.3247      5.6374          *                0.7158
     3     75.000000         1.5937      4.1972          *                4.4142          *
     4     62.000000         1.2719      1.5887                           5.7792          *
    21     70.000000         2.1768      3.6573          *               -6.2727          *


                                   Diagnostics Summary

                            Observation
                            Type           Proportion      Cutoff

                            Outlier            0.1905      3.0000
                            Leverage           0.1905      3.0575
```

Figure 62.11. Diagnostics

Figure 62.11 displays outlier and leverage point diagnostics with the constant $c = 3.5$. Besides observations 4 and 21, observations 1 and 3 are also detected as outliers.

LTS Estimation

If the data are contaminated in the x-space, M estimation does not do well. The following example shows how you can use LTS estimation to deal with this situation.

```
data hbk;
   input index$ x1 x2 x3 y @@;
   datalines;
1   10.1  19.6  28.3   9.7     39  2.1  0.0  1.2  -0.7
2    9.5  20.5  28.9  10.1     40  0.5  2.0  1.2  -0.5
3   10.7  20.2  31.0  10.3     41  3.4  1.6  2.9  -0.1
4    9.9  21.5  31.7   9.5     42  0.3  1.0  2.7  -0.7
5   10.3  21.1  31.1  10.0     43  0.1  3.3  0.9   0.6
6   10.8  20.4  29.2  10.0     44  1.8  0.5  3.2  -0.7
7   10.5  20.9  29.1  10.8     45  1.9  0.1  0.6  -0.5
8    9.9  19.6  28.8  10.3     46  1.8  0.5  3.0  -0.4
9    9.7  20.7  31.0   9.6     47  3.0  0.1  0.8  -0.9
10   9.3  19.7  30.3   9.9     48  3.1  1.6  3.0   0.1
11  11.0  24.0  35.0  -0.2     49  3.1  2.5  1.9   0.9
12  12.0  23.0  37.0  -0.4     50  2.1  2.8  2.9  -0.4
13  12.0  26.0  34.0   0.7     51  2.3  1.5  0.4   0.7
14  11.0  34.0  34.0   0.1     52  3.3  0.6  1.2  -0.5
15   3.4   2.9   2.1  -0.4     53  0.3  0.4  3.3   0.7
16   3.1   2.2   0.3   0.6     54  1.1  3.0  0.3   0.7
17   0.0   1.6   0.2  -0.2     55  0.5  2.4  0.9   0.0
18   2.3   1.6   2.0   0.0     56  1.8  3.2  0.9   0.1
19   0.8   2.9   1.6   0.1     57  1.8  0.7  0.7   0.7
20   3.1   3.4   2.2   0.4     58  2.4  3.4  1.5  -0.1
21   2.6   2.2   1.9   0.9     59  1.6  2.1  3.0  -0.3
22   0.4   3.2   1.9   0.3     60  0.3  1.5  3.3  -0.9
23   2.0   2.3   0.8  -0.8     61  0.4  3.4  3.0  -0.3
24   1.3   2.3   0.5   0.7     62  0.9  0.1  0.3   0.6
25   1.0   0.0   0.4  -0.3     63  1.1  2.7  0.2  -0.3
26   0.9   3.3   2.5  -0.8     64  2.8  3.0  2.9  -0.5
27   3.3   2.5   2.9  -0.7     65  2.0  0.7  2.7   0.6
28   1.8   0.8   2.0   0.3     66  0.2  1.8  0.8  -0.9
29   1.2   0.9   0.8   0.3     67  1.6  2.0  1.2  -0.7
30   1.2   0.7   3.4  -0.3     68  0.1  0.0  1.1   0.6
31   3.1   1.4   1.0   0.0     69  2.0  0.6  0.3   0.2
32   0.5   2.4   0.3  -0.4     70  1.0  2.2  2.9   0.7
33   1.5   3.1   1.5  -0.6     71  2.2  2.5  2.3   0.2
34   0.4   0.0   0.7  -0.7     72  0.6  2.0  1.5  -0.2
35   3.1   2.4   3.0   0.3     73  0.3  1.7  2.2   0.4
36   1.1   2.2   2.7  -1.0     74  0.0  2.2  1.6  -0.9
37   0.1   3.0   2.6  -0.6     75  0.3  0.4  2.6   0.2
38   1.5   1.2   0.2   0.9
;
```

The data set hbk is an artificial data set generated by Hawkins, Bradu, and Kass (1984). Both ordinary least squares (OLS) estimation and M estimation (not shown here) suggest that observations 11 to 14 are serious outliers. However, these four observations were generated from the underlying model, whereas observations 1 to 10 were contaminated. The reason that OLS estimation and M estimation do not pick up the contaminated observations is that they cannot distinguish good leverage points (observations 11 to 14) from bad leverage points (observations 1 to 10). In such cases, the LTS method identifies the true outliers.

The following statements invoke the ROBUSTREG procedure with the LTS estimation method.

```
proc robustreg data=hbk fwls method=lts;
   model y = x1 x2 x3 / diagnostics leverage;
   id index;
run;
```

```
                                 The ROBUSTREG Procedure

                                    Model Information

                         Data Set                          WORK.HBK
                         Dependent Variable                       y
                         Number of Covariates                     3
                         Number of Observations                  75
                         Method                      LTS Estimation


                                   Summary Statistics

                                                                      Standard
        Variable              Q1      Median          Q3        Mean     Deviation         MAD

        x1                0.8000      1.8000      3.1000      3.2067        3.6526      1.9274
        x2                1.0000      2.2000      3.3000      5.5973        8.2391      1.6309
        x3                0.9000      2.1000      3.0000      7.2307       11.7403      1.7791
        y                -0.5000      0.1000      0.7000      1.2787        3.4928      0.8896
```

Figure 62.12. Model Fitting Information and Summary Statistics

Figure 62.12 displays the model fitting information and summary statistics for the response variable and independent covariates.

```
                                 The ROBUSTREG Procedure

                                       LTS Profile

                     Total Number of Observations                75
                     Number of Squares Minimized                 57
                     Number of Coefficients                       4
                     Highest Possible Breakdown Value        0.2533
```

Figure 62.13. LTS Profile

Figure 62.13 displays information about the LTS fit, which includes the breakdown value of the LTS estimate. Roughly speaking, the breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness. In this example the LTS estimate minimizes the sum of 57 smallest squares of residuals. It can still pick up the right model if the remaining 18 observations are contaminated. This corresponds to the breakdown value around 0.25, which is set as the default.

```
                              The ROBUSTREG Procedure

                              LTS Parameter Estimates

                          Parameter         DF    Estimate

                          Intercept          1     -0.3431
                          x1                 1      0.0901
                          x2                 1      0.0703
                          x3                 1     -0.0731
                          Scale (sLTS)       0      0.7451
                          Scale (Wscale)     0      0.5749
```

Figure 62.14. LTS Parameter Estimates

Figure 62.14 displays parameter estimates for covariates and scale. Two robust estimates of the scale parameter are displayed. See the section "Final Weighted Scale Estimator" on page 4002 for how these estimates are computed. The weighted scale estimate (Wscale) is a more efficient estimate of the scale parameter.

```
                              The ROBUSTREG Procedure

                                    Diagnostics

                                      Robust                     Standardized
                        Mahalanobis      MCD                           Robust
  Obs      index           Distance  Distance     Leverage           Residual     Outlier

    1        1               1.9168   29.4424         *               17.0868        *
    3        2               1.8558   30.2054         *               17.8428        *
    5        3               2.3137   31.8909         *               18.3063        *
    7        4               2.2297   32.8621         *               16.9702        *
    9        5               2.1001   32.2778         *               17.7498        *
   11        6               2.1462   30.5892         *               17.5155        *
   13        7               2.0105   30.6807         *               18.8801        *
   15        8               1.9193   29.7994         *               18.2253        *
   17        9               2.2212   31.9537         *               17.1843        *
   19       10               2.3335   30.9429         *               17.8021        *
   21       11               2.4465   36.6384         *                0.0406
   23       12               3.1083   37.9552         *               -0.0874
   25       13               2.6624   36.9175         *                1.0776
   27       14               6.3816   41.0914         *               -0.7875


                                Diagnostics Summary

                        Observation
                        Type             Proportion      Cutoff

                        Outlier              0.1333      3.0000
                        Leverage             0.1867      3.0575
```

Figure 62.15. Diagnostics

Figure 62.15 displays outlier and leverage point diagnostics. The ID variable index is used to identify the observations. The first ten observations are identified as outliers and observations 11 to 14 are identified as good leverage points.

```
                              The ROBUSTREG Procedure

                  Parameter Estimates for Final Weighted Least Squares Fit

                                     Standard   95% Confidence     Chi-
Parameter       DF Estimate     Error       Limits       Square Pr > ChiSq

Intercept        1  -0.1805    0.1044  -0.3852   0.0242    2.99      0.0840
x1               1   0.0814    0.0667  -0.0493   0.2120    1.49      0.2222
x2               1   0.0399    0.0405  -0.0394   0.1192    0.97      0.3242
x3               1  -0.0517    0.0354  -0.1210   0.0177    2.13      0.1441
Scale            0   0.5572
```

Figure 62.16. Final Weighted LS Estimates

Figure 62.16 displays the final weighted least squares estimates. These estimates are least squares estimates computed after deleting the detected outliers.

Syntax

PROC ROBUSTREG < *options* > **;**
BY *variables* **;**
CLASS *variables* **;**
ID *variables* **;**
MODEL *response* = <*effects*> < / *options* > **;**
OUTPUT < **OUT=** *SAS-data-set* > < *options* > **;**
PERFORMANCE < *options* > **;**
TEST *'label' effects* **;**
WEIGHT *variable* **;**

The PROC ROBUSTREG statement invokes the procedure. The METHOD= option in the PROC ROBUSTREG statement selects one of the four estimation methods, M, LTS, S, and MM. By default, Huber M estimation is used. The MODEL statement is required and specifies the variables used in the regression. Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure. The CLASS statement specifies which explanatory variables are treated as categorical. These variables are allowed in the MODEL statement only for M estimation, and not for other estimation methods. The ID statement names variables to identify observations in the outlier diagnostics tables. The WEIGHT statement identifies a variable in the input data set whose values are used to weight the observations. The OUTPUT statement creates an output data set containing final weights, predicted values, and residuals. The TEST statement requests robust linear tests for the model parameters. The PERFORMANCE statement tunes the performance of the procedure by using single or multiple processors available on the hardware. In one invocation of PROC ROBUSTREG, multiple OUTPUT and TEST statements are allowed.

PROC ROBUSTREG Statement

PROC ROBUSTREG < *options* > **;**

The PROC ROBUSTREG statement invokes the procedure. You can specify the following options in the PROC ROBUSTREG statement.

COVOUT

saves the estimated covariance matrix in the OUTEST= data set for M estimation and MM estimation.

DATA=*SAS-data-set*

specifies the input SAS data set used by PROC ROBUSTREG. By default, the most recently created SAS data set is used.

FWLS

requests that final weighted least squares estimators be computed.

INEST= *SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= Data Set" on page 4011 for a detailed description of the contents of the INEST= data set.

ITPRINT

displays the iteration history for the iteratively reweighted least squares algorithm used by M and MM estimation. You can also use this option in the MODEL statement.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. The following table explains how PROC ROBUSTREG interprets values of the ORDER= option.

Table 62.1. Options for Order

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, refer to the chapter titled "The SORT Procedure" in the *SAS Procedures Guide*.

OUTEST=*SAS-data-set*

specifies an output SAS data set containing the parameter estimates, and, if the COVOUT option is specified, the estimated covariance matrix. See the section "OUTEST= Data Set" on page 4011 for a detailed description of the contents of the OUTEST= data set.

SEED=*number*

specifies the seed for the random number generator used to randomly select the subgroups and subsets for LTS and S estimation. By default or you specify zero, the ROBUSTREG procedure generates a seed between one and one billion.

METHOD= *method type* **< (** *options* **) >**

specifies the estimation method and $options$ specify some additional options for the estimation method. PROC ROBUSTREG provides four estimation methods: M estimation, LTS estimation, S estimation, and MM estimation. The default method is M estimation.

Since the LTS and S methods use subsampling algorithms, it is not suitable to apply these methods to an analysis with continuous independent variables which have only a few nonzero values or a few nonzero values within one BY group.

Options with METHOD=M

With METHOD=M, you can specify the following additional $options$:

ASYMPCOV=*H1* | *H2* | *H3*

specifies the type of asymptotic covariance computed for the M estimate. The three types are described in the section "Asymptotic Covariance and Confidence Intervals" on page 3997. By default, ASYMPCOV= H1.

CONVERGENCE=*criterion* $<$**(EPS=***value***)**$>$

specifies a convergence criterion for the M estimate.

Table 62.2. Options to Specify Convergence Criteria

Type	Option
residual	CONVERGENCE= RESID
weight	CONVERGENCE= WEIGHT
coefficient	CONVERGENCE= COEF

By default, CONVERGENCE = COEF. You can specify the precision of the convergence can be specified with the EPS= option. By default, EPS=1.E−8.

MAXITER=*n*

sets the maximum number of iterations during the parameter estimation. By default, MAXITER=1000.

SCALE=*scale type* | *value*

specifies the scale parameter or a method for estimating the scale parameter.

Table 62.3. Options to Specify Scale

Scale	Option	Default d
Median estimate	SCALE=MED	
Tukey estimate	SCALE=TUKEY<(D=d)>	2.5
Huber estimate	SCALE=HUBER<(D=d)>	2.5
Fixed constant	SCALE=*value*	

By default, SCALE = MED.

WF | WEIGHTFUNCTION=*function type*

specifies the weight function used for the M estimate. The ROBUSTREG procedure provides ten weight functions, which are listed in the following table. You can specify the parameters in these functions with the A=, B=, and C= options. These functions are described in the section "M Estimation" on page 3993. The default weight function is bisquare.

Table 62.4. Options to Specify Weight Functions

Weight Function	Option	Default a, b, c
andrews	WF = ANDREWS<(C=c)>	1.339
bisquare	WF = BISQUARE<(C=c)>	4.685
cauchy	WF = CAUCHY<(C=c)>	2.385
fair	WF = FAIR<(C=c)>	1.4
hampel	WF = HAMPEL<(<A=a> <B=b> <C=c>)>	2, 4, 8
huber	WF = HUBER<(C=c)>	1.345
logistic	WF = LOGISTIC<(C=c)>	1.205
median	WF = MEDIAN<(C=c)>	0.01
talworth	WF = TALWORTH<(C=c)>	2.795
welsch	WF = WELSCH<(C=c)>	2.985

Options with METHOD=LTS

With METHOD=LTS, you can specify the following additional $options$:

CSTEP=*n*

specifies the number of C-steps for the LTS estimate. See the section "LTS Estimate" on page 4000 for how the default value is determined.

IADJUST=ALL | NONE

requests (IADJUST=ALL) or suppresses (IADJUST=NONE) the intercept adjustment for all estimates in the LTS-algorithm. By default, the intercept adjustment is used for data sets with less than 10000 observations. See the section "Algorithm" on page 4001 for details.

H=*n*

specifies the quantile for the LTS estimate. See the section "LTS Estimate" on page 4000 for how the default value is determined.

NBEST=*n*

specifies the number of best solutions kept for each subgroup during the computation of the LTS estimate. The default number is 10, which is the maximum number allowed.

NREP=*n*

specifies the number of repeats of least squares fit in subgroups during the computation of the LTS estimate See the section "LTS Estimate" on page 4000 for how the default number is determined.

SUBANALYSIS

requests a display of the subgrouping information and parameter estimates within subgroups. This option may generate the following ODS tables:

Table 62.5. ODS Tables Available with SUBANALYSIS

ODS Table Name	Description
BestEstimates	Best final estimates for LTS
BestSubEstimates	Best estimates for each subgroup
CStep	C-Step information for LTS
Groups	Grouping information for LTS

Some of these tables are data dependent.

SUBGROUPSIZE=*n*

specifies the data set size of the subgroups in the computation of the LTS estimate. The default number is 300.

Options with METHOD=S

With METHOD=S, you can specify the following additional $options$:

ASYMPCOV=*H1* | *H2* | *H3* | *H4*

specifies the type of asymptotic covariance computed for the S estimate. The four types are described in the section "Asymptotic Covariance and Confidence Intervals" on page 4005. By default, ASYMPCOV= H4.

CHIF= TUKEY | YOHAI

specifies the χ function for the S estimate. PROC ROBUSTREG provides two χ functions, Tukey's BISQUARE function and Yohai's OPTIMAL function, which you can request with CHIF=TUKEY and CHIF=YOHAI, respectively. The default is Tukey's bisquare function.

EFF=*value*

specifies the efficiency for the S estimate. The parameter k_0 in the χ function is determined by this efficiency. The default efficiency is determined such that the consistent S estimate has the breakdown value of 25%.

MAXITER=*n*

sets the maximum number of iterations for computing the scale parameter of the S estimate. By default, MAXITER=1000.

NREP=*n*

specifies the number of repeats of subsampling in the computation of the S estimate. See the section "Algorithm" on page 4004 for how the default number of repeats is determined.

NOREFINE

suppresses the refinement for the S estimate. See the section "Algorithm" on page 4004 for details.

SUBSETSIZE=*n*

specifies the size of the subset for the S estimate. See the section "Algorithm" on page 4004 for how its default value is determined.

TOLERANCE=*value*

specifies the tolerance for the S estimate of the scale. The default value is .001.

Options with METHOD=MM

With METHOD=MM, you can specify the following additional $options$:

ASYMPCOV=*H1* | *H2* | *H3* | *H4*

specifies the type of asymptotic covariance computed for the MM estimate. The four types are described in the "Details" section. By default, ASYMPCOV= H4.

BIASTEST<(ALPHA= *number***)>**

requests the bias test for the final MM estimate. See the section "Bias Test" on page 4008 for details about this test.

CHIF= TUKEY | YOHAI

selects the χ function for the MM estimate. PROC ROBUSTREG provides two χ functions: Tukey's BISQUARE function and Yohai's OPTIMAL function, which you can request with CHIF=TUKEY and CHIF=YOHAI, respectively. The default is Tukey's bisquare function. This χ function is also used by the initial S estimate if you specify the INITEST=S option.

CONVERGENCE=*criterion* <**(EPS=***number***)**>

specifies a convergence criterion for the MM estimate.

Table 62.6. Options to Specify Convergence Criteria

Type	**Option**
residual	CONVERGENCE= RESID
weight	CONVERGENCE= WEIGHT
coefficient	CONVERGENCE= COEF

By default, CONVERGENCE = COEF. You can specify the precision of the convergence with the EPS= option. By default, EPS=1.E−8.

EFF=*value*

specifies the efficiency for the MM estimate. The parameter k_1 in the χ function is determined by this efficiency. The default efficiency is set to 85%, which corresponds to $k_1 = 3.440$ for CHIF=TUKEY or $k_1 = 0.868$ for CHIF=YOHAI.

INITH=*n*

specifies the integer h for the initial LTS estimator used by the MM estimator. See the section "Algorithm" on page 4007 for how to specify h and how the default is determined.

INITEST= LTS | S

specifies the initial estimator for the MM estimator. By default, the LTS estimator is used as the initial estimator for the MM estimator.

K0=*number*

specifies the parameter k_0 in the χ function for the MM estimate. For CHIF=TUKEY, the default is $k_0 = 2.9366$. For CHIF=YOHAI, the default is $k_0 = 0.7405$. These default values correspond to the 25% breakdown value of the MM estimator.

MAXITER=*n*

sets the maximum number of iterations during the parameter estimation. By default, MAXITER=1000.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ROBUSTREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ROBUSTREG procedure. The NOTSORTED option does not mean that the data are unsorted, but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

Explanatory variables that are classification variables rather than quantitative numeric variables must be listed in the CLASS statement. For each explanatory variable listed in the CLASS statement, indicator variables are generated for the levels assumed by the CLASS variable. If the CLASS statement is used, it must appear before the MODEL statement.

ID Statement

ID *variables* **;**

When the diagnostics table is requested with the DIAGNOSTICS option in the MODEL statement, the variables listed in the ID statement are displayed besides the observation number. These variables can be used to identify each observation. If the ID statement is omitted, the observation number is used to identify the observations.

MODEL Statement

<*label:*> **MODEL** *response* = <*effects*> < / *options* > **;**

Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure. Class variables are not allowed in the MODEL statement when you specify MM estimation or LTS estimation using the METHOD= option in the PROC statement.

The optional *label* is used to label output from the matching MODEL statement.

Options

You can specify the following options for the model fit.

ALPHA=*value*
specifies the significance level for the confidence intervals for regression parameters. The value must be between 0 and 1. By default, ALPHA = 0.05.

CORRB
produces the estimated correlation matrix of the parameter estimates.

COVB
produces the estimated covariance matrix of the parameter estimates.

CUTOFF=*value*
specifies the multiplier of the cutoff value for outlier detection. By default, CUTOFF = 3.

DIAGNOSTICS<(ALL)>
requests the outlier diagnostics. By default, only observations identified as outliers or leverage points are displayed. To request that all observations be displayed, specify the ALL option.

ITPRINT

displays the iteration history for the iteratively reweighted least squares algorithm used by M and MM estimation. You can also use this option in the PROC statement.

LEVERAGE<(CUTOFF=*value* **| CUTOFFALPHA=***value* **| QUANTILE=***n***)>**

requests an analysis of leverage points for the continuous covariates. The results are added to the diagnostics table, which you can request with the DIAGNOSTICS option in the MODEL statement. You can specify the cutoff value for leverage point detection with the CUTOFF= option. The default cutoff value is $\sqrt{\chi^2_{p;1-\alpha}}$, where α can be specified with the CUTOFFALPHA= option. By default, $\alpha = .025$. You can use the QUANTILE= option to specify the quantile to be minimized for the MCD algorithm used for the leverage point analysis. By default, QUANTILE=$[(3n + p + 1)/4]$, where n is the number of observations and p is the number of independent variables. The LEVERAGE option is ignored if the model includes class variables as covariates.

Since the MCD algorithm uses subsampling, it is not suitable to apply the leverage point analysis to continuous variables which have only a few nonzero values or a few nonzero values within one BY group.

NOGOODFIT

suppresses the computation of goodness-of-fit statistics.

NOINT

specifies no-intercept regression.

SINGULAR=*value*

specifies the tolerance for testing singularity of the information matrix and the crossproducts matrix for the initial least-squares estimates. Roughly, the test requires that a pivot be at least this value times the original diagonal value. By default, SINGULAR = 1.E−12.

OUTPUT Statement

OUTPUT <**OUT=***SAS-data-set*> *keyword***=***name* <*. . .keyword***=***name*> **;**

The OUTPUT statement creates an output SAS data set containing statistics calculated after fitting the model. At least one specification of the form *keyword=name* is required.

All variables in the original data set are included in the new data set, along with the variables created with *keyword* options in the OUTPUT statement. These new variables contain fitted values and estimated quantiles. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information on permanent SAS data sets).

The following specifications can appear in the OUTPUT statement:

OUT=*SAS-data-set* specifies the new data set. By default, the procedure uses the DATAn convention to name the new data set.

keyword=name specifies the statistics to include in the output data set and gives names to the new variables. Specify a keyword for each desired statistic (see the following list), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

LEVERAGE specifies a variable to indicate leverage points. To include this variable in the OUTPUT data set, you must specify the LEVERAGE option in the PROC statement. See the section "Leverage Point and Outlier Detection" on page 4010 for how to define LEVERAGE.

OUTLIER specifies a variable to indicate outliers. See the section "Leverage Point and Outlier Detection" on page 4010 for how to define OUTLIER.

PREDICTED | P specifies a variable to contain the estimated response.

RESIDUAL | R specifies a variable to contain the residuals

$$y_i - \mathbf{x}_i^T \mathbf{b}$$

SRESIDUAL | SR specifies a variable to contain the standardized residuals

$$\frac{y_i - \mathbf{x}_i^T \mathbf{b}}{\hat{\sigma}}$$

STDP specifies a variable to contain the estimates of the standard errors of the estimated response.

WEIGHT specifies a variable to contain the computed final weights.

PERFORMANCE Statement

You use the PERFORMANCE statement to specify options that tune the performance of PROC ROBUSTREG. By default these options are chosen to maximize performance. See Chen (2002) for some empirical results.

PERFORMANCE < *options* > **;**

The following option is available:

CPUCOUNT=*n*

specifies the number of threads to use in the computation of LTS or S estimation (initial LTS or S estimation for MM estimation). By default this will be equal to the number of processors on the hardware.

TEST Statement

<label:> **TEST** *effects* **;**

With M estimation and MM estimation, the TEST statement provides a means for obtaining a test for the canonical linear hypothesis concerning the model parameters:

$$\theta_j = 0, \;\; j = q+1, ..., p$$

where p is the total number of parameters in the model, and q is the number of parameters for testing of significance.

PROC ROBUSTREG provides two kinds of robust tests: the ρ-test and the R_n^2-test. They are described in the “Details” section. No test is available for LTS and S estimation.

The optional *label* is used to label output from the corresponding TEST statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a weight variable in the input data set.

If you want to use fixed weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with nonpositive or missing values for the weight variable do not contribute to the fit of the model.

Details

This section describes the statistical and computational aspects of the ROBUSTREG procedure. The following notation is used throughout this section.

Let $X = (x_{ij})$ denote an $n \times p$ matrix, $y = (y_1, ..., y_n)^T$ a given n-vector of responses, and $\theta = (\theta_1, ..., \theta_p)^T$ an unknown p-vector of parameters or coefficients whose components are to be estimated. The matrix X is called the design matrix. Consider the usual linear model

$$y = X\theta + e$$

where $e = (e_1, ..., e_n)^T$ is an n-vector of unknown errors. It is assumed that (for a given X) the components e_i of e are independent and identically distributed according to a distribution $L(\cdot/\sigma)$, where σ is a scale parameter (usually unknown). The vector of residuals for a given value of $\hat{\theta}$ is denoted by $r = (r_1, ..., r_n)^T$ and the ith row of the matrix X is denoted by x_i^T.

M Estimation

M estimation in the context of regression was first introduced by Huber (1973) as an result of making the least squares approach robust. Although M estimators are not robust with respect to leverage points, they are popular in applications where leverage points are not an issue.

Instead of minimizing a sum of squares of the residuals, a Huber-type M estimator $\hat{\theta}_M$ of θ minimizes a sum of less rapidly increasing functions of the residuals:

$$Q(\theta) = \sum_{i=1}^{n} \rho(\frac{r_i}{\sigma})$$

where $r = y - X\theta$. For the ordinary least squares estimation, ρ is the quadratic function.

If σ is known, by taking derivatives with respect to θ, $\hat{\theta}_M$ is also a solution of the system of p equations:

$$\sum_{i=1}^{n} \psi(\frac{r_i}{\sigma}) x_{ij} = 0, \; j = 1, ..., p$$

where $\psi = \rho'$. If ρ is convex, $\hat{\theta}_M$ is the unique solution.

The ROBUSTREG procedure solves this system by using iteratively reweighted least squares (IRLS). The weight function $w(x)$ is defined as

$$w(x) = \frac{\psi(x)}{x}$$

The ROBUSTREG procedure provides ten kinds of weight functions (corresponding to ten ρ-functions) through the WEIGHTFUNCTION= option in the MODEL statement. See the section "Weight Functions" on page 3995 for a complete discussion. You can specify the scale parameter σ with the SCALE= option in the PROC statement.

If σ is unknown, both θ and σ are estimated by minimizing the function

$$Q(\theta, \sigma) = \sum_{i=1}^{n} [\rho(\frac{r_i}{\sigma}) + a]\sigma, \; a > 0$$

The algorithm proceeds by alternately improving $\hat{\theta}$ in a location step and $\hat{\sigma}$ in a scale step.

For the scale step, three methods are available to estimate σ, which you can select with the SCALE= option.

1. (SCALE=HUBER<(D=d)>) Compute $\hat{\sigma}$ by the iteration

$$(\hat{\sigma}^{(m+1)})^2 = \frac{1}{nh}\sum_{i=1}^{n}\chi_d(\frac{r_i}{\hat{\sigma}^{(m)}})(\hat{\sigma}^{(m)})^2$$

   where

$$\chi_d(x) = \begin{cases} x^2/2 & \text{if } |x| < d \\ d^2/2 & \text{otherwise} \end{cases}$$

   is the Huber function and $h = \frac{n-p}{n}(d^2 + (1-d^2)\Phi(d) - .5 - d\sqrt{2\pi}e^{-\frac{1}{2}d^2})$ is the Huber constant (refer to Huber 1981, p. 179). You can specify d with the D= option. By default, $d = 2.5$.

2. (SCALE=TUKEY<(D=d)>) Compute $\hat{\sigma}$ by solving the supplementary equation

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi_d(\frac{r_i}{\sigma}) = \beta$$

   where

$$\chi_d(x) = \begin{cases} \frac{3x^2}{d^2} - \frac{3x^4}{d^4} + \frac{x^6}{d^6} & \text{if } |x| < d \\ 1 & \text{otherwise} \end{cases}$$

   Here $\psi = \frac{1}{6}\chi_1'$ is Tukey's biweight function, and $\beta = \int \chi_d(s)d\Phi(s)$ is the constant such that the solution $\hat{\sigma}$ is asymptotically consistent when $L(\cdot/\sigma) = \Phi(\cdot)$ (refer to Hampel et. al. 1986, p. 149). You can specify d with the D= option. By default, $d = 2.5$.

3. (SCALE=MED) Compute $\hat{\sigma}$ by the iteration

$$\hat{\sigma}^{(m+1)} = \text{median}\{|y_i - x_i^T\hat{\theta}^{(m)}|/\beta_0, i = 1, ..., n\}$$

   where $\beta_0 = \Phi^{-1}(.75)$ is the constant such that the solution $\hat{\sigma}$ is asymptotically consistent when $L(\cdot/\sigma) = \Phi(\cdot)$ (refer to Hampel et. al. 1986, p. 312).

Note that SCALE = MED is the default.

Algorithm

The basic algorithm for computing M estimates for regression is iteratively reweighted least squares (IRLS). As the name suggests, a weighted least squares fit is carried out inside an iteration loop. For each iteration, a set of weights for the observations is used in the least squares fit. The weights are constructed by applying a weight function to the current residuals. Initial weights are based on residuals from an initial fit. The ROBUSTREG procedure uses the unweighted least squares fit as a default initial fit. The iteration terminates when a convergence criterion is satisfied. The maximum number of iterations is set to 1000. You can specify the weight function and the convergence criteria.

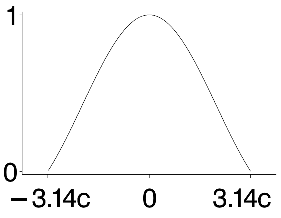
Weight Functions

You can specify the weight function for M estimation with the WEIGHTFUNCTION= option. The ROBUSTREG procedure provides ten weight functions. By default, the procedure uses the bisquare weight function. In most cases, M estimates are more sensitive to the parameters of these weight functions than to the type of the weight function. The median weight function is not stable and is seldom recommended in data analysis; it is included in the procedure for completeness. You can specify the parameters for these weight functions. Except for the hampel and median weight functions, default values for these parameters are defined such that the corresponding M estimates have 95% asymptotic efficiency in the location model with the Gaussian distribution (see Holland and Welsch (1977)).

The following list shows the weight functions available.

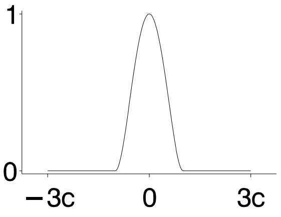
andrews

$$W(x,c) = \begin{cases} \frac{\sin(\frac{x}{c})}{\frac{x}{c}} & \text{if } |x| \le \pi c \\ 0 & \text{otherwise} \end{cases}$$



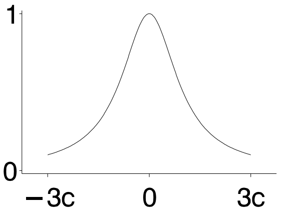
bisquare

$$W(x,c) = \begin{cases} (1-(\frac{x}{c})^2)^2 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases}$$



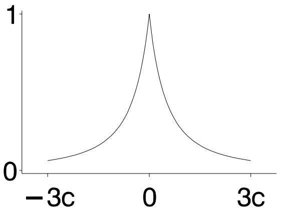
cauchy

$$W(x,c) = \frac{1}{1+(\frac{|x|}{c})^2}$$



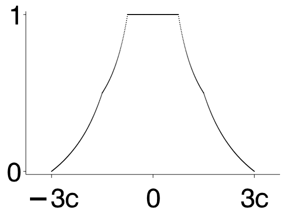
fair

$$W(x,c) = \frac{1}{(1+\frac{|x|}{c})}$$



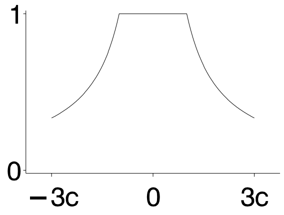
hampel

$$W(x,a,b,c) = \begin{cases} 1 & |x| < a \\ \frac{a}{|x|} & a < |x| \le b \\ \frac{a}{|x|}\frac{c-|x|}{c-b} & b < |x| \le c \\ 0 & \text{otherwise} \end{cases}$$



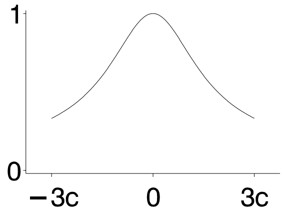
huber

$$W(x,c) = \begin{cases} 1 & \text{if } |x| < c \\ \frac{c}{|x|} & \text{otherwise} \end{cases}$$



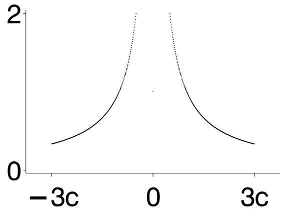
logistic

$$W(x,c) = \frac{\tanh(\frac{x}{c})}{\frac{x}{c}}$$



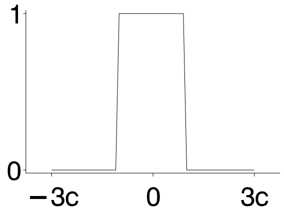
median

$$W(x,c) = \begin{cases} \frac{1}{c} & \text{if } x = 0 \\ \frac{1}{|x|} & \text{otherwise} \end{cases}$$

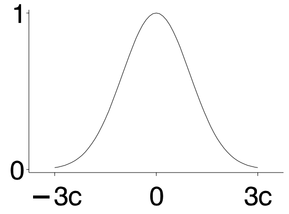


talworth

$$W(x,c) = \begin{cases} 1 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases}$$



welsch $W(x,c) = \exp(-\frac{1}{2}(\frac{x}{c})^2)$



See Table 62.4 on page 3985 for the default values of the constants in these weight functions.

Convergence Criteria

The following convergence criteria are available in PROC ROBUSTREG:

1. Relative change in the scaled residuals (CONVERGENCE= RESID)
2. Relative change in the coefficients (CONVERGENCE= COEF)
3. Relative change in weights (CONVERGENCE= W)

You can specify the criteria with the CONVERGENCE= option in the PROC statement. The default is CONVERGENCE= COEF.

You can specify the precision of the convergence criterion with the EPS= sub-option.

In addition to these convergence criteria, a convergence criterion based on scale-independent measure of the gradient is always checked. See Coleman, et. al. (1980). A warning is issued if this criterion is not satisfied.

Asymptotic Covariance and Confidence Intervals

The following three estimators of the asymptotic covariance of the robust estimator are available in PROC ROBUSTREG:

$$\text{H1: } K^2 \frac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]^2}(X^T X)^{-1}$$

$$\text{H2: } K \frac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]}W^{-1}$$

$$\text{H3: } K^{-1} \frac{1}{(n-p)}\sum(\psi(r_i))^2 W^{-1}(X^T X)W^{-1}$$

where $K = 1 + \frac{p}{n}\frac{var(\psi')}{(E\psi')^2}$ is a correction factor and $W_{jk} = \sum \psi'(r_i)x_{ij}x_{ik}$. Refer to Huber (1981, p. 173) for more details.

You can specify the asymptotic covariance estimate with the option ASYMPCOV= option. The ROBUSTREG procedure uses H1 as the default because of its simplicity and stability. Confidence intervals are computed from the diagonal elements of the estimated asymptotic covariance matrix.

R^2 and Deviance

The robust version of R^2 is defined as

$$R^2 = \frac{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}}) - \sum \rho(\frac{y_i - x_i^T \hat{\theta}}{\hat{s}})}{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}})}$$

and the robust deviance is defined as the optimal value of the objective function on the σ^2-scale:

$$D = 2(\hat{s})^2 \sum \rho(\frac{y_i - x_i^T \hat{\theta}}{\hat{s}})$$

where $\rho' = \psi$, $\hat{\theta}$ is the M estimator of θ, $\hat{\mu}$ is the M estimator of location, and $\hat{s}$ is the M estimator of the scale parameter in the full model.

Linear Tests

Two tests are available in PROC ROBUSTREG for the canonical linear hypothesis

$$\mathcal{H}_0 : \quad \theta_j = 0, \quad j = q+1, ..., p$$

The first test is a robust version of the F test, which is refered to as the ρ-test. Denote the M estimators in the full and reduced model as $\hat{\theta}_0 \in \Omega_0$ and $\hat{\theta}_1 \in \Omega_1$, respectively. Let

$$\begin{aligned} Q_0 &= Q(\hat{\theta}_0) = \min\{Q(\theta) | \theta \in \Omega_0\} \\ Q_1 &= Q(\hat{\theta}_1) = \min\{Q(\theta) | \theta \in \Omega_1\} \end{aligned}$$

with

$$Q = \sum_{i=1}^{n} \rho(\frac{r_i}{\sigma})$$

The robust F test is based on the test statistic

$$S_n^2 = \frac{2}{p-q}[Q_1 - Q_0]$$

Asymptotically $S_n^2 \sim \lambda\chi^2_{p-q}$ under $\mathcal{H}_0$, where the standardization factor is $\lambda = \int \psi^2(s)d\Phi(s)/\int \psi'(s)d\Phi(s)$ and Φ is the cumulative distribution function of the standard normal distribution. Large values of S_n^2 are significant. This test is a special case of the general τ-test of Hampel et. al. (1986, Section 7.2).

The second test is a robust version of the Wald test, which is refered to as R_n^2-test. The test uses a test statistic

$$R_n^2 = n(\hat{\theta}_{q+1}, ..., \hat{\theta}_p)H_{22}^{-1}(\hat{\theta}_{q+1}, ..., \hat{\theta}_p)^T$$

where $\frac{1}{n}H_{22}$ is the $(p-q) \times (p-q)$ lower right block of the asymptotic covariance matrix of the M estimate $\hat{\theta}_M$ of θ in a p-parameter linear model.

Under $\mathcal{H}_0$, the statistic R_n^2 has an asymptotic χ^2 distribution with $p-q$ degrees of freedom. Large absolute values of R_n^2 are significant. Refer to Hampel et. al. (1986, Chapter 7).

Model Selection

When M estimation is used, two criteria are available in PROC ROBUSTREG for model selection. The first criterion is a counterpart of the Akaike (1974) AIC criterion for robust regression, and it is defined as

$$AICR = 2\sum_{i=1}^{n} \rho(r_{i:p}) + \alpha p$$

where $r_{i:p} = (y_i - x_i^T\hat{\theta})/\hat{\sigma}$, $\hat{\sigma}$ is a robust estimate of σ and $\hat{\theta}$ is the M estimator with p-dimensional design matrix.

As with AIC, α is the weight of the penalty for dimensions. The ROBUSTREG procedure uses $\alpha = 2E\psi^2/E\psi'$ (Ronchetti (1985)) and estimates it using the final robust residuals.

The second criterion is a robust version of the Schwarz information criteria(BIC), and it is defined as

$$BICR = 2\sum_{i=1}^{n} \rho(r_{i:p}) + p\log(n)$$

High Breakdown Value Estimation

The *breakdown value* of an estimator is defined as the smallest fraction of contamination that can cause the estimator to take on values arbitrarily far from its value on the uncontamined data. The breakdown value of an estimator can be used as a measure of the robustness of the estimator. Rousseeuw and Leroy (1987) and others introduced the following high breakdown value estimators for linear regression.

LTS Estimate

The least trimmed squares (LTS) estimate proposed by Rousseeuw (1984) is defined as the p-vector

$$\hat{\theta}_{LTS} = \arg\min_{\theta} Q_{LTS}(\theta)$$

where

$$Q_{LTS}(\theta) = \sum_{i=1}^{h} r_{(i)}^2$$

$r_{(1)}^2 \leq r_{(2)}^2 \leq ... \leq r_{(n)}^2$ are the ordered squared residuals $r_i^2 = (y_i - x_i^T\theta)^2$, $i = 1, ..., n$, and h is defined in the range $\frac{n}{2} + 1 \leq h \leq \frac{3n+p+1}{4}$.

You can specify the parameter h with the H= option in the PROC statement. By default, $h = [\frac{3n+p+1}{4}]$. The breakdown value is $\frac{n-h}{n}$ for the LTS estimate.

The least median of squares (LMS) estimate is defined as the p-vector

$$\hat{\theta}_{LMS} = \arg\min_{\theta} Q_{LMS}(\theta)$$

where

$$Q_{LMS}(\theta) = r_{(h)}^2$$

$r_{(1)}^2 \leq r_{(2)}^2 \leq ... \leq r_{(n)}^2$ are the ordered squared residuals $r_i^2 = (y_i - x_i^T\theta)^2$, $i = 1, ..., n$, and h is defined in the range $\frac{n}{2} + 1 \leq h \leq \frac{3n+p+1}{4}$.

The breakdown value for the LMS estimate is also $\frac{n-h}{n}$. However the LTS estimate has several advantages over the LMS estimate. Its objective function is smoother, making the LTS estimate less “jumpy” (i.e. sensitive to local effects) than the LMS estimate. Its statistical efficiency is better, because the LTS estimate is asymptotically normal whereas the LMS estimate has a lower convergence rate (Rousseeuw and Leroy (1987)). Another important advantage is that, using the FAST-LTS algorithm of Rousseeuw and Van Driessen (2000), the LTS estimate takes less computing time and is more accurate.

The ROBUSTREG procedure computes LTS estimates using the FAST-LTS algorithm. The estimates are often used to detect outliers in the data, which are then downweighted in the resulting weighted LS regression.

Algorithm

Least trimmed squares (LTS) regression is based on the subset of h observations (out of a total of n observations) whose least squares fit possesses the smallest sum of squared residuals. The coverage h may be set between $\frac{n}{2}$ and n. The LTS method was proposed by Rousseeuw (1984, p. 876) as a highly robust regression estimator with breakdown value $\frac{n-h}{n}$. The ROBUSTREG procedure uses the FAST-LTS algorithm given by Rousseeuw and Van Driessen (1998). The intercept adjustment technique is also used in this implementation. However, because this adjustment is expensive to compute, it is optional. You can use the IADJUST option in the PROC statement to request or suppress the intercept adjustment. By default, PROC ROBUSTREG does intercept adjustment for data sets with less than 10000 observations. The algorithm is described briefly as follows. Refer to Rousseeuw and Van Driessen (2000) for details.

1. The default h is $[\frac{3n+p+1}{4}]$, where p is the number of independent variables. You can specify any integer h with $[\frac{n}{2}] + 1 \leq h \leq [\frac{3n+p+1}{4}]$ with the H= option in the MODEL statement. The breakdown value for LTS, $\frac{n-h}{n}$, is reported. The default h is a good compromise between breakdown value and statistical efficiency.
2. If $p = 1$ (single regressor) the procedure uses the exact algorithm of Rousseeuw and Leroy (1987, p. 172-172).
3. If $p \geq 2$, the procedure uses the following algorithm. If $n < 2$*ssubs*, where *ssubs* is the size of the subgroups (you can specify *ssubs* using the SUBGROUPSIZE= option in the PROC statement, by default, *ssubs* $= 300$), draw a random p-subset and compute the regression coefficients using these p points (if the regression is degenerate, draw another p-subset). Compute the absolute residuals for all observations in the data set and select the first h points with smallest absolute residuals. From this selected h-subset, carry out *nsteps* C-steps (Concentration step, see Rousseeuw and Van Driessen (2000) for details. You can specify *nsteps* with the CSTEP= option in the PROC statement, by default, *nsteps* $= 2$). Redraw p-subsets and repeat the preceding computing procedure *nrep* times and find the *nbsol* (at most) solutions with the lowest sums of h squared residuals. *nrep* can be specified with the NREP= option in the PROC statement. By default, NREP=$\min\{500, \binom{n}{p}\}$. For small n and p, all $\binom{n}{p}$ subsets are used and the NREP= option is ignored (Rousseeuw and Hubert (1996)). *nbsol* can be specified with the NBEST= option in the PROC statement. By default, NBEST=10. For each of these *nbsol* best solutions, take C-steps until convergence and find the best final solution.
4. If $n \geq 5$*ssubs*, construct 5 disjoint random subgroups with size *ssubs*. If 2*ssubs* $< n < 5$*ssubs*, the data are split into at most four subgroups with *ssubs* or more observations in each subgroup, so that each observation belongs to a subgroup and such that the subgroups have roughly the same size. Let *nsubs* denote the number of subgroups. Inside each subgroup, repeat the procedure in Step 3 $[\frac{nrep}{nsubs}]$ times and keep the *nbsol* best solutions. Pool the subgroups, yielding the merged set of size n_{merged}. In the merged set, for each of the *nsubs* × *nbsol* best solutions, carry out *nsteps* C-steps using n_{merged} and $h_{merged} = [n_{merged}\frac{h}{n}]$ and keep the *nbsol* best solutions. In the full data set, for

each of these *nbsol* best solutions, take C-steps using n and h until convergence and find the best final solution.

R^2

The robust version of R^2 for the LTS estimate is defined as

$$R^2_{LTS} = 1 - \frac{s^2_{LTS}(X, y)}{s^2_{LTS}(\mathbf{1}, y)}$$

for models with the intercept term and as

$$R^2_{LTS} = 1 - \frac{s^2_{LTS}(X, y)}{s^2_{LTS}(\mathbf{0}, y)}$$

for models without the intercept term, where

$$s_{LTS}(X, y) = d_{h,n}\sqrt{\frac{1}{h}\sum_{i=1}^{h} r^2_{(i)}}$$

s_{LTS} is a preliminary estimate of the parameter σ in the distribution function $L(\cdot/\sigma)$.

Here $d_{h,n}$ is chosen to make s_{LTS} consistent assuming a Gaussian model. Specifically,

$$\begin{aligned} d_{h,n} &= 1/\sqrt{1 - \frac{2n}{hc_{h,n}}\phi(1/c_{h,n})} \\ c_{h,n} &= 1/\Phi^{-1}(\frac{h+n}{2n}) \end{aligned}$$

with Φ and ϕ being the distribution function and the density function of the standard normal distribution, respectively.

Final Weighted Scale Estimator

The ROBUSTREG procedure displays two scale estimators, s_{LTS} and Wscale. The estimate Wscale is a more efficient scale estimate based on the preliminary estimate s_{LTS}, and it is defined as

$$\text{Wscale} = \sqrt{\frac{\sum_i w_i r_i^2}{\sum_i w_i - p}}$$

where

$$w_i = \begin{cases} 0 & \text{if } |r_i|/s_{LTS} > k \\ 1 & \text{otherwise} \end{cases}$$

You can specify k with the CUTOFF= option in the MODEL statement. By default, $k = 3$.

S Estimate

The S estimate proposed by Rousseeuw and Yohai (1984) is defined as the p-vector

$$\hat{\theta}_S = \arg\min_{\theta} S(\theta)$$

where the dispersion $S(\theta)$ is the solution of

$$\frac{1}{n-p}\sum_{i=1}^{n} \chi(\frac{y_i - x_i^T\theta}{S}) = \beta.$$

Here β is set to $\int \chi(s)d\Phi(s)$ such that $\hat{\theta}_S$ and $S(\hat{\theta}_S)$ are asymptotically consistent estimates of θ and σ for the Gaussian regression model. The breakdown value of the S estimate is

$$\frac{\beta}{\sup_s \chi(s)}$$

The ROBUSTREG procedure provides two choices for χ: the Tukey function and the Yohai function.

The Tukey function, which you can specify with the option CHIF=TUKEY, is

$$\chi_{k_0}(s) = \begin{cases} 3(\frac{s}{k_0})^2 - 3(\frac{s}{k_0})^4 + (\frac{s}{k_0})^6, & \text{if } |s| \le k_0 \\ 1 & \text{otherwise} \end{cases}$$

The constant k_0 controls the breakdown value and efficiency of the S estimate. By specifying the efficiency using the EFF= option, you can determine the corresponding k_0. The default k_0 is 2.9366 such that the breakdown value of the S estimate is 0.25 with a corresponding asymptotic efficiency for the Gaussian model of 75.9%.

The Yohai function, which you can specify with the option CHIF=YOHAI, is

$$\chi_{k_0}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \le 2k_0 \\ k_0^2[b_0 + b_1(\frac{s}{k_0})^2 + b_2(\frac{s}{k_0})^4 & \\ +b_3(\frac{s}{k_0})^6 + b_4(\frac{s}{k_0})^8] & \text{if } 2k_0 < |s| \le 3k_0 \\ 3.25k_0^2 & \text{if } |s| > 3k_0 \end{cases}$$

where $b_0 = 1.792$, $b_1 = -0.972$, $b_2 = 0.432$, $b_3 = -0.052$, and $b_4 = 0.002$. By specifying the efficiency using the EFF= option, you can determine the corresponding k_0. By default, k_0 is set to 0.7405 such that the breakdown value of the S estimate is 0.25 with a corresponding asymptotic efficiency for the Gaussian model of 72.7%.

Algorithm

The ROBUSTREG procedure implements the algorithm by Marazzi (1993) for the S estimate, which is a refined version of the algorithm proposed by Ruppert (1992). The refined algorithm is briefly described as follows.

Initialize $\mathrm{iter} = 1$.

1. Draw a random q-subset of the total n observations and compute the regression coefficients using these q observations (if the regression is degenerate, draw another q-subset), where $q \geq p$ can be specified with the SUBSIZE= option. By default, $q = p$.
2. Compute the residuals: $r_i = y_i - \sum_{i=1}^{p} x_{ij}\theta_j$ for $i = 1, ..., n$. If $\mathrm{iter} = 1$, set $s^* =$ 2median$\{|r_i|, i = 1, ..., n\}$; if $s^* = 0$, set $s^* = \min\{|r_i|, i = 1, ..., n\}$; while $\sum_{i=1}^{n} \chi(r_i/s^*) > (n-p)\beta$, set $s^* = 1.5s^*$; go to Step 3.
   If $\mathrm{iter} > 1$ and $\sum_{i=1}^{n} \chi(r_i/s^*) <= (n-p)\beta$, go to the Step 3; else go to Step 5.
3. Solve for s the equation

   $$\frac{1}{n-p}\sum_{i=1}^{n} \chi(r_i/s) = \beta$$

   using an iterative algorithm.
4. If $\mathrm{iter} > 1$ and $s > s^*$, go to Step 5. Otherwise, set $s^* = s$ and $\theta^* = \theta$. If $s^* < TOLS$, return s^* and θ^*; else go to Step 5.
5. if $\mathrm{iter} < NREP$, set $\mathrm{iter} = \mathrm{iter} + 1$ and return to Step 1; else return s^* and θ^*.

The ROBUSTREG procedure does the following refinement step by default. You can request this refinement not be done using the NOREFINE option in the PROC statement.

6. Let $\psi = \chi'$. Using the values s^* and θ^* from the previous steps, compute M estimates θ_M and σ_M of θ and σ with the setup for M estimation in the section "M Estimation" on page 3993. If $\sigma_M > s^*$, give a warning and return s^* and θ^*; otherwise, return σ_M and θ_M.

You can specify $TOLS$ with the TOLERANCE= option; by default, $TOLS = .001$. Alternately You can specify $NREP$ with the NREP= option. You can also use the options NREP= NREP0 or NREP= NREP1 to determine $NREP$ according to the following table. NREP= NREP0 is set as the default.

Table 62.7. Default NREP

P	NREP0	NREP1
1	150	500
2	300	1000
3	400	1500
4	500	2000
5	600	2500
6	700	3000
7	850	3000
8	1250	3000
9	1500	3000
>9	1500	3000

R^2 and Deviance

The robust version of R^2 for the S estimate is defined as

$$R_S^2 = 1 - \frac{(n-p)S_p^2}{(n-1)S_\mu^2}$$

for the model with the intercept term and

$$R_S^2 = 1 - \frac{(n-p)S_p^2}{nS_0^2}$$

for the model without the intercept term, where S_p is the S estimate of the scale in the full model, S_μ is the S estimate of the scale in the regression model with only the intercept term, and S_0 is the S estimate of the scale without any regressor. The deviance D is defined as the optimal value of the objective function on the σ^2-scale:

$$D = S_p^2$$

Asymptotic Covariance and Confidence Intervals

Since the S estimate satisfies the first-order necessary conditions as the M estimate, it has the same asymptotic covariance as that of the M estimate. All three estimators of the asymptotic covariance for the M estimate in the section "Asymptotic Covariance and Confidence Intervals" on page 3997 can be used for the S estimate. Besides, the weighted covariance estimator H4 described in the section "Asymptotic Covariance and Confidence Intervals" on page 4008 is also available and is set as the default. Confidence intervals for estimated parameters are computed from the diagonal elements of the estimated asymptotic covariance matrix.

MM Estimation

MM estimation is a combination of high breakdown value estimation and efficient estimation, which was introduced by Yohai (1987). It has three steps:

1. Compute an initial (consistent) high breakdown value estimate $\hat{\theta}'$. The ROBUSTREG procedure provides two kinds of estimates as the initial estimate, the LTS estimate and the S estimate. By default, the LTS estimate because of its speed, efficiency, and high breakdown value. The breakdown value of the final MM estimate is decided by the breakdown value of the initial LTS estimate and the constant k_0 in the χ function. To use the S estimate as the initial estimate, you specify the INITEST=S option in the PROC statement. In this case, the breakdown value of the final MM estimate is decided only by the constant k_0. Instead of computing the LTS estimate or the S estimate as initial estimates, you can also specify the initial estimate explicitly using the INEST= option in the PROC statement. See the section "INEST= Data Set" on page 4011 for details.

2. Find $\hat{\sigma}'$ such that

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi\left(\frac{y_i - x_i^T\hat{\theta}'}{\hat{\sigma}'}\right) = \beta$$

   where $\beta = \int \chi(s)d\Phi(s)$.

   The ROBUSTREG procedure provides two choices for χ: the Tukey function and the Yohai function.

   The Tukey function, which you can specify with the option CHIF=TUKEY, is

$$\chi_{k_0}(s) = \begin{cases} 3(\frac{s}{k_0})^2 - 3(\frac{s}{k_0})^4 + (\frac{s}{k_0})^6, & \text{if } |s| \le k_0 \\ 1 & \text{otherwise} \end{cases}$$

   where k_0 can be specified with the K0= option. The default $k_0 = 2.9366$ such that the asymptotically consistent scale estimate $\hat{\sigma}'$ has the breakdown value of 25%.

   The Yohai function, which you can specify with the option CHIF=YOHAI, is

$$\chi_{k_0}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \le 2k_0 \\ k_0^2[b_0 + b_1(\frac{s}{k_0})^2 + b_2(\frac{s}{k_0})^4 & \\ +b_3(\frac{s}{k_0})^6 + b_4(\frac{s}{k_0})^8] & \text{if } 2k_0 < |s| \le 3k_0 \\ 3.25k_0^2 & \text{if } |s| > 3k_0 \end{cases}$$

   where $b_0 = 1.792$, $b_1 = -0.972$, $b_2 = 0.432$, $b_3 = -0.052$, and $b_4 = 0.002$. You can specify k_0 with the K0= option. The default k_0 is .7405 such that the asymptotically consistent scale estimate $\hat{\sigma}'$ has the breakdown value of 25%.

3. Find a local minimum $\hat{\theta}_{MM}$ of

$$Q_{MM} = \sum_{i=1}^{n} \rho(\frac{y_i - x_i^T\theta}{\hat{\sigma}'})$$

such that $Q_{MM}(\hat{\theta}_{MM}) \leq Q_{MM}(\hat{\theta}')$. The algorithm for M estimation is used here.

The ROBUSTREG procedure provides two choices for χ: the Tukey function and the Yohai function.

The Tukey function, which you can specify with the option CHIF=TUKEY, is

$$\rho(s) = \chi_{k_1}(s) = \begin{cases} 3(\frac{s}{k_1})^2 - 3(\frac{s}{k_1})^4 + (\frac{s}{k_1})^6, & \text{if } |s| \leq k_1 \\ 1 & \text{otherwise} \end{cases}$$

where k_1 can be specified with the K1= option. The default k_1 is 3.440 such that the MM estimate has 85% asymptotic efficiency with the Gaussian distribution.

The Yohai function, which you can specify with the option CHIF=Yohai, is

$$\rho(s) = \chi_{k_1}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \leq 2k_1 \\ k_1^2[b_0 + b_1(\frac{s}{k_1})^2 + b_2(\frac{s}{k_1})^4 & \\ +b_3(\frac{s}{k_1})^6 + b_4(\frac{s}{k_1})^8] & \text{if } 2k_1 < |s| \leq 3k_1 \\ 3.25k_1^2 & \text{if } |s| > 3k_1 \end{cases}$$

where k_1 can be specified with the K1= option. The default k_1 is 0.868 such that the MM estimate has 85% asymptotic efficiency with the Gaussian distribution.

Algorithm

The initial LTS estimate is computed using the algorithm described in the section "LTS Estimate" on page 4000. You can control the quantile of the LTS estimate with the option INITH=h, where h is an integer between $[\frac{n}{2}]+1$ and $[\frac{3n+p+1}{4}]$. By default, $h = [\frac{3n+p+1}{4}]$, which corresponds to a breakdown value of around 25%.

The initial S estimate is computed using the algorithm described in the section "S Estimate" on page 4003. You can control the breakdown value and efficiency of this initial S estimate by the constant k_0 which can be specified with the K0 option.

The scale parameter σ is solved by an iterative algorithm

$$(\sigma^{(m+1)})^2 = \frac{1}{(n-p)\beta} \sum_{i=1}^{n} \chi_{k_0}(\frac{r_i}{\sigma^{(m)}})(\sigma^{(m)})^2$$

where $\beta = \int \chi_{k_0}(s)d\Phi(s)$.

Once the scale parameter is computed, the iteratively reweighted least squares (IRLS) algorithm with fixed scale parameter is used to compute the final MM estimator.

Convergence Criteria

In the iterative algorithm for the scale parameter, the relative change of the scale parameter controls the convergence.

In the iteratively reweighted least squares algorithm, the same convergence criteria for the M estimate used before are used here.

Bias Test

Although the final MM estimate inherits the high-breakdown-value properity, its bias due to the distortion of the outliers can be high. Yohai, Stahel, and Zamar (1991) introduced a bias test. The ROBUSTREG procedure implements this test when you specify the BIASTEST option in the PROC statement. This test bases on the initial scale estimate $\hat\sigma'$ and the final scale estimate $\hat\sigma'_1$, which is the solution of

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi(\frac{y_i - x_i^T\hat\theta_{MM}}{\hat\sigma'_1}) = \beta$$

Let $\psi_{k_0}(\cdot) = \chi'_{k_0}(\cdot)$ and $\psi_{k_1}(\cdot) = \chi'_{k_1}(\cdot)$, where $'$ denotes the derivative with respect to the argument. Compute

$$\begin{aligned}
\tilde r_i &= (y_i - x_i^T\hat\theta')/\hat\sigma' \quad \text{for } i = 1, ..., n \\
v_0 &= \frac{(1/n)\sum \psi'_{k_0}(\tilde r_i)}{(\hat\sigma'_1/n)\sum \psi_{k_0}(\tilde r_i)\tilde r_i} \\
p_i^{(0)} &= \frac{\psi_{k_0}(\tilde r_i)}{(1/n)\sum \psi'_{k_0}(\tilde r_i)} \quad \text{for } i = 1, ..., n \\
p_i^{(1)} &= \frac{\psi_{k_1}(\tilde r_i)}{(1/n)\sum \psi'_{k_1}(\tilde r_i)} \quad \text{for } i = 1, ..., n \\
d^2 &= \frac{1}{n}\sum (p_i^{(1)} - p_i^{(0)})^2 \\
T &= \frac{2n(\hat\sigma'_1 - \hat\sigma')}{v_0 d^2 (\hat\sigma')^2}
\end{aligned}$$

Standard asymptotic theory shows that T approximately follows a χ^2-distribution with p degrees of freedom. If T exceeds the α quantile χ^2_α of the χ^2-distribution with p degrees of freedom, then the ROBUSTREG procedure gives a warning and recommends to use other methods. Otherwise the final MM estimate and the initial scale estimate are reported. You can specify α with the ALPHA= option following the BIASTEST option. By default, ALPHA= 0.99.

Asymptotic Covariance and Confidence Intervals

Since the MM estimate is computed as a M estimate with a fixed scale in the last step, the asymptotic covariance for the M estimate can be used here for the asymptotic covariance of the MM estimate. Besides the three estimators H1, H2, and H3 as described in the section "Asymptotic Covariance and Confidence Intervals" on page 3997, a weighted covariance estimator H4 is available:

$$\text{H4: } K^2 \frac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]^2} W^{-1}$$

where $K = 1 + \frac{p}{n}\frac{var(\psi')}{(E\psi')^2}$ is the correction factor and $W_{jk} = \frac{1}{\bar{w}}\sum w_i x_{ij} x_{ik}$, $\bar{w} = \frac{1}{n}\sum w_i$.

You can specify these estimators with the option ASYMPCOV= [H1 | H2 | H3 | H4]. The ROBUSTREG procedure uses H4 as default. Confidence intervals for estimated parameters are computed from the diagonal elements of the estimated asymptotic covariance matrix.

R^2 and Deviance

The robust version of R^2 for the MM estimate is defined as

$$R^2 = \frac{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}}) - \sum \rho(\frac{y_i - x_i^T\hat{\theta}}{\hat{s}})}{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}})}$$

and the robust deviance is defined as the optimal value of the objective function on the σ^2-scale:

$$D = 2(\hat{s})^2 \sum \rho(\frac{y_i - x_i^T\hat{\theta}}{\hat{s}})$$

where $\rho' = \psi$, $\hat{\theta}$ is the MM estimator of θ, $\hat{\mu}$ is the MM estimator of location, and $\hat{s}$ is the MM estimator of the scale parameter in the full model.

Linear Tests

For MM estimation, the same ρ-test and R_n^2-test used for M estimation can be used. See the section "Linear Tests" on page 3998 for details.

Model Selection

For MM estimation, the same two model selection methods used for M estimation can be used. See the section "Model Selection" on page 3999 for details.

Robust Multivariate Location and Scale Estimates

The ROBUSTREG procedure uses the robust multivariate location and scale estimates for leverage points detection. The procedure provides the minimum covariance determinant (MCD) method, which was introduced by Rousseeuw (1984).

Algorithm

PROC ROBUSTREG implements the algorithm given by Rousseeuw and Van Driessen (1999) for MCD, which is similar to the algorithm for LTS.

Robust Distance

The Mahalanobis Distance is defined as

$$MD(x_i) = [(x_i - \bar{x})^T \bar{C}(X)^{-1}(x_i - \bar{x})]^{1/2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{C} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^T(x_i - \bar{x})$. Here $x_i = (x_{i1}, ..., x_{i(p-1)})^T$ do not include the constant variable. The relation between the Mahalanobis distance $MD(x_i)$ and the hat matrix $H = (h_{ij}) = X(X^TX)^{-1}X^T$ is

$$h_{ii} = \frac{1}{n-1}MD_i^2 + \frac{1}{n}$$

The Robust Distance is defined as

$$RD(x_i) = [(x_i - T(X))^T C(X)^{-1}(x_i - T(X))]^{1/2}$$

where $T(X)$ and $C(X)$ are the robust multivariate location and scale obtained by MCD.

These distances are used to detect leverage points.

Leverage Point and Outlier Detection

Let $C(p) = \sqrt{\chi^2_{p;1-\alpha}}$ be the cutoff value. The variable LEVERAGE is defined as

$$\text{LEVERAGE} = \begin{cases} 0 & \text{if } RD(x_i) \leq C(p) \\ 1 & \text{otherwise} \end{cases}$$

You can specify a cutoff value with the LEVERAGE option in the MODEL statement.

Residuals $r_i, i = 1, ..., n$ based on robust regression estimates are used to detect vertical outliers. The variable OUTLIER is defined as

$$\text{OUTLIER} = \begin{cases} 0 & \text{if } |r_i| \leq k\sigma \\ 1 & \text{otherwise} \end{cases}$$

You can specify the multiplier k of the cutoff value by using the CUTOFF= option in the MODEL statement.

An ODS table called DIAGNOSTICS contains these two variables.

INEST= Data Set

When you use the M or MM estimation, you cna use the INEST= data set to specify initial estimates for all the parameters in the model. The INEST= option is ignored if you specify LTS or S estimation using the METHOD=LTS or METHOD=S option or if you specify the INITEST= option after the METHOD=MM option in the PROC statement. The INEST= data set must contain the intercept variable (named Intercept) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in one BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value "PARMS" are used as starting values.

You can specify starting values for the iteratively reweighted least squares algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different modeling using the ROBUSTREG procedure. If the COVOUT option is specified, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates. Note that, if the ROBUSTREG procedure does not converge, the parameter estimates are set to missing in the OUTEST data set.

The OUTES= data set contains all variables specified in the MODEL statement and the BY statement. One observation consists of parameter values for the model with the dependent variable having the value -1. If the COVOUT option is specified, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL a character variable containing the label of the MODEL statement, if present. Otherwise, the variable's value is blank

NAME a character variable containing the name of the dependent variable for the parameter estimates observations or the name of the row for the covariance matrix estimates

TYPE a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates

METHOD a character variable of containing the type of estimation method, either M estimation, or LTS estimation, or S estimation, or MM estimation

STATUS	a character variable containing the status of model fitting, either Converged, or Warning, or Failed
INTERCEPT	a numeric variable containing the intercept parameter estimates and covariances
SCALE	a numeric variable containing the scale parameter estimates

Any BY variables specified are also added to the OUTEST= data set.

Computational Resources

The algorithms for the various different estimation methods need different amount of memory for working space. Let p be the number of parameters estimated and n be the number of observations used in the model estimation.

For M estimation, the minimum working space (in bytes) needed is

$$3n + 2p^2 + 30p$$

If sufficient space is available, the input data set is also kept in memory; otherwise, the input data set is reread for computing the iteratively reweighted least squares estimates and the execution time of the procedure increases substantially. For each reweighted least squares, $O(np^2 + p^3)$ multiplications and additions are required for computing the cross product matrix and its inverse. The $O(v)$ notation means that, for large values of the argument, v, $O(v)$ is approximately a constant times v.

Since the iteratively reweighted least squares algorithm converges very quickly (normally within less than 20 iterations), the computation of M estimates is fast.

LTS estimation is more expensive in computation. The minimum working space (in bytes) needed is

$$np + 12n + 4p^2 + 60p$$

The memory is mainly used to store the current data used by LTS for modeling. The LTS algorithm uses subsampling and spends much of its computing time on resampling and computing estimates for subsamples. Since it resamples if singularity is detected, it may take more time if the data set has serious singularities.

The MCD algorithm for high leverage point diagnostics is similar to the LTS algorithm.

ODS Table Names

The ROBUSTREG procedures assigns a name to each table it creates. You can specify these names when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

Table 62.8. ODS Tables Produced in PROC ROBUSTREG

ODS Table Name	Description	Statement	Option
BestEstimates	Best final estimates for LTS	PROC	SUBANALYSIS
BestSubEstimates	Best estimates for each subgroup	PROC	SUBANALYSIS*
BiasTest	Bias test for MM estimation	PROC	BIASTEST
ClassLevels	Class variable levels	CLASS	default*
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
CStep	C-Step for LTS fitting	PROC	SUBANALYSIS
Diagnostics	Outlier diagnostics	MODEL	DIAGNOSTICS
DiagSummary	Summary of the outlier diagnostics	MODEL	default
GoodFit	R2, deviance, AIC, and BIC	MODEL	default
InitLTSProfile	Profile for initial LTS estimate	PROC	METHOD
InitSProfile	Profile for initial S estimate	PROC	METHOD
IterHistory	Iteration history	PROC	ITPRINT
LTSEstimates	LTS parameter estimates	PROC	METHOD
LTSLocationScale	Location and scale for LTS	PROC	METHOD
LTSProfile	Profile for LTS estimate	PROC	METHOD
LTSRsquare	R2 for LTS estimate	PROC	METHOD
MMProfile	Profile for MM estimate	PROC	METHOD
ModelInfo	Model information	MODEL	default
NObs	Observations Summary	PROC	default
ParameterEstimates	Parameter estimates	MODEL	default
ParameterEstimatesF	Final weighted LS estimates	PROC	FWLS
ParameterEstimatesR	Reduced parameter estimates	TEST	default
ParmInfo	Parameter indices	MODEL	default
SProfile	Profile for S estimate	PROC	METHOD
Groups	Groups for LTS fitting	PROC	SUBANALYSIS*
SummaryStatistics	Summary statistics for model variables	MODEL	default
TestsProfile	Results for tests	TEST	default

* Depends on data.

ODS Graphics (Experimental)

Graphical displays are important in robust regression and outlier detection. Two plots are particularly useful for revealing outliers and leverage points. The first is a scatter plot of the standardized robust residuals against the robust distances (RDPLOT). The second is a scatter plot of the robust distances against the classical Mahalanobis distances (DDPLOT). See Figure 62.4 on page 3975 and Figure 62.5 on page 3975 for examples. In addition to these two plots, a histogram and a quantile-quantile plot of the standardized robust residuals are also helpful. See Figure 62.6 on page 3976 and Figure 62.7 on page 3976 for examples.

This section describes the use of ODS for creating these four plots with the ROBUSTREG procedure. These graphics are experimental in this release, meaning that both the graphical results and the syntax for specifying them are subject to change in a future release.

To request these plots you must specify the ODS GRAPHICS statement in addition to the PLOT= (or PLOTS=) option, which is described as follows. For more information on the ODS GRAPHICS statement, see Chapter 15, “Statistical Graphics Using ODS.”

You can specify the PLOT= or PLOTS= option in the PROC statement to request one or more plots:

PLOT=*keyword*
PLOTS=(*keyword-list***)**

requests plots for robust regression. You can specify one or more of the following *keywords*:

Table 62.9. Options for Plots

Keyword	**Plot**
DDPLOT	Robust distance - Mahalanobis distance
RDPLOT	Standardized robust residual - Robust distance
RESHISTOGRAM	Histogram of standardized robust residuals
RESQQPLOT	Q-Q plot of standardized robust residuals
ALL	All plots

With the RDPLOT and DDPLOT options, you can label the points on the plots by specifying the LABEL= suboption immediately after the keyword:

PLOT=DDPLOT<(LABEL=*label method***)>**
PLOT=RDPLOT<(LABEL=*label method***)>**

You can specify one of the following *label methods*:

Table 62.10. Label Methods

Value of LABEL=	**Label Method**
ALL	label all points
OUTLIER	label outliers
LEVERAGE	label leverage points
NONE	no labels

By default, the ROBUSTREG procedure labels both outliers and leverage points.

If you specify ID variables in the ID statement, the values of the first ID variable are used as labels; otherwise, observation numbers are used as labels.

The histogram is superimposed with a normal density curve and a kernel density curve.

ODS Graph Names

PROC ROBUSTREG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 62.11 on page 4015.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the PLOT= (or PLOTS=) option described in Table 62.9 on page 4014. For more information on the ODS GRAPHICS statement, see Chapter 15, "Statistical Graphics Using ODS."

Table 62.11. ODS Graphics Produced by PROC ROBUSTREG

ODS Graph Name	**Plot Description**	**Statement**	**PLOTS= Option**
DDPlot	Robust distance - Mahalanobis distance	PROC	DDPLOT
RDPlot	Standardized robust residual - Robust distance	PROC	RDPLOT
ResidualHistogram	Histogram of standardized robust residuals	PROC	RESHISTOGRAM
ResidualQQPlot	Q-Q plot of standardized robust residuals	PROC	RESQQPLOT

Examples

Example 62.1. Comparison of Robust Estimates

This example illustrates differences in the performance of robust estimates available in the ROBUSTREG procedure.

The following statements generate 1000 random observations. The first 900 observations are from a linear model and the last 100 observations are significantly biased in the y-direction. In other words, ten percent of the observations are contaminated with outliers.

```
data a (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 900 then y=100 + e;
      else y=10 + 5*x1 + 3*x2 + .5 * e;
      output;
   end;
run;

proc reg data=a;
   model y = x1 x2;
run;

proc robustreg data=a method=m ;
   model y = x1 x2;
run;

proc robustreg data=a method=mm;
   model y = x1 x2;
run;
```

Output 62.1.1. OLS Estimates for Data with 10% Contamination

```
                         The REG Procedure
                           Model: MODEL1
                       Dependent Variable: y

                        Parameter Estimates

                     Parameter       Standard
Variable     DF       Estimate          Error    t Value    Pr > |t|

Intercept     1       19.06712        0.86322      22.09      <.0001
x1            1        3.55485        0.86892       4.09      <.0001
x2            1        2.12341        0.83039       2.56      0.0107
```

Output 62.1.2. M Estimates for Data with 10% Contamination

```
                         The ROBUSTREG Procedure

                            Model Information

                 Data Set                          WORK.B
                 Dependent Variable                     y
                 Number of Covariates                   2
                 Number of Observations              1000
                 Method                      M Estimation


                           Parameter Estimates

                        Standard   95% Confidence     Chi-
Parameter DF Estimate    Error        Limits        Square Pr > ChiSq

Intercept  1  10.0024   0.0174   9.9683  10.0364  331908     <.0001
x1         1   5.0077   0.0175   4.9735   5.0420 82106.9     <.0001
x2         1   3.0161   0.0167   2.9834   3.0488 32612.5     <.0001
Scale      1   0.5780
```

Output 62.1.3. MM Estimates for Data with 10% Contamination

```
                         The ROBUSTREG Procedure

                            Model Information

                 Data Set                          WORK.B
                 Dependent Variable                     y
                 Number of Covariates                   2
                 Number of Observations              1000
                 Method                     MM Estimation


                           Parameter Estimates

                        Standard   95% Confidence     Chi-
Parameter DF Estimate    Error        Limits        Square Pr > ChiSq

Intercept  1  10.0035   0.0176   9.9690  10.0379  323947     <.0001
x1         1   5.0085   0.0178   4.9737   5.0433 79600.6     <.0001
x2         1   3.0181   0.0168   2.9851   3.0511 32165.0     <.0001
Scale      0   0.6733
```

The tables of parameter estimates generated by the ROBUSTREG procedure using M estimation and MM estimation are shown in Output 62.1.2 and Output 62.1.3. For comparison, the ordinary least squares (OLS) estimates produced by the REG procedure are shown in Output 62.1.1. Both the M estimate and the MM estimate correctly estimate the regression coefficients for the underlying model (10, 5, and 3), but the OLS estimate does not.

The next statements demonstrate that if the percentage of contamination is increased to 40%, the M estimates and MM estimates with default options fail to pick up the underlying model. However, by tuning the constant c for the M estimate and the constants INITH and K0 for the MM estimate, you can increase the breakdown values of these estimates and capture the right model. Output 62.1.4 and Output 62.1.5 display these estimates.

```
data b (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 600 then y=100 + e;
      else y=10 + 5*x1 +  3*x2 + .5 * e;
      output;
   end;
run;

proc robustreg data=b method=m(wf=bisquare(c=2));
   model y = x1 x2;
run;

proc robustreg data=b method=mm(inith=502 k0=1.8);
   model y = x1 x2;
run;
```

Output 62.1.4. M Estimates for Data with 40% Contamination

```
                                The ROBUSTREG Procedure

                                   Model Information

                         Data Set                          WORK.B
                         Dependent Variable                     y
                         Number of Covariates                   2
                         Number of Observations              1000
                         Method                      M Estimation


                                  Parameter Estimates

                              Standard   95% Confidence     Chi-
              Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

              Intercept  1  10.0137   0.0219   9.9708  10.0565  209688     <.0001
              x1         1   4.9905   0.0220   4.9473   5.0336 51399.1     <.0001
              x2         1   3.0399   0.0210   2.9987   3.0811 20882.4     <.0001
              Scale      1   1.0531
```

Output 62.1.5. MM Estimates for Data with 40% Contamination

```
                                The ROBUSTREG Procedure

                                   Model Information

                         Data Set                          WORK.B
                         Dependent Variable                     y
                         Number of Covariates                   2
                         Number of Observations              1000
                         Method                     MM Estimation


                                  Parameter Estimates

                              Standard   95% Confidence     Chi-
              Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

              Intercept  1  10.0103   0.0213   9.9686  10.0520  221639     <.0001
              x1         1   4.9890   0.0218   4.9463   5.0316 52535.7     <.0001
              x2         1   3.0363   0.0201   2.9970   3.0756 22895.4     <.0001
              Scale      0   1.8997
```

When there are bad leverage points, the M estimates fail to pick up the underlying model no matter what constant c you use. In this case, other estimates (LTS, S, and MM estimates) in PROC ROBUSTREG, which are robust to bad leverage points, will pick up the underlying model.

The following statements generate 1000 observations with 1% bad high leverage points.

```
data b (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 600 then y=100 + e;
      else y=10 + 5*x1 + 3*x2 + .5 * e;
      if i < 11 then x1=200 * rannor(1234);
      if i < 11 then x2=200 * rannor(1234);
      if i < 11 then y= 100*e;
      output;
   end;
run;

proc robustreg data=b method=s(k0=1.8);
   model y = x1 x2;
run;

proc robustreg data=b method=mm(inith=502 k0=1.8);
   model y = x1 x2;
run;
```

Output 62.1.6. S Estimates for Data with 1% Leverage Points

```
                          The ROBUSTREG Procedure

                             Model Information

                 Data Set                            WORK.C
                 Dependent Variable                       y
                 Number of Covariates                     2
                 Number of Observations                1000
                 Method                        S Estimation


                            Parameter Estimates

                             Standard   95% Confidence     Chi-
       Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

       Intercept  1   9.9808   0.0216   9.9383  10.0232  212532     <.0001
       x1         1   5.0303   0.0208   4.9896   5.0710 58656.3     <.0001
       x2         1   3.0217   0.0222   2.9782   3.0652 18555.7     <.0001
       Scale      0   2.2094
```

Output 62.1.7. MM Estimates for Data with 1% Leverage Points

```
                                 The ROBUSTREG Procedure

                                    Model Information

                          Data Set                             WORK.C
                          Dependent Variable                        y
                          Number of Covariates                      2
                          Number of Observations                 1000
                          Method                        MM Estimation


                                   Parameter Estimates

                               Standard   95% Confidence     Chi-
            Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

            Intercept  1   9.9820   0.0215   9.9398  10.0241  215369     <.0001
            x1         1   5.0303   0.0206   4.9898   5.0707 59469.1     <.0001
            x2         1   3.0222   0.0221   2.9789   3.0655 18744.9     <.0001
            Scale      0   2.2134
```

Output 62.1.6 displays the S estimates and Output 62.1.7 displays the MM estimates with initial LTS estimates.

Example 62.2. Robust ANOVA

The classical analysis of variance (ANOVA) technique based on least squares assumes that the underlying experimental errors are normally distributed. However, data often contain outliers due to recording or other errors. In other cases, extreme responses occurs when control variables in the experiments is set to extremes. It is important to distinguish these extreme points and determine whether they are outliers or important extreme cases. You can use the ROBUSTREG procedure for robust analysis of variance based on M estimation. Typically, there are no high leverage points in a well-designed experiment, so M estimation is appropriate.

The following example shows how to use the ROBUSTREG procedure for robust ANOVA.

An experiment was carried out to study the effects of two successive treatments (T1, T2) on the recovery time of mice with certain diseases. Sixteen mice were randomly assigned into four groups for the four different combinations of the treatments. The recovery times (time) were recorded (in hours).

```
data recover;
   input id T1 $ T2 $ time;
   datalines;
1  0 0 20.2
2  0 0 23.9
3  0 0 21.9
4  0 0 42.4
5  1 0 27.2
6  1 0 34.0
7  1 0 27.4
8  1 0 28.5
9  0 1 25.9
10 0 1 34.5
```

```
11 0 1 25.1
12 0 1 34.2
13 1 1 35.0
14 1 1 33.9
15 1 1 38.3
16 1 1 39.9
;
```

The following statements invoke the GLM procedure for a standard ANOVA.

```
proc glm data=recover;
    class T1 T2;
    model time = T1 T2 T1*T2;
run;
```

Output 62.2.1. Overall ANOVA

```
                                          The GLM Procedure

Dependent Variable: time

                                                        Sum of
                  Source                      DF       Squares     Mean Square    F Value    Pr > F

                  Model                        3   209.9118750      69.9706250       1.86    0.1905

                  Error                       12   451.9225000      37.6602083

                  Corrected Total             15   661.8343750


                                   R-Square     Coeff Var      Root MSE     time Mean

                                   0.317167      19.94488      6.136791      30.76875
```

Output 62.2.2. Model ANOVA

```
                                          The GLM Procedure

Dependent Variable: time

                  Source                      DF     Type I SS     Mean Square    F Value    Pr > F

                  T1                           1    81.4506250      81.4506250       2.16    0.1671
                  T2                           1   106.6056250     106.6056250       2.83    0.1183
                  T1*T2                        1    21.8556250      21.8556250       0.58    0.4609


                  Source                      DF   Type III SS     Mean Square    F Value    Pr > F

                  T1                           1    81.4506250      81.4506250       2.16    0.1671
                  T2                           1   106.6056250     106.6056250       2.83    0.1183
                  T1*T2                        1    21.8556250      21.8556250       0.58    0.4609
```

Output 62.2.1 indicates that the overall model effect is not significant at the 10% level and Output 62.2.2 indicates that neither treatment is significant at the 10% level.

The following statements invoke the ROBUSTREG procedure with the same model.

```
proc robustreg data=recover;
   class T1 T2;
   model time = T1 T2 T1*T2 / diagnostics;
   T1_T2: test T1*T2;
   output out=robout r=resid sr=stdres;
run;
```

Output 62.2.3 shows some basic information about the model and the response variable time.

Output 62.2.3. Model Fitting Information and Summary Statistics

```
                                 The ROBUSTREG Procedure

                                    Model Information

                    Data Set                                WORK.RECOVER
                    Dependent Variable                              time
                    Number of Covariates                               2
                    Number of Continuous Covariates                    0
                    Number of Discrete Covariates                      2
                    Number of Observations                            16
                    Method                                  M Estimation


                                    Summary Statistics

                                                                Standard
       Variable           Q1       Median          Q3        Mean      Deviation         MAD

       time          25.5000      31.2000     34.7500     30.7688         6.6425      6.8941
```

The Parameter Estimates table in Output 62.2.4 indicates that the main effects of both treatments are significant at the 5% level.

Output 62.2.4. Model Parameter Estimates

```
                                 The ROBUSTREG Procedure

                                   Parameter Estimates

                                    Standard   95% Confidence     Chi-
            Parameter      DF Estimate    Error       Limits       Square Pr > ChiSq

            Intercept       1  36.7655   2.0489  32.7497  40.7814  321.98     <.0001
            T1        0     1  -6.8307   2.8976 -12.5100  -1.1514    5.56     0.0184
            T1        1         0.0000   0.0000   0.0000   0.0000     .         .
            T2        0     1  -7.6755   2.8976 -13.3548  -1.9962    7.02     0.0081
            T2        1         0.0000   0.0000   0.0000   0.0000     .         .
            T1*T2     0 0   1  -0.2619   4.0979  -8.2936   7.7698    0.00     0.9490
            T1*T2     0 1       0.0000   0.0000   0.0000   0.0000     .         .
            T1*T2     1 0       0.0000   0.0000   0.0000   0.0000     .         .
            T1*T2     1 1       0.0000   0.0000   0.0000   0.0000     .         .
            Scale           1   3.5346
```

Output 62.2.5. Diagnostics

```
                                 The ROBUSTREG Procedure

                                       Diagnostics

                                   Standardized
                                         Robust
                           Obs         Residual      Outlier

                             4           5.7722         *


                                  Diagnostics Summary

                        Observation
                        Type              Proportion      Cutoff

                        Outlier               0.0625      3.0000
```

The reason for the difference between the traditional ANOVA and the robust ANOVA is explained by Output 62.2.5, which shows that the fourth observation is an outlier. Further investigation shows that the original value of 24.4 for the fourth observation was recorded incorrectly.

Output 62.2.6 displays the robust test results. The interaction between the two treatments is not significant. Output 62.2.7 displays the robust residuals and standardized robust residuals.

Output 62.2.6. Test of Significance

```
                                 The ROBUSTREG Procedure

                                   Robust Linear Tests

                                          T1_T2

                                Test                    Chi-
                  Test     Statistic   Lambda  DF   Square  Pr > ChiSq

                  Rho         0.0041   0.7977   1     0.01      0.9431
                  Rn2         0.0041            1     0.00      0.9490
```

Output 62.2.7. ROBUSTREG Output

```
                   Obs    T1     T2     time       resid      stdres

                     1     0      0     20.2     -1.7974    -0.50851
                     2     0      0     23.9      1.9026     0.53827
                     3     0      0     21.9     -0.0974    -0.02756
                     4     0      0     42.4     20.4026     5.77222
                     5     1      0     27.2     -1.8900    -0.53472
                     6     1      0     34.0      4.9100     1.38911
                     7     1      0     27.4     -1.6900    -0.47813
                     8     1      0     28.5     -0.5900    -0.16693
                     9     0      1     25.9     -4.0348    -1.14152
                    10     0      1     34.5      4.5652     1.29156
                    11     0      1     25.1     -4.8348    -1.36785
                    12     0      1     34.2      4.2652     1.20668
                    13     1      1     35.0     -1.7655    -0.49950
                    14     1      1     33.9     -2.8655    -0.81070
                    15     1      1     38.3      1.5345     0.43413
                    16     1      1     39.9      3.1345     0.88679
```

Example 62.3. Growth Study of De Long and Summers

Robust regression and outlier detection techniques have considerable applications to econometrics. The following example from Zaman, Rousseeuw, and Orhan (2001) shows how these techniques substantially improve the ordinary least squares (OLS) results for the growth study of De Long and Summers.

De Long and Summers (1991) studied the national growth of 61 countries from 1960 to 1985 using OLS.

```
data growth;
   input country$ GDP LFG EQP NEQ GAP @@;
   datalines;
Argentin  0.0089 0.0118 0.0214 0.2286 0.6079
Austria   0.0332 0.0014 0.0991 0.1349 0.5809
Belgium   0.0256 0.0061 0.0684 0.1653 0.4109
Bolivia   0.0124 0.0209 0.0167 0.1133 0.8634
Botswana  0.0676 0.0239 0.1310 0.1490 0.9474
Brazil    0.0437 0.0306 0.0646 0.1588 0.8498
Cameroon  0.0458 0.0169 0.0415 0.0885 0.9333
Canada    0.0169 0.0261 0.0771 0.1529 0.1783
Chile     0.0021 0.0216 0.0154 0.2846 0.5402
Colombia  0.0239 0.0266 0.0229 0.1553 0.7695
CostaRic  0.0121 0.0354 0.0433 0.1067 0.7043
Denmark   0.0187 0.0115 0.0688 0.1834 0.4079
Dominica  0.0199 0.0280 0.0321 0.1379 0.8293
Ecuador   0.0283 0.0274 0.0303 0.2097 0.8205
ElSalvad  0.0046 0.0316 0.0223 0.0577 0.8414
Ethiopia  0.0094 0.0206 0.0212 0.0288 0.9805
Finland   0.0301 0.0083 0.1206 0.2494 0.5589
France    0.0292 0.0089 0.0879 0.1767 0.4708
Germany   0.0259 0.0047 0.0890 0.1885 0.4585
Greece    0.0446 0.0044 0.0655 0.2245 0.7924
Guatemal  0.0149 0.0242 0.0384 0.0516 0.7885
Honduras  0.0148 0.0303 0.0446 0.0954 0.8850
HongKong  0.0484 0.0359 0.0767 0.1233 0.7471
India     0.0115 0.0170 0.0278 0.1448 0.9356
Indonesi  0.0345 0.0213 0.0221 0.1179 0.9243
Ireland   0.0288 0.0081 0.0814 0.1879 0.6457
Israel    0.0452 0.0305 0.1112 0.1788 0.6816
Italy     0.0362 0.0038 0.0683 0.1790 0.5441
IvoryCoa  0.0278 0.0274 0.0243 0.0957 0.9207
Jamaica   0.0055 0.0201 0.0609 0.1455 0.8229
Japan     0.0535 0.0117 0.1223 0.2464 0.7484
Kenya     0.0146 0.0346 0.0462 0.1268 0.9415
Korea     0.0479 0.0282 0.0557 0.1842 0.8807
Luxembou  0.0236 0.0064 0.0711 0.1944 0.2863
Madagasc -0.0102 0.0203 0.0219 0.0481 0.9217
Malawi    0.0153 0.0226 0.0361 0.0935 0.9628
Malaysia  0.0332 0.0316 0.0446 0.1878 0.7853
Mali      0.0044 0.0184 0.0433 0.0267 0.9478
Mexico    0.0198 0.0349 0.0273 0.1687 0.5921
Morocco   0.0243 0.0281 0.0260 0.0540 0.8405
Netherla  0.0231 0.0146 0.0778 0.1781 0.3605
```

```
Nigeria  -0.0047 0.0283 0.0358 0.0842 0.8579
Norway    0.0260 0.0150 0.0701 0.2199 0.3755
Pakistan  0.0295 0.0258 0.0263 0.0880 0.9180
Panama    0.0295 0.0279 0.0388 0.2212 0.8015
Paraguay  0.0261 0.0299 0.0189 0.1011 0.8458
Peru      0.0107 0.0271 0.0267 0.0933 0.7406
Philippi  0.0179 0.0253 0.0445 0.0974 0.8747
Portugal  0.0318 0.0118 0.0729 0.1571 0.8033
Senegal  -0.0011 0.0274 0.0193 0.0807 0.8884
Spain     0.0373 0.0069 0.0397 0.1305 0.6613
SriLanka  0.0137 0.0207 0.0138 0.1352 0.8555
Tanzania  0.0184 0.0276 0.0860 0.0940 0.9762
Thailand  0.0341 0.0278 0.0395 0.1412 0.9174
Tunisia   0.0279 0.0256 0.0428 0.0972 0.7838
U.K.      0.0189 0.0048 0.0694 0.1132 0.4307
U.S.      0.0133 0.0189 0.0762 0.1356 0.0000
Uruguay   0.0041 0.0052 0.0155 0.1154 0.5782
Venezuel  0.0120 0.0378 0.0340 0.0760 0.4974
Zambia   -0.0110 0.0275 0.0702 0.2012 0.8695
Zimbabwe  0.0110 0.0309 0.0843 0.1257 0.8875
;
```

The regression equation they used is:

$$GDP = \beta_0 + \beta_1 LFG + \beta_2 GAP + \beta_3 EQP + \beta_4 NEQ + \epsilon,$$

where the response variable is the growth in gross domestic product per worker (GDP) and the regressors are labor force growth (LFG), relative GDP gap (GAP), equipment investment (EQP), and non-equipment investment (NEQ).

The following statements invoke the REG procedure for the OLS analysis:

```
proc reg data=growth;
   model GDP  = LFG GAP EQP NEQ ;
run;
```

Output 62.3.1. OLS Estimates

```
                         The REG Procedure
                           Model: MODEL1
                      Dependent Variable: GDP

                        Parameter Estimates

                         Parameter       Standard
Variable      DF         Estimate          Error    t Value    Pr > |t|

Intercept      1         -0.01430        0.01028      -1.39      0.1697
LFG            1         -0.02981        0.19838      -0.15      0.8811
GAP            1          0.02026        0.00917       2.21      0.0313
EQP            1          0.26538        0.06529       4.06      0.0002
NEQ            1          0.06236        0.03482       1.79      0.0787
```

The OLS analysis of Output 62.3.1 indicates that GAP and EQP have a significant influence on GDP at the 5% level.

The following statements invoke the ROBUSTREG procedure with the default M estimation.

```
proc robustreg data=growth;
   model GDP  = LFG GAP EQP NEQ / diagnostics leverage;
   output out=robout r=resid sr=stdres;
run;
```

Output 62.3.2. Model Fitting Information and Summary Statistics

```
                               The ROBUSTREG Procedure

                                  Model Information

                       Data Set                     MYLIB.GROWTH
                       Dependent Variable                    GDP
                       Number of Covariates                    4
                       Number of Observations                 61
                       Method                       M Estimation


                                  Summary Statistics

                                                                  Standard
        Variable           Q1      Median          Q3        Mean    Deviation          MAD

        LFG            0.0118      0.0239      0.0281      0.0211      0.00979      0.00949
        GAP            0.5796      0.8015      0.8863      0.7258       0.2181       0.1778
        EQP            0.0265      0.0433      0.0720      0.0523       0.0296       0.0325
        NEQ            0.0956      0.1356      0.1812      0.1399       0.0570       0.0624
        GDP            0.0121      0.0231      0.0310      0.0224       0.0155       0.0150
```

Output 62.3.2 displays model information and summary statistics for variables in the model.

Output 62.3.3. M estimates

```
                               The ROBUSTREG Procedure

                                Parameter Estimates

                               Standard   95% Confidence     Chi-
               Parameter DF Estimate    Error        Limits       Square Pr > ChiSq

               Intercept  1  -0.0247   0.0097  -0.0437  -0.0058    6.53     0.0106
               LFG        1   0.1040   0.1867  -0.2619   0.4699    0.31     0.5775
               GAP        1   0.0250   0.0086   0.0080   0.0419    8.36     0.0038
               EQP        1   0.2968   0.0614   0.1764   0.4172   23.33     <.0001
               NEQ        1   0.0885   0.0328   0.0242   0.1527    7.29     0.0069
               Scale      1   0.0099
```

Output 62.3.4. Diagnostics

```
                              The ROBUSTREG Procedure

                                    Diagnostics

                             Robust                          Standardized
               Mahalanobis      MCD                                Robust
       Obs        Distance  Distance     Leverage                Residual      Outlier

         1          2.6083    4.0639         *                    -0.9424
         5          3.4351    6.7391         *                     1.4200
         8          3.1876    4.6843         *                    -0.1972
         9          3.6752    5.0599         *                    -1.8784
        17          2.6024    3.8186         *                    -1.7971
        23          2.1225    3.8238         *                     1.7161
        27          2.6461    5.0336         *                     0.0909
        31          2.9179    4.7140         *                     0.0216
        53          2.2600    4.3193         *                    -1.8082
        57          3.8701    5.4874         *                     0.1448
        58          2.5953    3.9671         *                    -0.0978
        59          2.9239    4.1663         *                     0.3573
        60          1.8562    2.7135                              -4.9798         *
        61          1.9634    3.9128         *                    -2.5959


                                Diagnostics Summary

                          Observation
                          Type             Proportion       Cutoff

                          Outlier              0.0164       3.0000
                          Leverage             0.2131       3.3382
```

Output 62.3.5. Goodness-of-Fit

```
                              The ROBUSTREG Procedure

                                  Goodness-of-Fit

                                Statistic        Value

                                R-Square        0.3178
                                AICR           80.2134
                                BICR           91.5095
                                Deviance        0.0070
```

Output 62.3.3 displays the M estimates. Besides GAP and EQP, the robust analysis also indicates that NEQ is significant. This new finding is explained by Output 62.3.4, which shows that Zambia, the sixtieth country in the data, is an outlier. Output 62.3.4 also identifies leverage points based the robust MCD distances; however, there are no serious high leverage points in this data set. Output 62.3.5 displays robust versions of goodness-of-fit statistics for the model.

The following statements invoke the ROBUSTREG procedure with LTS estimation, which was used by Zaman, Rousseeuw, and Orhan (2001). The results are consistent with those of M estimation.

```
proc robustreg method=lts(h=33) fwls data=growth;
   model GDP  = LFG GAP EQP NEQ / diagnostics leverage ;
   output out=robout r=resid sr=stdres;
run;
```

Output 62.3.6. LTS estimates

```
                         The ROBUSTREG Procedure

                               LTS Profile

              Total Number of Observations                61
              Number of Squares Minimized                 33
              Number of Coefficients                       5
              Highest Possible Breakdown Value        0.4590


                         LTS Parameter Estimates

                      Parameter         DF    Estimate

                      Intercept          1     -0.0249
                      LFG                1      0.1123
                      GAP                1      0.0214
                      EQP                1      0.2669
                      NEQ                1      0.1110
                      Scale (sLTS)       0      0.0076
                      Scale (Wscale)     0      0.0109
```

Output 62.3.6 displays the LTS estimates.

Output 62.3.7. Diagnostics and LTS-Rsquare

```
                         The ROBUSTREG Procedure

                               Diagnostics

                             Robust                    Standardized
             Mahalanobis        MCD                          Robust
     Obs        Distance   Distance      Leverage          Residual      Outlier

       1          2.6083     4.0639         *               -1.0715
       5          3.4351     6.7391         *                1.6574
       8          3.1876     4.6843         *               -0.2324
       9          3.6752     5.0599         *               -2.0896
      17          2.6024     3.8186         *               -1.6367
      23          2.1225     3.8238         *                1.7570
      27          2.6461     5.0336         *                0.2334
      31          2.9179     4.7140         *                0.0971
      53          2.2600     4.3193         *               -1.2978
      57          3.8701     5.4874         *                0.0605
      58          2.5953     3.9671         *               -0.0857
      59          2.9239     4.1663         *                0.4113
      60          1.8562     2.7135                         -4.4984         *
      61          1.9634     3.9128         *               -2.1201


                           Diagnostics Summary

                   Observation
                   Type           Proportion      Cutoff

                   Outlier            0.0164      3.0000
                   Leverage           0.2131      3.3382


                             R-Square for LTS
                                Estimation

                           R-Square      0.7418
```

Output 62.3.7 displays outlier and leverage point diagnostics based on the LTS estimates.

Output 62.3.8. Final Weighted LS estimates

```
                        The ROBUSTREG Procedure

        Parameter Estimates for Final Weighted Least Squares Fit

                           Standard   95% Confidence     Chi-
Parameter      DF Estimate    Error       Limits       Square Pr > ChiSq

Intercept       1  -0.0222   0.0093  -0.0405  -0.0039    5.65     0.0175
LFG             1   0.0446   0.1771  -0.3026   0.3917    0.06     0.8013
GAP             1   0.0245   0.0082   0.0084   0.0406    8.89     0.0029
EQP             1   0.2824   0.0581   0.1685   0.3964   23.60     <.0001
NEQ             1   0.0849   0.0314   0.0233   0.1465    7.30     0.0069
Scale           0   0.0116
```

Output 62.3.8 displays the final weighted lease squares estimates, which are identical to those reported in Zaman, Rousseeuw, and Orhan (2001).

References

Akaike, H. (1974), “A New Look at the Statistical Identification Model,” *IEEE Trans. Automat Control*, 19, 716–723.

Brownlee, K.A. (1965), *Statistical Theory and Methodology in Science and Engineering*, 2nd ed., New York: John Wiley & Sons, Inc.

Chen C. (2002), “Robust Regression and Outlier Detection with the ROBUSTREG Procedure,” *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Coleman, D. Holland, P., Kaden, N., Klema, V., and Peters, S.C. (1980), “A System of Subroutines for Iteratively Reweighted Least-Squares Computations,” *ACM Transactions on Mathematical Software*, 6, 327-336.

De Long, J.B., Summers, L.H. (1991), “Equipment Investment and Economic Growth,” *Quarterly Journal of Economics*, 106, 445-501.

Hampel, F. R., Ronchetti, E.M., Rousseeuw, P.J. and Stahel, W.A. (1986), *Robust Statistics, The Approach Based on Influence Functions*, New York: John Wiley & Sons, Inc.

Hawkins, D.M., Bradu, D. and Kass, G.V. (1984), “Location of Several Outliers in Multiple Regression Data Using Elemental Sets,” *Technometrics*, 26, 197-208.

Holland, P. and Welsch, R. (1977), “Robust Regression Using Interactively Reweighted Least-Squares,” *Commun. Statist. Theor. Meth.*, 6, 813-827.

Huber, P.J. (1973), “Robust Regression: Asymptotics, Conjectures and Monte Carlo,” *Ann. Stat.*, 1, 799-821.

Huber, P.J. (1981), *Robust Statistics*, New York: John Wiley & Sons, Inc.

Marazzi, A. (1993), *Algorithm, Routines, and S Functions for Robust Statistics*, Pacific Grove, CA: Wadsworth & Brooks / Cole.

Ronchetti, E. (1985), “Robust Model Selection in Regression,” *Statistics and Probability Letters*, 3, 21-23.

Rousseeuw, P.J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871-880.

Rousseeuw, P.J. and Hubert, M. (1996), "Recent Development in PROGRESS," *Computational Statistics and Data Analysis*, 21, 67-85.

Rousseeuw, P.J. and Leroy, A.M. (1987), *Robust Regression and Outlier Detection*, New York: John Wiley & Sons, Inc.

Rousseeuw, P.J. and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212-223.

Rousseeuw, P.J. and Van Driessen, K. (2000), "An Algorithm for Positive-Breakdown Regression Based on Concentration Steps," *Data Analysis: Scientific Modeling and Practical Application*, ed. W. Gaul, O. Opitz, and M. Schader. New York: Springer-Verlag, 335–346.

Rousseeuw, P.J. and Yohai, V. (1984), "Robust Regression by Means of S Estimators," in *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R.D. Martin, Lecture Notes in Statistics, 26, New York: Springer-Verlag, 256-274.

Ruppert, D. (1992), "Computing S Estimators for Regression and Multivariate Location/Dispersion," *Journal of Computational and Graphical Statistics*, 1, 253-270.

Yohai V.J. (1987), "High Breakdown Point and High Efficiency Robust Estimates for Regression," *Annals of Statistics*, 15, 642-656.

Yohai V.J., Stahel, W.A. and Zamar, R.H. (1991), "A Procedure for Robust Estimation and Inference in Linear Regression," in Stahel, W.A. and Weisberg, S.W., eds., *Directions in Robust Statistics and Diagnostics, Part II*, New York: Springer-Verlag.

Yohai, V.J. and Zamar, R.H. (1997), "Optimal Locally Robust M Estimate of Regression," *Journal of Statist. Planning and Inference*, 64, 309-323.

Zaman, A., Rousseeuw, P.J., Orhan, M. (2001), "Econometric Applications of High-Breakdown Robust Regression Techniques," *Econometrics Letters*, 71, 1-8.

Chapter 63
The RSREG Procedure

Chapter Contents

Chapter 63
The RSREG Procedure

Overview

The RSREG procedure uses the method of least squares to fit quadratic response surface regression models. Response surface models are a kind of general linear model in which attention focuses on characteristics of the fit response function and in particular, where optimum estimated response values occur.

In addition to fitting a quadratic function, you can use the RSREG procedure to

- test for lack of fit
- test for the significance of individual factors
- analyze the canonical structure of the estimated response surface
- compute the ridge of optimum response
- predict new values of the response

Comparison to Other SAS Software

Other SAS/STAT procedures can be used to fit the response surface, but the RSREG procedure is more specialized. The following statements model a three-factor response surface in PROC RSREG:

```
proc rsreg;
   model y=x1 x2 x3;
run;
```

These statements are more compact than the statements for other regression procedures in SAS/STAT software. For example, the equivalent statements for the GLM procedure are

```
proc glm;
   model y=x1 x1*x1
           x2 x1*x2 x2*x2
           x3 x1*x3 x2*x3 x3*x3;
run;
```

Additionally, PROC RSREG includes specialized methodology for analyzing the fitted response surface, such as canonical analysis and optimum response ridges.

Note that the ADX Interface in SAS/QC software provides an *interactive* environment for constructing and analyzing many different kinds of experiments, including response surface experiments. The ADX Interface is the preferred interactive SAS

System tool for analyzing experiments, since it includes facilities for checking underlying assumptions and graphically optimizing the response surface. The RSREG procedure is appropriate for analyzing experiments in a batch environment.

Terminology

Variables are referred to according to the following conventions:

factor variables — independent variables used in constructing the quadratic response surface. To estimate the necessary parameters, each variable must have at least three distinct values in the data. Independent variables must be numeric.

response variables — the dependent variables to which the quadratic response surface is fit. Dependent variables must be numeric.

covariates — additional independent variables for use in the regression but not in the formation of the quadratic response surface. Covariates must be numeric.

WEIGHT variable — a variable for weighting the observations in the regression. The WEIGHT variable must be numeric.

ID variables — variables not in the above lists that are transferred to an output data set containing statistics for each observation in the input data set. This data set is created using the OUT= option in the PROC RSREG statement. ID variables can be either character or numeric.

BY variables — variables for grouping observations. Separate analyses are obtained for each BY group. BY variables can be either character or numeric.

Getting Started

A Response Surface with a Simple Optimum

This example uses the three-factor quadratic model discussed in John (1971). Schneider and Stockett (1963) performed an experiment aimed at reducing the unpleasant odor of a chemical produced with several factors. The objective is to minimize the unpleasant odor of a chemical. The following statements read the data.

```
title 'Response Surface with a Simple Optimum';
data smell;
   input Odor T R H @@;
   label
      T = "Temperature"
      R = "Gas-Liquid Ratio"
      H = "Packing Height";
   datalines;
 66 40 .3 4     39 120 .3 4     43 40 .7 4     49 120 .7  4
```

```
    58 40 .5 2      17 120 .5 2      -5 40 .5 6     -40 120 .5  6
    65 80 .3 2       7  80 .7 2      43 80 .3 6     -22  80 .7  6
   -31 80 .5 4     -35  80 .5 4     -26 80 .5 4
   ;
```

The INPUT statement names the variables contained in the SAS data set smell; the variable Odor is the response, while the variables T, R, and H are the independent factors.

The following statements invoke PROC RSREG on the data set smell. Figure 63.1 through Figure 63.3 display the results of the analysis, including a lack-of-fit test requested with the LACKFIT option.

```
   proc rsreg data=smell;
      model Odor = T R H / lackfit;
   run;
```

```
                      Response Surface with a Simple Optimum

                               The RSREG Procedure

                 Coding Coefficients for the Independent Variables

                     Factor    Subtracted off      Divided by

                     T              80.000000       40.000000
                     R               0.500000        0.200000
                     H               4.000000        2.000000


                        Response Surface for Variable Odor

                     Response Mean                  15.200000
                     Root MSE                       22.478508
                     R-Square                          0.8820
                     Coefficient of Variation        147.8849


                                     Type I Sum
     Regression           DF      of Squares    R-Square    F Value    Pr > F

     Linear                3     7143.250000      0.3337       4.71    0.0641
     Quadratic             3           11445      0.5346       7.55    0.0264
     Crossproduct          3      293.500000      0.0137       0.19    0.8965
     Total Model           9           18882      0.8820       4.15    0.0657


                                       Sum of
    Residual              DF          Squares     Mean Square    F Value    Pr > F

    Lack of Fit            3      2485.750000      828.583333      40.75    0.0240
    Pure Error             2        40.666667       20.333333
    Total Error            5      2526.416667      505.283333
```

Figure 63.1. Summary Statistics and Analysis of Variance

Figure 63.1 displays the coding coefficients for the transformation of the independent variables to lie between -1 and 1, simple statistics for the response variable, hypothesis tests for linear, quadratic, and crossproduct terms, and the lack-of-fit test. The hypothesis tests can be used to gain a rough idea of importance of the effects; here the crossproduct terms are not significant. However, the lack-of-fit for the model is significant, so more complicated modeling or further experimentation with additional variables should be performed before firm statements are made concerning the underlying process.

```
                      Response Surface with a Simple Optimum

                               The RSREG Procedure

                                                                      Parameter
                                                                       Estimate
                                            Standard                 from Coded
Parameter   DF       Estimate                  Error   t Value   Pr > |t|          Data

Intercept    1     568.958333             134.609816      4.23     0.0083    -30.666667
T            1      -4.102083               1.489024     -2.75     0.0401    -12.125000
R            1   -1345.833333             335.220685     -4.01     0.0102    -17.000000
H            1     -22.166667              29.780489     -0.74     0.4902    -21.375000
T*T          1       0.020052               0.007311      2.74     0.0407     32.083333
R*T          1       1.031250               1.404907      0.73     0.4959      8.250000
R*R          1    1195.833333             292.454665      4.09     0.0095     47.833333
H*T          1       0.018750               0.140491      0.13     0.8990      1.500000
H*R          1      -4.375000              28.098135     -0.16     0.8824     -1.750000
H*H          1       1.520833               2.924547      0.52     0.6252      6.083333


                      Sum of
Factor     DF        Squares    Mean Square    F Value    Pr > F    Label

T           4    5258.016026    1314.504006       2.60    0.1613    Temperature
R           4          11045    2761.150641       5.46    0.0454    Gas-Liquid Ratio
H           4    3813.016026     953.254006       1.89    0.2510    Packing Height
```

Figure 63.2. Parameter Estimates and Hypothesis Tests

Parameter estimates and the factor ANOVA are shown in Figure 63.2. Looking at the parameter estimates, you can see that the crossproduct terms are not significantly different from zero, as noted previously. The “Estimate” column contains estimates based on the raw data, and the “Parameter Estimate from Coded Data” column contains those based on the coded data. The factor ANOVA table displays tests for all four parameters corresponding to each factor—the parameters corresponding to the linear effect, the quadratic effect, and the effects of the cross products with each of the other two factors. The only factor with a significant over-all effect is R, indicating that the level of noise left unexplained by the model is still too high to estimate the effects of T and H accurately. This may be due to the lack of fit.

```
                    Response Surface with a Simple Optimum

                             The RSREG Procedure
           Canonical Analysis of Response Surface Based on Coded Data

                         Critical Value
           Factor            Coded          Uncoded    Label

           T              0.121913       84.876502    Temperature
           R              0.199575        0.539915    Gas-Liquid Ratio
           H              1.770525        7.541050    Packing Height

                Predicted value at stationary point: -52.024631


                                       Eigenvectors
           Eigenvalues               T               R               H

             48.858807        0.238091        0.971116       -0.015690
             31.103461        0.970696       -0.237384        0.037399
              6.037732       -0.032594        0.024135        0.999177

                         Stationary point is a minimum.
```

Figure 63.3. Canonical Analysis and Eigenvectors

Figure 63.3 contains the canonical analysis and eigenvectors. The canonical analysis indicates that the directions of principle orientation for the predicted response surface are along the axes associated with the three factors, confirming the small interaction effect in the Regression ANOVA. The largest eigenvalue (48.8588) corresponds to the eigenvector $\{0.238091,\ 0.971116,\ -0.015690\}$, the largest component of which (0.971116) is associated with R; similarly, the second largest eigenvalue (31.1035) is associated with T. The third eigenvalue (6.0377), associated with H, is quite a bit smaller than the other two, indicating that the response surface is relatively insensitive to changes in this factor. The coded form of the canonical analysis indicates that the estimated response surface is at a minimum when T and R are both near the middle of their respective ranges and H is relatively high; in uncoded, terms, the model predicts that the unpleasant odor will be minimized when $T = 84.876502$, $R = 0.539915$, and $H = 7.541050$.

To plot the response surface with respect to two of the factor variables, first fix H, the least significant factor variable, at its estimated optimum value and generate a grid of points for T and R. To ensure that the grid data do not affect parameter estimates, the response variable (Odor) is set to missing. (See the "Missing Values" section on page 4048.) The following statements produce and graph the necessary data. Initial data steps creates a grid over T and R, with H set to a constant value, and combine this grid with the original data. Then, PROC RSREG is used to create predictions for the combined data. Finally, PROC G3D is used to create a surface plot of the predictions.

```
data grid;
   do;
      Odor =  .  ;
      H    = 7.541;
      do T = 20 to 140 by 5;
         do R = .1 to .9 by .05;
            output;
         end;
      end;
   end;
data grid;
   set smell grid;
run;

proc rsreg data=grid out=predict noprint;
   model Odor = T R H / predict;
run;

data plot;
   set predict;
   if H = 7.541;
proc g3d data=plot;
   plot T*R=Odor / rotate=38 tilt=75 xticknum=3 yticknum=3
   zmax=300 zmin=-60 ctop=red cbottom=blue caxis=black;
run;
```

The first DATA step creates grid points for T and R at H=7.541 and sets Odor to missing, and the second DATA step concatenates these grid points with the original data. Predicted values are created in the SAS data set predict by invoking the RSREG procedure with the PREDICT option in the MODEL statement. The analysis is not displayed due to the NOPRINT option. The third DATA step subsets the predicted values over just the grid points (excluding the predictions at the original points). PROC G3D is then used to create the three-dimensional plot shown in Figure 63.4.

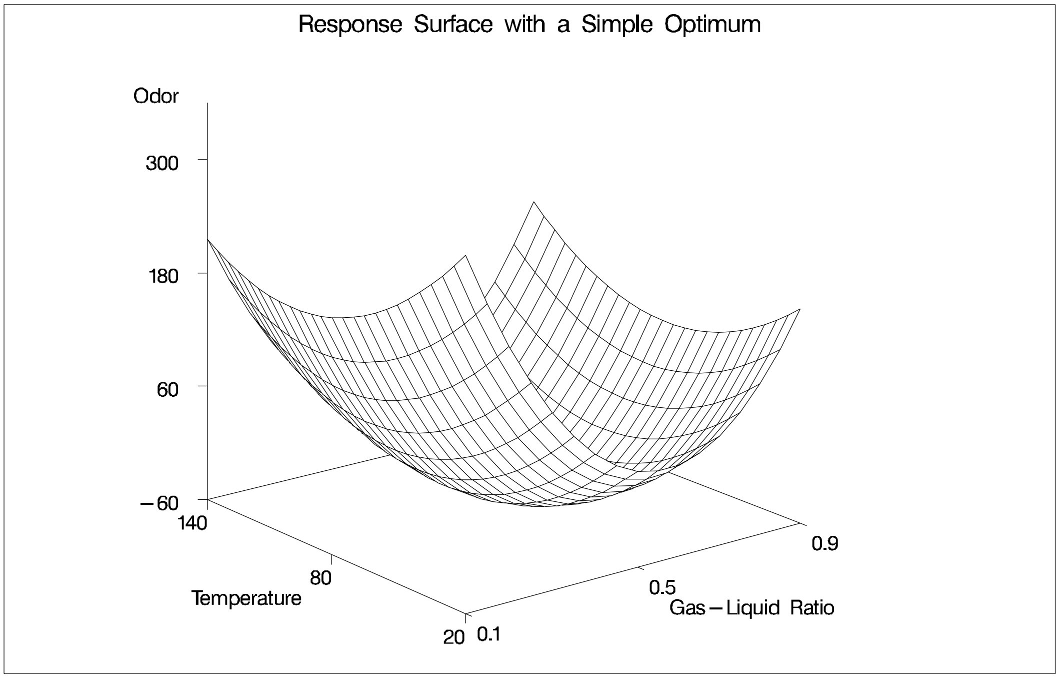


Figure 63.4. The Response Surface Obtained from the PREDICT Option

Syntax

The following statements are available in PROC RSREG.

PROC RSREG < *options* > **;**
  MODEL *responses= independents* < */ options* > **;**
  RIDGE < *options* > **;**
  WEIGHT *variable* **;**
  ID *variables* **;**
  BY *variables* **;**

The PROC RSREG and MODEL statements are required. The BY, ID, MODEL, RIDGE, and WEIGHT statements are described after the PROC RSREG statement, and they can appear in any order.

PROC RSREG Statement

PROC RSREG < *options* > **;**

The PROC RSREG statement invokes the procedure. You can specify the following options in the PROC RSREG statement.

DATA=*SAS-data-set*

specifies the input SAS data set that contains the data to be analyzed. By default, PROC RSREG uses the most recently created SAS data set.

NOPRINT

suppresses the normal display of results when only the output data set is required. For more information, see the description of the NOPRINT option in the "MODEL Statement" and "RIDGE Statement" sections. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 14, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*

creates an output SAS data set that contains statistics for each observation in the input data set. In particular, this data set contains the BY variables, the ID variables, the WEIGHT variable, the variables in the MODEL statement, and the output options requested in the MODEL statement. You must specify output options in the MODEL statement; otherwise, the output data set is created but contains no observations. To create a permanent SAS data set, you must specify a two-level name (refer to the discussion in *SAS Language Reference: Concepts* for more information on permanent SAS data sets). For details on the data set created by PROC RSREG, see the "Output Data Sets" section on page 4051.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC RSREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the RSREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement names variables that are to be transferred to the data set created by the OUT= option in the PROC RSREG statement.

MODEL Statement

MODEL *responses=independents < / options >* **;**

The MODEL statement lists response (dependent) variables followed by an equal sign and then lists independent variables, some of which may be covariates. The output options to the MODEL statement specify which statistics are output to the data set created using the OUT= option in the PROC RSREG statement. If none of the options are selected, the data set is created but contains no observations. The option keywords become values of the special variable _TYPE_ in the output data set. Any of the following options can be specified.

Task	**Options**
Analyze Original Data	NOCODE
Fit Model to First BY Group Only	BYOUT
Declare Covariates	COVAR=
Request Additional Statistics	PRESS
Request Additional Tests	LACKFIT
Suppress Displayed Output	NOANOVA NOOPTIMAL NOPRINT
Output Statistics	ACTUAL PREDICT RESIDUAL L95 U95 L95M U95M D

ACTUAL

specifies that the observed response values from the input data set be written to the output data set.

BYOUT

uses only the first BY group to estimate the model. Subsequent BY groups have scoring statistics computed in the output data set only. The BYOUT option is used only when a BY statement is specified.

COVAR=*n*

declares that the first n variables on the right-hand side of the model are simple linear regressors (covariates) and not factors in the quadratic response surface. By default, PROC RSREG forms quadratic and crossproduct effects for all regressor variables in the MODEL statement. See the "Handling Covariates" section on page 4050 for more details and Example 63.2 on page 4059 for an example using covariates.

D

specifies that Cook's D influence statistic be written to the output data set. See Chapter 2, "Introduction to Regression Procedures," for details and formulas.

LACKFIT

performs a lack-of-fit test. Refer to Draper and Smith (1981) for a discussion of lack-of-fit tests.

L95

specifies that the lower bound of a 95% confidence interval for an individual predicted value be written to the output data set. The variance used in calculating this bound is a function of both the mean square error and the variance of the parameter estimates. See Chapter 2 for details and formulas.

L95M

specifies that the lower bound of a 95% confidence interval for the expected value of the dependent variable be written to the output data set. The variance used in calculating this bound is a function of the variance of the parameter estimates. See Chapter 2 for details and formulas.

NOANOVA
NOAOV

suppresses the display of the analysis of variance and parameter estimates from the model fit.

NOCODE

performs the canonical and ridge analyses with the parameter estimates derived from fitting the response to the original values of the factors variables, rather than their coded values (see the "Coding the Factor Variables" section on page 4047 for more details.) Use this option if the data are already stored in a coded form.

NOOPTIMAL
NOOPT

suppresses the display of the canonical analysis for the quadratic response surface.

NOPRINT

suppresses the display of both the analysis of variance and the canonical analysis.

PREDICT

specifies that the values predicted by the model be written to the output data set.

PRESS

computes and displays the predicted residual sum of squares (PRESS) statistic for each dependent variable in the model. The PRESS statistic is added to the summary information at the beginning of the analysis of variance, so if the NOANOVA or

NOPRINT option is specified, PRESS has no effect. See Chapter 2 for details and formulas.

RESIDUAL

specifies that the residuals, calculated as ACTUAL − PREDICTED, be written to the output data set.

U95

specifies that the upper bound of a 95% confidence interval for an individual predicted value be written to the output data set. The variance used in calculating this bound is a function of both the mean square error and the variance of the parameter estimates. See Chapter 2 for details and formulas.

U95M

specifies that the upper bound of a 95% confidence interval for the expected value of the dependent variable be written to the output data set. The variance used in calculating this bound is a function of the variance of the parameter estimates. See Chapter 2 for details and formulas.

RIDGE Statement

RIDGE $<$ *options* $>$ **;**

A RIDGE statement computes the ridge of optimum response. The ridge starts at a given point $\mathbf{x}_0$, and the point on the ridge at radius r from $\mathbf{x}_0$ is the collection of factor settings that optimizes the predicted response at this radius. You can think of the ridge as climbing or falling as fast as possible on the surface of predicted response. Thus, the ridge analysis can be used as a tool to help interpret an existing response surface or to indicate the direction in which further experimentation should be performed.

The default starting point, $\mathbf{x}_0$, has each coordinate equal to the point midway between the highest and lowest values of the factor in the design. The default radii at which the ridge is computed are 0, 0.1, . . . , 0.9, 1. If, as usual, the ridge analysis is based on the response surface fit to coded values for the factor variables (see the “Coding the Factor Variables” section on page 4047 for details), then this results in a ridge that starts at the point with a coded zero value for each coordinate and extends toward, but not beyond, the edge of the range of experimentation. Alternatively, both the center point for the ridge and the radii at which it is to be computed can be specified.

You can specify the following options in the RIDGE statement:

CENTER=*uncoded-factor-values*

gives the coordinates of the point $\mathbf{x}_0$ from which to begin the ridge. The coordinates should be given in the original (uncoded) factor variable values and should be separated by commas. There must be as many coordinates specified as there are factors in the model, and the order of the coordinates must be the same as that used in the MODEL statement. This starting point should be well inside the range of experimentation. The default sets each coordinate equal to the value midway between the highest and lowest values for the associated factor.

MAXIMUM
MAX

computes the ridge of maximum response. Both the MIN and MAX options can be specified; at least one must be specified.

MINIMUM
MIN

computes the ridge of minimum response. Both the MIN and MAX options can be specified; at least one must be specified.

NOPRINT

suppresses the display of the ridge analysis when only an output data set is required.

OUTR=*SAS-data-set*

creates an output SAS data set containing the computed optimum ridge. For details, see the "Output Data Sets" section on page 4051.

RADIUS=*coded-radii*

gives the distances from the ridge starting point at which to compute the optimum. The values in the list represent distances between coded points. The list can take any of the following forms or can be composed of mixtures of them:

$m_1, m_2, \ldots, m_n$	several values
m TO n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m TO n BY i	a sequence where m equals the starting value, n equals the ending value, and i equals the increment

Mixtures of the preceding forms should be separated by commas. The default list runs from 0 to 1 by increments of 0.1. The following are examples of valid lists.

```
radius=0 to 5 by .5;
radius=0, .2, .25, .3, .5 to 1.0 by .1;
```

WEIGHT Statement

WEIGHT *variable* **;**

When a WEIGHT statement is used, a weighted residual sum of squares

$$\sum_i w_i(y_i - \hat{y}_i)^2$$

is minimized, where w_i is the value of the variable specified in the WEIGHT statement, y_i is the observed value of the response variable, and $\hat{y}_i$ is the predicted value of the response variable.

The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than zero. The WEIGHT statement has no effect on degrees of freedom or number of observations. If the weights for the observations are proportional to the reciprocals of the error variances, then the weighted least-squares estimates are best linear unbiased estimators (BLUE).

Details

Introduction to Response Surface Experiments

Many industrial experiments are conducted to discover which values of given factor variables optimize a response. If each factor is measured at three or more values, a quadratic response surface can be estimated by least-squares regression. The predicted optimal value can be found from the estimated surface if the surface is shaped like a simple hill or a valley. If the estimated surface is more complicated, or if the predicted optimum is far from the region of experimentation, then the shape of the surface can be analyzed to indicate the directions in which new experiments should be performed.

Suppose that a response variable y is measured at combinations of values of two factor variables, x_1 and x_2. The quadratic response-surface model for this variable is written as

$$y \quad = \quad \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon$$

The steps in the analysis for such data are

1. model fitting and analysis of variance to estimate parameters
2. canonical analysis to investigate the shape of the predicted response surface
3. ridge analysis to search for the region of optimum response

Model Fitting and Analysis of Variance

The first task in analyzing the response surface is to estimate the parameters of the model by least-squares regression and to obtain information about the fit in the form of an analysis of variance. The estimated surface is typically curved: a “hill” whose peak occurs at the unique estimated point of maximum response, a “valley,” or a “saddle-surface” with no unique minimum or maximum. Use the results of this phase of the analysis to answer the following questions:

- What is the contribution of each type of effect—linear, quadratic, and crossproduct—to the statistical fit? The ANOVA table with sources labeled “Regression” addresses this question.
- What part of the residual error is due to lack of fit? Does the quadratic response model adequately represent the true response surface? If you specify the LACKFIT option in the MODEL statement, then the ANOVA table with sources labeled “Residual” addresses this question.
- What is the contribution of each factor variable to the statistical fit? Can the response be predicted as well if the variable is removed? The ANOVA table with sources labeled “Factor” addresses this question.
- What are the predicted responses for a grid of factor values? (See the section “Plotting the Surface” on page 4048 and the “Searching for Multiple Response Conditions” section on page 4048.)

Lack-of-Fit Test

The test for lack-of-fit compares the variation around the model with "pure" variation within replicated observations. This measures the adequacy of the quadratic response surface model. In particular, if there are n_i replicated observations $Y_{i1}, \ldots, Y_{in_i}$ of the response all at the same values $\mathbf{x}_i$ of the factors, then we can predict the true response at $\mathbf{x}_i$ either by using the predicted value $\hat{Y}_i$ based on the model or by using the mean $\bar{Y}_i$ of the replicated values. The test for lack-of-fit decomposes the residual error into a component due to the variation of the replications around their mean value (the "pure" error), and a component due to the variation of the mean values around the model prediction (the "bias" error):

$$\sum_i \sum_{j=1}^{n_i} \left(Y_{ij} - \hat{Y}_i\right)^2 \;=\; \sum_i \sum_{j=1}^{n_i} (Y_{ij} - \bar{Y}_i)^2 + \sum_i n_i \left(\bar{Y}_i - \hat{Y}_i\right)^2$$

If the model is adequate, then both components estimate the nominal level of error; however, if the bias component of error is much larger than the pure error, then this constitutes evidence that there is significant lack of fit.

If some observations in your design are replicated, you can test for lack of fit by specifying the LACKFIT option in the MODEL statement. Note that, since all other tests use total error rather than pure error, you may want to hand-calculate the tests with respect to pure error if the lack-of-fit is significant. On the other hand, significant lack-of-fit indicates the quadratic model is inadequate, so if this is a problem you can also try to refine the model, possibly using PROC GLM for general polynomial modeling; refer to Chapter 32, "The GLM Procedure," for more information. Example 63.1 on page 4055 illustrates the use of the LACKFIT option.

Canonical Analysis

The second task in analyzing the response surface is to examine the overall shape of the curve and determine whether the estimated stationary point is a maximum, a minimum, or a saddle point. The canonical analysis can be used to answer the following questions:

- Is the surface shaped like a hill, a valley, a saddle surface, or a flat surface?
- If there is a unique optimum combination of factor values, where is it?
- To which factor or factors are the predicted responses most sensitive?

The eigenvalues and eigenvectors in the matrix of second-order parameters characterize the shape of the response surface. The eigenvectors point in the directions of principle orientation for the surface, and the signs and magnitudes of the associated eigenvalues give the shape of the surface in these directions. Positive eigenvalues indicate directions of upward curvature, and negative eigenvalues indicate directions of downward curvature. The larger an eigenvalue is in absolute value, the more pronounced is the curvature of the response surface in the associated direction. Often, all of the coefficients of an eigenvector except for one are relatively small, indicating that the vector points roughly along the axis associated with the factor corresponding to the single large coefficient. In this case, the canonical analysis can be used to determine the relative sensitivity of the predicted response surface to variations in that factor. (See the "Getting Started" section on page 4034 for an example.)

Ridge Analysis

If the estimated surface is found to have a simple optimum well within the range of experimentation, the analysis performed by the preceding two steps may be sufficient. In more complicated situations, further search for the region of optimum response is required. The method of ridge analysis computes the estimated ridge of optimum response for increasing radii from the center of the original design. The ridge analysis answers the following question:

- If there is not a unique optimum of the response surface within the range of experimentation, in which direction should further searching be done in order to locate the optimum?

You can use the RIDGE statement to compute the ridge of maximum or minimum response.

Coding the Factor Variables

For the results of the canonical and ridge analyses to be interpretable, the values of different factor variables should be comparable. This is because the canonical and ridge analyses of the response surface are not invariant with respect to differences in scale and location of the factor variables. The analysis of variance is not affected by these changes. Although the actual predicted surface does not change, its parameterization does. The usual solution to this problem is to code each factor variable so that its minimum in the experiment is -1 and its maximum is 1 and to carry through the analysis with the coded values instead of the original ones. This practice has the added benefit of making 1 a reasonable boundary radius for the ridge analysis since 1 represents approximately the edge of the experimental region. By default, PROC RSREG computes the linear transformation to perform this coding as the data are initially read in, and the canonical and ridge analyses are performed on the model fit to the coded data. The actual form of the coding operation for each value of a variable is

$$\textit{coded value} \quad = \quad (\textit{original value} - M)/S$$

where M is the average of the highest and lowest values for the variable in the design and S is half their difference.

Missing Values

If an observation has missing data for any of the variables used by the procedure, then that observation is not used in the estimation process. If one or more response variables are missing, but no factor or covariate variables are missing, then predicted values and confidence limits are computed for the output data set, but the residual and Cook's D statistic are missing.

Plotting the Surface

You can generate predicted values for a grid of points with the PREDICT option (see the "Getting Started" section on page 4034 for an example) and then use these values to create a contour plot or a three-dimensional plot of the response surface over a two-dimensional grid. Any two factor variables can be chosen to form the grid for the plot. Several plots can be generated by using different pairs of factor variables.

Searching for Multiple Response Conditions

Suppose you want to find the factor setting that produces responses in a certain region. For example, you have the following data with two factors and three responses:

```
data a;
   input x1 x2 y1 y2 y3;
   datalines;
-1      -1         1.8 1.940  3.6398
-1       1         2.6 1.843  4.9123
 1      -1         5.4 1.063  6.0128
 1       1         0.7 1.639  2.3629
 0       0         8.5 0.134  9.0910
 0       0         3.0 0.545  3.7349
 0       0         9.8 0.453 10.4412
 0       0         4.1 1.117  5.0042
 0       0         4.8 1.690  6.6245
 0       0         5.9 1.165  6.9420
 0       0         7.3 1.013  8.7442
 0       0         9.3 1.179 10.2762
 1.4142  0         3.9 0.945  5.0245
-1.4142  0         1.7 0.333  2.4041
 0       1.4142    3.0 1.869  5.2695
 0      -1.4142    5.7 0.099  5.4346
;
```

You want to find the values of x1 and x2 that maximize y1 subject to y2<2 and y3<y2+y1. The exact answer is not easy to obtain analytically, but you can obtain a practically feasible solution by checking conditions across a grid of values in the range of interest. First, append a grid of factor values to the observed data, with missing values for the responses.

```
data b;
   set a end=eof;
   output;
   if eof then do;
      y1=.;
      y2=.;
      y3=.;
      do x1=-2 to 2 by .1;
         do x2=-2 to 2 by .1;
            output;
         end;
      end;
   end;
run;
```

Next, use PROC RSREG to fit a response surface model to the data and to compute predicted values for both the observed data and the grid, putting the predicted values in a data set c.

```
proc rsreg data=b out=c;
   model y1 y2 y3=x1 x2 / predict;
run;
```

Finally, find the subset of predicted values that satisfy the constraints, sort by the unconstrained variable, and display the top five predictions.

```
data d;
   set c;
   if y2<2;
   if y3<y2+y1;

proc sort data=d;
   by descending y1;
run;

data d; set d;
   i = _n_;
proc print;
   where (i <= 5);
run;
```

The final results are displayed in Figure 63.5. They indicate that optimal values of the factors are around 0.3 for x1 and around -0.5 for x2.

```
Obs    x1     x2    _TYPE_       y1        y2        y3     i

  1   0.3    -0.5   PREDICT   6.92570   0.75784   7.60471   1
  2   0.3    -0.6   PREDICT   6.91424   0.74174   7.54194   2
  3   0.3    -0.4   PREDICT   6.91003   0.77870   7.64341   3
  4   0.4    -0.6   PREDICT   6.90769   0.73357   7.51836   4
  5   0.4    -0.5   PREDICT   6.90540   0.75135   7.56883   5
```

Figure 63.5. Top Five Predictions

Handling Covariates

Covariate regressors are added to a response surface model because they are believed to account for a sizable yet relatively uninteresting portion of the variation in the data. What the experimenter is really interested in is the response corrected for the effect of the covariates. A common example is the block effect in a block design. In the canonical and ridge analyses of a response surface, which estimate responses at hypothetical levels of the factor variables, the actual value of the predicted response is computed using the average values of the covariates. The estimated response values do optimize the estimated surface of the response corrected for covariates, but true prediction of the response requires actual values for the covariates. You can use the COVAR= option in the MODEL statement to include covariates in the response surface model. Example 63.2 on page 4059 illustrates the use of this option.

Computational Method

Canonical Analysis

For each response variable, the model can be written in the form

$$y_i = \mathbf{x}_i'\mathbf{A}\mathbf{x}_i + \mathbf{b}'\mathbf{x}_i + \mathbf{c}'\mathbf{z}_i + \epsilon_i$$

where

y_i is the ith observation of the response variable.

$\mathbf{x}_i$ $= (x_{i1}, x_{i2}, \ldots, x_{ik})'$ are the k factor variables for the ith observation.

$\mathbf{z}_i$ $= (z_{i1}, z_{i2}, \ldots, z_{iL})'$ are the L covariates, including the intercept term.

$\mathbf{A}$ is the $k \times k$ symmetrized matrix of quadratic parameters, with diagonal elements equal to the coefficients of the pure quadratic terms in the model and off-diagonal elements equal to half the coefficient of the corresponding cross product.

$\mathbf{b}$ is the $k \times 1$ vector of linear parameters.

$\mathbf{c}$ is the $L \times 1$ vector of covariate parameters, one of which is the intercept.

ϵ_i is the error associated with the ith observation. Tests performed by PROC RSREG assume that errors are independently and normally distributed with mean zero and variance σ^2.

The parameters in $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ are estimated by least squares. To optimize $\mathbf{y}$ with respect to $\mathbf{x}$, take partial derivatives, set them to zero, and solve:

$$\frac{\partial y}{\partial \mathbf{x}} = 2\mathbf{x}'\mathbf{A} + \mathbf{b}' = \mathbf{0} \quad \Longrightarrow \quad \mathbf{x} = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{b}$$

You can determine if the solution is a maximum or minimum by looking at the eigenvalues of $\mathbf{A}$:

If the eigenvalues...	**then the solution is...**
are all negative	a maximum
are all positive	a minimum
have mixed signs	a saddle point
contain zeros	in a flat area

Ridge Analysis

The eigenvector for the largest eigenvalue gives the direction of steepest ascent from the stationary point, if positive, or steepest descent, if negative. The eigenvectors corresponding to small or zero eigenvalues point in directions of relative flatness.

The point on the optimum response ridge at a given radius R from the ridge origin is found by optimizing

$$(\mathbf{x}_0 + \mathbf{d})'\mathbf{A}(\mathbf{x}_0 + \mathbf{d}) + \mathbf{b}'(\mathbf{x}_0 + \mathbf{d})$$

over $\mathbf{d}$ satisfying $\mathbf{d}'\mathbf{d} = R^2$, where $\mathbf{x}_0$ is the $k \times 1$ vector containing the ridge origin and $\mathbf{A}$ and $\mathbf{b}$ are as previously discussed. By the method of Lagrange multipliers, the optimal $\mathbf{d}$ has the form

$$\mathbf{d} = -(\mathbf{A} - \mu\mathbf{I})^{-1}(\mathbf{A}\mathbf{x}_0 + 0.5\mathbf{b})$$

where $\mathbf{I}$ is the $k \times k$ identity matrix and μ is chosen so that $\mathbf{d}'\mathbf{d} = R^2$. There may be several values of μ that satisfy this constraint; the right one depends on which sort of response ridge is of interest. If you are searching for the ridge of maximum response, then the appropriate μ is the unique one that satisfies the constraint and is greater than all the eigenvalues of $\mathbf{A}$. Similarly, the appropriate μ for the ridge of minimum response satisfies the constraint and is less than all the eigenvalues of $\mathbf{A}$. (Refer to Myers and Montgomery (1995) for details.)

Output Data Sets

OUT=SAS-data-set

An output data set containing statistics requested with options in the MODEL statement for each observation in the input data set is created whenever the OUT= option is specified in the PROC RSREG statement. The data set contains the following variables.

- the BY variables
- the ID variables
- the WEIGHT variable
- the independent variables in the MODEL statement

- the variable _TYPE_, which identifies the observation type in the output data set. _TYPE_ is a character variable with a length of eight, and it takes on the values ’ACTUAL’, ’PREDICT’, ’RESIDUAL’, ’U95M’, ’L95M’, ’U95’, ’L95’, and ’D’, corresponding to the options specified.
- the response variables containing special output values identified by the _TYPE_ variable

All confidence limits use the two-tailed Student’s t value.

OUTR=SAS-data-set

An output data set containing the optimum response ridge is created when the OUTR= option is specified in the RIDGE statement. The data set contains the following variables:

- the current values of the BY variables
- a character variable _DEPVAR_ containing the name of the dependent variable
- a character variable _TYPE_ identifying the type of ridge being computed, MINIMUM or MAXIMUM. If both MAXIMUM and MINIMUM are specified, the data set contains observations for the minimum ridge followed by observations for the maximum ridge.
- a numeric variable _RADIUS_ giving the distance from the ridge starting point
- the values of the model factors at the estimated optimum point at distance _RADIUS_ from the ridge starting point
- a numeric variable _PRED_, which is the estimated expected value of the dependent variable at the optimum
- a numeric variable _STDERR_, which is the standard error of the estimated expected value

Displayed Output

All estimates and hypothesis tests assume that the model is correctly specified and the errors are distributed according to classical statistical assumptions.

The output displayed by PROC RSREG includes the following.

Estimation and Analysis of Variance

- The actual form of the coding operation for each value of a variable is

$$\textit{coded value} = \frac{1}{S}(\textit{original value} - M)$$

where M is the average of the highest and lowest values for the variable in the design and S is half their difference. The Subtracted off column contains the M values for this formula for each factor variable, and S is found in the Divided by column.

- The summary table for the response variable contains the following information.

  - Response Mean is the mean of the response variable in the sample. When a WEIGHT statement is used, the mean $\bar{y}$ is calculated by

    $$\bar{y} = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

  - Root MSE estimates the standard deviation of the response variable and is calculated as the square root of the Total Error mean square.
  - The R-Square value is R^2, or the coefficient of determination. R^2 measures the proportion of the variation in the response that is attributed to the model rather than to random error.
  - The Coefficient of Variation is 100 times the ratio of the Root MSE to the Response Mean.

- A table analyzing the significance of the terms of the regression is displayed. Terms are brought into the regression in four steps: (1) the Intercept and any covariates in the model, (2) Linear terms like X1 and X2, (3) pure Quadratic terms like X1*X1 or X2*X2, and (4) Crossproduct terms like X1*X2.

  - The Degrees of Freedom should be the same as the number of corresponding parameters unless one or more of the parameters are not estimable.
  - Type I Sum of Squares, also called the sequential sums of squares, measure the reduction in the error sum of squares as sets of terms (Linear, Quadratic, and so forth) are added to the model.
  - R-Square measures the portion of total R^2 contributed as each set of terms (Linear, Quadratic, and so forth) is added to the model.
  - Each F Value tests the null hypothesis that all parameters in the term are zero using the Total Error mean square as the denominator. This item is a test of a Type I hypothesis, containing the usual F test numerator, conditional on the effects of subsequent variables not being in the model.
  - Pr > F is the significance value or probability of obtaining at least as great an F ratio given that the null hypothesis is true.

- The Total Error Sum of Squares can be partitioned into Lack of Fit and Pure Error. When Lack of Fit is significant, there is variation around the model other than random error (such as cubic effects of the factor variables).

  - The Total Error Mean Square estimates σ^2, the variance.
  - F Value tests the null hypothesis that the variation is adequately described by random error.

- A table containing the parameter estimates from the model is displayed.

  - The Parameter Estimate column contains the parameter estimates based on the *uncoded* values of the factor variables. If an effect is a linear combination of previous effects, the parameter for the effect is not estimable. When this happens, the degrees of freedom are zero, the parameter estimate is set to zero, and the estimates and tests on other parameters are conditional on this parameter being zero.

  - The Standard Error column contains the estimated standard deviations of the parameter estimates based on *uncoded* data.
  - The t Value column contains t values of a test of the null hypothesis that the true parameter is zero when the *uncoded* values of the factor variables are used.
  - Pr > |T| gives the significance value or probability of a greater absolute t ratio given that the true parameter is zero.
  - The Parameter Estimate from Coded Data column contains the parameter estimates based on the *coded* values of the factor variables. These are the estimates used in the subsequent canonical and ridge analyses.

- The sum of squares are partitioned by the Factors in the model, and an analysis table is displayed. The test on a factor, say X1, is a joint test on all the parameters involving that factor. For example, the test for X1 tests the null hypothesis that the true parameters for X1, X1*X1, and X1*X2 are all zero.

Canonical Analysis

- The Critical Value columns contains the values of the factor variables that correspond to the stationary point of the fitted response surface. The critical values can be at a minimum, maximum, or saddle point.
- The Eigenvalues and Eigenvectors are from the matrix of quadratic parameter estimates based on the coded data. They characterize the shape of the response surface.

Ridge Analysis

- Coded Radius is the distance from the coded version of the associated point to the coded version of the origin of the ridge. The origin is given by the point at radius zero.
- Estimated Response is the estimated value of the response variable at the associated point. The Standard Error of this estimate is also given. This quantity is useful for assessing the relative credibility of the prediction at a given radius. Typically, this standard error increases rapidly as the ridge moves up to and beyond the design perimeter, reflecting the inherent difficulty of making predictions beyond the range of experimentation.
- Uncoded Factor Values are the values of the uncoded factor variables that give the optimum response at this radius from the ridge origin.

ODS Table Names

PROC RSREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 63.1. ODS Tables Produced in PROC RSREG

ODS Table Name	Description	Statement
Coding	Coding coefficients for the independent variables	default
ErrorANOVA	Error analysis of variance	default
FactorANOVA	Factor analysis of variance	default
FitStatistics	Overall statistics for fit	default
ModelANOVA	Model analysis of variance	default
ParameterEstimates	Estimated linear parameters	default
Ridge	Ridge analysis for optimum response	RIDGE
Spectral	Spectral analysis	default
StationaryPoint	Stationary point of response surface	default

Examples

Example 63.1. A Saddle-Surface Response Using Ridge Analysis

Frankel (1961) reports an experiment aimed at maximizing the yield of *mercaptobenzothiazole* (MBT) by varying processing time and temperature. Myers (1976) uses a two-factor model in which the estimated surface does not have a unique optimum. A ridge analysis is used to determine the region in which the optimum lies. The objective is to find the settings of time and temperature in the processing of a chemical that maximize the yield. The following statements read the data and invoke PROC RSREG. These statements produce Output 63.1.1 through Output 63.1.5:

```
data d;
   input Time Temp MBT;
   label Time = "Reaction Time (Hours)"
         Temp = "Temperature (Degrees Centigrade)"
         MBT  = "Percent Yield Mercaptobenzothiazole";
   datalines;
 4.0   250   83.8
20.0   250   81.7
12.0   250   82.4
12.0   250   82.9
12.0   220   84.7
12.0   280   57.9
12.0   250   81.2
 6.3   229   81.3
 6.3   271   83.1
17.7   229   85.3
17.7   271   72.7
 4.0   250   82.0
;
proc sort;
   by Time Temp;
run;
```

```
proc rsreg;
   model MBT=Time Temp / lackfit;
   ridge max;
run;
```

Output 63.1.1. Coding and Response Variable Information

```
                                      The RSREG Procedure

                       Coding Coefficients for the Independent Variables

                           Factor    Subtracted off      Divided by

                           Time           12.000000        8.000000
                           Temp          250.000000       30.000000


            Response Surface for Variable MBT: Percent Yield Mercaptobenzothiazole

                           Response Mean                  79.916667
                           Root MSE                        4.615964
                           R-Square                          0.8003
                           Coefficient of Variation          5.7760
```

Output 63.1.2. Analyses of Variance

```
                                      The RSREG Procedure

                                        Type I Sum
            Regression          DF      of Squares    R-Square    F Value    Pr > F

            Linear               2      313.585803      0.4899       7.36    0.0243
            Quadratic            2      146.768144      0.2293       3.44    0.1009
            Crossproduct         1       51.840000      0.0810       2.43    0.1698
            Total Model          5      512.193947      0.8003       4.81    0.0410


                                         Sum of
          Residual           DF         Squares     Mean Square    F Value    Pr > F

          Lack of Fit         3      124.696053       41.565351      39.63    0.0065
          Pure Error          3        3.146667        1.048889
          Total Error         6      127.842720       21.307120


                                                                                 Parameter
                                                                                  Estimate
                                           Standard                              from Coded
    Parameter    DF        Estimate           Error    t Value    Pr > |t|             Data

    Intercept     1     -545.867976      277.145373      -1.97      0.0964        82.173110
    Time          1        6.872863        5.004928       1.37      0.2188        -1.014287
    Temp          1        4.989743        2.165839       2.30      0.0608        -8.676768
    Time*Time     1        0.021631        0.056784       0.38      0.7164         1.384394
    Temp*Time     1       -0.030075        0.019281      -1.56      0.1698        -7.218045
    Temp*Temp     1       -0.009836        0.004304      -2.29      0.0623        -8.852519


                     Sum of
  Factor   DF       Squares   Mean Square  F Value  Pr > F  Label

  Time      3     61.290957     20.430319     0.96  0.4704  Reaction Time (Hours)
  Temp      3    461.250925    153.750308     7.22  0.0205  Temperature (Degrees Centigrade)
```

Output 63.1.2 shows that the lack of fit for the model is highly significant. Since the quadratic model does not fit the data very well, firm statements about the underlying process should not be based only on the current analysis. Note from the analysis of variance for the model that the test for the time factor is not significant. If further experimentation is undertaken, it might be best to fix Time at a moderate to high value and to concentrate on the effect of temperature. In the actual experiment discussed here, extra runs were made that confirmed the results of the following analysis.

Output 63.1.3. Canonical Analysis

```
                                 The RSREG Procedure
                Canonical Analysis of Response Surface Based on Coded Data

                         Critical Value
     Factor              Coded          Uncoded    Label

     Time            -0.441758         8.465935    Reaction Time (Hours)
     Temp            -0.309976       240.700718    Temperature (Degrees Centigrade)

                      Predicted value at stationary point: 83.741940


                                            Eigenvectors
                        Eigenvalues             Time             Temp

                           2.528816         0.953223        -0.302267
                          -9.996940         0.302267         0.953223

                            Stationary point is a saddle point.
```

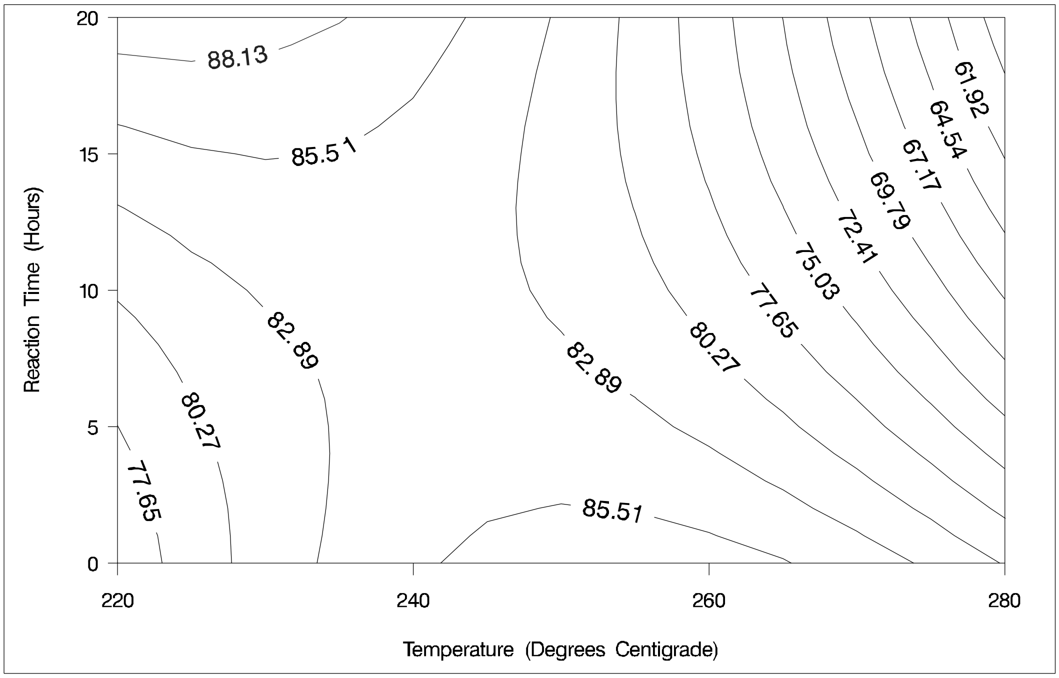
The canonical analysis (Output 63.1.3) indicates that the predicted response surface is shaped like a saddle. The eigenvalue of 2.5 shows that the valley orientation of the saddle is less curved than the hill orientation, with eigenvalue of -9.99. The coefficients of the associated eigenvectors show that the valley is more aligned with Time and the hill with Temp. Because the canonical analysis resulted in a saddle point, the estimated surface does not have a unique optimum.

Output 63.1.4. Ridge Analysis

```
                                 The RSREG Procedure

   Estimated Ridge of Maximum Response for Variable MBT: Percent Yield Mercaptobenzothiazole


            Coded        Estimated         Standard         Uncoded Factor Values
           Radius         Response            Error             Time             Temp

              0.0        82.173110         2.665023        12.000000       250.000000
              0.1        82.952909         2.648671        11.964493       247.002956
              0.2        83.558260         2.602270        12.142790       244.023941
              0.3        84.037098         2.533296        12.704153       241.396084
              0.4        84.470454         2.457836        13.517555       239.435227
              0.5        84.914099         2.404616        14.370977       237.919138
              0.6        85.390012         2.410981        15.212247       236.624811
              0.7        85.906767         2.516619        16.037822       235.449230
              0.8        86.468277         2.752355        16.850813       234.344204
              0.9        87.076587         3.130961        17.654321       233.284652
              1.0        87.732874         3.648568        18.450682       232.256238
```

However, the ridge analysis in Output 63.1.4 indicates that maximum yields will result from relatively high reaction times and low temperatures. A contour plot of the predicted response surface, shown in Output 63.1.5, confirms this conclusion.

Output 63.1.5. Contour Plot of Predicted Response Surface



The statements that produce this plot follow. Note that contour and three-dimensional plots can be created interactively using SAS/INSIGHT software or the ADX Interface in SAS/QC software. Initial DATA steps create a grid over Time and Temp and combine this grid with the original data, using a variable flag to indicate the grid. Then, PROC RSREG is used to create predictions for the combined data. Finally, PROC GCONTOUR to displays a contour plot of the predictions over just the grid.

```
data b;
   set d;
   flag=1;
   MBT=.;
   do Time=0 to 20 by 1;
      do Temp=220 to 280 by 5;
         output;
      end;
   end;
data c;
   set d b;
proc rsreg data=c out=e noprint;
  model MBT=Time Temp / predict;
  id flag;
run;
```

```
axis1 label=(angle=90) minor=none;
axis2 order=(220 to 280 by 20) minor=none;

proc gcontour data=e(where=(flag=1));
   plot Time*Temp=MBT
        / nlevels=12 vaxis=axis1 haxis=axis2 nolegend autolabel
          llevels=2 2 2   1 1 1   1 1 1   1 1 1   ;
run;
```

Example 63.2. Response Surface Analysis with Covariates

One way of viewing covariates is as extra sources of variation in the dependent variable that may mask the variation due to primary factors. This example demonstrates the use of the COVAR= option in PROC RSREG to fit a response surface model to the dependent variable values corrected for the covariates.

You have a chemical process with a yield that you hypothesize to be dependent on three factors: reaction time, reaction temperature, and reaction pressure. You perform an experiment to measure this dependence. You are willing to include up to 20 runs in your experiment, but you can perform no more than 8 runs on the same day, so the design for the experiment is composed of three blocks. Additionally, you know that the grade of raw material for the reaction has a significant impact on the yield. You have no control over this, but you keep track of it. The following statements create a SAS data set containing the results of the experiment:

```
data Experiment;
   input Day Grade Time Temp Pressure Yield;
   datalines;
1 67      -1     -1     -1         32.98
1 68      -1      1      1         47.04
1 70       1     -1      1         67.11
1 66       1      1     -1         26.94
1 74       0      0      0        103.22
1 68       0      0      0         42.94
2 75      -1     -1      1        122.93
2 69      -1      1     -1         62.97
2 70       1     -1     -1         72.96
2 71       1      1      1         94.93
2 72       0      0      0         93.11
2 74       0      0      0        112.97
3 69       1.633  0      0         78.88
3 67      -1.633  0      0         52.53
3 68       0      1.633  0         68.96
3 71       0     -1.633  0         92.56
3 70       0      0      1.633     88.99
3 72       0      0     -1.633    102.50
3 70       0      0      0         82.84
3 72       0      0      0        103.12
;
```

Your first analysis neglects to take the covariates into account. The following statements use PROC RSREG to fit a response surface to the observed yield, but note that Day and Grade are omitted.

```
proc rsreg data=Experiment;
   model Yield = Time Temp Pressure;
run;
```

The ANOVA results (shown in Output 63.2.1) indicate that *no* process variable effects are significantly larger than the background noise.

Output 63.2.1. Analysis of Variance Ignoring Covariates

```
                                   The RSREG Procedure

                                     Type I Sum
         Regression          DF      of Squares    R-Square    F Value    Pr > F

         Linear               3     1880.842426      0.1353       0.67    0.5915
         Quadratic            3     2370.438681      0.1706       0.84    0.5023
         Crossproduct         3      241.873250      0.0174       0.09    0.9663
         Total Model          9     4493.154356      0.3233       0.53    0.8226


                                                Sum of
                  Residual          DF         Squares     Mean Square

                  Total Error       10     9405.129724      940.512972
```

However, when the yields are adjusted for covariate effects of day and grade of raw material, very strong process variable effects are revealed. The following statements produce the ANOVA results in Output 63.2.2. Note that in order to include the effects of the classification factor Day as covariates, you need to create dummy variables indicating each day separately.

```
data Experiment; set Experiment;
   d1 = (Day = 1);
   d2 = (Day = 2);
   d3 = (Day = 3);
proc rsreg data=Experiment;
   model Yield = d1-d3 Grade Time Temp Pressure / covar=4;
run;
```

Output 63.2.2. Analysis of Variance Including Covariates

```
                              The RSREG Procedure

                                  Type I Sum
      Regression          DF     of Squares    R-Square    F Value     Pr > F

      Covariates           3          13695      0.9854     316957     <.0001
      Linear               3     156.524497      0.0113    3622.53     <.0001
      Quadratic            3      22.989775      0.0017     532.06     <.0001
      Crossproduct         3      23.403614      0.0017     541.64     <.0001
      Total Model         12          13898      1.0000    80413.2     <.0001


                                               Sum of
               Residual           DF          Squares     Mean Square

               Total Error         7         0.100820        0.014403
```

The results show very strong effects due to both the covariates and the process variables.

References

Box, G.E.P. (1954), "The Exploration and Exploitation of Response Surfaces: some General Considerations," *Biometrics*, 10, 16.

Box, G.E.P. (1987), *Empirical Model Building and Response Surfaces*, New York: John Wiley & Sons, Inc.

Box, G.E.P. and Draper, N.R. (1982), "Measures of Lack of Fit for Response Surface Designs and Predictor Variable Transformations," *Technometrics*, 24, 1–8.

Box, G.E.P. and Hunter, J.S. (1957), "Multifactor Experimental Designs for Exploring Response Surfaces," *Annals of Mathematical Statistics*, 28, 195–242.

Box, G.E.P., Hunter, W.G., and Hunter, J.S. (1978), *Statistics for Experimenters*, New York: John Wiley & Sons, Inc.

Box, G.E.P. and Wilson, K.J. (1951), "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society*, Ser. B, 13, 1–45.

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons, Inc.

Draper, N.R. (1963), "Ridge Analysis of Response Surfaces," *Technometrics* 5, 469–479.

Draper, N.R. and John, J.A. (1988), "Response Surface Designs for Quantitative and Qualitative Variables," *Technometrics*, 30 (4), 423–428.

Draper, N.R. and Smith, H. (1981), *Applied Regression Analysis, Second Edition*, New York: John Wiley & Sons, Inc.

John, P.W.M. (1971), *Statistical Design and Analysis of Experiments*, New York: Macmillan Publishing Co., Inc.

Mead, R. and Pike, D.J. (1975), "A review of Response Surface Methodology from a Biometric Point of View," *Biometrics*, 31, 803.

Meyer, D.C. (1963), “Response Surface Methodology in Education and Psychology,” *Journal of Experimental Education* 31, 329.

Myers, R.H. (1976), *Response Surface Methodology*, Blacksburg, VA: Virginia Polytechnic Institute and State University.

Myers, R.H. and Montgomery, D.C. (1995), *Response Surface Methodology*, New York: John Wiley & Sons, Inc.

Schneider, A.M. and Stockett, A.L. (1963), “An Experiment to Select Optimum Operating Conditions on the Basis of Arbitrary Preference Ratings,” *Chemical Engineering Progress Symposium Series* 59.

Chapter 64
The SCORE Procedure

Chapter Contents

Chapter 64

The SCORE Procedure

Overview

The SCORE procedure multiplies values from two SAS data sets, one containing coefficients (for example, factor-scoring coefficients or regression coefficients) and the other containing raw data to be scored using the coefficients from the first data set. The result of this multiplication is a SAS data set containing linear combinations of the coefficients and the raw data values.

Many statistical procedures output coefficients that PROC SCORE can apply to raw data to produce scores. The new score variable is formed as a linear combination of raw data and scoring coefficients. For each observation in the raw data set, PROC SCORE multiplies the value of a variable in the raw data set by the matching scoring coefficient from the data set of scoring coefficients. This multiplication process is repeated for each variable in the VAR statement. The resulting products are then summed to produce the value of the new score variable. This entire process is repeated for each observation in the raw data set. In other words, PROC SCORE cross multiplies part of one data set with another.

Raw Data Set

The raw data set can contain the original data used to calculate the scoring coefficients, or it can contain an entirely different data set. The raw data set must contain all the variables needed to produce scores. In addition, the scoring coefficients and the variables in the raw data set that are used in scoring must have the same names. See the section "Getting Started" beginning on page 4067.

Scoring Coefficients Data Set

The data set containing scoring coefficients must contain two special variables: the _TYPE_ variable and the _NAME_ or _MODEL_ variable.

- The _TYPE_ variable identifies the observations that contain scoring coefficients.
- The _NAME_ or _MODEL_ variable provides a SAS name for the new score variable.

PROC SCORE first looks for a _NAME_ variable in the SCORE= input data set. If there is such a variable, the variable's value is what SCORE uses to name the new score variable. If the SCORE= data set does not have a _NAME_ variable, then PROC SCORE looks for a _MODEL_ variable.

For example, PROC FACTOR produces an output data set that contains factor-scoring coefficients. In this output data set, the scoring coefficients are identified by

TYPE='SCORE'. For _TYPE_='SCORE', the _NAME_ variable has values of 'Factor1', 'Factor2', and so forth. PROC SCORE gives the new score variables the names Factor1, Factor2, and so forth.

As another example, the REG procedure produces an output data set that contains parameter estimates. In this output data set, the parameter estimates are identified by _TYPE_='PARMS'. The _MODEL_ variable contains the label used in the MODEL statement in PROC REG, or it uses MODELn if no label is specified. This label is the name PROC SCORE gives to the new score variable.

Standardization of Raw Data

PROC SCORE automatically standardizes or centers the DATA= variables for you, based on information from the original variables and analysis from the SCORE= data set.

If the SCORE= scoring coefficients data set contains observations with _TYPE_='MEAN' and _TYPE_='STD', then PROC SCORE standardizes the raw data before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP without the COV option.

If the SCORE= scoring coefficients data set contains observations with _TYPE_='MEAN' but _TYPE_='STD' is absent, then PROC SCORE centers the raw data (the means are substracted) before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP with the COV option.

If the SCORE= scoring coefficients data set does not contain observations with _TYPE_='MEAN' and _TYPE_='STD', or if you use the NOSTD option, then PROC SCORE does not center or standardize the raw data.

If the SCORE= scoring coefficients are obtained from observations with _TYPE_='USCORE', then PROC SCORE "standardizes" the raw data using the uncorrected standard deviations identified by _TYPE_='USTD', and the means are not subtracted from the raw data. For example, this type of SCORE= data set can come from PROC PRINCOMP with the NOINT option. For more information on _TYPE_='USCORE' scoring coefficients in TYPE=UCORR or TYPE=UCOV output data sets, see Appendix A, "Special SAS Data Sets."

You can use PROC SCORE to score the data that were also used to generate the scoring coefficents, although more typically, scoring results are directly obtained from the OUT= data set in a procedure that computes scoring coefficients. When scoring new data, it is important to realize that PROC SCORE assumes that the new data have approximately the same scales as the original data. For example, if you specify the COV option with PROC PRINCOMP for the original analysis, the scoring coefficients in the PROC PRINCOMP OUTSTAT= data set are not appropriate for standardized data. With the COV option, PROC PRINCOMP will not output _TYPE_='STD' observations to the OUTSTAT= data set, and PROC SCORE will only subtract the means of the original (not new) variables from the new variables before multiplying. Without the COV option in PROC PRINCOMP, both the original variable means and standard deviations will be in the OUTSTAT= data set, and PROC SCORE will subtract the original variable means from the new variables and divide them by the original variable standard deviations before multiplying.

In general, procedures that output scoring coefficients in their OUTSTAT= data sets provide the necessary information for PROC SCORE to determine the appropriate standardization. However, if you use PROC SCORE with a scoring coefficients data set that you constructed without _TYPE_='MEAN' and _TYPE_='STD' observations, you might have to do the relevant centering or standardization of the new data first. If you do this, you must use the means and standard deviations of the original variables, that is, the variables that were used to generate the coefficients, not the means and standard deviations of the variables to be scored.

See the section "Getting Started" on page 4067 for further illustration.

Getting Started

The SCORE procedure multiplies the values from two SAS data sets and creates a new data set to contain the results of the multiplication. The variables in the new data set are linear combinations of the variables in the two input data sets. Typically, one of these data sets contains raw data that you want to score, and the other data set contains scoring coefficients.

The following example demonstrates how to use the SCORE procedure to multiply values from two SAS data sets, one containing factor-scoring coefficients and the other containing raw data to be scored using the scoring coefficients.

Suppose you are interested in the performance of three different types of schools: private schools, state-run urban schools, and state-run rural schools. You want to compare the schools' performances as measured by student grades on standard tests in English, mathematics, and biology. You administer these tests and record the scores for each of the three types of schools.

The following DATA step creates the SAS data set Schools. The data are provided by Chaseling (1996).

```
data Schools;
   input Type $ English Math Biology @@;
   datalines;
p  52  55  45  p  42  49  40  p  63  64  54
p  47  50  51  p  64  69  47  p  63  67  54
p  59  63  42  p  56  61  41  p  41  44  72
p  39  42  45  p  56  63  44  p  63  73  42
p  62  68  46  p  51  61  51  p  45  56  54
p  63  66  63  p  65  67  57  p  49  50  47
p  47  48  34  p  53  54  46  p  49  40  43
p  50  41  50  p  82  72  80  p  68  61  62
p  68  61  46  p  63  53  48  p  77  72  74
p  50  47  60  p  61  49  48  p  64  54  45
p  60  53  40  p  80  69  75  p  76  69  77
p  55  48  51  p  85  76  80  p  70  64  48
p  61  51  61  p  51  47  58  p  78  72  79
```

```
   p  52  47  46  u  49  47  58  u  64  72  45
   u  36  44  46  u  32  43  46  u  52  57  42
   u  45  47  53  u  44  52  43  u  54  63  42
   u  39  45  49  u  48  51  46  u  53  61  54
   u  28  32  33  u  52  59  44  u  54  61  51
   u  60  65  66  u  60  63  63  u  47  52  49
   u  28  31  32  u  43  46  45  u  40  42  48
   u  66  51  48  u  79  68  77  u  58  52  49
   u  34  29  33  u  47  35  40  u  60  49  49
   u  62  50  51  u  69  50  47  u  59  41  52
   u  56  44  43  u  76  61  74  u  50  36  52
   u  69  56  52  u  57  41  55  u  56  44  51
   u  52  42  42  u  51  36  42  u  44  31  57
   u  79  68  77  u  61  44  41  r  38  28  22
   r  35  28  24  r  50  47  48  r  36  28  38
   r  69  65  53  r  55  44  41  r  62  58  45
   r  57  55  32  r  47  42  66  r  45  38  45
   r  56  55  42  r  39  36  33  r  63  51  42
   r  42  41  48  r  51  44  52  r  47  42  44
   r  53  42  47  r  62  59  48  r  80  74  81
   r  95  79  95  r  65  60  43  r  67  60  53
   r  42  43  50  r  70  68  55  r  63  56  48
   r  37  33  34  r  49  47  49  r  42  43  50
   r  44  46  47  r  62  55  44  r  67  64  52
   r  77  77  69  r  43  42  52  r  51  54  45
   r  67  65  45  r  65  73  49  r  34  29  32
   r  50  47  49  r  55  48  46  r  38  36  51
   ;
```

The data set Schools contains the character variable Type, which represents the type of school. Valid values are p (private schools), r (state-run rural schools), and u (state-run urban schools).

The three numeric variables in the data set are English, Math, and Biology, which represent the student scores for English, mathematics, and biology, respectively. The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values are read.

The following statements invoke the FACTOR procedure to compute the data set of factor scoring coefficients. The statements perform a principle components factor analysis using all three numeric variables in the SAS data set Schools. The OUTSTAT= option requests that PROC FACTOR output the factor scores to the data set Scores. The NOPRINT option suppresses display of the output.

```
   proc factor data=Schools score outstat=Scores noprint;
      var english math biology;
   run;
   proc score data=schools score=Scores out=New;
      var english math biology;
      id type;
   run;
```

The SCORE procedure is then invoked using Schools as the raw data set to be scored and Scores as the scoring data set. The OUT= option creates the SAS data set New to contain the linear combinations.

The VAR statement specifies that the variables English, Math, and Biology are used in computing scores. The ID statement copies the variable Type from the Schools data set to the output data set New.

The following statements print the SAS output data set Scores, the first two observations from the original data set Schools, and the first two observations of the resulting data set New.

```
title 'OUTSTAT= Data Set from PROC FACTOR';
   proc print data=Scores;
run;
title 'First Two Observations of the DATA= Data Set from PROC SCORE';
   proc print data=Schools(obs=2);
run;
title 'First Two Observations of the OUT= Data Set from PROC SCORE';
   proc print data=New(obs=2);
run;
```

Figure 64.1 displays the output data set Scores produced by the FACTOR procedure. The last observation (observation number 11) contains the scoring coefficients (_TYPE_='SCORE'). Only one factor has been retained. Figure 64.1 also lists the first two observations of the original SAS data set Schools and the first two observations of the output data set New from the SCORE procedure.

```
                       OUTSTAT= Data Set from PROC FACTOR

          Obs    _TYPE_      _NAME_     English       Math    Biology

            1    MEAN                    55.525     52.325     50.350
            2    STD                     12.949     12.356     12.239
            3    N                      120.000    120.000    120.000
            4    CORR        English      1.000      0.833      0.672
            5    CORR        Math         0.833      1.000      0.594
            6    CORR        Biology      0.672      0.594      1.000
            7    COMMUNAL                 0.881      0.827      0.696
            8    PRIORS                   1.000      1.000      1.000
            9    EIGENVAL                 2.405      0.437      0.159
           10    PATTERN     Factor1      0.939      0.910      0.834
           11    SCORE       Factor1      0.390      0.378      0.347



          First Two Observations of the DATA= Data Set from PROC SCORE

                   Obs    Type    English    Math    Biology

                     1     p         52       55        45
                     2     p         42       49        40



          First Two Observations of the OUT= Data Set from PROC SCORE

                            Obs    Type     Factor1

                              1     p      -0.17604
                              2     p      -0.80294
```

Figure 64.1. Views of the Scores, Schools, and New Data Sets

The score variable Factor1 in the New data set is named according to the value of the _NAME_ variable in the Scores data set. The values of the variable Factor1 are computed as follows: the original data set variables are standardized to a mean of 0 and a variance of 1 because the Scores data set contains observations with _TYPE_='MEAN' and _TYPE_='STD'. These standardized variables are then multiplied by their respective standardized scoring coefficients from the data set Scores. These products are summed over all three variables, and the sum is the value of the new variable Factor1. The first two values of the scored variable Factor1 are obtained as follows:

$$\left(\frac{(52-55.525)}{12.949}\times 0.390\right)+\left(\frac{(55-52.325)}{12.356}\times 0.378\right)+\left(\frac{(45-50.350)}{12.239}\times 0.347\right)=-0.17604$$

$$\left(\frac{(42-55.525)}{12.949}\times 0.390\right)+\left(\frac{(49-52.325)}{12.356}\times 0.378\right)+\left(\frac{(40-50.350)}{12.239}\times 0.347\right)=-0.80294$$

The following statements request that the GCHART procedure produce a horizontal bar chart of the variable Type. The length of each bar represents the mean of the variable Factor1.

```
proc gchart;
   hbar type/type=mean sumvar=Factor1;
run;
```

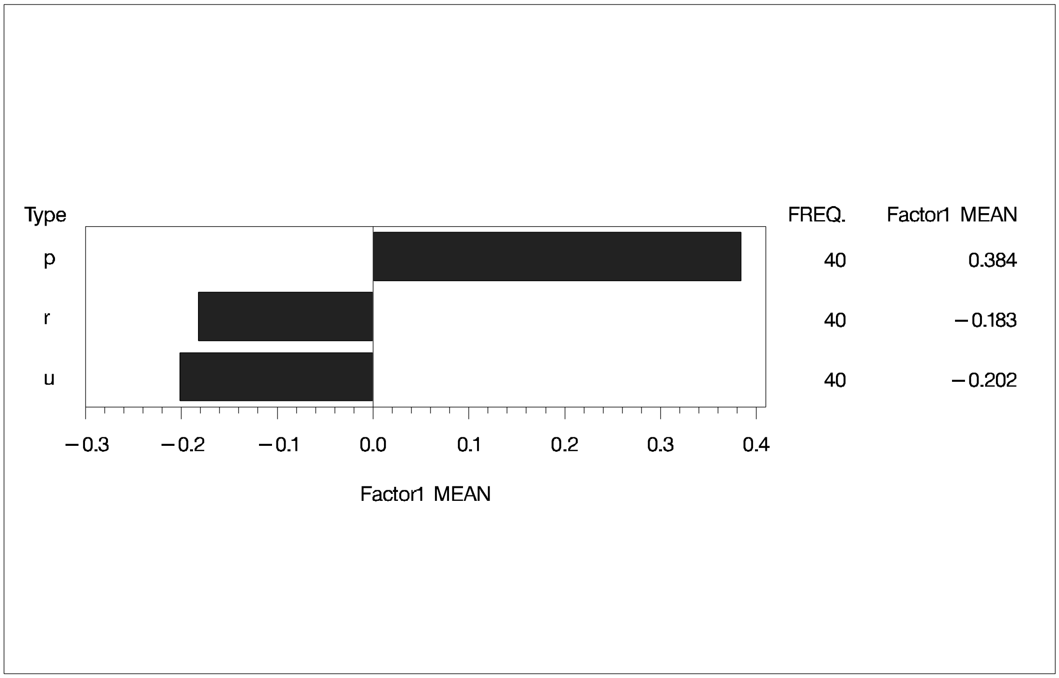


Figure 64.2. Bar Chart of School Type

Figure 64.2 displays the mean score of the variable Factor1 for each of the three school types. For private schools (Type=p), the average value of the variable Factor1 is 0.384, while for state-run schools the average value is much lower. The state-run urban schools (Type=u) have the lowest mean value of -0.202, and the state-run rural schools (Type=r) have a mean value of -0.183.

Syntax

The following statements are available in the SCORE procedure.

PROC SCORE *DATA= SAS-data-set* < *options* > **;**
  BY *variables* **;**
  ID *variables* **;**
  VAR *variables* **;**

The only required statement is the PROC SCORE statement. The BY, ID, and VAR statements are described following the PROC SCORE statement.

PROC SCORE Statement

PROC SCORE DATA= *SAS-data-set* < *options* > **;**

You can specify the following options in the PROC SCORE statement.

DATA=*SAS-data-set*

names the input SAS data set containing the raw data to score. This specification is required.

NOSTD

suppresses centering and scaling of the raw data. Ordinarily, if PROC SCORE finds _TYPE_='MEAN', _TYPE_= 'USCORE', _TYPE_='USTD', or _TYPE_='STD' observations in the SCORE= data set, the procedure uses these to standardize the raw data before scoring.

OUT=*SAS-data-set*

specifies the name of the SAS data set created by PROC SCORE. If you want to create a permanent SAS data set, you must specify a two-level name. (Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent SAS data sets.) If the OUT= option is omitted, PROC SCORE still creates an output data set and automatically names it according to the DATAn convention, just as if you omitted a data set name in a DATA statement.

PREDICT

specifies that PROC SCORE should treat coefficients of -1 in the SCORE= data set as 0. In regression applications, the dependent variable is coded with a coefficient of -1. Applied directly to regression results, PROC SCORE produces negative residuals (see the description of the RESIDUAL option, which follows); the PREDICT option produces predicted values instead.

RESIDUAL

reverses the sign of each score. Applied directly to regression results, PROC SCORE produces negative residuals (PREDICT$-$ACTUAL); the RESIDUAL option produces positive residuals (ACTUAL$-$PREDICT) instead.

SCORE=*SAS-data-set*

names the data set containing the scoring coefficients. If you omit the SCORE= option, the most recently created SAS data set is used. This data set must have two special variables: _TYPE_ and either _NAME_ or _MODEL_.

TYPE=*name* **|** *'string'*

specifies the observations in the SCORE= data set that contain scoring coefficients. The TYPE= procedure option is unrelated to the data set option that has the same name. PROC SCORE examines the values of the special variable _TYPE_ in the SCORE= data set. When the value of _TYPE_ matches TYPE=*name*, the observation in the SCORE= data set is used to score the raw data in the DATA= data set.

If you omit the TYPE= option, scoring coefficients are read from observations with either _TYPE_='SCORE' or _TYPE_='USCORE'. Because the default for PROC SCORE is TYPE=SCORE, you need not specify the TYPE= option for factor scoring or for computing scores from OUTSTAT= data sets from the CANCORR, CANDISC, PRINCOMP, or VARCLUS procedure. When you use regression coefficients from PROC REG, specify TYPE=PARMS.

The maximum length of the argument specified in the TYPE= option depends on the length defined by the VALIDVARNAME= SAS system option. For additional information, refer to *SAS Language Reference: Dictionary*.

Note that the TYPE= option setting is not case-sensitive. For example, the two option settings, TYPE='MyScore' and TYPE='myscore', are equivalent.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SCORE to obtain separate scoring for observations in groups defined by the BY variables. You can also specify a BY statement to apply separate groups of scoring coefficients to the entire DATA= data set.

If your SCORE= input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the SCORE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

If the DATA= data set does not contain any of the BY variables, the entire DATA= data set is scored by each BY group of scoring coefficients in the SCORE= data set.

If the DATA= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the DATA= data set as in the SCORE= data set, then PROC SCORE prints an error message and stops.

If all the BY variables appear in the DATA= data set with the same type and length as in the SCORE= data set, then each BY group in the DATA= data set is scored using scoring coefficients from the corresponding BY group in the SCORE= data set. The BY groups in the DATA= data set must be in the same order as in the SCORE= data set. All BY groups in the DATA= data set must also appear in the SCORE= data set. If you do not specify the NOTSORTED option, some BY groups can appear in the SCORE= data set but not in the DATA= data set; such BY groups are not used in computing scores.

ID Statement

ID *variables* **;**

The ID statement identifies variables from the DATA= data set to be included in the OUT= data set. If there is no ID statement, all variables from the DATA= data set are included in the OUT= data set. The ID variables can be character or numeric.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the variables to be used in computing scores. These variables must be in both the DATA= and SCORE= input data sets and must be numeric. If you do not specify a VAR statement, the procedure uses all numeric variables in the SCORE= data set. You should almost always specify a VAR statement with PROC SCORE because you would rarely use all the numeric variables in your data set to compute scores.

Details

Missing Values

If one of the scoring variables in the DATA= data set has a missing value for an observation, all the scores have missing values for that observation. The exception to this criterion is if the PREDICT option is specified, the variable with a coefficient of -1 can tolerate a missing value and still produce a prediction score. Also, a variable with a coefficient of 0 can tolerate a missing value.

If a scoring coefficient in the SCORE= data set has a missing value for an observation, the coefficient is not used in creating the new score variable for the observation. In other words, missing values of scoring coefficients are treated as zeros. This treatment affects only the observation in which the missing value occurs.

Regression Parameter Estimates from PROC REG

If the SCORE= data set is an OUTEST= data set produced by PROC REG and if you specify TYPE=PARMS, the interpretation of the new score variables depends on the PROC SCORE options chosen and the variables listed in the VAR statement. If the VAR statement contains only the independent variables used in a model in PROC REG, the new score variables give the predicted values. If the VAR statement contains the dependent variables and the independent variables used in a model in PROC REG, the interpretation of the new score variables depends on the PROC SCORE options chosen. If you omit both the PREDICT and the RESIDUAL options, the new score variables give negative residuals (PREDICT$-$ACTUAL). If you specify the RESIDUAL option, the new score variables give positive residuals (ACTUAL$-$PREDICT). If you specify the PREDICT option, the new score variables give predicted values.

Unless you specify the NOINT option for PROC REG, the OUTEST= data set contains the variable Intercept. The SCORE procedure uses the intercept value in computing the scores.

Output Data Set

PROC SCORE produces an output data set but displays no output. The output OUT= data set contains the following:

- the ID variables, if any
- all variables from the DATA= data set, if no ID variables are specified
- the BY variables, if any
- the new score variables, named from the _NAME_ or _MODEL_ values in the SCORE= data set

Computational Resources

Let

v = number of variables used in computing scores

s = number of new score variables

b = maximum number of new score variables in a BY group

n = number of observations

Memory

The array storage required is approximately $8(4v+(3+v)b+s)$ bytes. When you do not use BY processing, the array storage required is approximately $8(4v+(4+v)s)$ bytes.

Time

The time required to construct the scoring matrix is roughly proportional to vs and the time needed to compute the scores is roughly proportional to nvs.

Examples

The following three examples use a subset of the Fitness data set. The complete data set is given in Chapter 61, “The REG Procedure.”

Example 64.1. Factor Scoring Coefficients

This example shows how to use PROC SCORE with factor scoring coefficients. First, the FACTOR procedure produces an output data set containing scoring coefficients in observations identified by _TYPE_='SCORE'. These data, together with the original data set Fitness, are supplied to PROC SCORE, resulting in a data set containing scores Factor1 and Factor2. These statements produce Output 64.1.1 through Output 64.1.3:

```
/* This data set contains only the first 12 observations   */
/* from the full data set used in the chapter on PROC REG. */
data Fitness;
   input Age Weight Oxygen RunTime RestPulse RunPulse @@;
   datalines;
44 89.47  44.609 11.37 62 178     40 75.07  45.313 10.07 62 185
44 85.84  54.297  8.65 45 156     42 68.15  59.571  8.17 40 166
38 89.02  49.874  9.22 55 178     47 77.45  44.811 11.63 58 176
40 75.98  45.681 11.95 70 176     43 81.19  49.091 10.85 64 162
44 81.42  39.442 13.08 63 174     38 81.87  60.055  8.63 48 170
44 73.03  50.541 10.13 45 168     45 87.66  37.388 14.03 56 186
;

proc factor data=Fitness outstat=FactOut
            method=prin rotate=varimax score;
   var Age Weight RunTime RunPulse RestPulse;
   title 'FACTOR SCORING EXAMPLE';
   run;

proc print data=FactOut;
   title2 'Data Set from PROC FACTOR';
run;

proc score data=Fitness score=FactOut out=FScore;
   var Age Weight RunTime RunPulse RestPulse;
   run;

proc print data=FScore;
   title2 'Data Set from PROC SCORE';
run;
```

Output 64.1.1 shows the PROC FACTOR output. The scoring coefficients for the two factors are shown at the end of the PROC FACTOR output.

Output 64.1.1. Creating an OUTSTAT= Data Set with PROC FACTOR

```
                   FACTOR SCORING EXAMPLE

                    The FACTOR Procedure
        Initial Factor Method: Principal Components

Eigenvalues of the Correlation Matrix: Total = 5  Average = 1

        Eigenvalue    Difference    Proportion    Cumulative

   1    2.30930638    1.11710686        0.4619        0.4619
   2    1.19219952    0.30997249        0.2384        0.7003
   3    0.88222702    0.37965990        0.1764        0.8767
   4    0.50256713    0.38886717        0.1005        0.9773
   5    0.11369996                      0.0227        1.0000


                      Factor Pattern

                           Factor1         Factor2

         Age               0.29795         0.93675
         Weight            0.43282        -0.17750
         RunTime           0.91983         0.28782
         RunPulse          0.72671        -0.38191
         RestPulse         0.81179        -0.23344
```

```
                           The FACTOR Procedure
               Initial Factor Method: Principal Components

                     Variance Explained by Each Factor

                           Factor1         Factor2

                         2.3093064       1.1921995


               Final Communality Estimates: Total = 3.501506

       Age           Weight          RunTime         RunPulse       RestPulse

0.96628351      0.21883401      0.92893333      0.67396207      0.71349297



                           The FACTOR Procedure
                         Rotation Method: Varimax

                     Orthogonal Transformation Matrix

                                        1               2

                        1         0.92536         0.37908
                        2        -0.37908         0.92536


                         Rotated Factor Pattern

                                  Factor1         Factor2

                Age              -0.07939         0.97979
                Weight            0.46780        -0.00018
                RunTime           0.74207         0.61503
                RunPulse          0.81725        -0.07792
                RestPulse         0.83969         0.09172
```

```
                           The FACTOR Procedure
                         Rotation Method: Varimax

                      Variance Explained by Each Factor

                           Factor1         Factor2

                         2.1487753       1.3527306


                Final Communality Estimates: Total = 3.501506

          Age           Weight          RunTime         RunPulse        RestPulse

   0.96628351       0.21883401       0.92893333       0.67396207       0.71349297



                           The FACTOR Procedure
                         Rotation Method: Varimax

         Squared Multiple Correlations of the Variables with Each Factor

                           Factor1         Factor2

                         1.0000000       1.0000000


                     Standardized Scoring Coefficients

                                  Factor1         Factor2

                    Age          -0.17846         0.77600
                    Weight        0.22987        -0.06672
                    RunTime       0.27707         0.37440
                    RunPulse      0.41263        -0.17714
                    RestPulse     0.39952        -0.04793
```

Output 64.1.2 lists the OUTSTAT= data set from PROC FACTOR. Note that observations 18 and 19 have _TYPE_='SCORE'. Observations 1 and 2 have _TYPE_='MEAN' and _TYPE_='STD', respectively. These four observations are used by PROC SCORE.

Output 64.1.2. OUTSTAT= Data Set from PROC FACTOR Reproduced with PROC PRINT

```
                               FACTOR SCORING EXAMPLE
                              Data Set from PROC FACTOR

                                                                                Rest
Obs    _TYPE_      _NAME_            Age      Weight    RunTime    RunPulse     Pulse

  1    MEAN                      42.4167     80.5125    10.6483     172.917    55.6667
  2    STD                        2.8431      6.7660     1.8444       8.918     9.2769
  3    N                         12.0000     12.0000    12.0000      12.000    12.0000
  4    CORR        Age            1.0000      0.0128     0.5005      -0.095    -0.0080
  5    CORR        Weight         0.0128      1.0000     0.2637       0.173     0.2396
  6    CORR        RunTime        0.5005      0.2637     1.0000       0.556     0.6620
  7    CORR        RunPulse      -0.0953      0.1731     0.5555       1.000     0.4853
  8    CORR        RestPulse     -0.0080      0.2396     0.6620       0.485     1.0000
  9    COMMUNAL                   0.9663      0.2188     0.9289       0.674     0.7135
 10    PRIORS                     1.0000      1.0000     1.0000       1.000     1.0000
 11    EIGENVAL                   2.3093      1.1922     0.8822       0.503     0.1137
 12    UNROTATE    Factor1        0.2980      0.4328     0.9198       0.727     0.8118
 13    UNROTATE    Factor2        0.9368     -0.1775     0.2878      -0.382    -0.2334
 14    TRANSFOR    Factor1        0.9254     -0.3791      .            .          .
 15    TRANSFOR    Factor2        0.3791      0.9254      .            .          .
 16    PATTERN     Factor1       -0.0794      0.4678     0.7421       0.817     0.8397
 17    PATTERN     Factor2        0.9798     -0.0002     0.6150      -0.078     0.0917
 18    SCORE       Factor1       -0.1785      0.2299     0.2771       0.413     0.3995
 19    SCORE       Factor2        0.7760     -0.0667     0.3744      -0.177    -0.0479
```

Since the PROC SCORE statement does not contain the NOSTD option, the data in the Fitness data set are standardized before scoring. For each variable specified in the VAR statement, the mean and standard deviation are obtained from the FactOut data set. For each observation in the Fitness data set, the variables are then standardized. For example, for observation 1 in the Fitness data set, the variable Age is standardized to $0.5569 = [(44 - 42.4167)/2.8431]$.

After the data in the Fitness data set are standardized, the standardized values of the variables in the VAR statement are multiplied by the matching coefficients in the FactOut data set, and the resulting products are summed. This sum is output as a value of the new score variable.

Output 64.1.3 displays the FScore data set produced by PROC SCORE. This data set contains the variables Age, Weight, Oxygen, RunTime, RestPulse, and RunPulse from the Fitness data set. It also contains Factor1 and Factor2, the two new score variables.

Output 64.1.3. OUT= Data Set from PROC SCORE Reproduced with PROC PRINT

```
                              FACTOR SCORING EXAMPLE
                             Data Set from PROC SCORE

                                       Run     Rest    Run
 Obs    Age    Weight    Oxygen      Time    Pulse   Pulse     Factor1      Factor2

   1     44     89.47    44.609     11.37      62     178      0.82129      0.35663
   2     40     75.07    45.313     10.07      62     185      0.71173     -0.99605
   3     44     85.84    54.297      8.65      45     156     -1.46064      0.36508
   4     42     68.15    59.571      8.17      40     166     -1.76087     -0.27657
   5     38     89.02    49.874      9.22      55     178      0.55819     -1.67684
   6     47     77.45    44.811     11.63      58     176     -0.00113      1.40715
   7     40     75.98    45.681     11.95      70     176      0.95318     -0.48598
   8     43     81.19    49.091     10.85      64     162     -0.12951      0.36724
   9     44     81.42    39.442     13.08      63     174      0.66267      0.85740
  10     38     81.87    60.055      8.63      48     170     -0.44496     -1.53103
  11     44     73.03    50.541     10.13      45     168     -1.11832      0.55349
  12     45     87.66    37.388     14.03      56     186      1.20836      1.05948
```

Example 64.2. Regression Parameter Estimates

In this example, PROC REG computes regression parameter estimates for the Fitness data. (See Example 64.1 to create the Fitness data set.) The parameter estimates are output to a data set and used as scoring coefficients. For the first part of this example, PROC SCORE is used to score the Fitness data, which are the same data used in the regression.

In the second part of this example, PROC SCORE is used to score a new data set, Fitness2. For PROC SCORE, the TYPE= specification is PARMS, and the names of the score variables are found in the variable _MODEL_, which gets its values from the model label. The following code produces Output 64.2.1 through Output 64.2.3:

```
proc reg data=Fitness outest=RegOut;
OxyHat: model Oxygen=Age Weight RunTime RunPulse RestPulse;
   title 'REGRESSION SCORING EXAMPLE';
run;

proc print data=RegOut;
   title2 'OUTEST= Data Set from PROC REG';
run;

proc score data=Fitness score=RegOut out=RScoreP type=parms;
   var Age Weight RunTime RunPulse RestPulse;
run;

proc print data=RScoreP;
   title2 'Predicted Scores for Regression';
run;

proc score data=Fitness score=RegOut out=RScoreR type=parms;
```

```
   var Oxygen Age Weight RunTime RunPulse RestPulse;
run;

proc print data=RScoreR;
   title2 ’Negative Residual Scores for Regression’;
run;
```

Output 64.2.1 shows the PROC REG output. The column labeled “Parameter Estimates” lists the parameter estimates. These estimates are output to the RegOut data set.

Output 64.2.1. Creating an OUTEST= Data Set with PROC REG

```
                           REGRESSION SCORING EXAMPLE

                               The REG Procedure
                                 Model: oxyhat
                          Dependent Variable: Oxygen

                              Analysis of Variance

                                     Sum of           Mean
Source                   DF         Squares         Square    F Value    Pr > F

Model                     5       509.62201      101.92440      15.80    0.0021
Error                     6        38.70060        6.45010
Corrected Total          11       548.32261


             Root MSE              2.53970    R-Square     0.9294
             Dependent Mean       48.38942    Adj R-Sq     0.8706
             Coeff Var             5.24847


                              Parameter Estimates

                          Parameter       Standard
     Variable     DF       Estimate          Error    t Value    Pr > |t|

     Intercept     1      151.91550       31.04738       4.89      0.0027
     Age           1       -0.63045        0.42503      -1.48      0.1885
     Weight        1       -0.10586        0.11869      -0.89      0.4068
     RunTime       1       -1.75698        0.93844      -1.87      0.1103
     RunPulse      1       -0.22891        0.12169      -1.88      0.1090
     RestPulse     1       -0.17910        0.13005      -1.38      0.2176
```

Output 64.2.2 lists the RegOut data set. Notice that _TYPE_=’PARMS’ and _MODEL_=’OXYHAT’, which are from the label in the MODEL statement in PROC REG.

Output 64.2.2. OUTEST= Data Set from PROC REG Reproduced with PROC PRINT

```
                            REGRESSION SCORING EXAMPLE
                          OUTEST= Data Set from PROC REG


     Obs    _MODEL_    _TYPE_    _DEPVAR_     _RMSE_    Intercept       Age

      1     oxyhat     PARMS      Oxygen     2.53970     151.916     -0.63045

                                                  Rest
     Obs     Weight      RunTime    RunPulse      Pulse     Oxygen

      1     -0.10586    -1.75698    -0.22891    -0.17910      -1
```

Output 64.2.3 lists the data sets created by PROC SCORE. Since the SCORE= data set does not contain observations with _TYPE_='MEAN' or _TYPE_='STD', the data in the Fitness data set are not standardized before scoring. The SCORE= data set contains the variable Intercept, so this intercept value is used in computing the score. To produce the RScoreP data set, the VAR statement in PROC SCORE includes only the independent variables from the model in PROC REG. As a result, the OxyHat variable contains predicted values. To produce the RScoreR data set, the VAR statement in PROC SCORE includes both the dependent variables and the independent variables from the model in PROC REG. As a result, the OxyHat variable contains negative residuals (PREDICT−ACTUAL). If the RESIDUAL option is specified, the variable OxyHat contains positive residuals (ACTUAL−PREDICT). If the PREDICT option is specified, the OxyHat variable contains predicted values.

Output 64.2.3. Predicted and Residual Scores from the OUT= Data Set Created by PROC SCORE and Reproduced Using PROC PRINT

```
                        REGRESSION SCORING EXAMPLE
                      Predicted Scores for Regression

                                              Run      Rest     Run
     Obs     Age     Weight     Oxygen       Time     Pulse    Pulse      oxyhat

       1      44      89.47     44.609      11.37       62      178      42.8771
       2      40      75.07     45.313      10.07       62      185      47.6050
       3      44      85.84     54.297       8.65       45      156      56.1211
       4      42      68.15     59.571       8.17       40      166      58.7044
       5      38      89.02     49.874       9.22       55      178      51.7386
       6      47      77.45     44.811      11.63       58      176      42.9756
       7      40      75.98     45.681      11.95       70      176      44.8329
       8      43      81.19     49.091      10.85       64      162      48.6020
       9      44      81.42     39.442      13.08       63      174      41.4613
      10      38      81.87     60.055       8.63       48      170      56.6171
      11      44      73.03     50.541      10.13       45      168      52.1299
      12      45      87.66     37.388      14.03       56      186      37.0080



                        REGRESSION SCORING EXAMPLE
                  Negative Residual Scores for Regression

                                             Run      Rest     Run
    Obs     Age     Weight     Oxygen       Time     Pulse    Pulse      oxyhat

      1      44      89.47     44.609      11.37       62      178     -1.73195
      2      40      75.07     45.313      10.07       62      185      2.29197
      3      44      85.84     54.297       8.65       45      156      1.82407
      4      42      68.15     59.571       8.17       40      166     -0.86657
      5      38      89.02     49.874       9.22       55      178      1.86460
      6      47      77.45     44.811      11.63       58      176     -1.83542
      7      40      75.98     45.681      11.95       70      176     -0.84811
      8      43      81.19     49.091      10.85       64      162     -0.48897
      9      44      81.42     39.442      13.08       63      174      2.01935
     10      38      81.87     60.055       8.63       48      170     -3.43787
     11      44      73.03     50.541      10.13       45      168      1.58892
     12      45      87.66     37.388      14.03       56      186     -0.38002
```

The second part of this example uses the parameter estimates to score a new data set. The following code produces Output 64.2.4 and Output 64.2.5:

```
/* The FITNESS2 data set contains observations 13-16 from */
/* the FITNESS data set used in EXAMPLE 2 in the PROC REG */
/* chapter.                                                */
data Fitness2;
   input Age Weight Oxygen RunTime RestPulse RunPulse;
   datalines;
45  66.45  44.754  11.12  51  176
47  79.15  47.273  10.60  47  162
54  83.12  51.855  10.33  50  166
49  81.42  49.156   8.95  44  180
;

proc print data=Fitness2;
   title 'REGRESSION SCORING EXAMPLE';
   title2 'New Raw Data Set to be Scored';
run;

proc score data=Fitness2 score=RegOut out=NewPred type=parms
           nostd predict;
   var Oxygen Age Weight RunTime RunPulse RestPulse;
run;

proc print data=NewPred;
   title2 'Predicted Scores for Regression';
   title3 'for Additional Data from FITNESS2';
run;
```

Output 64.2.4 lists the Fitness2 data set.

Output 64.2.4. Listing of the Fitness2 Data Set

```
                         REGRESSION SCORING EXAMPLE
                       New Raw Data Set to be Scored

                                             Run      Rest      Run
       Obs     Age     Weight     Oxygen     Time     Pulse     Pulse

        1       45      66.45     44.754    11.12       51       176
        2       47      79.15     47.273    10.60       47       162
        3       54      83.12     51.855    10.33       50       166
        4       49      81.42     49.156     8.95       44       180
```

PROC SCORE scores the Fitness2 data set using the parameter estimates in the RegOut data set. These parameter estimates result from fitting a regression equation to the Fitness data set. The NOSTD option is specified, so the raw data are not standardized before scoring. (However, the NOSTD option is not necessary here. The SCORE= data set does not contain observations with _TYPE_='MEAN' or _TYPE_='STD', so standardization is not performed.) The VAR statement contains the dependent variables and the independent variables used in PROC REG. In addition, the PREDICT option is specified. This combination gives predicted values for the new score variable. The name of the new score variable is OxyHat, from the value of the _MODEL_ variable in the SCORE= data set. Output 64.2.5 shows the data set produced by PROC SCORE.

Output 64.2.5. Predicted Scores from the OUT= Data Set Created by PROC SCORE and Reproduced Using PROC PRINT

```
                         REGRESSION SCORING EXAMPLE
                      Predicted Scores for Regression
                     for Additional Data from FITNESS2

                                             Run      Rest      Run
  Obs      Age     Weight     Oxygen        Time     Pulse     Pulse       oxyhat

   1       45      66.45      44.754       11.12       51       176       47.5507
   2       47      79.15      47.273       10.60       47       162       49.7802
   3       54      83.12      51.855       10.33       50       166       43.9682
   4       49      81.42      49.156        8.95       44       180       47.5949
```

Example 64.3. Custom Scoring Coefficients

This example uses a specially created custom scoring data set and produces Output 64.3.1. The first scoring coefficient creates a variable that is Age−Weight; the second scoring coefficient evaluates the variable RunPulse−RstPulse; and the third scoring coefficient totals all six variables. Since the scoring coefficients data set (data set A) does not contain any observations with _TYPE_='MEAN' or _TYPE_='STD', the data in the Fitness data set (see Example 64.1) are not standardized before scoring.

```
data A;
   input _type_ $ _name_ $
         Age Weight RunTime RunPulse RestPulse;
   datalines;
SCORE   AGE_WGT   1 -1  0  0  0
SCORE   RUN_RST   0  0  0  1 -1
SCORE   TOTAL     1  1  1  1  1
;

proc print data=A;
   title 'CONSTRUCTED SCORING EXAMPLE';
   title2 'Scoring Coefficients';
run;

proc score data=Fitness score=A out=B;
   var Age Weight RunTime RunPulse RestPulse;
run;

proc print data=B;
   title2 'Scored Data';
run;
```

Output 64.3.1. Custom Scoring Data Set and Scored Fitness Data: PROC PRINT

```
                          CONSTRUCTED SCORING EXAMPLE
                              Scoring Coefficients

                                                          Run      Run       Rest
      Obs     _type_     _name_      Age     Weight      Time     Pulse     Pulse

       1      SCORE      AGE_WGT      1        -1          0        0          0
       2      SCORE      RUN_RST      0         0          0        1         -1
       3      SCORE      TOTAL        1         1          1        1          1
```

Output 64.3.2. Custom Scored Fitness Data: PROC PRINT

```
                          CONSTRUCTED SCORING EXAMPLE
                                  Scored Data

                                        Run     Rest     Run
Obs    Age    Weight    Oxygen     Time    Pulse    Pulse    AGE_WGT    RUN_RST     TOTAL

  1     44     89.47    44.609    11.37      62      178     -45.47       116      384.84
  2     40     75.07    45.313    10.07      62      185     -35.07       123      372.14
  3     44     85.84    54.297     8.65      45      156     -41.84       111      339.49
  4     42     68.15    59.571     8.17      40      166     -26.15       126      324.32
  5     38     89.02    49.874     9.22      55      178     -51.02       123      369.24
  6     47     77.45    44.811    11.63      58      176     -30.45       118      370.08
  7     40     75.98    45.681    11.95      70      176     -35.98       106      373.93
  8     43     81.19    49.091    10.85      64      162     -38.19        98      361.04
  9     44     81.42    39.442    13.08      63      174     -37.42       111      375.50
 10     38     81.87    60.055     8.63      48      170     -43.87       122      346.50
 11     44     73.03    50.541    10.13      45      168     -29.03       123      340.16
 12     45     87.66    37.388    14.03      56      186     -42.66       130      388.69
```

References

Chaseling, J. (1996), “Standard Test Results of Students at Three Types of Schools,” sample data, Faculty of Environmental Sciences, Griffith University: Queensland, Australia.

Chapter 65
The SIM2D Procedure

Chapter Contents

Chapter 65
The SIM2D Procedure

Overview

The SIM2D procedure produces a spatial simulation for a Gaussian random field with a specified mean and covariance structure in two dimensions using an LU decomposition technique.

The simulation can be conditional or unconditional. If it is conditional, a set of coordinates and associated field values are read from a SAS data set. The resulting simulation honors these data values.

You can specify the mean structure as a quadratic in the coordinates. You can specify the covariance by naming the form and supplying the associated parameters.

PROC SIM2D can handle anisotropic and nested semivariogram models. Three covariance models are supported: Gaussian, exponential, and spherical. A single nugget effect is also supported.

You can specify the locations of simulation points in a GRID statement or they can be read from a SAS data set. The grid specification is most suitable for a regular grid; the data set specification can handle any irregular pattern of points.

The SIM2D procedure writes the simulated values for each grid point to an output data set. The SIM2D procedure does not produce any displayed output.

Introduction to Spatial Simulation

The purpose of spatial simulation is to produce a set of partial realizations of a spatial random field (SRF) $Z(\mathbf{s}), \mathbf{s} \in D \subset \mathcal{R}^2$ in a way that preserves a specified mean $\mu(\mathbf{s}) = E\left[Z(\mathbf{s})\right]$ and covariance structure $C_z(\mathbf{s}_1 - \mathbf{s}_2) = cov\left(Z(\mathbf{s}_1), Z(\mathbf{s}_2)\right)$.

The realizations are partial in the sense that they occur only at a finite set of locations $(\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n)$. These locations are typically on a regular grid, but they can be arbitrary locations in the plane.

There are a number of different types of spatial simulation and associated computational methods. PROC SIM2D produces simulations for continuous processes in two dimensions. This means that the possible values of the measured quantity $Z(\mathbf{s}_0)$ at location $\mathbf{s}_0 = (x_0, y_0)$ can vary continuously over a certain range.

An additional assumption, needed for computational purposes, is that the spatial random field $Z(\mathbf{s})$ is Gaussian.

Spatial simulation is different from spatial prediction, where the emphasis is on producing a point estimate at a given grid location. In this sense, spatial prediction is local. In contrast, spatial simulation is global; the emphasis is on the entire realization $(Z(\mathbf{s}_1), Z(\mathbf{s}_2), \cdots, Z(\mathbf{s}_n))$.

Given the correct mean $\mu(\mathbf{s})$ and covariance structure $C_z(\mathbf{s}_1 - \mathbf{s}_2)$, SRF quantities that are difficult or impossible to calculate in a spatial prediction context can easily be approximated by repeated simulations.

Getting Started

Spatial simulation, just like spatial prediction, requires a model of spatial dependence, usually in terms of the covariance $C_z(h)$. For a given set of spatial data $Z(\mathbf{s}_i), i = 1, \cdots, n$, the covariance structure (both the form and parameter values) can be found by the VARIOGRAM procedure. This example uses the coal seam thickness data that is also used in the “Getting Started” section of Chapter 80, “The VARIOGRAM Procedure.”

Preliminary Spatial Data Analysis

In this example, the data consist of coal seam thickness measurements (in feet) taken over an approximately square area. The coordinates are offsets from a point in the southwest corner of the measurement area, with the north and east distances in units of thousands of feet.

It is instructive to see the locations of the measured points in the area where you want to perform spatial simulations. It is generally desirable to have these locations scattered evenly around the simulation area.

First, the data are input and the sample locations plotted.

```
data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;

proc gplot data=thick;
   title 'Locations of Measured Samples';
   plot north*east / frame cframe=ligr haxis=axis1
                     vaxis=axis2;
   symbol1 v=dot color=blue;
   axis1 minor=none;
   axis2 minor=none label=(angle=90 rotate=0);
   label east   = 'East'
         north  = 'North'
   ;
run;
```

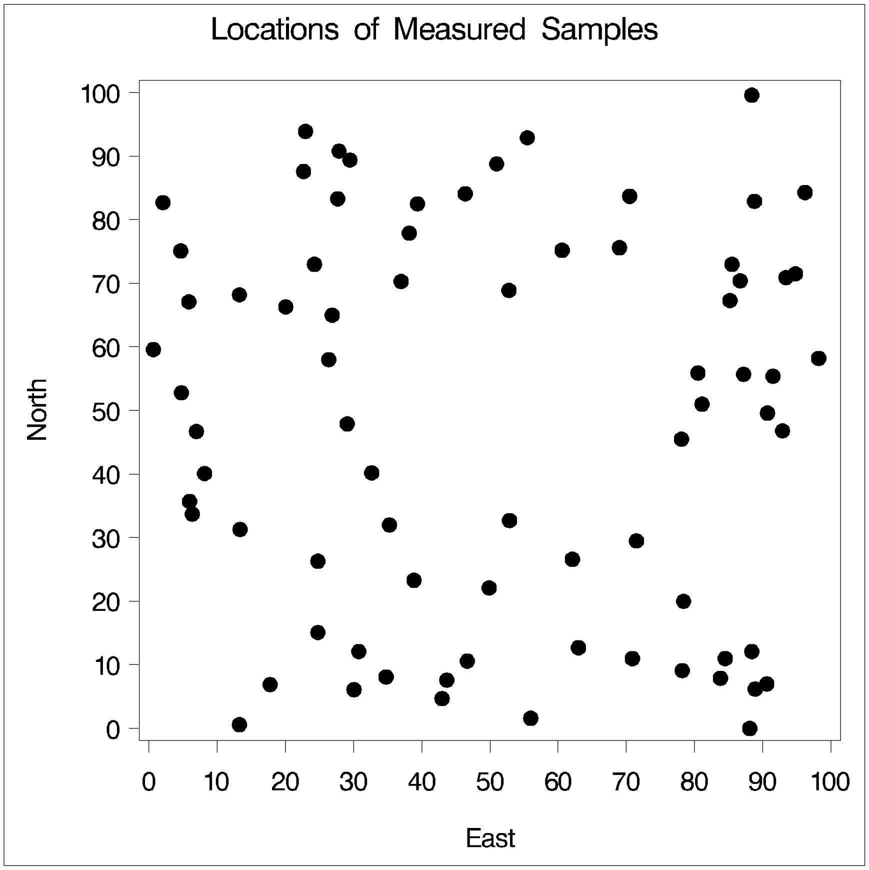


Figure 65.1. Locations of Measured Samples

```
proc g3d data=thick;
   title 'Surface Plot of Coal Seam Thickness';
   scatter east*north=thick / xticknum=5 yticknum=5
```

```
      grid zmin=20 zmax=65;
   label east  = 'East'
         north = 'North'
         thick = 'Thickness'
   ;
run;
```

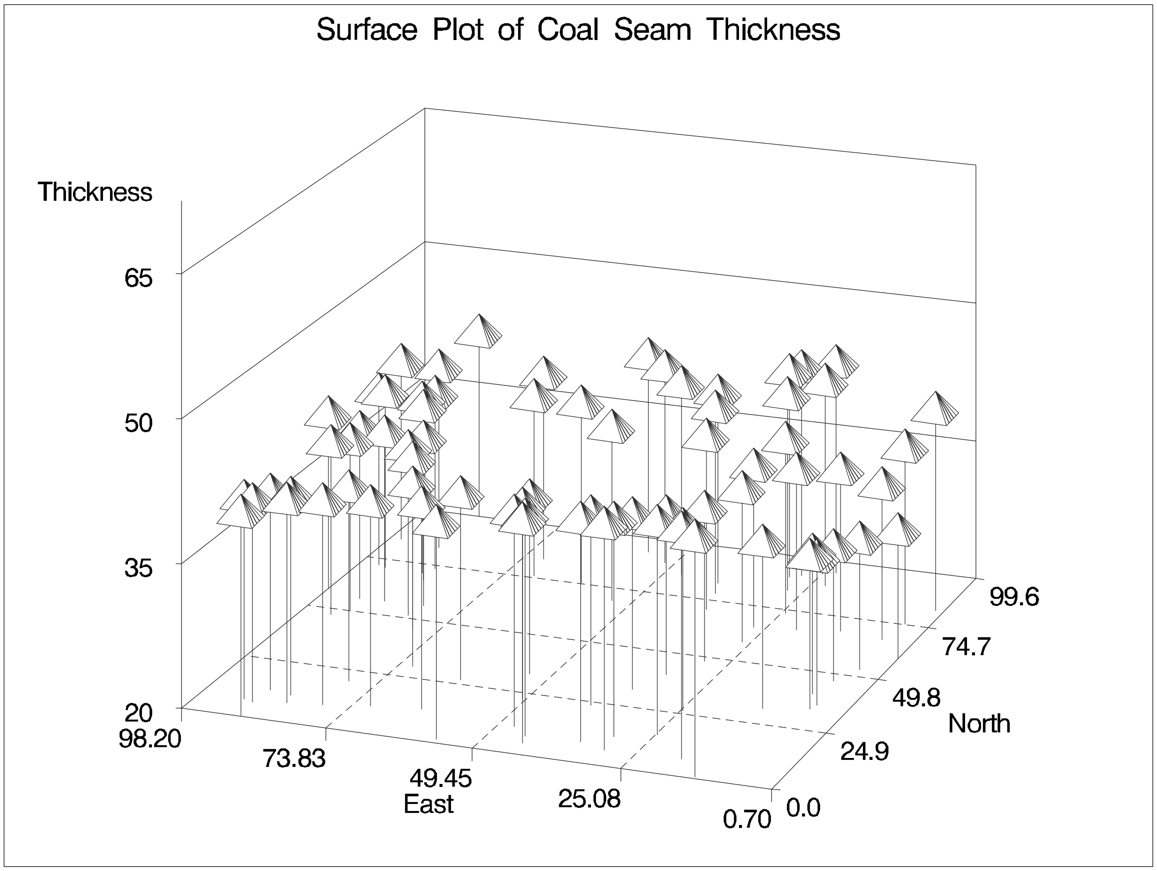


Figure 65.2. Surface Plot of Coal Seam Thickness

Figure 65.2 shows the small scale variation typical of spatial data, but there does not appear to be any surface trend. Hence, you can work with the original thickness data rather than residuals from a trend surface fit. In fact, a reasonable approximation of the spatial process generating the coal seam data is given by

$$Z(\mathbf{s}) = \mu + \varepsilon(\mathbf{s})$$

where the $\varepsilon(\mathbf{s})$ is a Gaussian SRF with Gaussian covariance structure

$$C_z(h) = c_0 \exp\left(-\frac{h^2}{a_0^2}\right)$$

Note that the term “Gaussian” is used in two ways in this description. For a set of locations $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n$, the random vector

$$\mathbf{Z}(\mathbf{s}) = \begin{bmatrix} Z(\mathbf{s}_1) \\ Z(\mathbf{s}_2) \\ \vdots \\ Z(\mathbf{s}_n) \end{bmatrix}$$

has a multivariate Gaussian or normal distribution $N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The (i,j)th element of $\boldsymbol{\Sigma}$ is computed by $C_z(\mathbf{s}_i - \mathbf{s}_j)$, which happens to be a Gaussian functional form. Any functional form for $C_z(\mathbf{h})$ yielding a valid covariance matrix $\boldsymbol{\Sigma}$ can be used. Both the functional form of $C_z(\mathbf{h})$ and the parameter values

$$\mu = 40.14$$
$$c_0 = 7.5$$
$$a_0 = 30.0$$

are visually estimated using PROC VARIOGRAM, a DATA step, and the GPLOT procedure. Refer to the “Getting Started” section beginning on page 4852 in the chapter on the VARIOGRAM procedure for details on how these parameter values are obtained.

The choice of a Gaussian functional form for $C_z(\mathbf{h})$ is simply based on the data, and it is not at all crucial to the simulation. However, it *is* crucial to the simulation method used in PROC SIM2D that $Z(\mathbf{s})$ be a Gaussian SRF. For details, see the section “Computational and Theoretical Details of Spatial Simulation” beginning on page 4106.

Investigating Variability by Simulation

The variability of $Z(\mathbf{s})$, modeled by

$$Z(\mathbf{s}) = \mu + \varepsilon(\mathbf{s})$$

with the Gaussian covariance structure $C_z(\mathbf{h})$ found previously is not obvious from the covariance model form and parameters. The variation around the mean of the surface is relatively small, making it difficult visually to pick up differences in surface plots of simulated realizations. Instead, you investigate variations at selected grid points.

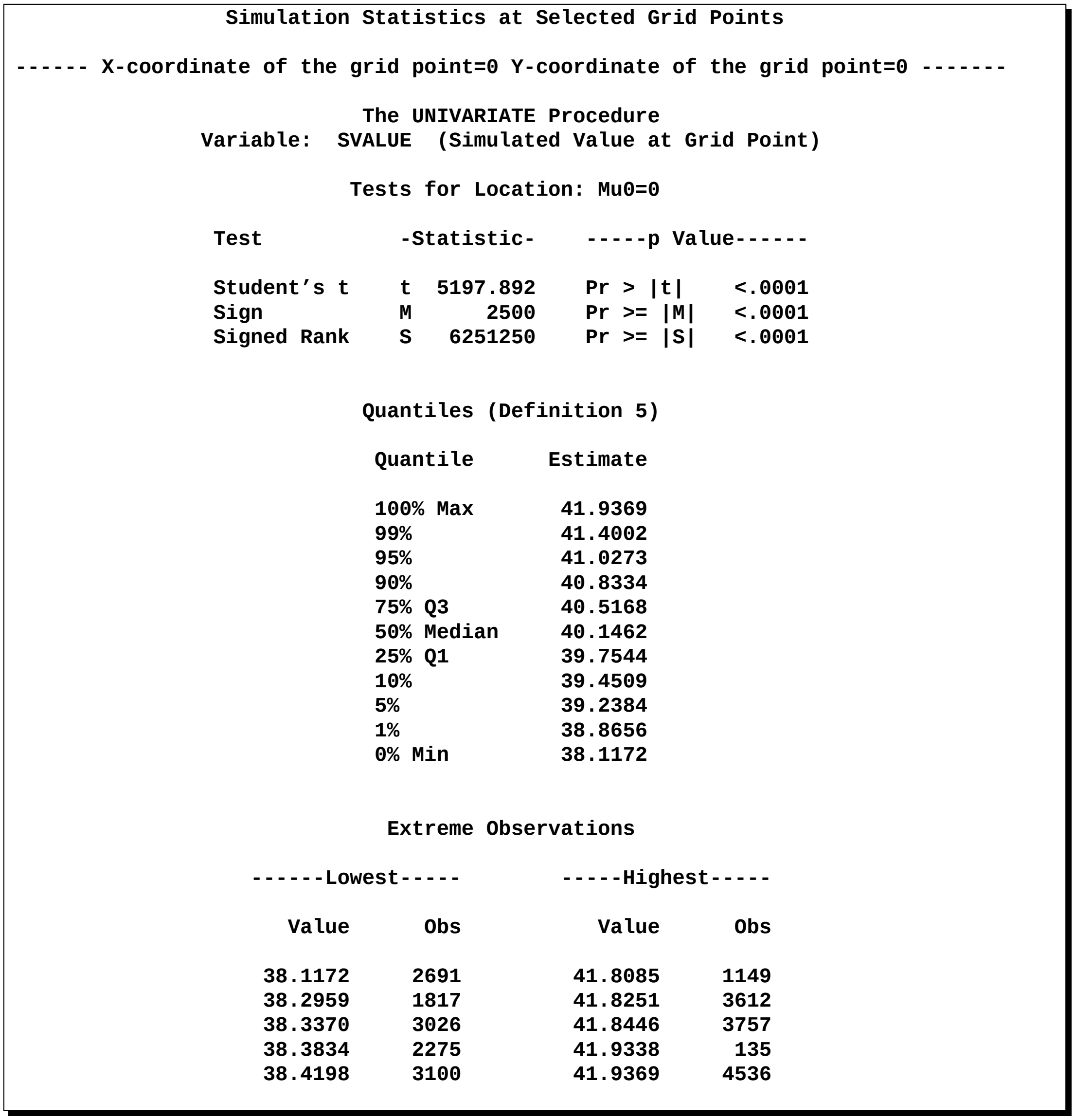
To do this investigation, this example uses PROC SIM2D and specifies the Gaussian model with the parameters found previously. Five thousand simulations (iterations) are performed on two points: the extreme south-west point of the region and a point towards the north-east corner of the region. Because of the irregular nature of these points, a GDATA= data set is produced with the coordinates of the selected points.

Summary statistics are computed for each of these grid points by using a BY statement in PROC UNIVARIATE.

```
data grid;
   input xc yc;
   datalines;
   0   0
   75  75
run;

proc sim2d data=thick outsim=sim1;
   simulate var=thick numreal=5000 seed=79931
      scale=7.5 range=30.0 form=gauss;
   mean 40.14;
   coordinates xc=east yc=north;
   grid gdata=grid xc=xc yc=yc;
run;

proc sort data=sim1;
   by gxc gyc;
run;

proc univariate data=sim1;
   var svalue;
   by gxc gyc;
   title 'Simulation Statistics at Selected Grid Points';
run;
```

```
                  Simulation Statistics at Selected Grid Points

------ X-coordinate of the grid point=0 Y-coordinate of the grid point=0 -------

                             The UNIVARIATE Procedure
                Variable:  SVALUE  (Simulated Value at Grid Point)

                                     Moments

        N                        5000    Sum Weights               5000
        Mean               40.1387121    Sum Observations    200693.561
        Std Deviation      0.54603592    Variance            0.29815523
        Skewness           -0.0217334    Kurtosis            -0.0519914
        Uncorrected SS     8057071.54    Corrected SS          1490.478
        Coeff Variation    1.36037231    Std Error Mean      0.00772211


                           Basic Statistical Measures

                  Location                    Variability

             Mean     40.13871     Std Deviation            0.54604
             Median   40.14620     Variance                 0.29816
             Mode       .          Range                    3.81973
                                   Interquartile Range      0.76236
```

Figure 65.3. Simulation Statistics at Grid Point (XC=0, YC=0)



```
                   Simulation Statistics at Selected Grid Points

------ X-coordinate of the grid point=0 Y-coordinate of the grid point=0 -------

                            The UNIVARIATE Procedure
               Variable:  SVALUE  (Simulated Value at Grid Point)

                           Tests for Location: Mu0=0

                Test           -Statistic-    -----p Value------

                Student's t    t  5197.892    Pr > |t|    <.0001
                Sign           M      2500    Pr >= |M|   <.0001
                Signed Rank    S   6251250    Pr >= |S|   <.0001


                            Quantiles (Definition 5)

                             Quantile      Estimate

                             100% Max       41.9369
                             99%            41.4002
                             95%            41.0273
                             90%            40.8334
                             75% Q3         40.5168
                             50% Median     40.1462
                             25% Q1         39.7544
                             10%            39.4509
                             5%             39.2384
                             1%             38.8656
                             0% Min         38.1172


                              Extreme Observations

                   ------Lowest-----         -----Highest-----

                      Value      Obs            Value      Obs

                    38.1172     2691          41.8085     1149
                    38.2959     1817          41.8251     3612
                    38.3370     3026          41.8446     3757
                    38.3834     2275          41.9338      135
                    38.4198     3100          41.9369     4536
```

Figure 65.4. Simulation Statistics at Grid Point (XC=75, YC=75)

Syntax

The following statements are available in PROC SIM2D.

> **PROC SIM2D** *options* ;
>   **COORDINATES** *coordinate-variables* ;
>   **GRID** *grid-options* ;
>   **SIMULATE** *simulate-options* ;
>   **MEAN** *mean-options* ;

The SIMULATE and MEAN statements are hierarchical; you can specify any number of SIMULATE statements, but you must specify at least one. If you specify a MEAN statement, it refers to the preceding SIMULATE statement. If you do not specify a MEAN statement, a zero mean model is simulated.

You must specify a single COORDINATES statement to identify the x and y coordinate variables in the input data set when you perform a conditional simulation. You must also specify a single GRID statement to specify the grid information.

The following table outlines the options available in PROC SIM2D classified by function.

Table 65.1. Options Available in the SIM2D Procedure

Task	Statement	Option
Data Set Options		
specify input data set	PROC SIM2D	DATA=
specify grid data set	GRID	GDATA=
specify quadratic form data set	MEAN	QDATA=
write simulated values	PROC SIM2D	OUTSIM=
Declaring the Role of Variables		
specify the conditioning variable	SIMULATE	VAR=
specify the x and y coordinate variables in the DATA= data set	COORDINATES	XC= YC=
specify the x and y coordinate variables in the GDATA= data set	GRID	XC= YC=
specify the constant coefficient variable in the QDATA= data set	MEAN	CONST=
specify the linear x coefficient variable in the QDATA= data set	MEAN	CX=
specify the linear y coefficient variable in the QDATA= data set	MEAN	CY=
specify the quadratic x coefficient variable in the QDATA= data set	MEAN	CXX=
specify the quadratic y coefficient variable in the QDATA= data set	MEAN	CYY=
specify the quadratic xy coefficient variable in the QDATA= data set	MEAN	CXY=
Controlling the Simulation		
specify the number of realizations	SIMULATE	NUMREAL=
specify the seed value for the random generator	SIMULATE	SEED=
Controlling the Mean Quadratic Surface		
specify the CONST term	MEAN	CONST=
specify the linear x term	MEAN	CX=
specify the linear y term	MEAN	CY=
specify the quadratic x term	MEAN	CXX=
specify the quadratic y term	MEAN	CYY=
specify the quadratic cross term	MEAN	CXY=

Table 65.1. (continued)

Task	Statement	Option
Controlling the Semivariogram Model		
specify a nugget effect	SIMULATE	NUGGET=
specify a functional form	SIMULATE	FORM=
specify nested functional forms	SIMULATE	FORM=($f_1, \cdots, f_k$)
specify a range parameter	SIMULATE	RANGE=
specify nested range parameters	SIMULATE	RANGE=($r_1, \cdots, r_k$)
specify a scale parameter	SIMULATE	SCALE=
specify nested scale parameters	SIMULATE	SCALE=($s_1, \cdots, s_k$)
specify an angle for an anisotropic model	SIMULATE	ANGLE=
specify nested angles	SIMULATE	ANGLE=($a_1, \cdots, a_k$)
specify a minor-major axis ratio for an anisotropic model	SIMULATE	RATIO=
specify nested minor-major axis ratios	SIMULATE	RATIO=($ra_1, \cdots, ra_k$)

PROC SIM2D Statement

PROC SIM2D *options* **;**

You can specify the following options with the PROC SIM2D statement.

DATA=*SAS-data-set*

specifies a SAS data set containing the x and y coordinate variables and the SIMULATE VAR= variables. This data set is required if any of the SIMULATE statements are conditional, that is, if you specify the VAR= option. If none of the SIMULATE statements are conditional then you do not need the DATA= option, and this option is ignored if you specify it.

NARROW

restricts the variables included in the OUTSIM= data set. When you specify the NARROW option, only four variables are included. This option is useful when a large number of simulations are produced. Including only four variables reduces the memory required for the OUTSIM= data set. For details on the variables that are excluded with the NARROW option, see the section "Output Data Set" on page 4110.

OUTSIM=*SAS-data-set*

specifies a SAS data set to store the simulation values, iteration number, simulate statement label, variable name, and grid location. For details, see the section "Output Data Set" on page 4110.

COORDINATES Statement

COORDINATES *coordinate-variables* **;**

The following two options give the name of the variables in the DATA= data set containing the values of the x and y coordinates of the conditioning data.

Only one COORDINATES statement is allowed, and it is applied to all SIMULATE statements that have a VAR= specification. In other words, it is assumed that all the VAR= variables in all SIMULATE statements have the same x and y coordinates.

You can abbreviate the COORDINATES statement as COORD.

XCOORD=*(variable-name)*
XC=*(variable-name)*
gives the name of the variable containing the x coordinate of the data in the DATA= data set.

YCOORD=*(variable-name)*
YC=*(variable-name)*
gives the name of the variable containing the y coordinate of the data locations in the DATA= data set.

GRID Statement

GRID *grid-options* **;**

The following options can be used to specify the grid of spatial locations at which to perform the simulations. A single GRID statement is required and is applied to all SIMULATE statements.

There are two basic methods for specifying the grid. You can specify the x and y coordinates explicitly, or they can be read from a SAS data set. The options for the explicit specification of grid locations are as follows.

X=*number*
X=$x_1, \ldots, x_m$
X=x_1 *to* x_m
X=x_1 *to* x_m *by* δx
specifies the x coordinate of the grid locations.

Y=*number*
Y=$y_1, \ldots, y_m$
Y=y_1 *to* y_m
Y=y_1 *to* y_m *by* δy
specifies the y coordinate of the grid locations.

For example, the following two GRID statements are equivalent:

```
grid x=1,2,3,4,5  y=0,2,4,6,8,10;
grid x=1 to 5 y=0 to 10 by 2;
```

To specify grid locations from a SAS data set, you must provide the name of the data set and the variables containing the values of the x and y coordinates.

GRIDDATA=*SAS-data-set*
GDATA=*SAS-data-set*
specifies a SAS data set containing the x and y grid coordinates.

XCOORD=*(variable-name)*
XC=*(variable-name)*

gives the name of the variable containing the x coordinate of the grid locations in the GRIDDATA= data set.

YCOORD=*(variable-name)*
YC=*(variable-name)*

gives the name of the variable containing the y coordinate of the grid locations in the GRIDDATA= data set.

SIMULATE Statement

SIMULATE *simulate-options* **;**

The SIMULATE statement specifies details on the simulation and the covariance model used in the simulation. You can specify the following options with a SIMULATE statement, which can be abbreviated by SIM.

NUMREAL=*number*
NUMR=*number*
NR=*number*

specifies the number of realizations to produce for the spatial process specified by the covariance model. Note that the number of observations in the OUTSIM= data set contributed by a given SIMULATE statement is the product of the NUMREAL= value with the number of grid points. This can cause the OUTSIM= data set to become large even for moderate values of the NUMREAL= option.

VAR= *(variable-name)*

specifies the single numeric variable used as the conditioning variable in the simulation. In other words, the simulation is conditional on the values of the VAR= variable found in the DATA= data set. If you omit the VAR= option, the simulation is *unconditional*. Since multiple SIMULATE statements are allowed, you can perform both unconditional and conditional simulations with a single PROC SIM2D statement.

Covariance Model Specification

There are two ways to specify a semivariogram or covariance model. In the first method, you can specify the required parameters SCALE, RANGE, and FORM, and possibly the optional parameters NUGGET, ANGLE, and RATIO, explicitly in the SIMULATE statement.

In the second method, you can specify an MDATA= data set. This data set contains variables corresponding to the required SCALE, RANGE, and FORM parameters, and, optionally, variables for the NUGGET, ANGLE, and RATIO parameters.

The two methods are exclusive; either you specify all parameters explicitly, or they are all are read from the MDATA= data set.

ANGLE=*angle*
ANGLE=*(angle$_1$,. . . ,angle$_k$)*

specifies the angle of the major axis for anisotropic models, measured in degrees clockwise from the N-S axis. In the case of a nested semivariogram model, you can specify an angle for each nesting. The default is ANGLE=0.

FORM=*form_spec*
FORM=*(form_spec$_1$, form_spec$_2$,. . . ,form_spec$_k$)*

specifies the functional form or forms of the semivariogram model, where *form_spec* can take only the values SPHERICAL, EXPONENTIAL, and GAUSSIAN. The two ways of specifying the FORM= parameter allows specification of both nested and nonnested models. The following abbreviations are permitted. For the spherical model, you can specify the *form_spec* as FORM=SPHERICAL, FORM=SPH, or FORM=S. For the exponential model, you can specify the *form_spec* as FORM=EXPONENTIAL, FORM=EXP, or FORM=E. For the Gaussian model, you can specify the *form_spec* as FORM=GAUSSIAN, FORM=GAUSS, or FORM=G.

MDATA=*SAS-data-set*

specifies the input data set that contains parameter values for the covariance or semivariogram model. The MDATA= data set must contain variables named SCALE, RANGE, and FORM, and it can optionally contain the variables NUGGET, ANGLE, and RATIO.

The FORM variables must be character, and they can assume the same values allowed in the explicit FORM= syntax described previously. The RANGE and SCALE variables must be numeric. The optional variables ANGLE, RATIO, and NUGGET must also be numeric if present.

The number of observations present in the MDATA= data set corresponds to the level of nesting of the covariance or semivariogram model. For example, to specify a nonnested model using a spherical covariance, an MDATA= data set might look like the following.

```
data md1;
   input scale range form $;
   datalines;
   25 10 sph
run;
```

The PROC SIM2D statement to use the MDATA= specification is of the form

```
proc sim2d data=...;
   sim var=.... mdata=md1;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc sim2d data=...;
   sim var=.... scale=25 range=10 form=sph;
run;
```

The following MDATA= data set is an example of an anisotropic nested model:

```
data md2;
   input scale range form $ nugget angle ratio;
   datalines;
   20 8 S  5 35 .7
   12 3 G  5 0  .8
   4  1 G  5 45 .5
   ;

proc sim2d data=...;
   sim var=.... mdata=md2;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc sim2d data=...;
   sim var=.... scale=(20,12,4) range=(8,3,1) form=(S,G,G)
                angle=(35,0,45) ratio=(.7,.8,.5) nugget=5;
run;
```

This example is somewhat artificial in that it is usually hard to detect different anisotropy directions and ratios for different nestings using an experimental semivariogram. Note that the NUGGET value is the same for all nestings. This is always the case; the nugget effect is a single additive term for all models. For further details, refer to the section “The Nugget Effect” on page 2051 in Chapter 37, “The KRIGE2D Procedure.”

The SIMULATE statement can be given a label. This is useful for identification in the OUTSIM= data set when multiple SIMULATE statements are specified.

For example,

```
proc sim2d data=...;
   gauss1: sim var=.... form=gauss;
   mean ....;
   gauss2: sim var=.... form gauss;
   mean ....;
   exp1: sim var=.... form=exp;
   mean ....;
   exp2: sim var=.... form=exp;
   mean ....;
run;
```

In the OUTSIM= data set, the values ’GAUSS1’, ’GAUSS2’, ’EXP1’, and ’EXP2’ for the LABEL variable help to identify the realizations corresponding to the four SIMULATE statements. If you do not provide a label for a SIMULATE statement, a default label of SIMn is given, where n is the number of unlabeled SIMULATE statements seen so far.

NUGGET=*number*

specifies the nugget effect for the model. This effect is due to a discontinuity in the semivariogram as determined by plotting the sample semivariogram (refer to the section "The Nugget Effect" on page 2051 in the chapter on the KRIGE2D procedure for details). For models without any nugget effect, the NUGGET= option is left out. The default is NUGGET=0.

RANGE=*range*
RANGE=*(range$_1$,. . . ,range$_k$)*

specifies the range parameter in the semivariogram models. In the case of a nested semivariogram model, you must specify a range for each nesting.

The range parameter is the divisor in the exponent in all supported models. It has the units of distance or distance squared for these models, and it is related to the correlation scale for the underlying spatial process. Refer to the section "Theoretical Semivariogram Models" beginning on page 2045 in the chapter on the KRIGE2D procedure for details on how the RANGE= values are determined.

RATIO=*ratio*
RATIO=*(ratio$_1$,. . . ,ratio$_k$)*

specifies the ratio of the length of the minor axis to the length of the major axis for anisotropic models. The value of the RATIO= option must be between 0 and 1. In the case of a nested semivariogram model, you can specify a ratio for each nesting. The default is RATIO=1.

SCALE=*scale*
SCALE=*(scale$_1$,. . . ,scale$_k$)*

specifies the scale parameter in semivariogram models. In the case of a nested semivariogram model, you must specify a scale for each nesting.

The scale parameter is the multiplicative factor in all supported models; it has the same units as the variance of the VAR= variable. Refer to the section "Theoretical Semivariogram Models" beginning on page 2045 in the chapter on the KRIGE2D procedure for details on how the SCALE= values are determined.

SEED=*seed value*

specifies the seed to use for the random number generator. If you omit the SEED= value, the system clock is used.

SINGULAR=*number*

gives the singularity criteria for solving the set of linear equations involved in the computation of the mean and covariance of the conditional distribution associated with a given SIMULATE statement. The larger the value of the SINGULAR= option, the easier it is for the covariance matrix system to be declared singular. The default is SINGULAR=1E-8. For more details on the use of the SINGULAR= option, see the section "Computational and Theoretical Details of Spatial Simulation" beginning on page 4106.

MEAN Statement

MEAN *spec*$_1$,. . . ,*spec*$_6$ **;**

MEAN QDATA=*SAS-data-set CONST=var*$_1$ *CX=var*$_2$ *CY=var*$_3$
CXX=var$_4$ *CYY=var*$_5$ *CXY=var*$_6$ **;**

MEAN QDATA=*SAS-data-set* **;**

A mean function $\mu(s)$ that is a quadratic in the coordinates can be written

$$\mu(s) = \mu(x, y) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy$$

The MEAN statement is used to specify the quadratic surface to use as the mean function for the simulated SRF. There are three ways to specify the MEAN statement. The MEAN statement allows the specification of the coefficients $\beta_0, \cdots, \beta_5$ either explicitly or through a QDATA= data set.

An example of an explicit specification is

```
mean 1.4 + 2.5*x + 3.6*y + .47*x*x + .58*y*y + .69*x*y;
```

In this example, all terms have a nonzero coefficient. Any term with a zero coefficient is simply left out of the specification. For example,

```
mean 1.4;
```

is a valid quadratic form with all terms having zero coefficients except the constant term.

An equivalent way of specifying the mean function is through the QDATA= data set. For example, the following MEAN statement

```
mean 1.4 + 2.5*x + 3.6*y + .47*x*x + .58*y*y + .69*x*y;
```

can be alternatively specified by the following DATA step and MEAN statement:

```
data q1;
   input c1 c2 c3 c4 c5 c6;
   datalines;
   1.4 2.5 3.6 .47 .58 .69
run;
proc sim2d data=....;
   simulate ...;
   mean qdata=q1 const=c1 cx=c2 cy=c3 cxx=c4
           cyy=c5 cxy=c6;
run;
```

The QDATA= data set specifies the data set containing the coefficients. The parameters CONST=, CX=, CY=, CXX=, CYY=, and CYX= specify the variables in the QDATA= data set that correspond to the constant, linear x, linear y, and so on. For any coefficient not specified in this list, the QDATA= data set is checked for the presence of variables with default names of CONST, CX, CY, CXX, CYY, and CXY. If these variables are present, their values are taken as the corresponding coefficients. Hence, you can rewrite the previous example as

```
data q1;
   input const cx cy cxx cyy cxy;
   datalines;
   1.4 2.5 3.6 .47 .58 .69
   ;
proc sim2d data=....;
   simulate ...;
   mean qdata=q1;
run;
```

If a given coefficient does not appear in the list or in the data set with the default name, a value of zero is assumed.

Details

Computational and Theoretical Details of Spatial Simulation

Introduction

There are a number of approaches to simulating spatial random fields or, more generally, simulating sets of dependent random variables. This includes sequential indicator methods, turning bands, and the Karhunen-Loeve Expansion. Refer to Christakos (1992, Chapter 8) and Duetsch and Journel (1992, Chapter V) for details.

A particularly simple method available for Gaussian spatial random fields is the LU decomposition method. This method is computationally efficient. For a given covariance matrix, the $LU = \mathbf{L}\mathbf{L}^T$ decomposition is computed once, and the simulation proceeds by repeatedly generating a vector of independent $N(0,1)$ random variables and multiplying by the **L** matrix.

One problem with this technique is memory requirements; memory is required to hold the full data and grid covariance matrix in core. While this is especially limiting in the three-dimensional case, you can use PROC SIM2D, which handles only two-dimensional data, for moderately sized simulation problems.

Theoretical Development

It is a simple matter to produce an $N(0,1)$ random number, and by stacking k $N(0,1)$ random numbers in a column vector, you can obtain a vector with independent standard normal components $\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$. The meaning of the terms *independence* and *randomness* in the context of a deterministic algorithm required for the generation of these numbers is a little subtle; refer to Knuth (1981, Vol. 2, Chapter 3) for details.

Rather than $\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$, what is required is the generation of a vector $\mathbf{Z} \sim N_k(\mathbf{0}, \mathbf{C})$, that is,

$$\mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_k \end{bmatrix}$$

with covariance matrix

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1k} \\ C_{21} & C_{22} & \cdots & C_{2k} \\ & \ddots & & \\ C_{k1} & C_{k2} & \cdots & C_{kk} \end{pmatrix}$$

If the covariance matrix is symmetric and positive definite, it has a Cholesky root $\mathbf{L}$ such that $\mathbf{C}$ can be factored as

$$\mathbf{C} = \mathbf{L}\mathbf{L}^T$$

where $\mathbf{L}$ is lower triangular. Refer to Ralston and Rabinowitz (1978, Chapter 9, Section 3-3) for details. This vector $\mathbf{Z}$ can be generated by the transformation $\mathbf{Z} = \mathbf{L}\mathbf{W}$. Note that this is where the assumption of a Gaussian SRF is crucial. When $\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$, then $\mathbf{Z} = \mathbf{L}\mathbf{W}$ is also Gaussian. The mean of $\mathbf{Z}$ is

$$E(\mathbf{Z}) = \mathbf{L}(E(\mathbf{W})) = \mathbf{0}$$

and the variance is

$$\text{Var}(\mathbf{Z}) = \text{Var}(\mathbf{L}\mathbf{W}) = E(\mathbf{L}\mathbf{W}\mathbf{W}^T\mathbf{L}^T) = \mathbf{L}E(\mathbf{W}\mathbf{W}^T)\mathbf{L}^T = \mathbf{L}\mathbf{L}^T = \mathbf{C}$$

Consider now an SRF $Z(\mathbf{s}), \mathbf{s} \in D \subset \mathcal{R}^2$, with spatial covariance function $C(\mathbf{h})$. Fix locations $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_k$, and let $\mathbf{Z}$ denote the random vector

$$\mathbf{Z} = \begin{bmatrix} Z(\mathbf{s}_1) \\ Z(\mathbf{s}_2) \\ \vdots \\ Z(\mathbf{s}_k) \end{bmatrix}$$

with corresponding covariance matrix

$$\mathbf{C}_z = \begin{pmatrix} C(\mathbf{0}) & C(\mathbf{s}_1 - \mathbf{s}_2) & \cdots & C(\mathbf{s}_1 - \mathbf{s}_k) \\ C(\mathbf{s}_2 - \mathbf{s}_1) & C(\mathbf{0}) & \cdots & C(\mathbf{s}_2 - \mathbf{s}_k) \\ & & \ddots & \\ C(\mathbf{s}_k - \mathbf{s}_1) & C(\mathbf{s}_k - \mathbf{s}_2) & \cdots & C(\mathbf{0}) \end{pmatrix}$$

Since this covariance matrix is symmetric and positive definite, it has a Cholesky root, and the $Z(\mathbf{s}_i), i = 1, \cdots, k$ can be simulated as described previously. This is how the SIM2D procedure implements unconditional simulation in the zero mean case. More generally,

$$Z(\mathbf{s}) = \mu(\mathbf{s}) + \varepsilon(\mathbf{s})$$

with $\mu(\mathbf{s})$ being a quadratic form in the coordinates $\mathbf{s} = (x, y)$, and the $\varepsilon(\mathbf{s})$ being an SRF having the same covariance matrix $\mathbf{C}_z$ as previously. In this case, the $\mu(\mathbf{s}_i), i = 1, \cdots, k$ is computed once and added to the simulated vector $\varepsilon(\mathbf{s}_i), i = 1, \cdots, k$ for each realization.

For a conditional simulation, this distribution of

$$\mathbf{Z} = \left[\begin{array}{c} Z(\mathbf{s}_1) \\ Z(\mathbf{s}_2) \\ \vdots \\ Z(\mathbf{s}_k) \end{array} \right]$$

must be conditioned on the observed data. The relevant general result concerning conditional distributions of multivariate normal random variables is the following. Let $\mathbf{X} \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\mathbf{X} = \left[\begin{array}{c} \mathbf{X}_1 \\ \mathbf{X}_2 \end{array} \right]$$

$$\boldsymbol{\mu} = \left[\begin{array}{c} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{array} \right]$$

and

$$\boldsymbol{\Sigma} = \left(\begin{array}{cc} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{array} \right)$$

The subvector $\mathbf{X}_1$ is $k \times 1$, $\mathbf{X}_2$ is $n \times 1$, $\boldsymbol{\Sigma}_{11}$ is $k \times k$, $\boldsymbol{\Sigma}_{22}$ is $n \times n$, and $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^T$ is $k \times n$, with $k + n = m$. The full vector $\mathbf{X}$ is partitioned into two subvectors $\mathbf{X}_1$ and $\mathbf{X}_2$, and $\boldsymbol{\Sigma}$ is similarly partitioned into covariances and cross covariances.

With this notation, the distribution of $\mathbf{X}_1$ conditioned on $\mathbf{X}_2 = \mathbf{x}_2$ is $N_k(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$, with

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)$$

and

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$$

Refer to Searle (1971, pp. 46–47) for details. The correspondence with the conditional spatial simulation problem is as follows. Let the coordinates of the observed data points be denoted $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_n$, with values $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_n$. Let $\tilde{\mathbf{Z}}$ denote the random vector

$$\tilde{\mathbf{Z}} = \begin{bmatrix} Z(\tilde{\mathbf{s}}_1) \\ Z(\tilde{\mathbf{s}}_2) \\ \vdots \\ Z(\tilde{\mathbf{s}}_n) \end{bmatrix}$$

The random vector $\tilde{\mathbf{Z}}$ corresponds to $\mathbf{X}_2$, while $\mathbf{Z}$ corresponds to $\mathbf{X}_1$. Then $\left(\mathbf{Z} \mid \tilde{\mathbf{Z}} = \tilde{\mathbf{z}}\right) \sim N_k(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{C}})$ as in the previous distribution. The matrix

$$\tilde{\mathbf{C}} = \mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$$

is again positive definite, so a Cholesky factorization can be performed.

The dimension n for $\tilde{\mathbf{Z}}$ is simply the number of nonmissing observations for the VAR= variable; the values $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_n$ are the values of this variable. The coordinates $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_n$ are also found in the DATA= data set, with the variables corresponding to the x and y coordinates identified in the COORDINATES statement. Note that all VAR= variables use the same set of conditioning coordinates; this fixes the matrix $\mathbf{C}_{22}$ for all simulations.

The dimension k for $\mathbf{Z}$ is the number of grid points specified in the GRID statement. Since there is a single GRID statement, this fixes the matrix $\mathbf{C}_{11}$ for all simulations. Similarly, $\mathbf{C}_{12}$ is fixed.

The Cholesky factorization $\tilde{\mathbf{C}} = \mathbf{L}\mathbf{L}^T$ is computed once, as is the mean correction

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \mathbf{C}_{12}\mathbf{C}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2)$$

Note that the means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are computed using the grid coordinates $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_k$, the data coordinates $\tilde{\mathbf{s}}_1, \tilde{\mathbf{s}}_2, \cdots, \tilde{\mathbf{s}}_n$, and the quadratic form specification from the MEAN statement. The simulation is now performed exactly as in the unconditional case. A $k \times 1$ vector of independent standard $N(0,1)$ random variables is generated and multiplied by $\mathbf{L}$, and $\tilde{\boldsymbol{\mu}}$ is added to the transformed vector. This is repeated N times, where N is the value specified for the NR= option.

Computational Details

In the computation of $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\Sigma}$ described in the previous section, the inverse $\boldsymbol{\Sigma}_{22}^{-1}$ is never actually computed; an equation of the form

$$\boldsymbol{\Sigma}_{22}\mathbf{A} = \mathbf{B}$$

is solved for $\mathbf{A}$ using a modified Gaussian elimination algorithm that takes advantage of the fact that $\boldsymbol{\Sigma}_{22}$ is symmetric with constant diagonal $C_z(0)$ that is larger than all off-diagonal elements. The SINGULAR= option pertains to this algorithm. The value specified for the SINGULAR= option is scaled by $C_z(0)$ before comparison with the pivot element.

Memory Usage

For conditional simulations, the largest matrix held in core at any one time depends on the number of grid points and data points. Using the previous notation, the data-data covariance matrix $\mathbf{C}_{22}$ is $n \times n$, where n is the number of nonmissing observations for the VAR= variable in the DATA= data set. The grid-data cross covariance $\mathbf{C}_{12}$ is $n \times k$, where k is the number of grid points. The grid-grid covariance $\mathbf{C}_{11}$ is $k \times k$. The maximum memory required at any one time for storing these matrices is

$$\max\left(k(k+1), n(n+1) + 2(n \times k)\right) \times \text{sizeof}(\text{double})$$

There are additional memory requirements that add to the total memory usage, but usually these matrix calculations dominate, especially when the number of grid points is large.

Output Data Set

The SIM2D procedure produces a single output data set: the OUTSIM=*SAS-data-set*. The OUTSIM= data set contains all the needed information to uniquely identify the simulated values.

The OUTSIM= data set contains the following variables:

- LABEL, which is the label for the current SIMULATE statement
- VARNAME, which is the name of the conditioning variable for the current SIMULATE statement
- _ITER_, which is the iteration number within the current SIMULATE statement
- GXC, which is the x-coordinate for the current grid point
- GYC, which is the y-coordinate for the current grid point
- SVALUE, which is the value of the simulated variable

If you specify the NARROW option in the PROC SIM2D statement, the LABEL and VARNAME variables are not included in the OUTSIM= data set. This option is useful in the case where the number of data points, grid points, and realizations are such that they generate a very large OUTSIM= data set. The size of the OUTSIM= data set is reduced when these variables are not included.

In the case of an unconditional simulation, the VARNAME variable is not included. In the case of mixed conditional and unconditional simulations (that is, when multiple SIMULATE statements are specified and one or more contain a VAR= specification and one or more do *not* contain a VAR= specification), the VARNAME variable is included but is given a missing value for those observations corresponding to an unconditional simulation.

Example

Example 65.1. Simulation

Continuing with the coal seam thickness example from the "Getting Started" section beginning on page 4092, this example asks a more complicated question. This question is economic in nature, and the (approximate) answer requires the use of simulation.

Simulating a Subregion for Economic Feasibility

The coal seam must be of a minimum thickness, called a *cutoff value*, for a mining operation to be profitable. Suppose that, for a subregion of the measured area, the cost of mining is higher than the remaining areas due to the geology of the overburden. This higher cost results in a higher thickness cutoff value for the subregion. Suppose also that it is determined from a detailed cost analysis that at least 60 percent of the subregion must exceed a seam thickness of 39.7 feet for profitability.

How can you use the SRF model (μ and $C_z(\mathbf{s})$) and the measured seam thickness values $Z(\mathbf{s}_i), i = 1, \cdots, 75$ to determine, in some approximate way, if at least 60 percent of the subregion exceeds this minimum?

Spatial prediction does not appear to be helpful in answering this question. While it is easy to determine if a predicted value at a location in the subregion is above the 39.7 feet cutoff value, it is not clear how to incorporate the standard error associated with the predicted value. The standard error is what characterizes the stochastic nature of the prediction (and the underlying SRF). It is clear that it must be included in any realistic approach to the problem.

A conditional simulation, on the other hand, seems to be a natural way of obtaining an approximate answer. By simulating the SRF on a sufficiently fine grid in the subregion, you can determine the proportion of grid points in which the mean value over realizations exceeds the 39.7 feet cutoff and compare it with the 60 percent value needed for profitability.

It is desirable in any simulation study that the quantity being estimated (in this case, the proportion exceeding the 39.7 feet cutoff) not depend on the number of simulations performed. For example, suppose that the maximum seam thickness is simulated. It is likely that the maximum value increases as the number of simulations performed increases. Hence, a simulation is not useful for such an estimate. A simulation is useful for determining the *distribution* of the maximum, but there are general theoretical results for such distributions, making such a simulation unnecessary. Refer to Leadbetter, Lindgren, and Rootzen (1983) for details.

In the case of simulating the proportion exceeding the 39.7 feet cutoff, it is expected that this quantity will settle down to a fixed value as the number of realizations increases. At a fixed grid point, the quantity being compared with the cutoff value is the mean over all simulated realizations; this mean value settles down to a fixed number as the number of realizations increases. In the same manner, the proportion of the grid where the mean values exceed the cutoff also becomes constant. This can be tested using PROC SIM2D.

A crucial, nonprovable assumption in applying SRF theory to the coal seam thickness data is that the values $Z(\mathbf{s}_i), i = 1, \cdots, 75$ represent a *single* realization from the set of all possible realizations consistent with the SRF model (μ and $C_z(\mathbf{h})$). A conditional simulation repeatedly produces other possible simulated realizations consistent with the model and data. However, the only concern of the mining company is with this single unique realization. It is not concerned with similar coal fields to be mined sometime in the future; it may never see another coal field remotely similar to this one, or it may not be in business in the future.

Hence the proportion found by generating repeated simulated realizations must somehow relate back to the unique realization that is the coal field (seam thickness). This is done by interpreting the proportion found from a simulation to the spatial mean proportion for the unique realization. The term “spatial mean” is simply an appropriate integral over the fixed (but unknown) spatial function $z(\mathbf{s})$. (The SRF is denoted $Z(\mathbf{s})$; a particular realization, a deterministic function of the spatial coordinates, is denoted $z(\mathbf{s})$.)

This interpretation requires an ergodic assumption, which is also needed in the original estimation of $C_z(\mathbf{s})$. Refer to Cressie (1993, pp. 53–58) for a discussion of ergodicity and Gaussian SRFs.

Implementation Using PROC SIM2D

The subregion to be considered is the southeast corner of the field, which is a square region with length 40 distance units (in thousands of feet). PROC SIM2D is run on the entire data set for conditioning, while the simulation grid covers only this subregion. It is convenient to be able to vary the seed, the grid increment, and the number of simulations performed. The following macro implements the computation of the percent area exceeding the cutoff value by using the seed, the grid increment, and the number of simulated realizations as macro arguments.

The data set produced by PROC SIM2D is transposed so that each grid location is a separate variable. The MEANS procedure is then used to average the simulated value at each grid point over all realizations. It is this average that is compared to the cutoff value. The last DATA step does the comparison and determines the percent of the grid locations that exceed this cutoff value and writes the results to the listing file in the form of a report.

The macro is first invoked with a relatively coarse grid (grid increment of 10 distance units) and a small number of realizations (5). The next invocation uses a finer grid and 50 realizations, and the final invocation uses the same grid increment and 500 realizations. Each time, the macro is invoked with a different seed. The simulations indicate that around 87 percent of the subregion exceeds the cutoff value.

The number of grid points in the simulation increases with the square of the decrease in the grid increment, leading to long CPU processing times. Increasing the number of realizations results in a linear increase in processing times. Hence, using as coarse a grid as possible allows more realizations and experimentation with different seeds.

```
/*- Set the covariance model parameters and cutoff value -*/
```

```
%let cc0=7.5;
%let aa0=30.0;
%let form=gauss;
%let cut=39.7;

data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;


%macro area_sim(seed=,nr=,ginc=);

   %let ngrid=%eval(40/&ginc+1);
   %let tgrid=%eval(&ngrid*&ngrid);

   proc sim2d data=thick outsim=sim1;
      simulate var=thick numreal=&nr seed=&seed
         scale=&cc0 range=&aa0 form=&form;
      mean 40.14;
      coordinates xc=east yc=north;
      grid x=60 to 100 by &ginc
           y=0 to 40 by &ginc;
   run;

   proc transpose data=sim1 out=sim2 prefix=sims;
      by _iter_;
      var svalue;
   run;
   proc means data=sim2 noprint n mean;
      var sims1-sims&tgrid;
      output out=msim n=numsim mean=ms1-ms&tgrid;
   run;

   /*- Determine the percentage of sites exceeding cutoff -*/
   data _null_;
      file print;
      array simss ms1-ms&tgrid;
      set msim;

   /*- Loop over the grid sites to test cutoff  -*/
      cflag=0;
      do ss=1 to &tgrid;
         tempv=simss[ss];
         if simss[ss] > &cut then do;
            cflag + 1;
         end;
      end;

      area_per=100*(cflag/&tgrid);
      put // +5 'Conditional Simulation of Coal Seam'
                ' Thickness for Subregion';
      put / +5 'Subregion is South-East Corner 40 by 40'
               ' distance units';
      put / +5 "Seed:&seed" +2 "Grid Increment:&ginc";
      put / +5 "Total Number of Grid Points:&tgrid" +2
               "Number of Simulations:&nr";
      put / +5 "Percent of Subregion Exceeding Cutoff of
                &cut ft.:"
                +2 area_per 5.2;
   run;
%mend area_sim;



%area_sim(seed=12345,nr=5,ginc=10);
%area_sim(seed=54321,nr=50,ginc=1);
%area_sim(seed=655311,nr=500,ginc=1);
```

Output 65.1.1. Conditional Simulation of Coal Seam Thickness

```
Conditional Simulation of Coal Seam Thickness for Subregion

Subregion is South-East Corner 40 by 40 distance units

Seed:12345  Grid Increment:10

Total Number of Grid Points:25  Number of Simulations:5

Percent of Subregion Exceeding Cutoff of 39.7 ft.:  80.00
```

```
      Conditional Simulation of Coal Seam Thickness for Subregion

      Subregion is South-East Corner 40 by 40 distance units

      Seed:54321  Grid Increment:1

      Total Number of Grid Points:1681  Number of Simulations:50

      Percent of Subregion Exceeding Cutoff of 39.7 ft.:  88.34
```

```
      Conditional Simulation of Coal Seam Thickness for Subregion

      Subregion is South-East Corner 40 by 40 distance units

      Seed:655311  Grid Increment:1

      Total Number of Grid Points:1681  Number of Simulations:500

      Percent of Subregion Exceeding Cutoff of 39.7 ft.:  87.63
```

References

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Cressie, N.A.C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons, Inc.

Duetsch, C.V. and Journel, A.G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Knuth, D.E., (1981), *The Art of Computer Programming: Seminumerical Algorithms*, Vol. 2, Second Edition, Reading, MA: Addison-Wesley.

Leadbetter, M.R., Lindgren, G. and Rootzen, H. (1983), *Extremes and Related Properties of Random Sequences and Processes*, New York: Springer-Verlag.

Ralston, A. and Rabinowitz, P. (1978), *A First Course in Numerical Analysis*, Second Edition, New York: McGraw-Hill, Inc.

Searle, S.R. (1971), *Linear Models*, New York: John Wiley & Sons, Inc.

Chapter 66
The STDIZE Procedure

Chapter Contents

Chapter 66
The STDIZE Procedure

Overview

The STDIZE procedure standardizes one or more numeric variables in a SAS data set by subtracting a location measure and dividing by a scale measure. A variety of location and scale measures are provided, including estimates that are resistant to outliers and clustering. Some of the well-known standardization methods such as mean, median, std, range, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate are available in the STDIZE procedure.

In addition, you can multiply each standardized value by a constant and add a constant. Thus, the final output value is

$$result = add + multiply \times \frac{original - location}{scale}$$

where

result = final output value
add = constant to add (ADD= option)
multiply = constant to multiply by (MULT= option)
original = original input value
location = location measure
scale = scale measure

PROC STDIZE can also find quantiles in one pass of the data, a capability that is especially useful for very large data sets. With such data sets, the UNIVARIATE procedure may have high or excessive memory or time requirements.

Getting Started

The following example demonstrates how you can use the STDIZE procedure to obtain location and scale measures of your data.

In the following hypothetical data set, a random sample of grade 12 students is selected from a number of co-educational schools. Each school is classified as one of two types: Urban or Rural. There are 40 observations.

The variables are id (student identification), Type (type of school attended: 'urban'=urban area and 'rural'=rural area), and total (total assessment scores in History, Geometry, and Chemistry).

The following DATA step creates the SAS data set TotalScores.

```
data TotalScores;
   title ’High School Scores Data’;
   input id Type $  total;
   datalines;
 1      rural        135
 2      rural        125
 3      rural        223
 4      rural        224
 5      rural        133
 6      rural        253
 7      rural        144
 8      rural        193
 9      rural        152
10      rural        178
11      rural        120
12      rural        180
13      rural        154
14      rural        184
15      rural        187
16      rural        111
17      rural        190
18      rural        128
19      rural        110
20      rural        217
21      urban        192
22      urban        186
23      urban         64
24      urban        159
25      urban        133
26      urban        163
27      urban        130
28      urban        163
29      urban        189
30      urban        144
31      urban        154
32      urban        198
33      urban        150
34      urban        151
35      urban        152
36      urban        151
37      urban        127
38      urban        167
39      urban        170
40      urban        123
;
run;
```

Suppose you would now like to standardize the total scores in different types of schools prior to any further analysis. Before standardizing the total scores, you can use the Schematic Plots from PROC UNIVARIATE to summarize the total scores for both types of schools.

```
proc univariate data=TotalScores plot;
   var total;
   by Type;
run;
```

The PLOT option in the PROC UNIVARIATE statement creates the Schematic Plots and several other types of plots. The Schematic Plots display side-by-side box plots for each BY group (Figure 66.1). The vertical axis represents the total scores, and the horizontal axis displays two box plots: the one on the left is for the rural scores and the one on the right is for the urban scores.

```
       High School Scores Data

       The UNIVARIATE Procedure
           Variable:  total

           Schematic Plots

     |
 260 +
     |            |
     |            |
     |            |
 240 +            |
     |            |
     |            |
     |            |
 220 +            |
     |            |
     |            |
     |            |
 200 +            |           |
     |            |           |
     |        +-----+         |
     |        |     |         |
 180 +        |     |         |
     |        |     |         |
     |        |     |     +-----+
     |        *--+--*     |     |
 160 +        |     |     |     |
     |        |     |     *--+--*
     |        |     |     |     |
     |        |     |     |     |
 140 +        |     |     +-----+
     |        |     |         |
     |        +-----+         |
     |            |           |
 120 +            |
     |            |
     |            |
     |
 100 +
     |
     |
     |
  80 +
     |
     |
     |                        0
  60 +
      ------------+-----------+-----------


       High School Scores Data

       The UNIVARIATE Procedure
           Variable:  total

           Schematic Plots

Type             rural       urban
```

Figure 66.1. Schematic Plots from PROC UNIVARIATE

Inspection reveals that one urban score is a low outlier. Also, if you compare the lengths of two boxplots, there seems to be twice as much dispersion for the rural scores as for the urban scores.

```
                            High School Scores Data

---------------------------------- Type=urban ----------------------------------

                            The UNIVARIATE Procedure
                                Variable:  total

                              Extreme Observations

                      ----Lowest----        ----Highest---

                      Value      Obs        Value      Obs

                         64       23          170       39
                        123       40          186       22
                        127       37          189       29
                        130       27          192       21
                        133       25          198       32
```

Figure 66.2. Table for Extreme Observations When Type=urban

Figure 66.2 displays the table from PROC UNIVARIATE for the lowest and highest five total scores for urban schools. The outlier (Obs = 3), marked in Figure 66.1 by the symbol ‘0’, has a score of 64.

The following statements use the traditional standardization method (METHOD=STD) to compute the location and scale measures:

```
proc stdize data=totalscores method=std pstat;
   title2 ’METHOD=STD’;
   var total;
   by Type;
run;
```

```
                            High School Scores Data
                                   METHOD=STD

---------------------------------- Type=rural ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name         Location           Scale           N

               total      167.050000       41.956713          20


                            High School Scores Data
                                   METHOD=STD

---------------------------------- Type=urban ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name         Location           Scale           N

               total      153.300000       30.066768          20
```

Figure 66.3. Location and Scale Measures Table When METHOD=STD

Figure 66.3 displays the table of location and scale measures from the PROC STDIZE statement. PROC STDIZE uses the mean as the location measure and the standard deviation as the scale measure for standardizing. The PSTAT option displays this table; otherwise, no display is created.

The ratio of the scale of rural scores to the scale of urban scores is approximately 1.4 (41.96/30.07). This ratio is smaller than the dispersion ratio observed in the previous Schematic Plots.

The STDIZE procedure provides several location and scale measures that are resistant to outliers. The following statements invoke three different standardization methods and display the Location and Scale Measures tables:

```
proc stdize data=totalscores method=mad pstat;
   title2 'METHOD=MAD';
   var total;
   by Type;
run;
```

```
proc stdize data=totalscores method=iqr pstat;
   title2 'METHOD=IQR';
   var total;
   by Type;
run;

proc stdize data=totalscores method=abw(4) pstat;
   title2 'METHOD=ABW(4)';
   var total;
   by Type;
run;
```

The results from this analysis are displayed in the following figures.

```
                             High School Scores Data
                                   METHOD=MAD

---------------------------------- Type=rural ----------------------------------

                               The STDIZE Procedure

                Location and Scale Measures

             Location = median     Scale = median abs dev from median

                Name         Location           Scale           N

                total      166.000000       32.000000          20


                             High School Scores Data
                                   METHOD=MAD

---------------------------------- Type=urban ----------------------------------

                               The STDIZE Procedure

                Location and Scale Measures

             Location = median     Scale = median abs dev from median

                Name         Location           Scale           N

                total      153.000000       15.500000          20
```

Figure 66.4. Location and Scale Measures Table When METHOD=MAD

Figure 66.4 displays the table of location and scale measures when the standardization method is MAD. The location measure is the median, and the scale measure is the median absolute deviation from median. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5) and is close to the dispersion ratio observed in the previous Schematic Plots.

```
                            High School Scores Data
                                   METHOD=IQR

---------------------------------- Type=rural ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

               Location = median     Scale = interquartile range

               Name         Location           Scale           N

               total      166.000000       61.000000          20


                            High School Scores Data
                                   METHOD=IQR

---------------------------------- Type=urban ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

               Location = median     Scale = interquartile range

               Name         Location           Scale           N

               total      153.000000       30.000000          20
```

Figure 66.5. Location and Scale Measures Table When METHOD=IQR

Figure 66.5 displays the table of location and scale measures when the standardization method is IQR. The location measure is the median, and the scale measure is the interquartile range. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.03 (61/30) and is, in fact, the dispersion ratio observed in the previous Schematic Plots.

```
                            High School Scores Data
                                 METHOD=ABW(4)

---------------------------------- Type=rural ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

     Location = biweight 1-step M-estimate     Scale = biweight A-estimate

               Name         Location           Scale           N

               total      162.889603       56.662855          20


                            High School Scores Data
                                 METHOD=ABW(4)

---------------------------------- Type=urban ----------------------------------

                              The STDIZE Procedure

               Location and Scale Measures

     Location = biweight 1-step M-estimate     Scale = biweight A-estimate

               Name         Location           Scale           N

               total      156.014608       28.615980          20
```

Figure 66.6. Location and Scale Measures Table When METHOD=ABW

Figure 66.6 displays the table of location and scale measures when the standardization method is ABW. The location measure is the biweight 1-step M-estimate, and the scale measure is the biweight A-estimate. Note that the initial estimate for ABW is MAD. The tuning constant (4) of ABW is obtained by the following steps:

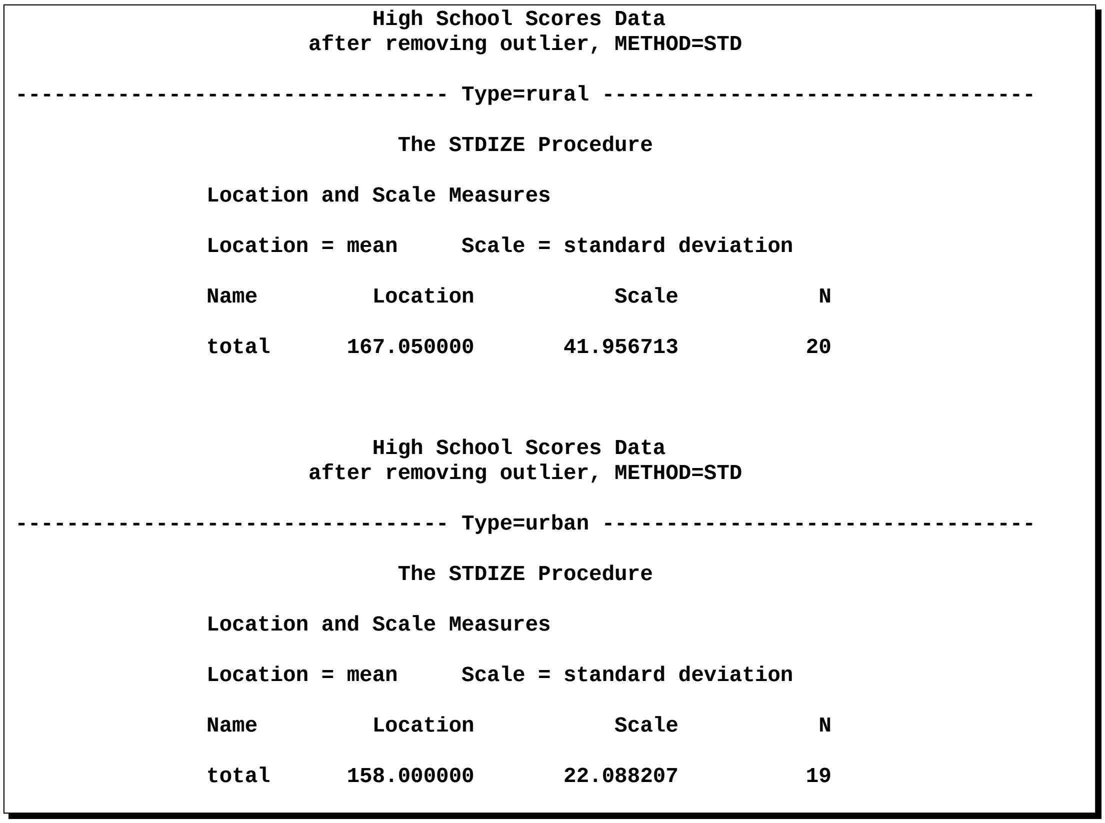
1. For rural scores, the location estimate for MAD is 166.0 and the scale estimate for MAD is 32.0. The maximum of the rural scores is 253 (not shown), and the minimum is 110 (not shown). Thus, the tuning constant needs to be 3 so that it does not reject any observation that has a score between 110 to 253.
2. For urban scores, the location estimate for MAD is 153.0 and the scale estimate for MAD is 15.5. The maximum of the rural scores is 198, and the minimum (also an outlier) is 64. Thus, the tuning constant needs to be 4 so that it rejects the outlier (64) but includes the maximum (198) as an normal observation.
3. The maximum of the tuning constants, obtained in steps 1 and 2, is 4.

Refer to Goodall (1983, Chapter 11) for details on the tuning constant. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5). It is also close to the dispersion ratio observed in the previous Schematic Plots.

The preceding analysis shows that METHOD=MAD, METHOD=IQR, and METHOD=ABW all provide better dispersion ratios than does METHOD=STD.

You can recompute the standard deviation after deleting the outlier from the original data set for comparison. The following statements create a DATA set NoOutlier that excludes the outlier from the TotalScores data set and invoke PROC STDIZE with METHOD=STD.

```
data NoOutlier;
   set totalscores;
   if (total = 64) then delete;
run;

proc stdize data=NoOutlier method=std pstat;
   title2 'after removing outlier, METHOD=STD';
   var total;
   by Type;
run;
```



```
                           High School Scores Data
                      after removing outlier, METHOD=STD

--------------------------------- Type=rural ---------------------------------

                             The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name         Location           Scale           N

               total      167.050000       41.956713          20


                           High School Scores Data
                      after removing outlier, METHOD=STD

--------------------------------- Type=urban ---------------------------------

                             The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name         Location           Scale           N

               total      158.000000       22.088207          19
```

Figure 66.7. After Deleting the Outlier, Location and Scale Measures Table When METHOD=STD

Figure 66.7 displays the location and scale measures after deleting the outlier. The lack of resistance of the standard deviation to outliers is clearly illustrated: if you delete the outlier, the sample standard deviation of urban scores changes from 30.07 to 22.09. The new ratio of the scale of rural scores to the scale of urban scores is approximately 1.90 (41.96/22.09).

Syntax

The following statements are available in the STDIZE procedure.

PROC STDIZE < *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
LOCATION *variables* **;**
SCALE *variables* **;**
VAR *variables* **;**
WEIGHT *variable* **;**

The PROC STDIZE statement is required. The BY, LOCATION, FREQ, VAR, SCALE, and WEIGHT statements are described in alphabetical order following the PROC STDIZE statement.

PROC STDIZE Statement

PROC STDIZE < *options* > **;**

The PROC STDIZE statement invokes the procedure. You can specify the following options in the PROC STDIZE statement.

Table 66.1. Summary of PROC STDIZE Statement Options

Task	**Options**	**Description**
Specify standardization methods	METHOD=	specifies the name of the standardization method
	INITIAL=	specifies the method for computing initial estimates for the A estimates
Unstandardize variables	UNSTD	unstandardizes variables when you also specify the METHOD=IN option
Process missing values	NOMISS	omits observations with any missing values from computation
	MISSING=	specifies the method or a numeric value for replacing missing values
	REPLACE	replaces missing data by zero in the standardized data
	REPONLY	replaces missing data by the location measure (does not standardize the data)
Specify data set details	DATA=	specifies the input data set
	OUT=	specifies the output data set
	OUTSTAT=	specifies the output statistic data set
Specify computational settings	VARDEF=	specifies the variances divisor
	NMARKERS=	specifies the number of markers when you also specify PCTLMTD=ONEPASS

Table 66.1. (continued)

Task	Options	Description
	MULT=	specifies the constant to multiply each value by after standardizing
	ADD=	specifies the constant to add to each value after standardizing and multiplying by the value specified in the MULT= option
	FUZZ=	specifies the relative fuzz factor for writing the output
Specify percentiles	PCTLDEF=	specifies the definition of percentiles when you also specify the PCTLMTD=ORD_STAT option
	PCTLMTD=	specifies the method used to estimate percentiles
	PCTLPTS=	writes observations containing percentiles to the data set specified in the OUTSTAT= option
Normalize scale estimators	NORM	normalizes the scale estimator to be consistent for the standard deviation of a normal distribution
	SNORM	normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution
Specify output	PSTAT	displays the location and scale measures

These options and their abbreviations are described, in alphabetical order, in the remainder of this section.

ADD= *c*

specifies a constant, *c*, to add to each value after standardizing and multiplying by the value you specify in the MULT= option. The default value is 0.

DATA=*SAS-data-set*

specifies the input data set to be standardized. If you omit the DATA= option, the most recently created data set is used.

FUZZ=*c*

specifies the relative fuzz factor. The default value is 1E-14. For the OUT= data set, the score is computed as follows:

$$\text{if } |\text{Result}| < \text{Scale} \times \text{Fuzz, then Result} = 0$$

For the OUTSTAT= data set and the Location and Scale table, the scale and location values are computed as follows:

$$\text{if Scale} < |\text{Location}| \times \text{Fuzz, then Scale} = 0$$

Otherwise,

$$\text{if } |\text{Location}| < \text{Scale} \times \text{Fuzz, then Location} = 0$$

INITIAL=*method*

specifies the method for computing initial estimates for the A estimates (ABW, AWAVE, and AHUBER). The following methods are not allowed: INITIAL=ABW, INITIAL=AHUBER, INITIAL=AWAVE, and INITIAL=IN. The default is INITIAL=MAD.

METHOD=*name*

specifies the name of the method for computing location and scale measures. Valid values for *name* are as follows: MEAN, MEDIAN, SUM, EUCLEN, USTD, STD, RANGE, MIDRANGE, MAXABS, IQR, MAD, ABW, AHUBER, AWAVE, AGK, SPACING, L, and IN.

For details on these methods, see the descriptions in the "Standardization Methods" section on page 4136. The default is METHOD=STD.

MISSING= *method*
MISSING= *value*

specifies the method (or a numeric value) for replacing missing values. If you omit the MISSING= option, the REPLACE option replaces missing values with the location measure given by the METHOD= option. Specify the MISSING= option when you want to replace missing values with a different value. You can specify any name that is valid in the METHOD= option except the name IN. The corresponding location measure is used to replace missing values.

If a numeric value is given, the value replaces missing values after standardizing the data. However, you can specify the REPONLY option with the MISSING= option to suppress standardization for cases in which you want only to replace missing values.

MULT= *c*

specifies a constant, *c*, by which to multiply each value after standardizing. The default value is 1.

NMARKERS= *n*

specifies the number of markers used when you specify the one-pass algorithm (PCTLMTD=ONEPASS). The value n must be greater than or equal to 5. The default value is 105.

NOMISS

omits observations with missing values for any of the analyzed variables from calculation of the location and scale measures. If you omit the NOMISS option, all nonmissing values are used.

NORM

normalizes the scale estimator to be consistent for the standard deviation of a normal distribution when you specify the option METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING.

OUT=*SAS-data-set*

specifies the name of the SAS data set created by PROC STDIZE. The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Note that analyzed variables are those specified in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement. See the section "Output Data Sets" on page 4141 for more information.

If you want to create a permanent SAS data set, you must specify a two-level name. (Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information on permanent SAS data sets.)

If you omit the OUT= option, PROC STDIZE creates an output data set named according to the DATAn convention.

OUTSTAT=*SAS-data-set*

specifies the name of the SAS data set containing the location and scale measures and other computed statistics. See the section "Output Data Sets" on page 4141 for more information.

PCTLDEF= *percentiles*

specifies which of five definitions is used to calculate percentiles when you specify the option PCTLMTD=ORD_STAT. By default, PCTLDEF=5.

Note that the option PCTLMTD=ONEPASS implies a specification of PCTLDEF=5. See the section "Computational Methods for the PCTLDEF= Option" on page 4140 for details on the PCTLDEF= option.

PCTLMTD=ORD_STAT
PCTLMTD=ONEPASS | P2

specifies the method used to estimate percentiles. Specify the PCTLMTD=ORD_STAT option to compute the percentiles by the order statistics method. The PCTLMTD=ONEPASS option modifies an algorithm invented by Jain and Chlamtac (1985). See the "Computing Quantiles" section on page 4139 for more details on this algorithm.

The PCTLMTD=ONEPASS option modifies an algorithm invented by Jain and Chlamtac (1985). See the "Computing Quantiles" section on page 4139 for more details on this algorithm.

PCTLPTS= *n*

writes percentiles to the OUTSTAT= data set. Values of *n* can be any decimal number between 0 and 100, inclusive.

A requested percentile is identified by the _TYPE_ variable in the OUTSTAT= data set with a value of P*n*. For example, suppose you specify the option PCTLPTS=10, 30. The corresponding observations in the OUTSTAT= data set that contain the 10th and the 30th percentiles would then have values _TYPE_=P10 and _TYPE_=P30, respectively.

PSTAT

displays the location and scale measures.

REPLACE

replaces missing data with the value 0 in the standardized data (this value corresponds to the location measure before standardizing). To replace missing data by other values, see the preceding description of the MISSING= option. You cannot specify both the REPLACE and REPONLY options.

REPONLY

replaces missing data only; PROC STDIZE does not standardize the data. Missing values are replaced with the location measure unless you also specify the MISSING=*value* option, in which case missing values are replaced with *value*. You cannot specify both the REPLACE and REPONLY options.

SNORM

normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution when you specify the METHOD=SPACING option.

UNSTD
UNSTDIZE

unstandardizes variables when you specify the METHOD=IN(*ds*) option. The location and scale measures, along with constants for addition and multiplication that the unstandardization is based upon, are identified by the _TYPE_ variable in the ds data set.

The ds data set must have a _TYPE_ variable and contain the following two observations: a _TYPE_= 'LOCATION' observation and a _TYPE_= 'SCALE' observation. The variable _TYPE_ can also contain the optional observations, 'ADD' and 'MULT'; if these observations are not found in the ds data set, the constants specified in the ADD= and MULT= options (or their default values) are used for unstandardization.

See the "OUTSTAT= Data Set" section on page 4141 for details on the statistics that each value of _TYPE_ represents. The formula used for unstandardization is as follows: If the final output value from the previous standardization is calculated as

$$result = add + multiply \times \frac{original - location}{scale}$$

$$original = scale \times \frac{result - add}{multiply} + location$$

VARDEF= DF
VARDEF= N
VARDEF= WDF
VARDEF= WEIGHT | WGT

specifies the divisor to be used in the calculation of variances. By default, VARDEF=DF. The values and associated divisors are as follows.

Value	Divisor	Formula
DF	degrees of freedom	$n-1$
N	number of observations	n
WDF	sum of weights minus 1	$(\sum_i w_i) - 1$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC STDIZE to obtain separate standardization for observations in groups defined by the BY variables.

If your DATA= input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the STDIZE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

When you specify the option METHOD=IN(ds), the following rules are applied to BY-group processing:

- If the ds data set does not contain any of the BY variables, the entire DATA= data set is standardized by the location and scale measures (along with the constants for addition and multiplication) in the ds data set.
- If the ds data set contains some, but not all, of the BY variables or if some BY variables do not have the same type or length in the ds data set that they have in the DATA= data set, PROC STDIZE displays an error message and stops.
- If all of the BY variables appear in the ds data set with the same type and length as in the DATA= data set, each BY group in the DATA= data set is standardized using the location and scale measures (along with the constants for addition and multiplication) from the corresponding BY group in the ds data set. The BY groups in the ds data set must be in the same order as they appear in the DATA= data set. All BY groups in the DATA= data set must also appear in the ds data set. If you do not specify the NOTSORTED option, some BY groups can appear in the ds data set but not in the DATA= data set; such BY groups are not used in standardizing data.

FREQ Statement

FREQ | **FREQUENCY** *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable name in a FREQ statement. PROC STDIZE treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value. If the FREQ variable has a value that is less than 1 or is missing, the observation is not used in the analysis.

LOCATION Statement

LOCATION *variables* **;**

The LOCATION statement specifies a list of numeric variables that contain location measures in the input data set specified by the METHOD=IN option.

SCALE Statement

SCALE *variables* **;**

The SCALE statement specifies the list of numeric variables containing scale measures in the input data set specified by the METHOD=IN option.

VAR Statement

VAR | **VARIABLES** *variables* **;**

The VAR statement lists numeric variables to be standardized. If you omit the VAR statement, all numeric variables not listed in the BY, FREQ, and WEIGHT statements are used.

WEIGHT Statement

WGT | **WEIGHT** *variable* **;**

The WEIGHT statement specifies a numeric variable in the input data set with values that are used to weight each observation. Only one variable can be specified.

The WEIGHT variable values can be nonintegers. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero. The WEIGHT variable applies only when you specify the option METHOD=MEAN, METHOD=SUM, METHOD=EUCLEN, METHOD=USTD, METHOD=STD, METHOD=AGK, or METHOD=L.

PROC STDIZE uses the value of the WEIGHT variable w_i, as follows.

The sample mean and (uncorrected) sample variances are computed as

$$\overline{x}_w = \sum_i w_i x_i / \sum_i w_i$$

$$us_w{}^2 = \sum_i w_i x_i{}^2 / d$$

$$s_w{}^2 = \sum_i w_i (x_i - \overline{x}_w)^2 / d$$

where w_i is the weight value of the ith observation, x_i is the value of the ith observation, and d is the divisor controlled by the VARDEF= option (see the VARDEF= option for details).

PROC STDIZE uses the value of the WEIGHT variable to calculate the following statistics:

MEAN the weighted mean, $\overline{x}_w$

SUM the weighted sum, $\sum_i w_i x_i$

USTD the weighted uncorrected standard deviation, $\sqrt{us_w^2}$

STD the weighted standard deviation, $\sqrt{s_w^2}$

EUCLEN the weighted Euclidean length, computed as the square root of the weighted uncorrected sum of squares:

$$\sqrt{\sum_i w_i x_i{}^2}$$

AGK the AGK estimate. This estimate is documented further in the ACECLUS procedure as the METHOD=COUNT option. See the discussion of the WEIGHT statement in Chapter 16, “The ACECLUS Procedure,” for information on how the WEIGHT variable is applied to the AGK estimate.

L the L_p estimate. This estimate is documented further in the FASTCLUS procedure as the LEAST= option. See the discussion of the WEIGHT statement in Chapter 28, “The FASTCLUS Procedure,” for information on how the WEIGHT variable is used to compute weighted cluster means. Note that the number of clusters is always 1.

Details

Standardization Methods

The following table lists standardization methods and their corresponding location and scale measures available with the METHOD= option.

Table 66.2. Available Standardization Methods

Method	Location	Scale
MEAN	mean	1
MEDIAN	median	1
SUM	0	sum
EUCLEN	0	Euclidean length
USTD	0	standard deviation about origin
STD	mean	standard deviation
RANGE	minimum	range
MIDRANGE	midrange	range/2
MAXABS	0	maximum absolute value
IQR	median	interquartile range
MAD	median	median absolute deviation from median
ABW(c)	biweight 1-step M-estimate	biweight A-estimate
AHUBER(c)	Huber 1-step M-estimate	Huber A-estimate
AWAVE(c)	Wave 1-step M-estimate	Wave A-estimate
AGK(p)	mean	AGK estimate (ACECLUS)
SPACING(p)	mid minimum-spacing	minimum spacing
L(p)	L(p)	L(p)
IN(ds)	read from data set	read from data set

For METHOD=ABW(c), METHOD=AHUBER(c), or METHOD=AWAVE(c), c is a positive numeric tuning constant.

For METHOD=AGK(p), p is a numeric constant giving the proportion of pairs to be included in the estimation of the within-cluster variances.

For METHOD=SPACING(p), p is a numeric constant giving the proportion of data to be contained in the spacing.

For METHOD=L(p), p is a numeric constant greater than or equal to 1 specifying the power to which differences are to be raised in computing an L(p) or Minkowski metric.

For METHOD=IN(ds), ds is the name of a SAS data set that meets either one of the following two conditions:

- contains a _TYPE_ variable. The observation that contains the location measure corresponds to the value _TYPE_= 'LOCATION' and the observation that contains the scale measure corresponds to the value _TYPE_= 'SCALE'. You can also use a data set created by the OUTSTAT= option from another PROC STDIZE statement as the ds data set. See the section "Output Data Sets" on page 4141 for the contents of the OUTSTAT= data set.
- contains the location and scale variables specified by the LOCATION and SCALE statements.

PROC STDIZE reads in the location and scale variables in the ds data set by first looking for the _TYPE_ variable in the ds data set. If it finds this variable, PROC

STDIZE continues to search for all variables specified in the VAR statement. If it does not find the _TYPE_ variable, PROC STDIZE searches for the location variables specified in the LOCATION statement and the scale variables specified in the SCALE statement.

For robust estimators, refer to Goodall (1983) and Iglewicz (1983). The MAD method has the highest breakdown point (50%), but it is somewhat inefficient. The ABW, AHUBER, and AWAVE methods provide a good compromise between breakdown and efficiency. The L(p) location estimates are increasingly robust as p drops from 2 (corresponding to least squares, or mean estimation) to 1 (corresponding to least absolute value, or median estimation). However, the L(p) scale estimates are not robust.

The SPACING method is robust to both outliers and clustering (Jannsen et al. 1995) and is, therefore, a good choice for cluster analysis or nonparametric density estimation. The mid-minimum spacing method estimates the mode for small p. The AGK method is also robust to clustering and more efficient than the SPACING method, but it is not as robust to outliers and takes longer to compute. If you expect g clusters, the argument to METHOD=SPACING or METHOD=AGK should be $\frac{1}{g}$ or less. The AGK method is less biased than the SPACING method for small samples. As a general guide, it is reasonable to use AGK for samples of size 100 or less and SPACING for samples of size 1000 or more, with the treatment of intermediate sample sizes depending on the available computer resources.

Computation of the Statistics

Formulas for statistics of METHOD=MEAN, METHOD=MEDIAN, METHOD=SUM, METHOD=USTD, METHOD=STD, METHOD=RANGE, and METHOD=IQR are given in the chapter on elementary statistics procedures in the *SAS Procedures Guide*.

Note that the computations of median and upper and lower quartiles depend on the PCTLMTD= option.

The other statistics listed in Table 66.2, except for METHOD=IN, are described as follows:

EUCLEN

Euclidean length.
$\sqrt{\sum_{i=1}^{n} {x_i}^2}$ where x_i is the ith observation and n is the total number of observations in the sample.

L(p)

Minkowski metric. This metric is documented as the LEAST=p option in the PROC FASTCLUS statement of the FASTCLUS procedure (see Chapter 28, “The FASTCLUS Procedure,”).

If you specify METHOD=L(p) in the PROC STDIZE statement, your results are similar to those obtained from PROC FASTCLUS if you specify the LEAST=*p* option with MAXCLUS=1 (and use the default values of the MAXITER= option). The difference between the two types of calculations concerns the maximum number of iterations. In PROC STDIZE, it is a criteria for convergence on all variables; In PROC FASTCLUS, it is a criteria for convergence on a single variable.

The location and scale measures for L(p) are output to the OUTSEED= data set in PROC FASTCLUS.

MIDRANGE $(\text{maximum} + \text{minimum})/2$

ABW(c) Tukey's biweight. Refer to Goodall (1983, pp. 376–378, p. 385) for the biweight 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416–418) for the biweight A-estimate.

AHUBER(c) Hubers. Refer to Goodall (1983, pp. 371–374) for the Huber 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416–418) for the Huber A-estimate of scale.

AWAVE(c) Andrews' Wave. Refer to Goodall (1983, p. 376) for the Wave 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416 –418) for the Wave A-estimate of scale.

AGK(p) The noniterative univariate form of the estimator described by Art, Gnanadesikan, and Kettenring (1982).

The AGK estimate is documented in the section on the METHOD= option in the PROC ACECLUS statement of the ACECLUS procedure (also see the "Background" section on page 388 in Chapter 16, "The ACECLUS Procedure,"). Specifying METHOD=AGK(p) in the PROC STDIZE statement is the same as specifying METHOD=COUNT and P=p in the PROC ACECLUS statement.

SPACING(p) The absolute difference between two data values. The minimum spacing for a proportion p is the minimum absolute difference between two data values that contain a proportion p of the data between them. The mid minimum-spacing is the mean of these two data values.

Computing Quantiles

PROC STDIZE offers two methods for computing quantiles: the one-pass approach and the order-statistics approach (like that used in the UNIVARIATE procedure).

The one-pass approach used in PROC STDIZE modifies the P^2 algorithm for histograms proposed by Jain and Chlamtac (1985). The primary difference comes from the movement of markers. The one-pass method allows a marker to move to the right (or left) by more than one position (to the largest possible integer) as long as it does not result in two markers being in the same position. The modification is necessary in order to incorporate the FREQ variable.

You may obtain inaccurate results if you use the one-pass approach to estimate quantiles beyond the quartiles (that is, when you estimate quantiles $<$ P25 or $>$ P75). A large sample size (10,000 or more) is often required if the tail quantiles (quantiles <= P10 or >= P90) are requested. Note that, for variables with highly skewed or heavy-tailed distributions, tail quantile estimates may be inaccurate.

The order-statistics approach for estimating quantiles is faster than the one-pass method but requires that the entire data set be stored in memory. The accuracy in estimating the quantiles is comparable for both methods when the requested percentiles are between the lower and upper quartiles. The default is PCTLMTD=ORD_STAT if enough memory is available; otherwise, PCTLMTD=ONEPASS.

Computational Methods for the PCTLDEF= Option

You can specify one of five methods for computing quantile statistics when you use the order-statistics approach (PCTLMTD=ORD_STAT); otherwise, the PCTLDEF=5 method is used when you use the one-pass approach (PCTLMTD=ONEPASS).

Let n be the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the variable. For the tth percentile, let $p = t/100$. In the following definitions numbered 1, 2, 3, and 5, let

$$np = j + g$$

where j is the integer part and g is the fractional part of np. For definition 4, let

$$(n+1)p = j + g$$

Given the preceding definitions, the tth percentile, y, is defined as follows:

PCTLDEF=1 weighted average at x_{np}

$$y = (1-g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1

PCTLDEF=2 observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$ if $g \neq 1/2$. If $g = 1/2$, then
$y = x_j$ if j is even, or
$y = x_{j+1}$ if j is odd

PCTLDEF=3 empirical distribution function

$$y = x_j \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

PCTLDEF=4 weighted average aimed at $x_{p(n+1)}$

$$y = (1-g)x_j + gx_{j+1}$$

where x_{n+1} is taken to be x_n

PCTLDEF=5 empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

Missing Values

Missing values can be replaced by the location measure or by any specified constant (see the REPLACE option and the MISSING= option). You can also suppress standardization if you want only to replace missing values (see the REPONLY option).

If you specify the NOMISS option, PROC STDIZE omits observations with any missing values in the analyzed variables from computation of the location and scale measures.

Output Data Sets

OUT= Data Set

The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement.

OUTSTAT= Data Set

The new data set contains the following variables:

- the BY variables, if any
- _TYPE_, a character variable
- the analyzed variables

Each observation in the new data set contains a type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

TYPE	
LOCATION	location measure of each variable
SCALE	scale measure of each variable
ADD	constant specified in the ADD= option. This value is the same for each variable.
MULT	constant specified in the MULT= option. This value is the same for each variable.
N	total number of nonmissing positive frequencies of each variable
NORM	norm measure of each variable. This observation is produced only when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING.
NObsRead	number of physical records read
NObsUsed	number of physical records used in the analysis
NObsMiss	number of physical records containing missing values

SumFreqsRead sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations read

SumFreqsUsed sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations used in the analysis

SumWeightsRead sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations read

SumWeightsUsed sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations used in the analysis

P*n* percentiles of each variable, as specified by the PCTLPTS= option. The argument *n* is any real number such that $0 \leq n \leq 100$.

Displayed Output

If you specify the PSTAT option, PROC STDIZE displays the following statistics for each variable:

- the name of the variable, Name
- the location estimate, Location
- the scale estimate, Scale
- the norm estimate, Norm (when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING)
- the total nonmissing positive frequencies, N

ODS Table Names

PROC STDIZE assigns a name to the single table it creates. You can use this name to reference the table when using the Output Delivery System (ODS) to select output or create an output data set. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 66.3. ODS Table Produced in PROC STDIZE

ODS Table Name	Description	Option
Statistics	Location and Scale Measures	PSTAT

Examples

Example 66.1. Standardization of Variables in Cluster Analysis

To illustrate the effect of standardization in cluster analysis, this example uses the Fish data set described in the "Getting Started" section of Chapter 28, "The FASTCLUS Procedure." The numbers are measurements taken on 159 fish caught from the same lake (Laengelmavesi) near Tampere in Finland; this data set is available from the Data Archive of the *Journal of Statistics Education*. The complete data set is displayed in Chapter 67, "The STEPDISC Procedure."

The species (Bream, Parkki, Pike, Perch, Roach, Smelt, and Whitefish), weight, three different length measurements (measured from the nose of the fish to the beginning of its tail, the notch of its tail, and the end of its tail), height, and width of each fish are recorded. The height and width are recorded as percentages of the third length variable.

Several new variables are created in the Fish data set: Weight3, Height, Width, and logLengthRatio. The weight of a fish indicates its size—a heavier Tuna tends to be larger than a lighter Tuna. To get a one dimensional measure of the size of a fish, take the cubic root of the weight (Weight3). The variables Height, Width, Length1, Length2, and Length3 are rescaled in order to adjust for dimensionality. The logLengthRatio variable measures the tail length.

Because the new variables Weight3–logLengthRatio depend on the variable Weight, observations with missing values for Weight are not added to the data set. Consequently, there are 157 observations in the SAS data set Fish.

Before you perform a cluster analysis on coordinate data, it is necessary to consider scaling or transforming the variables since variables with large variances tend to have a larger effect on the resulting clusters than those with small variances.

This example uses three different approaches to standardize or transform the data prior to the cluster analysis. The first approach uses several standardization methods provided in the STDIZE procedure. However, since standardization is not always appropriate prior to the clustering (refer to Milligan and Cooper, 1987, for a Monte Carlo study on various methods of variable standardization), the second approach performs the cluster analysis with no standardization. The third approach invokes the ACECLUS procedure to transform the data into a within-cluster covariance matrix.

The clustering is performed by the FASTCLUS procedure to find seven clusters. Note that the variables Length2 and Length3 are eliminated from this analysis since they both are significantly and highly correlated with the variable Length1. The correlation coefficients are 0.9958 and 0.9604, respectively. An output data set is created, and the FREQ procedure is invoked to compare the clusters with the species classification.

The DATA step is as follows:

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
data Fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;

   if Weight <= 0 or Weight=. then delete;
   Weight3=Weight**(1/3);
   Height=HtPct*Length3/(Weight3*100);
   Width=WidthPct*Length3/(Weight3*100);
   Length1=Length1/Weight3;
   Length2=Length2/Weight3;
   Length3=Length3/Weight3;
   logLengthRatio=log(Length3/Length1);

   format Species specfmt.;
   symbol = put(Species, specfmt2.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4
1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1
1  363.0 26.3 29.0 33.5 38.0 13.3
 ... [155 more records]
;
run;
```

The following macro, Std, standardizes the Fish data. The macro reads a single argument, mtd, which selects the METHOD= specification to be used in PROC STDIZE.

```
/*--- macro for standardization ---*/

%macro Std(mtd);
title2 "Data is standardized by PROC STDIZE with
        METHOD= &mtd";
   proc stdize data=fish out=sdzout method=&mtd;
      var Length1 logLengthRatio Height Width Weight3;
   run;
%mend Std;
```

The following macro, FastFreq, includes a PROC FASTCLUS statement for performing cluster analysis and a PROC FREQ statement for cross-tabulating species with the cluster membership information that is derived from the previous PROC FASTCLUS statement. The macro reads a single argument, ds, which selects the input data set to be used in PROC FASTCLUS.

```
/*--- macro for clustering and cross-tabulating ---*/
/*--- cluster membership with species           ---*/
%macro FastFreq(ds);
   proc fastclus data=&ds out=clust maxclusters=7 maxiter=100 noprint;
      var Length1 logLengthRatio Height Width Weight3;
   run;

   proc freq data=clust;
      tables species*cluster;
   run;
%mend FastFreq;
```

The following analysis, (labeled 'Approach 1') includes 18 different methods of standardization followed by clustering. Since there is a large amount of output from this approach, only results from METHOD=STD, METHOD=RANGE, METHOD=AGK(.14), and METHOD=SPACING(.14) are shown. The following statements produce Output 66.1.1 through Output 66.1.4.

```
/**********************************************************/
/*                                                        */
/*     Approach 1: data is standardized by PROC STDIZE    */
/*                                                        */
/**********************************************************/

%Std(MEAN);
%FastFreq(sdzout);

%Std(MEDIAN);
%FastFreq(sdzout);

%Std(SUM);
%FastFreq(sdzout);

%Std(EUCLEN);
%FastFreq(sdzout);

%Std(USTD);
%FastFreq(sdzout);

%Std(STD);
%FastFreq(sdzout);

%Std(RANGE);
%FastFreq(sdzout);

%Std(MIDRANGE);
%FastFreq(sdzout);

%Std(MAXABS);
%FastFreq(sdzout);

%Std(IQR);
```

```
%FastFreq(sdzout);

%Std(MAD);
%FastFreq(sdzout);

%Std(AGK(.14));
%FastFreq(sdzout);

%Std(SPACING(.14));
%FastFreq(sdzout);

%Std(ABW(5));
%FastFreq(sdzout);

%Std(AWAVE(5));
%FastFreq(sdzout);

%Std(L(1));
%FastFreq(sdzout);

%Std(L(1.5));
%FastFreq(sdzout);

%Std(L(2));
%FastFreq(sdzout);
```

Output 66.1.1. Data Is Standardized by PROC STDIZE with METHOD=STD

```
                              Fish Measurement Data
                   Data is standardized by PROC STDIZE with METHOD= STD

                                 The FREQ Procedure

                              Table of Species by CLUSTER

          Species     CLUSTER(Cluster)

          Frequency |
          Percent   |
          Row Pct   |
          Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Bream     |      0 |      0 |      0 |      0 |      0 |     34 |      0 |     34
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  21.66 |   0.00 |  21.66
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Roach     |      0 |      0 |      0 |      0 |      0 |      0 |     19 |     19
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  12.10 |  12.10
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |
                    |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  38.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Whitefish |      0 |      2 |      0 |      1 |      0 |      0 |      3 |      6
                    |   0.00 |   1.27 |   0.00 |   0.64 |   0.00 |   0.00 |   1.91 |   3.82
                    |   0.00 |  33.33 |   0.00 |  16.67 |   0.00 |   0.00 |  50.00 |
                    |   0.00 |  10.53 |   0.00 |   7.69 |   0.00 |   0.00 |   6.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Parkki    |      0 |      0 |      0 |      0 |     11 |      0 |      0 |     11
                    |   0.00 |   0.00 |   0.00 |   0.00 |   7.01 |   0.00 |   0.00 |   7.01
                    |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
                    |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Perch     |      0 |     17 |      0 |     12 |      0 |      0 |     27 |     56
                    |   0.00 |  10.83 |   0.00 |   7.64 |   0.00 |   0.00 |  17.20 |  35.67
                    |   0.00 |  30.36 |   0.00 |  21.43 |   0.00 |   0.00 |  48.21 |
                    |   0.00 |  89.47 |   0.00 |  92.31 |   0.00 |   0.00 |  54.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Pike      |     17 |      0 |      0 |      0 |      0 |      0 |      0 |     17
                    |  10.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  10.83
                    | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                    | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Smelt     |      0 |      0 |     13 |      0 |      0 |      0 |      1 |     14
                    |   0.00 |   0.00 |   8.28 |   0.00 |   0.00 |   0.00 |   0.64 |   8.92
                    |   0.00 |   0.00 |  92.86 |   0.00 |   0.00 |   0.00 |   7.14 |
                    |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   2.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Total           17       19       13       13       11       34       50      157
                       10.83    12.10     8.28     8.28     7.01    21.66    31.85   100.00
```

Output 66.1.2. Data Is Standardized by PROC STDIZE with METHOD=RANGE

```
                              Fish Measurement Data
                 Data is standardized by PROC STDIZE with METHOD= RANGE

                                The FREQ Procedure

                           Table of Species by CLUSTER

            Species     CLUSTER(Cluster)

            Frequency |
            Percent   |
            Row Pct   |
            Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Bream     |      0 |      0 |     34 |      0 |      0 |      0 |      0 |     34
                      |   0.00 |   0.00 |  21.66 |   0.00 |   0.00 |   0.00 |   0.00 |  21.66
                      |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Roach     |      0 |      0 |      0 |     19 |      0 |      0 |      0 |     19
                      |   0.00 |   0.00 |   0.00 |  12.10 |   0.00 |   0.00 |   0.00 |  12.10
                      |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |
                      |   0.00 |   0.00 |   0.00 |  61.29 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Whitefish |      0 |      0 |      0 |      3 |      3 |      0 |      0 |      6
                      |   0.00 |   0.00 |   0.00 |   1.91 |   1.91 |   0.00 |   0.00 |   3.82
                      |   0.00 |   0.00 |   0.00 |  50.00 |  50.00 |   0.00 |   0.00 |
                      |   0.00 |   0.00 |   0.00 |   9.68 |  13.04 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Parkki    |      0 |      0 |      0 |      0 |      0 |     11 |      0 |     11
                      |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   7.01 |   0.00 |   7.01
                      |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
                      |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Perch     |      0 |      0 |      0 |      9 |     20 |      0 |     27 |     56
                      |   0.00 |   0.00 |   0.00 |   5.73 |  12.74 |   0.00 |  17.20 |  35.67
                      |   0.00 |   0.00 |   0.00 |  16.07 |  35.71 |   0.00 |  48.21 |
                      |   0.00 |   0.00 |   0.00 |  29.03 |  86.96 |   0.00 | 100.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Pike      |     17 |      0 |      0 |      0 |      0 |      0 |      0 |     17
                      |  10.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  10.83
                      | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Smelt     |      0 |     14 |      0 |      0 |      0 |      0 |      0 |     14
                      |   0.00 |   8.92 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   8.92
                      |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Total           17       14       34       31       23       11       27      157
                         10.83     8.92    21.66    19.75    14.65     7.01    17.20   100.00
```

Output 66.1.3. Data Is Standardized by PROC STDIZE with METHOD=AGK(.14)

```
                                Fish Measurement Data
                  Data is standardized by PROC STDIZE with METHOD= AGK(.14)

                                  The FREQ Procedure

                              Table of Species by CLUSTER

            Species     CLUSTER(Cluster)

            Frequency |
            Percent   |
            Row Pct   |
            Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Bream     |      0 |      0 |     34 |      0 |      0 |      0 |      0 |     34
                      |   0.00 |   0.00 |  21.66 |   0.00 |   0.00 |   0.00 |   0.00 |  21.66
                      |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Roach     |      0 |      0 |      0 |     17 |      0 |      0 |      2 |     19
                      |   0.00 |   0.00 |   0.00 |  10.83 |   0.00 |   0.00 |   1.27 |  12.10
                      |   0.00 |   0.00 |   0.00 |  89.47 |   0.00 |   0.00 |  10.53 |
                      |   0.00 |   0.00 |   0.00 |  73.91 |   0.00 |   0.00 |   5.71 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Whitefish |      0 |      0 |      0 |      3 |      0 |      3 |      0 |      6
                      |   0.00 |   0.00 |   0.00 |   1.91 |   0.00 |   1.91 |   0.00 |   3.82
                      |   0.00 |   0.00 |   0.00 |  50.00 |   0.00 |  50.00 |   0.00 |
                      |   0.00 |   0.00 |   0.00 |  13.04 |   0.00 |  13.04 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Parkki    |     11 |      0 |      0 |      0 |      0 |      0 |      0 |     11
                      |   7.01 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   7.01
                      | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Perch     |      0 |      0 |      0 |      3 |      0 |     20 |     33 |     56
                      |   0.00 |   0.00 |   0.00 |   1.91 |   0.00 |  12.74 |  21.02 |  35.67
                      |   0.00 |   0.00 |   0.00 |   5.36 |   0.00 |  35.71 |  58.93 |
                      |   0.00 |   0.00 |   0.00 |  13.04 |   0.00 |  86.96 |  94.29 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Pike      |      0 |      0 |      0 |      0 |     17 |      0 |      0 |     17
                      |   0.00 |   0.00 |   0.00 |   0.00 |  10.83 |   0.00 |   0.00 |  10.83
                      |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
                      |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Smelt     |      0 |     14 |      0 |      0 |      0 |      0 |      0 |     14
                      |   0.00 |   8.92 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   8.92
                      |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                      |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
            ----------+--------+--------+--------+--------+--------+--------+--------+
            Total           11       14       34       23       17       23       35      157
                          7.01     8.92    21.66    14.65    10.83    14.65    22.29   100.00
```

Output 66.1.4. Data Is Standardized by PROC STDIZE with METHOD=SPACING(.14)

```
                              Fish Measurement Data
              Data is standardized by PROC STDIZE with METHOD= SPACING(.14)

                                The FREQ Procedure

                           Table of Species by CLUSTER

         Species     CLUSTER(Cluster)

         Frequency |
         Percent   |
         Row Pct   |
         Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Bream     |      0 |      0 |      0 |      0 |      0 |      0 |     34 |     34
                   |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  21.66 |  21.66
                   |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |
                   |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Roach     |      0 |      0 |      0 |     17 |      0 |      2 |      0 |     19
                   |   0.00 |   0.00 |   0.00 |  10.83 |   0.00 |   1.27 |   0.00 |  12.10
                   |   0.00 |   0.00 |   0.00 |  89.47 |   0.00 |  10.53 |   0.00 |
                   |   0.00 |   0.00 |   0.00 |  85.00 |   0.00 |   5.26 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Whitefish |      3 |      0 |      0 |      3 |      0 |      0 |      0 |      6
                   |   1.91 |   0.00 |   0.00 |   1.91 |   0.00 |   0.00 |   0.00 |   3.82
                   |  50.00 |   0.00 |   0.00 |  50.00 |   0.00 |   0.00 |   0.00 |
                   |  13.04 |   0.00 |   0.00 |  15.00 |   0.00 |   0.00 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Parkki    |      0 |      0 |     11 |      0 |      0 |      0 |      0 |     11
                   |   0.00 |   0.00 |   7.01 |   0.00 |   0.00 |   0.00 |   0.00 |   7.01
                   |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                   |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Perch     |     20 |      0 |      0 |      0 |      0 |     36 |      0 |     56
                   |  12.74 |   0.00 |   0.00 |   0.00 |   0.00 |  22.93 |   0.00 |  35.67
                   |  35.71 |   0.00 |   0.00 |   0.00 |   0.00 |  64.29 |   0.00 |
                   |  86.96 |   0.00 |   0.00 |   0.00 |   0.00 |  94.74 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Pike      |      0 |     17 |      0 |      0 |      0 |      0 |      0 |     17
                   |   0.00 |  10.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  10.83
                   |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
                   |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Smelt     |      0 |      0 |      0 |      0 |     14 |      0 |      0 |     14
                   |   0.00 |   0.00 |   0.00 |   0.00 |   8.92 |   0.00 |   0.00 |   8.92
                   |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
                   |   0.00 |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |
         ----------+--------+--------+--------+--------+--------+--------+--------+
         Total           23       17       11       20       14       38       34      157
                      14.65    10.83     7.01    12.74     8.92    24.20    21.66   100.00
```

The following analysis (labeled ‘Approach 2’) applies the cluster analysis directly to the original data. The following statements produce Output 66.1.5.

```
/*************************************************************/
/*                                                           */
/*         Approach 2: data is untransformed                 */
/*                                                           */
/*************************************************************/

title2 'Data is untransformed';
%FastFreq(fish);
```

Output 66.1.5. Untransformed Data

```
                           Fish Measurement Data
                           Data is untransformed

                            The FREQ Procedure

                         Table of Species by CLUSTER

          Species     CLUSTER(Cluster)

          Frequency |
          Percent   |
          Row Pct   |
          Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Bream     |     13 |      0 |      0 |      0 |      0 |      0 |     21 |     34
                    |   8.28 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  13.38 |  21.66
                    |  38.24 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  61.76 |
                    |  44.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  47.73 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Roach     |      3 |      4 |      0 |      0 |     12 |      0 |      0 |     19
                    |   1.91 |   2.55 |   0.00 |   0.00 |   7.64 |   0.00 |   0.00 |  12.10
                    |  15.79 |  21.05 |   0.00 |   0.00 |  63.16 |   0.00 |   0.00 |
                    |  10.34 |  25.00 |   0.00 |   0.00 |  30.77 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Whitefish |      3 |      0 |      0 |      0 |      0 |      0 |      3 |      6
                    |   1.91 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   1.91 |   3.82
                    |  50.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  50.00 |
                    |  10.34 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   6.82 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Parkki    |      2 |      3 |      0 |      0 |      6 |      0 |      0 |     11
                    |   1.27 |   1.91 |   0.00 |   0.00 |   3.82 |   0.00 |   0.00 |   7.01
                    |  18.18 |  27.27 |   0.00 |   0.00 |  54.55 |   0.00 |   0.00 |
                    |   6.90 |  18.75 |   0.00 |   0.00 |  15.38 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Perch     |      8 |      9 |      0 |      1 |     20 |      0 |     18 |     56
                    |   5.10 |   5.73 |   0.00 |   0.64 |  12.74 |   0.00 |  11.46 |  35.67
                    |  14.29 |  16.07 |   0.00 |   1.79 |  35.71 |   0.00 |  32.14 |
                    |  27.59 |  56.25 |   0.00 |   6.67 |  51.28 |   0.00 |  40.91 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Pike      |      0 |      0 |     10 |      0 |      1 |      4 |      2 |     17
                    |   0.00 |   0.00 |   6.37 |   0.00 |   0.64 |   2.55 |   1.27 |  10.83
                    |   0.00 |   0.00 |  58.82 |   0.00 |   5.88 |  23.53 |  11.76 |
                    |   0.00 |   0.00 | 100.00 |   0.00 |   2.56 | 100.00 |   4.55 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Smelt     |      0 |      0 |      0 |     14 |      0 |      0 |      0 |     14
                    |   0.00 |   0.00 |   0.00 |   8.92 |   0.00 |   0.00 |   0.00 |   8.92
                    |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |
                    |   0.00 |   0.00 |   0.00 |  93.33 |   0.00 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Total           29       16       10       15       39        4       44      157
                       18.47    10.19     6.37     9.55    24.84     2.55    28.03   100.00
```

The following analysis (labeled ‘Approach 3’) transforms the original data with the ACECLUS procedure and creates a TYPE=ACE output data set that is used as an input data set for the cluster analysis. The following statements produce Output 66.1.6.

```
/**********************************************************/
/*                                                        */
/*    Approach 3: data is transformed by PROC ACECLUS     */
/*                                                        */
/**********************************************************/

title2 ’Data is transformed by PROC ACECLUS’;
proc aceclus data=fish out=ace p=.02 noprint;
   var Length1 logLengthRatio Height Width Weight3;
run;
%FastFreq(ace);
```

Output 66.1.6. Data Is Transformed by PROC ACECLUS

```
                                Fish Measurement Data
                         Data is transformed by PROC ACECLUS

                                  The FREQ Procedure

                             Table of Species by CLUSTER

          Species     CLUSTER(Cluster)

          Frequency |
          Percent   |
          Row Pct   |
          Col Pct   |       1|       2|       3|       4|       5|       6|       7|  Total
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Bream     |     13 |      0 |      0 |      0 |      0 |      0 |     21 |     34
                    |   8.28 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  13.38 |  21.66
                    |  38.24 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  61.76 |
                    |  44.83 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  47.73 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Roach     |      3 |      4 |      0 |      0 |     12 |      0 |      0 |     19
                    |   1.91 |   2.55 |   0.00 |   0.00 |   7.64 |   0.00 |   0.00 |  12.10
                    |  15.79 |  21.05 |   0.00 |   0.00 |  63.16 |   0.00 |   0.00 |
                    |  10.34 |  25.00 |   0.00 |   0.00 |  30.77 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Whitefish |      3 |      0 |      0 |      0 |      0 |      0 |      3 |      6
                    |   1.91 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   1.91 |   3.82
                    |  50.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |  50.00 |
                    |  10.34 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   6.82 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Parkki    |      2 |      3 |      0 |      0 |      6 |      0 |      0 |     11
                    |   1.27 |   1.91 |   0.00 |   0.00 |   3.82 |   0.00 |   0.00 |   7.01
                    |  18.18 |  27.27 |   0.00 |   0.00 |  54.55 |   0.00 |   0.00 |
                    |   6.90 |  18.75 |   0.00 |   0.00 |  15.38 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Perch     |      8 |      9 |      0 |      1 |     20 |      0 |     18 |     56
                    |   5.10 |   5.73 |   0.00 |   0.64 |  12.74 |   0.00 |  11.46 |  35.67
                    |  14.29 |  16.07 |   0.00 |   1.79 |  35.71 |   0.00 |  32.14 |
                    |  27.59 |  56.25 |   0.00 |   6.67 |  51.28 |   0.00 |  40.91 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Pike      |      0 |      0 |     10 |      0 |      1 |      4 |      2 |     17
                    |   0.00 |   0.00 |   6.37 |   0.00 |   0.64 |   2.55 |   1.27 |  10.83
                    |   0.00 |   0.00 |  58.82 |   0.00 |   5.88 |  23.53 |  11.76 |
                    |   0.00 |   0.00 | 100.00 |   0.00 |   2.56 | 100.00 |   4.55 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Smelt     |      0 |      0 |      0 |     14 |      0 |      0 |      0 |     14
                    |   0.00 |   0.00 |   0.00 |   8.92 |   0.00 |   0.00 |   0.00 |   8.92
                    |   0.00 |   0.00 |   0.00 | 100.00 |   0.00 |   0.00 |   0.00 |
                    |   0.00 |   0.00 |   0.00 |  93.33 |   0.00 |   0.00 |   0.00 |
          ----------+--------+--------+--------+--------+--------+--------+--------+
          Total           29       16       10       15       39        4       44      157
                       18.47    10.19     6.37     9.55    24.84     2.55    28.03   100.00
```

Table 66.4 displays a table summarizing each classification results. In this table, the first column represents the standardization method, the second column represents the number of clusters that the 7 species are classified into, and the third column represents the total number of observations that are misclassified.

Table 66.4. Summary of Clustering Results

Method of Standardization	Number of Clusters	Misclassification
MEAN	5	71
MEDIAN	5	71
SUM	6	51
EUCLEN	6	45
USTD	6	45
STD	5	33
RANGE	7	32
MIDRANGE	7	32
MAXABS	7	26
IQR	5	28
MAD	4	35
ABW(5)	6	34
AWAVE(5)	6	29
AGK(.14)	7	28
SPACING(.14)	7	25
L(1)	6	41
L(1.5)	5	33
L(2)	5	33
untransformed	5	71
PROC ACECLUS	5	71

Consider the results displayed in Output 66.1.1. In that analysis, the method of standardization is STD, and the number of clusters and the number of misclassifications are computed as shown in Table 66.5.

Table 66.5. Computations of Numbers of Clusters and Misclassification When Standardization Method Is STD

Species	Cluster Number	Misclassification in Each Species
Bream	6	0
Roach	7	0
Whitefish	7	3
Parkki	5	0
Perch	7	29
Pike	1	0
Smelt	3	1

In Output 66.1.1, the Bream species is classified as cluster 6 since all 34 Bream fish are categorized into cluster 6 with no misclassification. A similar pattern is seen with the Roach, Parkki, Pike, and Smelt species.

For the Whitefish species, two fish are categorized into cluster 2, one fish is categorized into cluster 4, and three fish are categorized into cluster 7. Because the majority of this species is categorized into cluster 7, it is recorded in Table 66.5 as being classified as cluster 7 with 3 misclassifications. A similar pattern is seen with the Perch species: it is classified as cluster 7 with 29 misclassifications.

In summary, when the standardization method is STD, seven species of fish are classified into only 5 clusters and the total number of misclassified observations is 33.

The result of this analysis demonstrates that when variables are standardized by the STDIZE procedure with methods including RANGE, MIDRANGE, MAXABS, AGK(.14), and SPACING(.14), the FASTCLUS procedure produces the correct number of clusters and less misclassification than it does when other standardization methods are used. The SPACING method attains the best result, probably because the variables Length1 and Height both exhibit marked groupings (bimodality) in their distributions.

References

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), “Data-based Metrics for Cluster Analysis,” *Utilitas Mathematica*, 21A, 75–99.

Goodall, C. (1983), “M-Estimators of Location: An Outline of Theory,” in Hoaglin, D.C., Mosteller, M. and Tukey, J.W., eds., *Understanding Robust and Exploratory Data Analysis*, New York: John Wiley & Sons, Inc.

Iglewicz, B. (1983), “Robust Scale Estimators and Confidence Intervals for Location,” in Hoaglin, D.C., Mosteller, M. and Tukey, J.W., eds., *Understanding Robust and Exploratory Data Analysis*, New York: John Wiley & Sons, Inc.

Jannsen, P., Marron, J.S., Veraverbeke, N, and Sarle, W.S. (1995), “Scale Measures for Bandwidth Selection,” *J. of Nonparametric Statistics*, 5(4), 359–380.

Jain R. and Chlamtac I. (1985), “The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations,” *Communications of the ACM*, 28(10), 1076-1085.

Journal of Statistics Education, “Fish Catch Data Set,” [http://www.stat.ncsu.edu/info/jse], accessed 4 December 1997.

Milligan, G.W. and Cooper, M.C. (1987), “A Study of Variable Standardization,” *College of Administrative Science Working Paper Series*, 87–63, Columbus, OH: Ohio State University.

Chapter 67
The STEPDISC Procedure

Chapter Contents

Chapter 67
The STEPDISC Procedure

Overview

Given a classification variable and several quantitative variables, the STEPDISC procedure performs a stepwise discriminant analysis to select a subset of the quantitative variables for use in discriminating among the classes. The set of variables that make up each class is assumed to be multivariate normal with a common covariance matrix. The STEPDISC procedure can use forward selection, backward elimination, or stepwise selection (Klecka 1980). The STEPDISC procedure is a useful prelude to further analyses using the CANDISC procedure or the DISCRIM procedure.

With PROC STEPDISC, variables are chosen to enter or leave the model according to one of two criteria:

- the significance level of an F-test from an analysis of covariance, where the variables already chosen act as covariates and the variable under consideration is the dependent variable
- the squared partial correlation for predicting the variable under consideration from the CLASS variable, controlling for the effects of the variables already selected for the model

Forward selection begins with no variables in the model. At each step, PROC STEPDISC enters the variable that contributes most to the discriminatory power of the model as measured by Wilks' Lambda, the likelihood ratio criterion. When none of the unselected variables meets the entry criterion, the forward selection process stops.

Backward elimination begins with all variables in the model except those that are linearly dependent on previous variables in the VAR statement. At each step, the variable that contributes least to the discriminatory power of the model as measured by Wilks' Lambda is removed. When all remaining variables meet the criterion to stay in the model, the backward elimination process stops.

Stepwise selection begins, like forward selection, with no variables in the model. At each step, the model is examined. If the variable in the model that contributes least to the discriminatory power of the model as measured by Wilks' lambda fails to meet the criterion to stay, then that variable is removed. Otherwise, the variable not in the model that contributes most to the discriminatory power of the model is entered. When all variables in the model meet the criterion to stay and none of the other variables meet the criterion to enter, the stepwise selection process stops. Stepwise selection is the default method of variable selection.

It is important to realize that, in the selection of variables for entry, only one variable can be entered into the model at each step. The selection process does not take into account the relationships between variables that have not yet been selected. Thus, some important variables could be excluded in the process. Also, Wilks' Lambda may not be the best measure of discriminatory power for your application. However, if you use PROC STEPDISC carefully, in combination with your knowledge of the data and careful cross-validation, it can be a valuable aid in selecting variables for a discrimination model.

As with any stepwise procedure, it is important to remember that, when many significance tests are performed, each at a level of, for example, 5% (0.05), the overall probability of rejecting at least one true null hypothesis is much larger than 5%. If you want to prevent including any variables that do not contribute to the discriminatory power of the model in the population, you should specify a very small significance level. In most applications, all variables considered have some discriminatory power, however small. To choose the model that provides the best discrimination using the sample estimates, you need only to guard against estimating more parameters than can be reliably estimated with the given sample size.

Costanza and Afifi (1979) use Monte Carlo studies to compare alternative stopping rules that can be used with the forward selection method in the two-group multivariate normal classification problem. Five different numbers of variables, ranging from 10 to 30, are considered in the studies. The comparison is based on conditional and estimated unconditional probabilities of correct classification. They conclude that the use of a moderate significance level, in the range of 10 percent to 25 percent, often performs better than the use of a much larger or a much smaller significance level.

The significance level and the squared partial correlation criteria select variables in the same order, although they may select different numbers of variables. Increasing the sample size tends to increase the number of variables selected when using significance levels, but it has little effect on the number selected using squared partial correlations.

See Chapter 6, "Introduction to Discriminant Procedures," for more information on discriminant analysis.

Getting Started

The data in this example are measurements on 159 fish caught in Finland's lake Laengelmavesi; this data set is available from the Data Archive of the *Journal of Statistics Education*. For each of the seven species (bream, parkki, pike, perch, roach, smelt, and whitefish), the weight, length, height, and the width of each fish are tallied. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail, from the nose to the notch of its tail, and from the nose to the end of its tail. The height and width are recorded as percentages of the third length variable. PROC STEPDISC will select a subset of the six quantitative variables that may be useful for differentiating between the fish species. This subset is used in conjunction with PROC CANDISC and PROC DISCRIM to develop discrimination models.

The following program creates the data set fish and uses PROC STEPDISC to select a subset of potential discriminator variables. By default, PROC STEPDISC uses stepwise selection on all numeric variables that are not listed in other statements, and the significance levels for a variable to enter the subset and to stay in the subset are set to 0.15.

```
proc format;
   value specfmt
      1=’Bream’
      2=’Roach’
      3=’Whitefish’
      4=’Parkki’
      5=’Perch’
      6=’Pike’
      7=’Smelt’;
data fish (drop=HtPct WidthPct);
   title ’Fish Measurement Data’;
   input Species Weight Length1 Length2 Length3 HtPct WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1 1  363.0 26.3 29.0 33.5 38.0 13.3
1  430.0 26.5 29.0 34.0 36.6 15.1 1  450.0 26.8 29.7 34.7 39.2 14.2
1  500.0 26.8 29.7 34.5 41.1 15.3 1  390.0 27.6 30.0 35.0 36.2 13.4
1  450.0 27.6 30.0 35.1 39.9 13.8 1  500.0 28.5 30.7 36.2 39.3 13.7
1  475.0 28.4 31.0 36.2 39.4 14.1 1  500.0 28.7 31.0 36.2 39.7 13.3
1  500.0 29.1 31.5 36.4 37.8 12.0 1     .  29.5 32.0 37.3 37.3 13.6
1  600.0 29.4 32.0 37.2 40.2 13.9 1  600.0 29.4 32.0 37.2 41.5 15.0
1  700.0 30.4 33.0 38.3 38.8 13.8 1  700.0 30.4 33.0 38.5 38.8 13.5
1  610.0 30.9 33.5 38.6 40.5 13.3 1  650.0 31.0 33.5 38.7 37.4 14.8
1  575.0 31.3 34.0 39.5 38.3 14.1 1  685.0 31.4 34.0 39.2 40.8 13.7
1  620.0 31.5 34.5 39.7 39.1 13.3 1  680.0 31.8 35.0 40.6 38.1 15.1
1  700.0 31.9 35.0 40.5 40.1 13.8 1  725.0 31.8 35.0 40.9 40.0 14.8
1  720.0 32.0 35.0 40.6 40.3 15.0 1  714.0 32.7 36.0 41.5 39.8 14.1
1  850.0 32.8 36.0 41.6 40.6 14.9 1 1000.0 33.5 37.0 42.6 44.5 15.5
1  920.0 35.0 38.5 44.1 40.9 14.3 1  955.0 35.0 38.5 44.0 41.1 14.3
1  925.0 36.2 39.5 45.3 41.4 14.9 1  975.0 37.4 41.0 45.9 40.6 14.7
1  950.0 38.0 41.0 46.5 37.9 13.7
2   40.0 12.9 14.1 16.2 25.6 14.0 2   69.0 16.5 18.2 20.3 26.1 13.9
2   78.0 17.5 18.8 21.2 26.3 13.7 2   87.0 18.2 19.8 22.2 25.3 14.3
2  120.0 18.6 20.0 22.2 28.0 16.1 2    0.0 19.0 20.5 22.8 28.4 14.7
2  110.0 19.1 20.8 23.1 26.7 14.7 2  120.0 19.4 21.0 23.7 25.8 13.9
2  150.0 20.4 22.0 24.7 23.5 15.2 2  145.0 20.5 22.0 24.3 27.3 14.6
2  160.0 20.5 22.5 25.3 27.8 15.1 2  140.0 21.0 22.5 25.0 26.2 13.3
2  160.0 21.1 22.5 25.0 25.6 15.2 2  169.0 22.0 24.0 27.2 27.7 14.1
2  161.0 22.0 23.4 26.7 25.9 13.6 2  200.0 22.1 23.5 26.8 27.6 15.4
2  180.0 23.6 25.2 27.9 25.4 14.0 2  290.0 24.0 26.0 29.2 30.4 15.4
2  272.0 25.0 27.0 30.6 28.0 15.6 2  390.0 29.5 31.7 35.0 27.1 15.3
3  270.0 23.6 26.0 28.7 29.2 14.8 3  270.0 24.1 26.5 29.3 27.8 14.5
3  306.0 25.6 28.0 30.8 28.5 15.2 3  540.0 28.5 31.0 34.0 31.6 19.3
3  800.0 33.7 36.4 39.6 29.7 16.6 3 1000.0 37.3 40.0 43.5 28.4 15.0
4   55.0 13.5 14.7 16.5 41.5 14.1 4   60.0 14.3 15.5 17.4 37.8 13.3
4   90.0 16.3 17.7 19.8 37.4 13.5 4  120.0 17.5 19.0 21.3 39.4 13.7
4  150.0 18.4 20.0 22.4 39.7 14.7 4  140.0 19.0 20.7 23.2 36.8 14.2
4  170.0 19.0 20.7 23.2 40.5 14.7 4  145.0 19.8 21.5 24.1 40.4 13.1
```

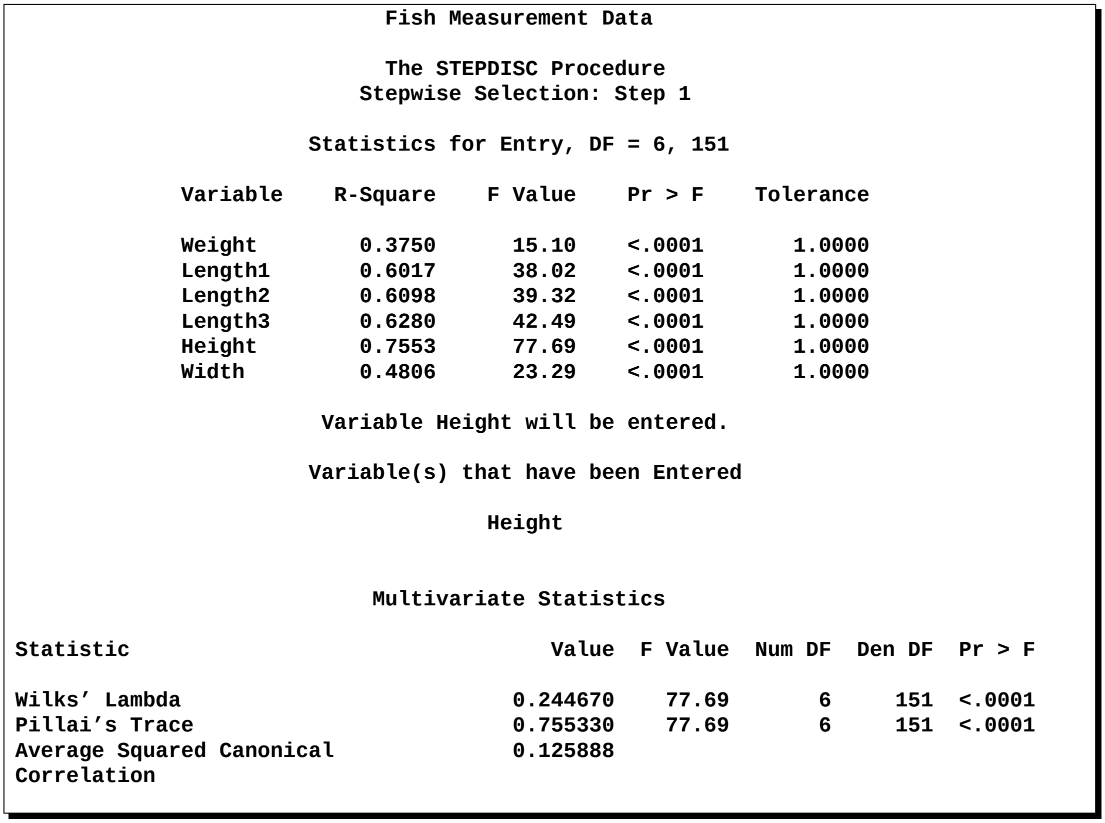
```
4  200.0 21.2 23.0 25.8 40.1 14.2 4  273.0 23.0 25.0 28.0 39.6 14.8
4  300.0 24.0 26.0 29.0 39.2 14.6
5    5.9  7.5  8.4  8.8 24.0 16.0 5   32.0 12.5 13.7 14.7 24.0 13.6
5   40.0 13.8 15.0 16.0 23.9 15.2 5   51.5 15.0 16.2 17.2 26.7 15.3
5   70.0 15.7 17.4 18.5 24.8 15.9 5  100.0 16.2 18.0 19.2 27.2 17.3
5   78.0 16.8 18.7 19.4 26.8 16.1 5   80.0 17.2 19.0 20.2 27.9 15.1
5   85.0 17.8 19.6 20.8 24.7 14.6 5   85.0 18.2 20.0 21.0 24.2 13.2
5  110.0 19.0 21.0 22.5 25.3 15.8 5  115.0 19.0 21.0 22.5 26.3 14.7
5  125.0 19.0 21.0 22.5 25.3 16.3 5  130.0 19.3 21.3 22.8 28.0 15.5
5  120.0 20.0 22.0 23.5 26.0 14.5 5  120.0 20.0 22.0 23.5 24.0 15.0
5  130.0 20.0 22.0 23.5 26.0 15.0 5  135.0 20.0 22.0 23.5 25.0 15.0
5  110.0 20.0 22.0 23.5 23.5 17.0 5  130.0 20.5 22.5 24.0 24.4 15.1
5  150.0 20.5 22.5 24.0 28.3 15.1 5  145.0 20.7 22.7 24.2 24.6 15.0
5  150.0 21.0 23.0 24.5 21.3 14.8 5  170.0 21.5 23.5 25.0 25.1 14.9
5  225.0 22.0 24.0 25.5 28.6 14.6 5  145.0 22.0 24.0 25.5 25.0 15.0
5  188.0 22.6 24.6 26.2 25.7 15.9 5  180.0 23.0 25.0 26.5 24.3 13.9
5  197.0 23.5 25.6 27.0 24.3 15.7 5  218.0 25.0 26.5 28.0 25.6 14.8
5  300.0 25.2 27.3 28.7 29.0 17.9 5  260.0 25.4 27.5 28.9 24.8 15.0
5  265.0 25.4 27.5 28.9 24.4 15.0 5  250.0 25.4 27.5 28.9 25.2 15.8
5  250.0 25.9 28.0 29.4 26.6 14.3 5  300.0 26.9 28.7 30.1 25.2 15.4
5  320.0 27.8 30.0 31.6 24.1 15.1 5  514.0 30.5 32.8 34.0 29.5 17.7
5  556.0 32.0 34.5 36.5 28.1 17.5 5  840.0 32.5 35.0 37.3 30.8 20.9
5  685.0 34.0 36.5 39.0 27.9 17.6 5  700.0 34.0 36.0 38.3 27.7 17.6
5  700.0 34.5 37.0 39.4 27.5 15.9 5  690.0 34.6 37.0 39.3 26.9 16.2
5  900.0 36.5 39.0 41.4 26.9 18.1 5  650.0 36.5 39.0 41.4 26.9 14.5
5  820.0 36.6 39.0 41.3 30.1 17.8 5  850.0 36.9 40.0 42.3 28.2 16.8
5  900.0 37.0 40.0 42.5 27.6 17.0 5 1015.0 37.0 40.0 42.4 29.2 17.6
5  820.0 37.1 40.0 42.5 26.2 15.6 5 1100.0 39.0 42.0 44.6 28.7 15.4
5 1000.0 39.8 43.0 45.2 26.4 16.1 5 1100.0 40.1 43.0 45.5 27.5 16.3
5 1000.0 40.2 43.5 46.0 27.4 17.7 5 1000.0 41.1 44.0 46.6 26.8 16.3
6  200.0 30.0 32.3 34.8 16.0  9.7 6  300.0 31.7 34.0 37.8 15.1 11.0
6  300.0 32.7 35.0 38.8 15.3 11.3 6  300.0 34.8 37.3 39.8 15.8 10.1
6  430.0 35.5 38.0 40.5 18.0 11.3 6  345.0 36.0 38.5 41.0 15.6  9.7
6  456.0 40.0 42.5 45.5 16.0  9.5 6  510.0 40.0 42.5 45.5 15.0  9.8
6  540.0 40.1 43.0 45.8 17.0 11.2 6  500.0 42.0 45.0 48.0 14.5 10.2
6  567.0 43.2 46.0 48.7 16.0 10.0 6  770.0 44.8 48.0 51.2 15.0 10.5
6  950.0 48.3 51.7 55.1 16.2 11.2 6 1250.0 52.0 56.0 59.7 17.9 11.7
6 1600.0 56.0 60.0 64.0 15.0  9.6 6 1550.0 56.0 60.0 64.0 15.0  9.6
6 1650.0 59.0 63.4 68.0 15.9 11.0
7    6.7  9.3  9.8 10.8 16.1  9.7 7    7.5 10.0 10.5 11.6 17.0 10.0
7    7.0 10.1 10.6 11.6 14.9  9.9 7    9.7 10.4 11.0 12.0 18.3 11.5
7    9.8 10.7 11.2 12.4 16.8 10.3 7    8.7 10.8 11.3 12.6 15.7 10.2
7   10.0 11.3 11.8 13.1 16.9  9.8 7    9.9 11.3 11.8 13.1 16.9  8.9
7    9.8 11.4 12.0 13.2 16.7  8.7 7   12.2 11.5 12.2 13.4 15.6 10.4
7   13.4 11.7 12.4 13.5 18.0  9.4 7   12.2 12.1 13.0 13.8 16.5  9.1
7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;
proc stepdisc data=fish;
   class Species;
run;
```

PROC STEPDISC begins by displaying summary information about the analysis; see Figure 67.1. This information includes the number of observations with nonmissing values, the number of classes in the classification variable (specified by the CLASS statement), the number of quantitative variables under consideration, the significance criteria for variables to enter and to stay in the model, and the method of variable selection being used. The frequency of each class is also displayed.

```
                              Fish Measurement Data

                             The STEPDISC Procedure

                  The Method for Selecting Variables is STEPWISE

      Observations        158           Variable(s) in the Analysis        6
      Class Levels          7           Variable(s) will be Included       0
                                        Significance Level to Enter     0.15
                                        Significance Level to Stay      0.15


                             Class Level Information

                         Variable
         Species         Name            Frequency        Weight     Proportion

         Bream           Bream                  34       34.0000       0.215190
         Parkki          Parkki                 11       11.0000       0.069620
         Perch           Perch                  56       56.0000       0.354430
         Pike            Pike                   17       17.0000       0.107595
         Roach           Roach                  20       20.0000       0.126582
         Smelt           Smelt                  14       14.0000       0.088608
         Whitefish       Whitefish               6        6.0000       0.037975
```

Figure 67.1. Summary Information

For each entry step, the statistics for entry are displayed for all variables not currently selected; see Figure 67.2. The variable selected to enter at this step (if any) is displayed, as well as all the variables currently selected. Next are multivariate statistics that take into account all previously selected variables and the newly entered variable.



```
                              Fish Measurement Data

                              The STEPDISC Procedure
                            Stepwise Selection: Step 1

                        Statistics for Entry, DF = 6, 151

             Variable    R-Square    F Value    Pr > F    Tolerance

             Weight        0.3750      15.10    <.0001       1.0000
             Length1       0.6017      38.02    <.0001       1.0000
             Length2       0.6098      39.32    <.0001       1.0000
             Length3       0.6280      42.49    <.0001       1.0000
             Height        0.7553      77.69    <.0001       1.0000
             Width         0.4806      23.29    <.0001       1.0000

                         Variable Height will be entered.

                        Variable(s) that have been Entered

                                      Height


                             Multivariate Statistics

Statistic                                 Value  F Value  Num DF  Den DF  Pr > F

Wilks' Lambda                          0.244670    77.69       6     151  <.0001
Pillai's Trace                         0.755330    77.69       6     151  <.0001
Average Squared Canonical              0.125888
Correlation
```

Figure 67.2. Step 1: Variable HEIGHT Selected for Entry

For each removal step (Figure 67.3), the statistics for removal are displayed for all variables currently entered. The variable to be removed at this step (if any) is displayed. If no variable meets the criterion to be removed and the maximum number of steps as specified by the MAXSTEP= option has not been attained, then the procedure continues with another entry step.

```
                              Fish Measurement Data

                             The STEPDISC Procedure
                           Stepwise Selection: Step 2

                      Statistics for Removal, DF = 6, 151

                    Variable    R-Square    F Value    Pr > F

                    Height        0.7553      77.69    <.0001

                          No variables can be removed.


                       Statistics for Entry, DF = 6, 150

                          Partial
              Variable    R-Square    F Value    Pr > F    Tolerance

              Weight        0.7388      70.71    <.0001       0.4690
              Length1       0.9220     295.35    <.0001       0.6083
              Length2       0.9229     299.31    <.0001       0.5892
              Length3       0.9173     277.37    <.0001       0.5056
              Width         0.8783     180.44    <.0001       0.3699

                       Variable Length2 will be entered.

                       Variable(s) that have been Entered

                                 Length2 Height


                            Multivariate Statistics

Statistic                                  Value  F Value  Num DF  Den DF  Pr > F

Wilks' Lambda                           0.018861   157.04      12     300  <.0001
Pillai's Trace                          1.554349    87.78      12     302  <.0001
Average Squared Canonical               0.259058
Correlation
```

Figure 67.3. Step 2: No Variable is Removed; Variable Length1 Added

The stepwise procedure terminates either when no variable can be removed and no variable can be entered or when the maximum number of steps as specified by the MAXSTEP= option has been attained. In this example at Step 7 no variables can be either removed or entered (Figure 67.4). Steps 3 through 6 are not displayed in this document.

```
                           Fish Measurement Data

                           The STEPDISC Procedure
                         Stepwise Selection: Step 7

                    Statistics for Removal, DF = 6, 146

                                Partial
                    Variable    R-Square    F Value    Pr > F

                    Weight        0.4521      20.08    <.0001
                    Length1       0.2987      10.36    <.0001
                    Length2       0.5250      26.89    <.0001
                    Length3       0.7948      94.25    <.0001
                    Height        0.7257      64.37    <.0001
                    Width         0.5757      33.02    <.0001

                        No variables can be removed.

                       No further steps are possible.
```

Figure 67.4. Step 7: No Variables Entered or Removed

PROC STEPDISC ends by displaying a summary of the steps.

```
                                  Fish Measurement Data

                                 The STEPDISC Procedure

                               Stepwise Selection Summary

                                                                                Average
                                                                                Squared
         Number                      Partial                        Wilks'     Pr <    Canonical     Pr >
  Step       In  Entered   Removed  R-Square  F Value  Pr > F       Lambda   Lambda   Correlation     ASCC

     1        1  Height               0.7553    77.69  <.0001   0.24466983   <.0001    0.12588836   <.0001
     2        2  Length2              0.9229   299.31  <.0001   0.01886065   <.0001    0.25905822   <.0001
     3        3  Length3              0.8826   186.77  <.0001   0.00221342   <.0001    0.38427100   <.0001
     4        4  Width                0.5775    33.72  <.0001   0.00093510   <.0001    0.45200732   <.0001
     5        5  Weight               0.4461    19.73  <.0001   0.00051794   <.0001    0.49488458   <.0001
     6        6  Length1              0.2987    10.36  <.0001   0.00036325   <.0001    0.51744189   <.0001
```

Figure 67.5. Step Summary

All the variables in the data set are found to have potential discriminatory power. These variables are used to develop discrimination models in both the CANDISC and DISCRIM procedure chapters.

Syntax

The following statements are available in PROC STEPDISC.

PROC STEPDISC < *options* > **;**
  CLASS *variable* **;**

  BY *variables* **;**
  FREQ *variable* **;**
  VAR *variables* **;**
  WEIGHT *variable* **;**

The BY, CLASS, FREQ, VAR, and WEIGHT statements are described after the PROC STEPDISC statement.

PROC STEPDISC Statement

PROC STEPDISC < *options* > **;**

The PROC STEPDISC statement invokes the STEPDISC procedure. The PROC STEPDISC statement has the following options.

Table 67.1. STEPDISC Procedure Options

Task	**Options**
Specify Data Set	DATA=
Select Method	METHOD=
Selection Criterion	SLENTRY=
	SLSTAY=
	PR2ENTRY=
	PR2STAY=
Selection Process	INCLUDE=
	MAXSTEP=
	START=
	STOP=
Determine Singularity	SINGULAR=
Control Displayed Output	
Correlations	BCORR
	PCORR
	TCORR
	WCORR
Covariances	BCOV
	PCOV
	TCOV
	WCOV
SSCP Matrices	BSSCP
	PSSCP

Table 67.1. (continued)

Task	Options
	TSSCP WSSCP
Miscellaneous	ALL SIMPLE STDMEAN
Suppress Output	SHORT

ALL

activates all of the display options.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. The between-class covariances should be interpreted in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

DATA=*SAS-data-set*

specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. These specially structured data sets include TYPE=CORR, COV, CSSCP, and SSCP. If the DATA= option is omitted, the procedure uses the most recently created SAS data set.

INCLUDE=*n*

includes the first n variables in the VAR statement in every model. By default, INCLUDE=0.

MAXSTEP=*n*

specifies the maximum number of steps. By default, MAXSTEP= two times the number of variables in the VAR statement.

METHOD=BACKWARD | BW
METHOD=FORWARD | FW
METHOD=STEPWISE | SW

specifies the method used to select the variables in the model. The BACKWARD method specifies backward elimination, FORWARD specifies forward selection, and STEPWISE specifies stepwise selection. By default, METHOD=STEPWISE.

PCORR

displays pooled within-class correlations (partial correlations based on the pooled within-class covariances).

PCOV

displays pooled within-class covariances.

PR2ENTRY=*p*
PR2E=*p*

specifies the partial R^2 for adding variables in the forward selection mode, where $p \leq 1$.

PR2STAY=*p*
PR2S=*p*

specifies the partial R^2 for retaining variables in the backward elimination mode, where $p \leq 1$.

PSSCP

displays the pooled within-class corrected SSCP matrix.

SHORT

suppresses the displayed output from each step.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the singularity criterion for entering variables, where $0 < p < 1$. PROC STEPDISC precludes the entry of a variable if the squared multiple correlation of the variable with the variables already in the model exceeds $1 - p$. With more than one variable already in the model, PROC STEPDISC also excludes a variable if it would cause any of the variables already in the model to have a squared multiple correlation (with the entering variable and the other variables in the model) exceeding $1 - p$. By default, SINGULAR= 1E$-$8.

SLENTRY=*p*
SLE=*p*

specifies the significance level for adding variables in the forward selection mode, where $0 \leq p \leq 1$. The default value is 0.15.

SLSTAY=*p*
SLS=*p*

specifies the significance level for retaining variables in the backward elimination mode, where $0 \leq p \leq 1$. The default value is 0.15.

START=*n*

specifies that the first n variables in the VAR statement be used to begin the selection process. When you specify METHOD=FORWARD or METHOD=STEPWISE, the default value is 0; when you specify METHOD=BACKWARD, the default value is the number of variables in the VAR statement.

STDMEAN
displays total-sample and pooled within-class standardized class means.

STOP=*n*
specifies the number of variables in the final model. The STEPDISC procedure stops the selection process when a model with *n* variables is found. This option applies only when you specify METHOD=FORWARD or METHOD=BACKWARD. When you specify METHOD=FORWARD, the default value is the number of variables in the VAR statement; when you specify METHOD=BACKWARD, the default value is 0.

TCORR
displays total-sample correlations.

TCOV
displays total-sample covariances.

TSSCP
displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC STEPDISC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the STEPDISC procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The values of the CLASS variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The CLASS variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the name of the variable in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the quantitative variables eligible for selection. The default is all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the WEIGHT variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is assumed.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details

Missing Values

Observations containing missing values are omitted from the analysis.

Input Data Sets

The input data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. For more information on these data sets, see Appendix A, "Special SAS Data Sets." The BY variable in these data sets becomes the CLASS variable in PROC STEPDISC. These specially structured data sets include

- TYPE=CORR data sets created by PROC CORR using a BY statement
- TYPE=COV data sets created by PROC PRINCOMP using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG using both the OUTSSCP= option and a BY statement

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, the STEPDISC procedure reads the number of observations for each class from the observations with _TYPE_='N' and the variable means in each class from the observations with _TYPE_='MEAN'. The procedure then reads the within-class correlations from the observations with _TYPE_='CORR', the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and crossproducts from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When the data set does not include any observations with _TYPE_='CORR' (data set TYPE=CORR), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CSSCP' (data set TYPE=CSSCP) for each class, PROC STEPDISC reads the pooled within-class information from the data set. In this case, the STEPDISC procedure reads the pooled within-class correlations from the observations with _TYPE_='PCORR', the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR), the pooled within-class covariances from the observations with _TYPE_='PCOV' (data set TYPE=COV), or the pooled within-class corrected SSCP matrix from the observations with_TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, the STEPDISC procedure reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations from the variable INTERCEPT in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and crossproducts from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_=*variablenames*.

Computational Resources

In the following discussion, let

$$\begin{aligned} n &= \text{number of observations} \\ c &= \text{number of class levels} \\ v &= \text{number of variables in the VAR list} \\ l &= \text{length of the CLASS variable} \\ t &= v + c - 1. \end{aligned}$$

Memory Requirements

The amount of memory in bytes for temporary storage needed to process the data is

$$c(4v^2 + 28v + 3l + 4c + 72) + 16v^2 + 92v + 4t^2 + 20t + 4l$$

Additional temporary storage of 72 bytes at each step is also required to store the results.

Time Requirements

The following factors determine the time requirements of a stepwise discriminant analysis.

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. The STEPDISC procedure must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n\ln(c)$.
- The time needed for stepwise discriminant analysis is proportional to the number of steps required to select the set of variables in the discrimination model. The number of steps required depends on the data set itself and the selection method and criterion used in the procedure. Each forward or backward step takes time proportional to $(v + c)^2$.

Displayed Output

The STEPDISC procedure displays the following output:

- Class Level Information, including the values of the classification variable, the Frequency of each value, the Weight of each value, and the Proportion of each value in the total sample

Optional output includes

- Within-Class SSCP Matrices for each group
- Pooled Within-Class SSCP Matrix
- Between-Class SSCP Matrix
- Total-Sample SSCP Matrix
- Within-Class Covariance Matrices for each group
- Pooled Within-Class Covariance Matrix
- Between-Class Covariance Matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-Sample Covariance Matrix
- Within-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled Within-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-Class Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the between-class population correlation coefficients are zero
- Total-Sample Correlation Coefficients and $\mathrm{Pr} > |r|$ to test the hypothesis that the total population correlation coefficients are zero
- descriptive Simple Statistics including N (the number of observations), Sum, Mean, Variance, and Standard Deviation for the total sample and within each class
- Total-Sample Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the total-sample standard deviation
- Pooled Within-Class Standardized Class Means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation

At each step, the following statistics are displayed:

- for each variable considered for entry or removal: Partial R-Square, the squared (partial) correlation, the F statistic, and $\mathrm{Pr} > F$, the probability level, from a one-way analysis of covariance

- the minimum Tolerance for entering each variable. A variable is entered only if its tolerance and the tolerances for all variables already in the model are greater than the value specified in the SINGULAR= option. The tolerance for the entering variable is $1 - R^2$ from regressing the entering variable on the other variables already in the model. The tolerance for a variable already in the model is $1 - R^2$ from regressing that variable on the entering variable and the other variables already in the model. With m variables already in the model, for each entering variable, $m + 1$ multiple regressions are performed using the entering variable and each of the m variables already in the model as a dependent variable. These $m + 1$ tolerances are computed for each entering variable, and the minimum tolerance is displayed for each.

  The tolerance is computed using the total-sample correlation matrix. It is customary to compute tolerance using the pooled within-class correlation matrix (Jennrich 1977), but it is possible for a variable with excellent discriminatory power to have a high total-sample tolerance and a low pooled within-class tolerance. For example, PROC STEPDISC enters a variable that yields perfect discrimination (that is, produces a canonical correlation of one), but a program using pooled within-class tolerance does not.

- the variable Label, if any
- the name of the variable chosen
- the variables already selected or removed
- Wilks' Lambda and the associated F approximation with degrees of freedom and $\mathrm{Pr} < F$, the associated probability level after the selected variable has been entered or removed. Wilks' lambda is the likelihood ratio statistic for testing the hypothesis that the means of the classes on the selected variables are equal in the population (see the "Multivariate Tests" section in Chapter 2, "Introduction to Regression Procedures.") Lambda is close to zero if any two groups are well separated.
- Pillai's Trace and the associated F approximation with degrees of freedom and $\mathrm{Pr} > F$, the associated probability level after the selected variable has been entered or removed. Pillai's trace is a multivariate statistic for testing the hypothesis that the means of the classes on the selected variables are equal in the population (see the "Multivariate Tests" section in Chapter 2).
- Average Squared Canonical Correlation (ASCC). The ASCC is Pillai's trace divided by the number of groups minus 1. The ASCC is close to 1 if all groups are well separated and if all or most directions in the discriminant space show good separation for at least two groups.
- Summary to give statistics associated with the variable chosen at each step. The summary includes the following:
  - Step number
  - Variable Entered or Removed
  - Number In, the number of variables in the model
  - Partial R-Square
  - the F Value for entering or removing the variable

- $\mathrm{Pr} > F$, the probability level for the F statistic
- Wilks’ Lambda
- $\mathrm{Pr} < \mathrm{Lambda}$ based on the F approximation to Wilks’ Lambda
- Average Squared Canonical Correlation
- $\mathrm{Pr} > \mathrm{ASCC}$ based on the F approximation to Pillai’s trace
- the variable Label, if any

ODS Table Names

PROC STEPDISC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 67.2. ODS Tables Produced in PROC STEPDISC

ODS Table Name	Description	PROC STEPWISE Option
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
Counts	Number of observations, variables, classes, df	default
CovDF	DF for covariance matrices, not printed	any *COV option
Levels	Class level information	default
Messages	Entry/removal messages	default
Multivariate	Multivariate statistics	default
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
SimpleStatistics	Simple statistics	SIMPLE
Steps	Stepwise selection entry/removal	default
Summary	Stepwise selection summary	default
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
Variables	Variable lists	default
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Example

Example 67.1. Performing a Stepwise Discriminant Analysis

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on fifty iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*.

```
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
data iris;
   title ’Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
```

```
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
```

A stepwise discriminant analysis is performed using stepwise selection.

In the PROC STEPDISC statement, the BSSCP and TSSCP options display the between-class SSCP matrix and the total-sample corrected SSCP matrix. By default, the significance level of an F test from an analysis of covariance is used as the selection criterion. The variable under consideration is the dependent variable, and the variables already chosen act as covariates. The following SAS statements produce Output 67.1.1 through Output 67.1.8:

```
proc stepdisc data=iris bsscp tsscp;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

Output 67.1.1. Iris Data: Summary Information

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure

                        The Method for Selecting Variables is STEPWISE

             Observations       150         Variable(s) in the Analysis        4
             Class Levels         3         Variable(s) will be Included       0
                                            Significance Level to Enter     0.15
                                            Significance Level to Stay      0.15


                                   Class Level Information

                              Variable
               Species        Name          Frequency       Weight    Proportion

               Setosa         Setosa               50      50.0000      0.333333
               Versicolor     Versicolor           50      50.0000      0.333333
               Virginica      Virginica            50      50.0000      0.333333
```

Output 67.1.2. Iris Data: Between-Class and Total-Sample SSCP Matrices

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure

                                   Between-Class SSCP Matrix

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.      6321.21333    -1995.26667    16524.84000     7127.93333
 SepalWidth    Sepal Width in mm.      -1995.26667     1134.49333    -5723.96000    -2293.26667
 PetalLength   Petal Length in mm.     16524.84000    -5723.96000    43710.28000    18677.40000
 PetalWidth    Petal Width in mm.       7127.93333    -2293.26667    18677.40000     8041.33333


                                   Total-Sample SSCP Matrix

 Variable      Label                  SepalLength     SepalWidth    PetalLength     PetalWidth

 SepalLength   Sepal Length in mm.     10216.83333     -632.26667    18987.30000     7692.43333
 SepalWidth    Sepal Width in mm.       -632.26667     2830.69333    -4911.88000    -1812.42667
 PetalLength   Petal Length in mm.     18987.30000    -4911.88000    46432.54000    19304.58000
 PetalWidth    Petal Width in mm.       7692.43333    -1812.42667    19304.58000     8656.99333
```

In Step 1, the tolerance is 1.0 for each variable under consideration because no variables have yet entered the model. Variable PetalLength is selected because its F statistic, 1180.161, is the largest among all variables.

Output 67.1.3. Iris Data: Stepwise Selection Step 1

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure
                                  Stepwise Selection: Step 1

                              Statistics for Entry, DF = 2, 147

       Variable       Label                  R-Square    F Value    Pr > F     Tolerance

       SepalLength    Sepal Length in mm.      0.6187     119.26    <.0001        1.0000
       SepalWidth     Sepal Width in mm.       0.4008      49.16    <.0001        1.0000
       PetalLength    Petal Length in mm.      0.9414    1180.16    <.0001        1.0000
       PetalWidth     Petal Width in mm.       0.9289     960.01    <.0001        1.0000

                            Variable PetalLength will be entered.

                               Variable(s) that have been Entered

                                          PetalLength


                                   Multivariate Statistics

Statistic                                       Value    F Value    Num DF    Den DF    Pr > F

Wilks’ Lambda                                0.058628    1180.16         2       147    <.0001
Pillai’s Trace                               0.941372    1180.16         2       147    <.0001
Average Squared Canonical Correlation        0.470686
```

In Step 2, with variable PetalLength already in the model, PetalLength is tested for removal before selecting a new variable for entry. Since PetalLength meets the criterion to stay, it is used as a covariate in the analysis of covariance for variable selection. Variable SepalWidth is selected because its F statistic, 43.035, is the largest among all variables not in the model and its associated tolerance, 0.8164, meets the criterion to enter. The process is repeated in Steps 3 and 4. Variable PetalWidth is entered in Step 3, and variable SepalLength is entered in Step 4.

Output 67.1.4. Iris Data: Stepwise Selection Step 2

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure
                                  Stepwise Selection: Step 2

                              Statistics for Removal, DF = 2, 147

              Variable       Label                 R-Square    F Value    Pr > F

              PetalLength    Petal Length in mm.     0.9414    1180.16    <.0001

                                 No variables can be removed.


                              Statistics for Entry, DF = 2, 146

                                              Partial
       Variable       Label                  R-Square    F Value    Pr > F    Tolerance

       SepalLength    Sepal Length in mm.      0.3198      34.32    <.0001       0.2400
       SepalWidth     Sepal Width in mm.       0.3709      43.04    <.0001       0.8164
       PetalWidth     Petal Width in mm.       0.2533      24.77    <.0001       0.0729

                             Variable SepalWidth will be entered.

                              Variable(s) that have been Entered

                                    SepalWidth PetalLength


                                   Multivariate Statistics

Statistic                                       Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda                                0.036884     307.10         4       292    <.0001
Pillai's Trace                               1.119908      93.53         4       294    <.0001
Average Squared Canonical Correlation        0.559954
```

Output 67.1.5. Iris Data: Stepwise Selection Step 3

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure
                                  Stepwise Selection: Step 3

                              Statistics for Removal, DF = 2, 146

                                                     Partial
              Variable       Label                  R-Square    F Value    Pr > F

              SepalWidth     Sepal Width in mm.       0.3709      43.04    <.0001
              PetalLength    Petal Length in mm.      0.9384    1112.95    <.0001

                                 No variables can be removed.


                              Statistics for Entry, DF = 2, 145

                                              Partial
       Variable       Label                  R-Square    F Value    Pr > F    Tolerance

       SepalLength    Sepal Length in mm.      0.1447      12.27    <.0001       0.1323
       PetalWidth     Petal Width in mm.       0.3229      34.57    <.0001       0.0662

                             Variable PetalWidth will be entered.

                               Variable(s) that have been Entered

                               SepalWidth PetalLength PetalWidth


                                   Multivariate Statistics

Statistic                                       Value    F Value    Num DF    Den DF    Pr > F

Wilks’ Lambda                                0.024976     257.50         6       290    <.0001
Pillai’s Trace                               1.189914      71.49         6       292    <.0001
Average Squared Canonical Correlation        0.594957
```

Output 67.1.6. Iris Data: Stepwise Selection Step 4

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure
                                  Stepwise Selection: Step 4

                              Statistics for Removal, DF = 2, 145

                                                     Partial
              Variable       Label                  R-Square    F Value    Pr > F

              SepalWidth     Sepal Width in mm.       0.4295      54.58    <.0001
              PetalLength    Petal Length in mm.      0.3482      38.72    <.0001
              PetalWidth     Petal Width in mm.       0.3229      34.57    <.0001

                                 No variables can be removed.


                              Statistics for Entry, DF = 2, 144

                                              Partial
       Variable       Label                  R-Square    F Value    Pr > F    Tolerance

       SepalLength    Sepal Length in mm.      0.0615       4.72    0.0103       0.0320

                            Variable SepalLength will be entered.

                               All variables have been entered.


                                   Multivariate Statistics

Statistic                                       Value    F Value    Num DF    Den DF    Pr > F

Wilks’ Lambda                                0.023439     199.15         8       288    <.0001
Pillai’s Trace                               1.191899      53.47         8       290    <.0001
Average Squared Canonical Correlation        0.595949
```

Since no more variables can be added to or removed from the model, the procedure stops at Step 5 and displays a summary of the selection process.

Output 67.1.7. Iris Data: Stepwise Selection Step 5

```
                                    Fisher (1936) Iris Data

                                    The STEPDISC Procedure
                                  Stepwise Selection: Step 5

                              Statistics for Removal, DF = 2, 144

                                                     Partial
              Variable       Label                  R-Square    F Value    Pr > F

              SepalLength    Sepal Length in mm.      0.0615       4.72    0.0103
              SepalWidth     Sepal Width in mm.       0.2335      21.94    <.0001
              PetalLength    Petal Length in mm.      0.3308      35.59    <.0001
              PetalWidth     Petal Width in mm.       0.2570      24.90    <.0001

                                 No variables can be removed.

                                No further steps are possible.
```

Output 67.1.8. Iris Data: Stepwise Selection Summary

```
                                          Fisher (1936) Iris Data

                                          The STEPDISC Procedure

                                         Stepwise Selection Summary

                                                                                                   Average
                                                                                                   Squared
        Number                                              Partial                    Wilks'   Pr <  Canonical    Pr >
   Step     In Entered     Removed     Label               R-Square F Value Pr > F     Lambda Lambda Correlation   ASCC

      1      1 PetalLength             Petal Length in mm.   0.9414 1180.16 <.0001 0.05862828 <.0001  0.47068586 <.0001
      2      2 SepalWidth              Sepal Width in mm.    0.3709   43.04 <.0001 0.03688411 <.0001  0.55995394 <.0001
      3      3 PetalWidth              Petal Width in mm.    0.3229   34.57 <.0001 0.02497554 <.0001  0.59495691 <.0001
      4      4 SepalLength             Sepal Length in mm.   0.0615    4.72 0.0103 0.02343863 <.0001  0.59594941 <.0001
```

References

Costanza, M.C. and Afifi, A.A. (1979), “Comparison of Stopping Rules in Forward Stepwise Discriminant Analysis,” *Journal of the American Statistical Association*, 74, 777–785.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Jennrich, R.I. (1977), “Stepwise Discriminant Analysis,” in *Statistical Methods for Digital Computers*, eds. K. Enslein, A. Ralston, and H. Wilf, New York: John Wiley & Sons, Inc.

Journal of Statistics Education, “Fish Catch Data Set,” [http://www.stat.ncsu.edu/info/jse], accessed 4 December 1997

Klecka, W.R. (1980), *Discriminant Analysis*, Sage University Paper Series on Quantitative Applications in the Social Sciences, Series No. 07-019, Beverly Hills: Sage Publications.

Rencher, A.C. and Larson, S.F. (1980), “Bias in Wilks’s Λ in Stepwise Discriminant Analysis,” *Technometrics*, 22, 349–356.

Chapter 68
The SURVEYFREQ Procedure

Chapter Contents

Chapter 68
The SURVEYFREQ Procedure

Overview

The SURVEYFREQ procedure produces one-way to *n*-way frequency and crosstabulation tables from sample survey data. These tables include estimates of population totals and proportions, and the corresponding standard errors. PROC SURVEYFREQ computes the variance estimates based on the sample design used to obtain the survey data. The design can be a complex multistage survey design with stratification, clustering, and unequal weighting. PROC SURVEYFREQ also provides design-based tests of independence and association between variables.

PROC SURVEYFREQ uses the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method is appropriate for all designs where the first-stage sample is selected with replacement, or where the first-stage sampling fraction is small, as it often is in practice. The Taylor expansion method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971, Fuller 1975). When there are clusters or primary sampling units (PSUs) in the sample design, the procedure estimates variance from the variation among PSUs. When the design is stratified, the procedure combines stratum variance estimates to compute the overall variance estimate.

Getting Started

The following example shows how you can use PROC SURVEYFREQ to analyze sample survey data. The example uses data from a customer satisfaction survey for a student information system (SIS), a software product that provides modules for student registration, class scheduling, attendance, grade reporting, and other functions.

The software company conducted a survey of school personnel who use the SIS. A probability sample of SIS users was selected from the study population, which included SIS users at middle schools and high schools in the three-state area of Georgia, South Carolina, and North Carolina. The sample design for this survey was a two-stage stratified design. A first-stage sample of schools was selected from the list of schools using the SIS in the three-state area. The list of schools, or the first-stage sampling frame, was stratified by state and by customer status (whether the school was a new user of the system, or a renewal user). Within the first-stage strata, schools were selected with probability proportional to size and with replacement, where the size measure was school enrollment. From each sample school, five staff members were randomly selected to complete the SIS satisfaction questionnaire. These staff members included three teachers, and two administrators or guidance staff members.

The SAS data set SIS_Survey contains the survey results, as well as the sample design information needed to analyze the data. This data set includes an observation for each school staff member responding to the survey. The variable Response contains the staff member's response on overall satisfaction with the system.

The variable State contains the school's state, and the variable NewUser contains the school's customer status ('New Customer' or 'Renewal Customer'). These two variables determine the first stage strata from which schools were selected. The variable School contains the school identification code and identifies the first-stage sampling units, or clusters. The variable SamplingWeight contains the overall sampling weight for each respondent. Overall sampling weights were computed from the selection probabilities at each stage of sampling and were adjusted for nonresponse.

Other variables in the data set SIS_Survey include SchoolType and Department. The variable SchoolType identifies the school as a high school or a middle school. The variable Department identifies the staff member as a teacher, or an administrator or guidance department member.

The following PROC SURVEYFREQ statements request a one-way table for the variable Response.

```
title 'School Information System Survey';
proc surveyfreq data=SIS_Survey;
   tables  Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The PROC SURVEYFREQ statement invokes the procedure and identifies the input data set to be analyzed. The TABLES statement requests a one-way table for the variable Response. The table request syntax for PROC SURVEYFREQ is very similar to the PROC FREQ table request syntax. This example shows a request for a single one-way table, but you can also request two-way tables or multiway tables. As in PROC FREQ, you can request more than one table in the same TABLES statement, and you can use multiple TABLES statements in the same invocation of PROC SURVEYFREQ.

The STRATA, CLUSTER, and WEIGHT statements provide sample design information to the procedure, so that the analysis is done according to the sample design used for the survey, and the estimates apply to the study population. The STRATA statement names the variables State and NewUser, which identify the first-stage strata. Note that the design for this example also includes stratification at the second stage of selection (by type of school personnel), but you specify only the first-stage strata for PROC SURVEYFREQ. The CLUSTER statement identifies School as the cluster or first-stage sampling unit. The WEIGHT statement names the sampling weight variable.

Figure 68.1 and Figure 68.2 display the output produced by PROC SURVEYFREQ, which includes the Data Summary table and the one-way Table of Response. The Data Summary table is produced by default unless you specify the NOSUMMARY option. This table shows there are are 6 strata, 370 clusters or schools, and 1850 observations or respondents in the SIS_Survey data set. The sum of the sampling weights is approximately 39,000, which estimates the total number of school personnel using the SIS in the study area.

```
                     School Information System Survey

                         The SURVEYFREQ Procedure

                               Data Summary

                   Number of Strata                      6
                   Number of Clusters                  370
                   Number of Observations             1850
                   Sum of Weights               38899.6482
```

Figure 68.1. SIS_Survey Data Summary

Figure 68.2 displays the one-way table for Response, which provides estimates of the population total (weighted frequency) and the population percentage for each category, or level, of Response. The response level ’Very Unsatisfied’ has a frequency of 304, which means that 304 sample respondents fall into this category. It is estimated that 17.17% of all school personnel in the study population fall into this category, and the standard error of this estimate is 1.29%. Note that the estimates apply to the population of all SIS users in the study area, as opposed to describing only the sample of 1850 respondents. The estimate of the total number of school personnel ’Very Unsatisfied’ is 6,678, with a standard deviation of 502. The standard errors computed by PROC SURVEYFREQ are based on the multistage stratified design used for the survey. This differs from some of the traditional analysis procedures, which assume the design is simple random sampling from an infinite population.

```
                         School Information System Survey

                                 Table of Response

                                    Weighted   Std Dev of                Std Err of
        Response     Frequency     Frequency     Wgt Freq    Percent       Percent
-------------------------------------------------------------------------------
Very Unsatisfied           304          6678    501.61039    17.1676        1.2872
     Unsatisfied           326          6907    495.94101    17.7564        1.2712
         Neutral           581         12291    617.20147    31.5965        1.5795
       Satisfied           455          9309    572.27868    23.9311        1.4761
  Very Satisfied           184          3714    370.66577     9.5483        0.9523

           Total          1850         38900    129.85268    100.000
-------------------------------------------------------------------------------
```

Figure 68.2. One-Way Table of Response

The following PROC SURVEYFREQ statements request confidence limits for the percentage estimates and a chi-square goodness-of-fit test for the one-way table of Response.

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  Response / cl nowt chisq;
   Strata  State  NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The NOSUMMARY option in the PROC statement suppresses the Data Summary table. In the TABLES statement, the CL option requests confidence limits for the percentages in the one-way table. The NOWT option suppresses display of the weighted frequencies and their standard deviations. The CHISQ option requests a Rao-Scott chi-square goodness-of-fit test.

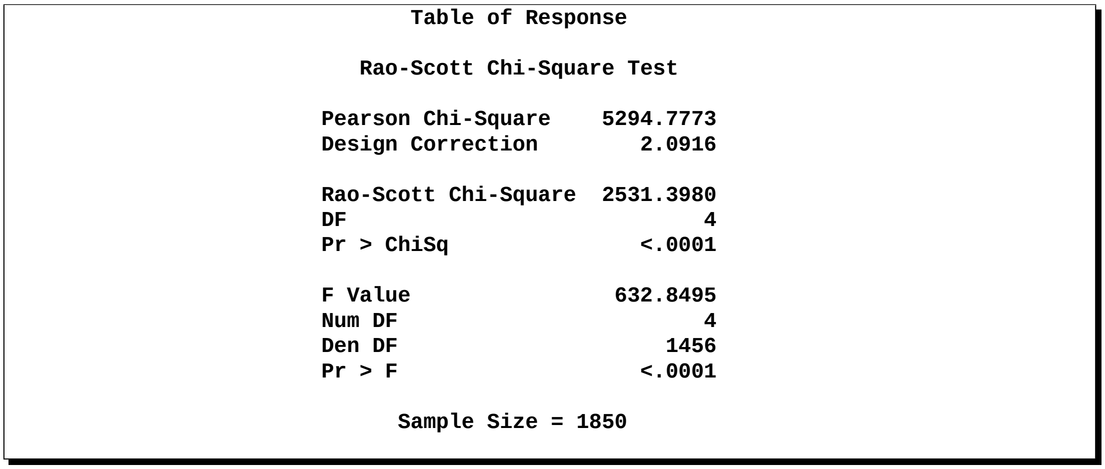
Figure 68.3 shows the one-way table of Response, which includes confidence limits for the percentages. The 95% confidence limits for the percentage of users that are 'Very Unsatisfied' are 14.64% and 19.70%. To change the α level of the confidence limits, which equals 5% by default, you can use the ALPHA= option. As for the other estimates and standard errors produced by PROC SURVEYFREQ, these confidence limit computations take into account the complex sample design used for the survey, and the results apply to the entire study population.

```
                         School Information System Survey

                             The SURVEYFREQ Procedure

                                 Table of Response

                                                 Std Err of    95% Confidence Limits
        Response      Frequency     Percent        Percent          for Percent
--------------------------------------------------------------------------------
Very Unsatisfied            304     17.1676         1.2872    14.6364        19.6989
     Unsatisfied            326     17.7564         1.2712    15.2566        20.2562
         Neutral            581     31.5965         1.5795    28.4904        34.7026
       Satisfied            455     23.9311         1.4761    21.0285        26.8338
  Very Satisfied            184      9.5483         0.9523     7.6756        11.4210

           Total           1850     100.000
--------------------------------------------------------------------------------
```

Figure 68.3. Confidence Limits for Response Percentages

Figure 68.4 shows the chi-square goodness-of-fit results for the table of Response. The null hypothesis for this test is equal proportions for the levels of the one-way table. (To test a null hypothesis of specified proportions instead of equal proportions, you can use the TESTP= option to specify null hypothesis proportions.)

The chi-square test invoked by the CHISQ option is the Rao-Scott design-adjusted chi-square test, which takes the survey design into account and provides inferences for the entire study population. To produce the Rao-Scott chi-square statistic, PROC SURVEYFREQ first computes the usual Pearson chi-square statistic based on the weighted frequencies, and then adjusts this value with a design correction. An F approximation is also provided. For the table of Response, the F value is 632.85 with a *p*-value $< .0001$, which leads to rejection of the null hypothesis of equal proportions for all response levels.



```
       Table of Response

   Rao-Scott Chi-Square Test

Pearson Chi-Square    5294.7773
Design Correction        2.0916

Rao-Scott Chi-Square  2531.3980
DF                            4
Pr > ChiSq               <.0001

F Value                632.8495
Num DF                        4
Den DF                     1456
Pr > F                   <.0001

      Sample Size = 1850
```

Figure 68.4. Chi-Square Goodness-of-Fit Test for Response

Continuing to analyze the SIS_Survey data, the following PROC SURVEYFREQ statements request a two-way table for the variables SchoolType by Response.

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  SchoolType * Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The STRATA, CLUSTER and WEIGHT statements do not change from the one-way table example, since the survey design and the input data set are the same. These SURVEYFREQ statements request a different table, but specify the same sample design information.

Figure 68.5 shows the two-way table produced. The first variable named in the two-way table request, SchoolType, is referred to as the *row variable*, and the second variable named, Response, is referred to as the *column variable*. Two-way tables display all column variable levels for each row variable level. So this two-way table lists all levels of the column variable Response for each level of the row variable SchoolType, ’Middle School’ and ’High School’. Also SchoolType = ’Total’ shows the distribution of Response overall for both types of schools. And Response = ’Total’ provides totals over all levels of response, for each type of school and overall. To suppress these totals, you can use the NOTOTAL option.

By default, without any other TABLES statement options, a two-way table displays the frequency, weighted frequency and its standard deviation, and percentage and its standard error for each table cell, or combination of row and column variable levels. But there are several options available to customize your table display by adding more information or suppressing some of the default information.

```
                               School Information System Survey

                                  The SURVEYFREQ Procedure

                                Table of SchoolType by Response

                                                          Weighted    Std Dev of                Std Err of
    SchoolType             Response       Frequency      Frequency      Wgt Freq     Percent       Percent
   ------------------------------------------------------------------------------------------------------
   Middle School    Very Unsatisfied            116           2496     351.43834      6.4155        0.9030
                         Unsatisfied            109           2389     321.97957      6.1427        0.8283
                             Neutral            234           4856     504.20553     12.4847        1.2953
                           Satisfied            197           4064     443.71188     10.4467        1.1417
                      Very Satisfied             94           1952     302.17144      5.0193        0.7758

                               Total            750          15758          1000     40.5089        2.5691
   ------------------------------------------------------------------------------------------------------
     High School    Very Unsatisfied            188           4183     431.30589     10.7521        1.1076
                         Unsatisfied            217           4518     446.31768     11.6137        1.1439
                             Neutral            347           7434     574.17175     19.1119        1.4726
                           Satisfied            258           5245     498.03221     13.4845        1.2823
                      Very Satisfied             90           1762     255.67158      4.5290        0.6579

                               Total           1100          23142          1003     59.4911        2.5691
   ------------------------------------------------------------------------------------------------------
           Total    Very Unsatisfied            304           6678     501.61039     17.1676        1.2872
                         Unsatisfied            326           6907     495.94101     17.7564        1.2712
                             Neutral            581          12291     617.20147     31.5965        1.5795
                           Satisfied            455           9309     572.27868     23.9311        1.4761
                      Very Satisfied            184           3714     370.66577      9.5483        0.9523

                               Total           1850          38900     129.85268     100.000
   ------------------------------------------------------------------------------------------------------
```

Figure 68.5. Two-Way Table of SchoolType by Response

The following PROC SURVEYFREQ statements request a two-way table of SchoolType by Response with row percentages, and also request a chi-square test for association between the two variables.

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  SchoolType * Response / row nowt chisq;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

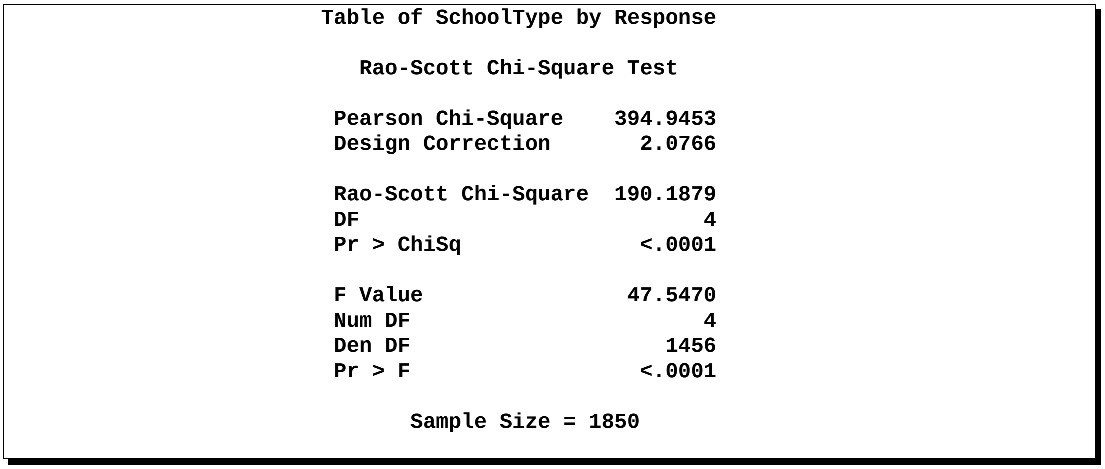
The ROW option in the TABLES statement requests row percentages, which display the distribution of Response as a percentage of each level of the row variable SchoolType. The NOWT option suppresses display of the weighted frequencies and their standard deviations. The CHISQ option requests a Rao-Scott chi-square test of association between SchoolType and Response.

Figure 68.6 displays the two-way table produced. For middle schools, it is estimated that 25.79% of school personnel are satisfied with the school information system, and 12.39% are very satisfied. For high schools, these estimates are 22.67% and 7.61%, respectively.

```
                           School Information System Survey

                              The SURVEYFREQ Procedure

                            Table of SchoolType by Response

                                                           Std Err of        Row    Std Err of
SchoolType            Response     Frequency     Percent      Percent    Percent   Row Percent
-------------------------------------------------------------------------------------------------
Middle School    Very Unsatisfied        116      6.4155       0.9030    15.8373        1.9920
                      Unsatisfied        109      6.1427       0.8283    15.1638        1.8140
                          Neutral        234     12.4847       1.2953    30.8196        2.5173
                        Satisfied        197     10.4467       1.1417    25.7886        2.2947
                   Very Satisfied         94      5.0193       0.7758    12.3907        1.7449

                            Total        750     40.5089       2.5691    100.000
-------------------------------------------------------------------------------------------------
  High School    Very Unsatisfied        188     10.7521       1.1076    18.0735        1.6881
                      Unsatisfied        217     11.6137       1.1439    19.5218        1.7280
                          Neutral        347     19.1119       1.4726    32.1255        2.0490
                        Satisfied        258     13.4845       1.2823    22.6663        1.9240
                   Very Satisfied         90      4.5290       0.6579     7.6128        1.0557

                            Total       1100     59.4911       2.5691    100.000
-------------------------------------------------------------------------------------------------
        Total    Very Unsatisfied        304     17.1676       1.2872
                      Unsatisfied        326     17.7564       1.2712
                          Neutral        581     31.5965       1.5795
                        Satisfied        455     23.9311       1.4761
                   Very Satisfied        184      9.5483       0.9523

                            Total       1850     100.000
-------------------------------------------------------------------------------------------------
```

Figure 68.6. Two-Way Table with Row Percentages

Figure 68.7 displays the chi-square test results. The Rao-Scott chi-square statistic equals 190.19, and the corresponding F value is 47.55 with a *p*-value $< .0001$. This indicates a significant association between school type (middle school or high school) and satisfaction with the student information system.



```
                         Table of SchoolType by Response

                            Rao-Scott Chi-Square Test

                          Pearson Chi-Square    394.9453
                          Design Correction       2.0766

                          Rao-Scott Chi-Square  190.1879
                          DF                           4
                          Pr > ChiSq              <.0001

                          F Value                47.5470
                          Num DF                       4
                          Den DF                    1456
                          Pr > F                  <.0001

                                Sample Size = 1850
```

Figure 68.7. Chi-Square Test of No Association

Syntax

The following statements are available in PROC SURVEYFREQ.

PROC SURVEYFREQ < *options* > **;**
  BY *variables* **;**
  CLUSTER *variables* **;**
  STRATA *variables* < **/** *option* > **;**
  TABLES *requests* < **/** *options* > **;**
  WEIGHT *variable* **;**

The PROC SURVEYFREQ statement invokes the procedure, identifies the data set to be analyzed, and provides sample design information. The PROC SURVEYFREQ statement is required.

The TABLES statement specifies frequency or crosstabulation tables and requests tests and statistics for those tables. The STRATA statement lists the variables that form the strata in a stratified sample design. The CLUSTER statement specifies cluster identification variables in a clustered sample design. The WEIGHT statement names the sampling weight variable. You can use a BY statement with PROC SURVEYFREQ to obtain separate analyses for groups defined by the BY variables.

All statements can appear multiple times except the PROC SURVEYFREQ statement and the WEIGHT statement, which can appear only once.

The rest of this section gives detailed syntax information for the BY, CLUSTER, STRATA, TABLES, and WEIGHT statements in alphabetical order after the description of the PROC SURVEYFREQ statement.

PROC SURVEYFREQ Statement

PROC SURVEYFREQ < *options* > **;**

The PROC SURVEYFREQ statement invokes the procedure. In this statement, you identify the data set to be analyzed and specify sample design information. The DATA= option names the input data set to be analyzed. If your analysis includes a finite population correction factor, you can input either the sampling rate or the population total using the RATE= or TOTAL= option. If your design is stratified, with different sampling rates or totals for different strata, then you can input these stratum rates or totals in a SAS data set containing the stratification variables.

You can specify the following options in the PROC SURVEYFREQ statement:

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC SURVEYFREQ. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MISSING

requests that the procedure treat missing values as a valid category for all categorical variables, which include TABLES variables, STRATA variables, and CLUSTER variables. For more information, see the section "Missing Values" on page 4205.

NOSUMMARY

suppresses the display of the Data Summary table, which PROC SURVEYFREQ produces by default. For a description of this table, see the section "Data and Sample Design Summary Table" on page 4225.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which the values of the frequency and crosstabulation table variables are to be reported. The following table shows how PROC SURVEYFREQ interprets values of the ORDER= option:

DATA	orders values according to their order in the input data set.
FORMATTED	orders values by their formatted values. This order is operating-environment dependent. By default, the order is ascending.
FREQ	orders values by descending frequency count. The frequency count of a variable value is its (nonweighted) frequency of occurrence or sample size, and not its weighted frequency.
INTERNAL	orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is operating-environment dependent.

By default, ORDER=INTERNAL.

PAGE

displays only one table per page. Otherwise, PROC SURVEYFREQ displays multiple tables per page as space permits.

RATE=*value* **|** *SAS-data-set*
R=*value* **|** *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or identifies an input data set that gives the stratum sampling rates in a variable named _RATE_. The procedure uses this information to compute a finite population correction for variance estimation. If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Population Totals and Sampling Rates" on page 4204 for more details.

The sampling rate *value* must be a nonnegative number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYFREQ will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

TOTAL=*value* **|** *SAS-data-set*
N=*value* **|** *SAS-data-set*

specifies the total number of primary sampling units (PSUs) in the study population as a positive *value*, or identifies an input data set that gives the stratum population totals in a variable named _TOTAL_. The procedure uses this information to compute a finite population correction for variance estimation.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Population Totals and Sampling Rates" on page 4204 for more details.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYFREQ to obtain separate analyses on observations in groups defined by the BY variables. The *variables* are one or more variables in the input data set.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, the difference being that in domain analysis the total number of units in the subpopulation is not known with certainty. You should include the domain variable(s) in your TABLES request to obtain domain analysis. See the section "Domain Analysis" on page 4205 for more details.

If you specify more than one BY statement, the procedure uses only the last BY statement and ignores any previous BY statements.

When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FREQ procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement names variables that identify the first-stage clusters, or PSUs, in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should specify only the first-stage clusters or primary sampling units (PSUs) in the CLUSTER statement. See the section "Specifying the Sample Design" on page 4203 for more information.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can either be character or numeric, but the procedure treats them as categorical variables. The formatted values of the CLUSTER variables determine the

CLUSTER variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can use multiple CLUSTER statements to specify CLUSTER variables. The procedure uses variables from all CLUSTER statements to create clusters.

STRATA Statement

STRATA *variables* < **/** *option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of categories of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Specifying the Sample Design" on page 4203 for more information.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical. The formatted values of the STRATA variables determine the STRATA variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following option in the STRATA statement after a slash (**/**):

LIST

displays a "Stratum Information" table, which lists all strata together with the corresponding values of the STRATA variables. This table provides the number of observations and number of clusters for each stratum, as well as the sampling fraction if you specify the RATE= or the TOTAL= option. See the section "Stratum Information Table" on page 4225 for more information.

TABLES Statement

TABLES *requests* < **/** *options* > **;**

The TABLES statement requests one-way to *n*-way frequency and crosstabulation tables and statistics for those tables.

If you omit the TABLES statement, PROC SURVEYFREQ generates one-way frequency tables for all data set variables that are not listed in the other statements.

The following argument is required in the TABLES statement.

requests

specify the frequency and crosstabulation tables to produce. A *request* is composed of one variable name or several variable names separated by asterisks. To request a one-way frequency table, use a single variable. To request a two-way crosstabulation table, use an asterisk between two variables. To request a multiway table (an *n*-way table, where *n*>2), separate the desired variables with asterisks. The unique values of these variables form the rows, columns, and layers of the table.

For two-way tables to multiway tables, the values of the last variable form the crosstabulation table columns, while the values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one layer. PROC SURVEYFREQ produces a separate crosstabulation table for each layer. For example, a specification of A*B*C*D in a TABLES statement produces *k* tables, where *k* is the number of different combinations of levels for A and B. Each table lists the levels for D (columns) within each level of C (rows).

You can use multiple TABLES statements in the PROC SURVEYFREQ step. You can also specify any number of table requests in a single TABLES statement. To specify multiple table requests quickly, use a grouping syntax by placing parentheses around several variables and joining other variables or variable combinations. For example, the following statements illustrate grouping syntax:

Table 68.1. Grouping Syntax

Request	**Equivalent to**
tables A*(B C);	tables A*B A*C;
tables (A B)*(C D);	tables A*C B*C A*D B*D;
tables (A B C)*D;	tables A*D B*D C*D;
tables A – – C;	tables A B C;
tables (A – – C)*D;	tables A*D B*D C*D;

The TABLES statement variables are one or more variables from the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical variables. PROC SURVEYFREQ uses the formatted values of the TABLES variable to determine the categorical variable levels. So if you assign a format to a variable with a FORMAT statement, PROC SURVEYFREQ formats the values before dividing observations into the levels of a frequency or crosstabulation table. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

The frequency or crosstabulation table lists the values of both character and numeric variables in ascending order based on internal (unformatted) variable values unless you change the order with the ORDER= option. To list the values in ascending order by formatted value, use ORDER=FORMATTED in the PROC SURVEYFREQ statement.

Without Options

If you request a frequency or crosstabulation table without specifying options, PROC SURVEYFREQ produces the following for each table level or cell:

- frequency (sample size)
- weighted frequency (estimated total)
- standard error of weighted frequency
- percentage (estimated proportion)
- standard error of percentage

The table displays weighted frequencies if your analysis includes a WEIGHT statement, or if you specify the WTFREQ option in the TABLES statement. The table also displays the number of observations with missing values. See the section "One-Way Frequency Tables" on page 4226 and the section "Crosstabulation Tables" on page 4227 for more information.

Options

The following table lists the options available with the TABLES statement. Descriptions follow in alphabetical order.

Table 68.2. TABLES Statement Options

Option	Description
Control Statistical Analysis	
ALPHA=	sets the level for confidence limits
CHISQ	requests Rao-Scott chi-square test
CHISQ1	requests Rao-Scott modified chi-square test
DDF=	specifies denominator DF for Wald chi-square tests
LRCHISQ	requests Rao-Scott likelihood ratio test
LRCHISQ1	requests Rao-Scott modified likelihood ratio test
TESTP=	specifies null proportions for one-way chi-square tests
WCHISQ	requests Wald chi-square test
WLLCHISQ	requests Wald log-linear chi-square test
Control Additional Table Information	
CL	displays confidence limits for percents
CLWT	displays confidence limits for weighted frequencies
COL	displays column percents and standard errors
CV	displays coefficients of variation for percents
CVWT	displays coefficients of variation for weighted frequencies
DEFF	displays design effects for percents
EXPECTED	displays expected weighted frequencies for two-way tables
ROW	displays row percents and standard errors
VAR	displays variances for percents
VARWT	displays variances for weighted frequencies
WTFREQ	displays weighted frequencies and standard errors when there is no WEIGHT statement
Control Displayed Output	
NOFREQ	suppresses display of frequency counts
NOPERCENT	suppresses display of percents
NOPRINT	suppresses display of tables but displays statistical tests
NOSPARSE	suppresses display of zero rows and columns in two-way tables
NOSTD	suppresses display of standard errors for all estimates
NOTOTAL	suppresses display of row and column totals
NOWT	suppresses display of weighted frequencies

You can specify the following options in a TABLES statement:

ALPHA=α

sets the level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default is 0.05. A confidence level of α produces $100(1-\alpha)$% confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

You request confidence limits for percentages with the CL option, and you request confidence limits for weighted frequencies with the CLWT option. See the section "Confidence Limits" on page 4213 for more information.

CHISQ

requests the Rao-Scott chi-square test. This test applies a design effect correction to the Pearson chi-square statistic computed from the weighted frequencies. See the section "Rao-Scott Chi-Square Test" on page 4216 for more information.

By default for one-way tables, the CHISQ option provides a design-based goodness-of-fit test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

The CHISQ option uses proportion estimates to compute the design effect correction. To use null hypothesis proportions instead, specify the CHISQ1 option.

CHISQ1

requests the Rao-Scott modified chi-square test. This test applies a design effect correction to the Pearson chi-square statistic computed from the weighted frequencies, and bases the design effect correction on null hypothesis proportions. See the section "Rao-Scott Chi-Square Test" on page 4216 for more information. To compute the design effect correction from proportion estimates instead of null proportions, specify the CHISQ option.

By default for one-way tables, the CHISQ option provides a design-based goodness-of-fit test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

CL

requests confidence limits for the percentages, or proportion estimates, in the crosstabulation table. PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. See the section "Confidence Limits" on page 4213 for more information.

CLWT

requests confidence limits for the weighted frequencies, or estimated totals, in the crosstabulation table. PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. See the section "Confidence Limits" on page 4213 for more information.

COL

displays the column percentage, or estimated proportion of the column total, for each cell in a two-way table. The COL option also displays the standard errors of the column percentages. See the section "Row and Column Proportions" on page 4212 for more information. This option has no effect for one-way tables.

CV

displays the coefficient of variation for each percentage, or proportion estimate, in the crosstabulation table. See the section "Coefficient of Variation" on page 4214 for more information.

CVWT

displays the coefficient of variation for each weighted frequency, or estimated total, in the crosstabulation table. See the section "Coefficient of Variation" on page 4214 for more information.

DDF=*df*

specifies the denominator degrees of freedom for the F-statistics used in the Wald chi-square tests. By default, the denominator degrees of freedom is the number of clusters minus the number of strata. See the section "Wald Chi-Square Test" on page 4221 and the section "Wald Log-Linear Chi-Square Test" on page 4223 for more information. You request the Wald chi-square test with the WCHISQ option, and you request the Wald log-linear chi-square test with the WLLCHISQ option.

DEFF

displays the design effect for each overall proportion estimate in the crosstabulation table. See the section "Design Effect" on page 4215 for more information.

EXPECTED

displays expected weighted frequencies for the table cells in a two-way table. The expected frequencies are computed under the null hypothesis that the row and column variables are independent. See the section "Expected Weighted Frequency" on page 4215 for more information. This option has no effect for one-way tables.

LRCHISQ

requests the Rao-Scott likelihood ratio chi-square test. This test applies a design effect correction to the likelihood ratio chi-square statistic computed from the weighted frequencies. See the section "Rao-Scott Likelihood Ratio Chi-Square Test" on page 4219 for more information.

By default for one-way tables, the LRCHISQ option provides a design-based test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

The LRCHISQ option uses proportion estimates to compute the design effect correction. To use null hypothesis proportions instead, specify the LRCHISQ1 option.

LRCHISQ1

requests the Rao-Scott modified likelihood ratio chi-square test. This test applies a design effect correction to the likelihood ratio chi-square statistic computed from the weighted frequencies, and bases the design effect correction on null hypothesis proportions. See the section "Rao-Scott Likelihood Ratio Chi-Square Test" on page

4219 for more information. To compute the design effect correction from proportion estimates instead of null proportions, specify the LRCHISQ option.

By default for one-way tables, the LRCHISQ option provides a design-based test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

NOFREQ

suppresses the display of cell frequencies in the crosstabulation table. The NOFREQ option also suppresses the display of row, column, and overall table frequencies.

NOPERCENT

suppresses the display of cell percentages in the crosstabulation table. The NOPERCENT option also suppresses the display of standard errors of the percentages.

NOPRINT

suppresses the display of frequency and crosstabulation tables but displays all requested statistical tests. Note that this option disables the Output Delivery System (ODS) for the suppressed tables. For more information, see Chapter 14, "Using the Output Delivery System."

NOSPARSE

suppresses the display of variable levels with zero frequency in two-way tables. By default, the procedure displays all levels of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. For multiway tables, the procedure displays all levels of the row variable for each layer of the table by default, including any row variable levels with zero frequency for the layer. Also by default, the procedure displays all variable levels that occur in the input data set, including those levels with no observations actually used in the analysis due to missing or nonpositive weights or missing values. See the section "Missing Values" on page 4205 for details.

NOSTD

suppresses the display of all standard errors in the crosstabulation table.

NOTOTAL

suppresses the display of row totals, column totals, and overall totals in the crosstabulation table.

NOWT

suppresses the display of weighted frequencies in the crosstabulation table. The NOWT option also suppresses the display of standard errors of the weighted frequencies.

ROW

displays the row percentage, or estimated proportion of the row total, for each cell in a two-way table. The ROW option also displays the standard errors of the row percentages. See the section "Row and Column Proportions" on page 4212 for more information. This option has no effect for one-way tables.

TESTP=(*values*)

specifies null hypothesis proportions, or test percentages, for one-way chi-square tests. You can separate *values* with blanks or commas. Specify *values* in probability form as numbers between 0 and 1, where the proportions sum to 1. Or specify *values* in percentage form as numbers between 0 and 100, where the percentages sum to 100. PROC SURVEYFREQ treats the value 1 as the percentage form 1%. The number of TESTP= values must equal the number of variable levels in the one-way table. List these values in the order in which the corresponding variable levels appear in the output.

When you specify the TESTP= option, PROC SURVEYFREQ displays the specified test percentages in the one-way frequency table. The TESTP= option has no effect for two-way tables.

PROC SURVEYFREQ uses the TESTP= values for all one-way chi-square tests you request in the TABLES statement. The available one-way chi-square tests include the Rao-Scott (Pearson) chi-square test and the Rao-Scott likelihood ratio chi-square test and their modified versions, requested by options CHISQ, CHISQ1, LRCHISQ, and LRCHISQ1. See the section "Rao-Scott Chi-Square Test" on page 4216 and the section "Rao-Scott Likelihood Ratio Chi-Square Test" on page 4219 for more details.

VAR

displays the variance estimate for each percentage in the crosstabulation table. See the section "Proportions" on page 4210 for details.

VARWT

displays the variance estimate for each weighted frequency, or estimated total, in the crosstabulation table. See the section "Totals" on page 4209 for details.

WCHISQ

requests the Wald chi-square test. See the section "Wald Chi-Square Test" on page 4221 for more information. By default, the denominator degrees of freedom for the Wald test *F*-statistic is the number of clusters minus the number of strata. Alternatively, you can specify the denominator degrees of freedom with the DDF= option.

WLLCHISQ

requests the Wald log-linear chi-square test. See the section "Wald Log-Linear Chi-Square Test" on page 4223 for more information. By default, the denominator degrees of freedom for the Wald test *F*-statistic is the number of clusters minus the number of strata. Alternatively, you can specify the denominator degrees of freedom with the DDF= option.

WTFREQ

displays the weighted frequencies and their standard errors when you do not specify a WEIGHT statement. PROC SURVEYFREQ displays the weighted frequencies by default when the analysis includes a WEIGHT statement. Without a WEIGHT statement, PROC SURVEYFREQ assigns all observations a weight of 1.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric. If you do not specify a WEIGHT statement, PROC SURVEYFREQ assigns all observations a weight of 1. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 4205 for more information. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

Details

Specifying the Sample Design

PROC SURVEYFREQ produces tables and statistics based on the sample design used to obtain the survey data. The procedure uses the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. See the section "Statistical Computations" on page 4206 for details. This method is appropriate for all designs where the first-stage sample is selected with replacement, or where the first-stage sampling fraction is small, as it often is in practice.

PROC SURVEYFREQ can be used for single-stage designs or for multistage designs, with or without stratification, and with or without unequal weighting. You provide sample design information with the STRATA, CLUSTER, and WEIGHT statements, and with the RATE= or TOTAL= option in the PROC SURVEYFREQ statement.

When there are clusters, or PSUs, in the sample design, the procedure estimates variance from the variance among PSUs. For a multistage sample design, the variance estimation method depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

Stratification

If your sample design is stratified at the first stage of sampling, use the STRATA statement to name variables that form the strata. The combinations of categories of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently. If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. If you do not specify a STRATA statement, PROC SURVEYFREQ assumes there is no stratification at the first stage.

Clustering

If your sample design selects clusters at the first stage of sampling, use the CLUSTER statement to name variables that identify the first-stage clusters, or primary sampling units (PSUs). The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should specify only the first-stage clusters, or PSUs, in the CLUSTER statement. PROC SURVEYFREQ assumes that each cluster defined by the CLUSTER statement variables represents a PSU in the sample, and that each observation belongs to one PSU. If you do not specify a CLUSTER statement, the procedure treats each observation as a PSU.

Weighting

If your sample design includes unequal weighting, use the WEIGHT statement to name the variable that contains the sampling weights. If you do not specify a WEIGHT statement, PROC SURVEYFREQ assigns all observations a weight of 1. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 4205 for more information.

Population Totals and Sampling Rates

If your analysis needs to include a finite population correction (*fpc*), you can input either the sampling rate or the population total using the RATE= option or the TOTAL= option in the PROC SURVEYFREQ statement. (You cannot specify both of these options in the same PROC SURVEYFREQ statement.) If you do not specify one of these options, the procedure does not use the *fpc* when computing variance estimates. For fairly small sampling fractions, it is appropriate to ignore this correction. Refer to Cochran (1977) and Kish (1965).

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you should use the RATE=*value* option or the TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in the strata, then you can use the RATE=*SAS-data-set* option or the TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. Furthermore, the BY groups must appear in the same order as in the primary data set. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Alternatively, if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYFREQ will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Domain Analysis

PROC SURVEYFREQ provides domain analysis through its multiway table capability. *Domain Analysis* refers to the computation of statistics for subpopulations, or domains, in addition to the computation of statistics for the entire study population. Formation of these domains may be unrelated to the sample design, so the domain sample sizes may actually be random variables. Domain analysis takes into account this variability, using the entire sample when estimating variance for domain estimates. This is also known as subgroup analysis, subpopulation analysis, or subdomain analysis. For more information on domain analysis, refer to Lohr (1999), Cochran (1977), and Fuller et al. (1989).

To request domain analysis with PROC SURVEYFREQ, you should include the domain variable(s) in your TABLES statement request. For example, specifying DOMAIN * A * B in a TABLES statement produces separate two-way tables of A by B for each level of DOMAIN. If your domains are formed by more than one variable, you can specify DomainVariable_1 * DomainVariable_2 * A * B, for example, to obtain two-way tables of A by B for each domain formed by the different combinations of levels for DomainVariable_1 and DomainVariable_2.

Including the domain variables in a TABLES statement request gives a different analysis from that obtained by using a BY statement, which provides completely separate analyses of the BY groups. The BY statement can also be used to analyze the dataset by subgroups, but it is critical to note that this will *not* produce a valid domain analysis. The BY statement is only appropriate when the number of units in each subgroup is known with certainty; when the subgroup sample size is a random variable, include the domain variables in your TABLES statement request.

Missing Values

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then PROC SURVEYFREQ excludes that observation from the analysis.

An observation is also excluded from the analysis if it has a missing value for any STRATA or CLUSTER variable, unless you specify the MISSING option in the PROC SURVEYFREQ statement. The MISSING option requests that the procedure treat missing values as a valid category for all categorical variables, which include strata variables, cluster variables, and classification or table variables.

Additionally, PROC SURVEYFREQ excludes an observation from a crosstabulation table (and any associated analyses) if that observation has a missing value for any of the table variables, unless you specify the MISSING option. When the procedure excludes observations with missing values from a table, it displays the total frequency of missing observations below that table. With the MISSING option, the procedure treats the missing values as a valid category and includes them in calculations of percentages and other statistics.

If all values in a stratum are excluded from the analysis of a table as missing values, then that stratum is called an *empty stratum*. Empty strata are not counted in the total number of strata for the table, which is used to determine the degrees of freedom for confidence limits and tests. Similarly, empty clusters and missing observations are not included in the total counts of clusters and observations used in the analysis of the table.

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS summary table that contains the number of (nonmissing) observations, strata, and clusters that are included in the analysis of the requested table. When there are missing observations, empty strata, or empty clusters for the requested table, then these numbers in the “Table Summary” differ from the total number of observations, strata, and clusters that are present in the input data set and reported in the “Data Summary.” See Example 68.3 on page 4236 for more information on the “Table Summary.”

If you have missing values in your survey data for any reason (such as nonresponse), this can compromise the quality of your survey results. An observation without missing values is called a *complete respondent*, and an observation with missing values is called an *incomplete respondent*. If the complete respondents are different from the incomplete respondents with regard to a survey effect or outcome, then survey estimates will be biased and will not accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. Once data collection is complete, you can use imputation to replace missing values with acceptable values, and you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYFREQ. Refer to Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more details.

Statistical Computations

The SURVEYFREQ procedure uses the Taylor series expansion method to estimate standard errors of estimators of proportions for crosstabulation tables. For sample survey data, the proportion estimator is a ratio estimator formed from estimators of totals. For example, to estimate the proportion in a crosstabulation table cell, the procedure uses the ratio of the estimator of the cell total frequency to the estimator of the overall population total, where these totals are linear statistics computed from the survey data. The Taylor series expansion method obtains a first-order linear approximation for the ratio estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971, Fuller 1975).

When there are clusters, or PSUs, in the sample design, the procedure estimates variance from the variance among PSUs. When the design is stratified, the procedure combines stratum variance estimates to compute the overall variance estimate. For a multistage sample design, the variance estimation method depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling. This variance estimation method assumes that the first-stage sampling fraction is small, or the first-stage sample is drawn with replacement, as it often is in practice.

In addition to this required sample design information, you also need to specify the sampling weights for a valid analysis, if the weights are not equal. Quite often in complex surveys, respondents have unequal weights, which reflect unequal selection probabilities and adjustments for nonresponse.

For more information on the analysis of sample survey data, refer to Lohr (1999), Särndal, Swenson, and Wretman (1992), Lee, Forthoffer, and Lorimor (1989), Cochran (1977), Kish (1965), and Hansen, Hurwitz, and Madow (1953).

Definitions and Notation

For a stratified clustered sample design, define the following:

$$
\begin{array}{rcl}
h & = & 1, 2, \ldots, H \quad \text{is the stratum number, with a total of } H \text{ strata} \\[6pt]
i & = & 1, 2, \ldots, n_h \quad \text{is the cluster number within stratum } h, \\
 & & \text{with a total of } n_h \text{ sample clusters from stratum } h \\[6pt]
j & = & 1, 2, \ldots, m_{hi} \quad \text{is the unit number within cluster } i \text{ of stratum } h, \\
 & & \text{with a total of } m_{hi} \text{ sample units from cluster } i \text{ of stratum } h \\[6pt]
n & = & \displaystyle\sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi} \quad \text{is the total number of observations in the sample} \\[6pt]
f_h & = & \text{first-stage sampling rate for stratum } h \\[6pt]
W_{hij} & = & \text{sampling weight of unit } j \text{ in cluster } i \text{ of stratum } h
\end{array}
$$

The sampling rate f_h is the fraction of first-stage units (PSUs) selected for the sample. You can specify the stratum sampling rates with the RATE= option. Or if you specify population totals with the TOTAL= option, PROC SURVEYFREQ computes f_h as the ratio of stratum sample size to the stratum total, in terms of PSUs. See the section "Population Totals and Sampling Rates" on page 4204 for details. If you do not specify the RATE= option or the TOTAL= option, then the procedure assumes that the stratum sampling rates f_h are negligible and does not use a finite population correction when computing variances.

This notation is also applicable to other sample designs. For example, for a design without stratification, you can let $H = 1$; for a sample design without clustering, you can let $m_{hi} = 1$ for every h and i, replacing clusters with individual sampling units.

For a two-way table representing the crosstabulation of two variables, define the following, where there are R levels of the row variable and C levels of the column variable:

$$
\begin{aligned}
r &= 1, 2, \ldots, R \quad \text{is the row number, with a total of } R \text{ rows} \\
c &= 1, 2, \ldots, C \quad \text{is the column number, with a total of } C \text{ columns} \\
N_{rc} &= \text{is the population total in row } r \text{ and column } c \\
N_{r\cdot} &= \sum_{c=1}^{C} N_{rc} \quad \text{is the total in row } r \\
N_{\cdot c} &= \sum_{r=1}^{R} N_{rc} \quad \text{is the total in column } c \\
N &= \sum_{r=1}^{R} \sum_{c=1}^{C} N_{rc} \quad \text{is the overall total} \\
P_{rc} &= N_{rc} \;/\; N \quad \text{is the population proportion in row } r \text{ and column } c \\
P_{r.} &= N_{r\cdot} \;/\; N \quad \text{is the proportion in row } r \\
P_{.c} &= N_{\cdot c} \;/\; N \quad \text{is the proportion in column } c \\
P_{rc}^{\,r} &= N_{rc} \;/\; N_{r\cdot} \quad \text{is the row proportion for cell } (r,c) \\
P_{rc}^{\,c} &= N_{rc} \;/\; N_{\cdot c} \quad \text{is the column proportion for cell } (r,c)
\end{aligned}
$$

For a specified observation (identified by stratum, cluster, and unit number within the cluster), define the following to indicate whether or not that observation belongs to cell (r,c), row r and column c, of the two-way table, for $r = 1, 2, \ldots, R$ and $c = 1, 2, \ldots, C$:

$$
\delta_{hij}(r,c) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in cell } (r,c) \\ 0 & \text{otherwise} \end{cases}
$$

Similarly, define the following functions to indicate the observation's row classification and the observation's column classification.

$$
\delta_{hij}(r) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in row } r \\ 0 & \text{otherwise} \end{cases}
$$

$$
\delta_{hij}(c) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in column } c \\ 0 & \text{otherwise} \end{cases}
$$

Totals

PROC SURVEYFREQ estimates population frequency totals for the specified crosstabulation tables, including totals for two-way table cells, rows, columns, and overall totals. The procedure computes the estimate of the total frequency in table cell (r, c) as the weighted frequency sum

$$\widehat{N}_{rc} \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c) \;\; W_{hij}$$

Similarly, PROC SURVEYFREQ computes estimates of the row totals, column totals, and overall totals as

$$\widehat{N}_{r\cdot} \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r) \;\; W_{hij}$$

$$\widehat{N}_{\cdot c} \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(c) \;\; W_{hij}$$

$$\widehat{N} \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \;\; W_{hij}$$

The estimators of totals are linear sample statistics, and so their variances can be estimated directly, without the Taylor series approximation that is used for proportions. PROC SURVEYFREQ estimates the variance of the total frequency in table cell (r, c) as

$$\widehat{\text{Var}}(\widehat{N}_{rc}) \quad = \quad \sum_{h=1}^{H} \widehat{\text{Var}}_h(\widehat{N}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{\text{Var}}_h(\widehat{N}_{rc}) \quad &= \quad \frac{n_h(1-f_h)}{n_h - 1} \;\; \sum_{i=1}^{n_h} (n_{rc}^{\;hi} - \bar{n}_{rc}^{\;h}\,)^2 \\
n_{rc}^{hi} \quad &= \quad \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c) \;\; W_{hij} \\
\bar{n}_{rc}^{\;h} \quad &= \quad \sum_{i=1}^{n_h} n_{rc}^{\;hi} \; / \; n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{\text{Var}}_h(\widehat{N}_{rc}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard deviation of the total is computed as

$$\text{Std}(\widehat{N}_{rc}) \quad = \quad \sqrt{\widehat{\text{Var}}(\widehat{N}_{rc})}$$

The variances and standard deviations are computed in a similar manner for row totals, column totals, and overall table totals.

Covariance of Totals

PROC SURVEYFREQ estimates the covariance between total frequency estimates for table cells (r, c) and (a, b) as

$$\widehat{\text{Cov}}(\widehat{N}_{rc},\ \widehat{N}_{ab}) \quad = \quad \sum_{h=1}^{H} \left(\frac{n_h(1-f_h)}{n_h - 1} \ \sum_{i=1}^{n_h} (n_{rc}^{\ hi} - \bar{n}_{rc}^{\ h}\ )\ (n_{ab}^{\ hi} - \bar{n}_{ab}^{\ h}\ ) \right)$$

The estimated covariance matrix of the table cell totals $\widehat{N}_{rc}$ is an $rc \times rc$ matrix $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$, which contains the pair-wise table cell covariances $\widehat{\text{Cov}}(\widehat{N}_{rc},\ \widehat{N}_{ab})$, for $r = 1, \cdots, R$; $c = 1, \cdots, C$; $a = 1, \cdots, R$; and $b = 1, \cdots, C$.

Proportions

PROC SURVEYFREQ computes the estimate of the proportion in table cell (r, c) as the ratio of the estimated total for the table cell to the estimated overall total,

$$\begin{aligned} \widehat{P}_{rc} \quad &= \quad \widehat{N}_{rc}\ /\ \widehat{N} \\ &= \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\ \ W_{hij}\ \ /\ \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}}\ W_{hij} \end{aligned}$$

Using the Taylor series expansion method, PROC SURVEYFREQ estimates the variance of this proportion estimate as

$$\widehat{\text{Var}}(\widehat{P}_{rc}) \quad = \quad \sum_{h=1}^{H} \widehat{\text{Var}}_h(\widehat{P}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned} \widehat{\text{Var}}_h(\widehat{P}_{rc}) \quad &= \quad \frac{n_h(1-f_h)}{n_h - 1} \ \sum_{i=1}^{n_h} (\ e_{rc}^{\ hi} - \bar{e}_{rc}^{\ h}\ )^2 \\ e_{rc}^{\ hi} \quad &= \quad \left(\sum_{j=1}^{m_{hi}} (\ \delta_{hij}(r,c) - \widehat{P}_{rc}\ )\ \ W_{hij} \right) \ /\ \widehat{N} \\ \bar{e}_{rc}^{\ h} \quad &= \quad \sum_{i=1}^{n_h} e_{rc}^{\ hi}\ /\ n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{\text{Var}}_h(\widehat{P}_{rc}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard error of the proportion is computed as

$$\text{StdErr}(\widehat{P}_{rc}) \quad = \quad \sqrt{\widehat{\text{Var}}(\widehat{P}_{rc})}$$

Similarly, the estimate of the proportion in row r is

$$\widehat{P}_{r\cdot} \quad = \quad \widehat{N}_{rc} \ / \ \widehat{N}$$

And its variance estimate is

$$\widehat{\text{Var}}(\widehat{P}_{r\cdot}) \quad = \quad \sum_{h=1}^{H} \widehat{\text{Var}}_h(\widehat{P}_{r\cdot})$$

where if $n_h > 1$,

$$\begin{aligned} \widehat{\text{Var}}_h(\widehat{P}_{r\cdot}) \quad &= \quad \frac{n_h(1-f_h)}{n_h - 1} \ \sum_{i=1}^{n_h} (\ e_{r\cdot}^{hi} - \bar{e}_{r\cdot}^{h}\ )^2 \\ e_{r\cdot}^{hi} \quad &= \quad \left(\sum_{j=1}^{m_{hi}} (\ \delta_{hij}(r) - \widehat{P}_{r\cdot}\ )\ \ W_{hij} \right) \ / \ \widehat{N} \\ \bar{e}_{r\cdot}^{h} \quad &= \quad \sum_{i=1}^{n_h} e_{r\cdot}^{hi} \ / \ n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{\text{Var}}_h(\widehat{P}_{r\cdot}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard error of the proportion in row r is computed as

$$\text{StdErr}(\widehat{P}_{r\cdot}) \quad = \quad \sqrt{\widehat{\text{Var}}(\widehat{P}_{r\cdot})}$$

Computations for the proportion in column c are done in the same way.

Row and Column Proportions

PROC SURVEYFREQ computes the estimate of the row proportion for table cell (r, c) as the ratio of the estimated total for the table cell to the estimated total for row r,

$$\begin{aligned} \widehat{P}^{\,r}_{rc} &= \widehat{N}_{rc} \;/\; \widehat{N}_{r\cdot} \\ &= \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c) \;\; W_{hij} \;\;/\;\; \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r) \;\; W_{hij} \end{aligned}$$

Again using the Taylor series expansion method, PROC SURVEYFREQ estimates the variance of this row proportion estimate as

$$\widehat{\text{Var}}(\widehat{P}^{\,r}_{rc}) \;=\; \sum_{h=1}^{H} \widehat{\text{Var}}_h(\widehat{P}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned} \widehat{\text{Var}}_h(\widehat{P}^{\,r}_{rc}) &= \frac{n_h(1-f_h)}{n_h - 1} \; \sum_{i=1}^{n_h} (\; g^{\,hi}_{rc} - \bar{g}^{\,h}_{rc} \;)^2 \\ g^{\,hi}_{rc} &= \left(\sum_{j=1}^{m_{hi}} (\; \delta_{hij}(r,c) - \widehat{P}^{\,r}_{rc} \;\; \delta_{hij}(r) \;) \;\; W_{hij} \right) \;\;/\;\; \widehat{N}_{r\cdot} \\ \bar{g}^{\,h}_{rc} &= \sum_{i=1}^{n_h} g^{\,hi}_{rc} \;/\; n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{\text{Var}}_h(\widehat{P}^{\,r}_{rc}) \;=\; \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard error of the row proportion is computed as

$$\text{StdErr}(\widehat{P}^{\,r}_{rc}) \;=\; \sqrt{\widehat{\text{Var}}(\widehat{P}^{\,r}_{rc})}$$

Similarly, PROC SURVEYFREQ estimates the column proportion for table cell (r, c) as the ratio of the estimated total for the table cell to the estimated total for column c,

$$\begin{aligned} \widehat{P}^{\,c}_{rc} &= \widehat{N}_{rc} \;/\; \widehat{N}_{\cdot c} \\ &= \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c) \;\; W_{hij} \;\;/\;\; \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(c) \;\; W_{hij} \end{aligned}$$

The variance estimate for the column proportion is computed as described above for the row proportion, but with

$$g_{rc}^{hi} \;=\; \left(\sum_{j=1}^{m_{hi}} \left(\, \delta_{hij}(r,c) - \widehat{P}_{rc}^{\,c} \;\, \delta_{hij}(c) \, \right) \; W_{hij} \right) \;\; / \;\; \widehat{N}_{\cdot c}$$

Confidence Limits

If you specify the CL option in the TABLES statement, PROC SURVEYFREQ computes confidence limits for the proportions in the frequency and crosstabulation tables. The confidence coefficient is determined according to the value of the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

For the proportion in table cell (r,c), the confidence limits are computed as

$$\widehat{P}_{rc} \;\; \pm \;\; t_{df,\alpha/2} \;\cdot\; \mathrm{StdErr}(\widehat{P}_{rc})$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r,c), StdErr$(\widehat{P}_{rc})$ is the standard error of the estimate, and $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the t distribution with df degrees of freedom calculated as described in the "Degrees of Freedom" section on page 4214. The confidence limits for row proportions and column proportions are computed similarly to the confidence limits for table cell proportions.

If you specify the CLWT option in the TABLES statement, PROC SURVEYFREQ computes confidence limits for the weighted frequencies, or totals, in the crosstabulation tables.

For the total in table cell (r,c), the confidence limits are computed as

$$\widehat{N}_{rc} \;\; \pm \;\; t_{df,\alpha/2} \;\cdot\; \mathrm{StdErr}(\widehat{N}_{rc})$$

where $\widehat{N}_{rc}$ is the estimate of the total frequency in table cell (r,c), StdErr$(\hat{N}_{rc})$ is the standard error of the estimate, and $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$ percentile of the t distribution with df degrees of freedom calculated as described in the "Degrees of Freedom" section on page 4214. The confidence limits for row totals, column totals, and the overall total are computed similarly to the confidence limits for table cell totals.

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS summary table that contains the number of (nonmissing) observations, strata, and clusters that are included in the analysis of the requested table. When you request confidence limits, this table also contains the degrees of freedom df and the value of $t_{df,\,\alpha/2}$ used to compute the confidence limits. See Example 68.3 on page 4236 for more information on the table summary.

Degrees of Freedom

To compute confidence limits for proportions and totals, PROC SURVEYFREQ uses the $100(1-\alpha/2)$ percentile from the t distribution with df degrees of freedom. PROC SURVEYFREQ calculates the degrees of freedom for t as the number of clusters minus the number of strata. If there are no clusters, then df equals the number of observations minus the number of strata. If the design is not stratified, then df equals the number of clusters minus one. If missing values or missing weights are present in the data, the number of strata, the number of observations, and the number of clusters are counted based on the observations in nonempty strata. See the section "Missing Values" on page 4205 for details.

For the Wald F statistics, PROC SURVEYFREQ also calculates the denominator degrees of freedom as the number of clusters minus the number of strata. Alternatively you can specify the denominator degrees of freedom for these tests with the DDF= option in the TABLES statement. See the section "Wald Chi-Square Test" on page 4221 and the section "Wald Log-Linear Chi-Square Test" on page 4223 for details.

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS summary table that contains the number of (nonmissing) observations, strata, and clusters that are included in the analysis of the requested table. When you request confidence limits or chi-square tests, this table also contains the degrees of freedom df. See Example 68.3 on page 4236 for more information on the table summary.

Coefficient of Variation

If you specify the CV option in the TABLES statement, PROC SURVEYFREQ computes the coefficients of variation for the proportion estimates in the frequency and crosstabulation tables. The coefficient of variation is the ratio of the standard error to the estimate.

For the proportion in table cell (r, c), the coefficient of variation is computed as

$$\mathrm{CV}(\widehat{P}_{rc}) \quad = \quad \mathrm{StdErr}(\widehat{P}_{rc}) \ / \ \widehat{P}_{rc}$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), and StdErr$(\widehat{P}_{rc})$ is the standard error of the estimate. The coefficients of variation for row proportions and column proportions are computed similarly.

If you specify the CVWT option in the TABLES statement, PROC SURVEYFREQ computes the coefficients of variation for the weighted frequencies, or estimated totals, in the crosstabulation tables. For the total in table cell (r, c), the coefficient of variation is computed as

$$\mathrm{CV}(\widehat{N}_{rc}) \quad = \quad \mathrm{StdErr}(\widehat{N}_{rc}) \ / \ \widehat{N}_{rc}$$

where $\widehat{N}_{rc}$ is the estimate of the total in table cell (r, c), and StdErr$(\widehat{N}_{rc})$ is the standard error of the estimate. The coefficients of variation for row totals, column totals, and the overall total are computed similarly.

Design Effect

If you specify the DEFF option in the TABLES statement, PROC SURVEYFREQ computes design effects for the overall proportion estimates in the frequency and crosstabulation tables. The design effect for an estimate is the ratio of the actual variance (estimated based on the sample design) to the variance of a simple random sample with the same number of observations. Refer to Lohr (1999) and Kish (1965).

The design effect for the proportion in table cell (r, c) is computed as

$$\begin{aligned} \text{DEFF}(\widehat{P}_{rc}) &= \widehat{\text{Var}}(\widehat{P}_{rc}) \;/\; \widehat{\text{Var}}_{\text{SRS}}(\widehat{P}_{rc}) \\ &= \widehat{\text{Var}}(\widehat{P}_{rc}) \;/\; \Big\{ (\,1 - f\,)\; \widehat{P}_{rc}\; (\,1 - \widehat{P}_{rc}\,) \,/\, (\,n - 1\,) \Big\} \end{aligned}$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), $\widehat{\text{Var}}(\widehat{P}_{rc})$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample.

PROC SURVEYFREQ determines the value of f, the overall sampling fraction, based on the RATE= and TOTAL= options. If you do not specify either of these options, then PROC SURVEYFREQ assumes the value of f is negligible and does not use a finite population correction in the analysis, as described in the section "Population Totals and Sampling Rates" on page 4204. If you specify RATE=*value*, then PROC SURVEYFREQ uses this value for the overall sampling fraction f. If you specify TOTAL=*value*, then PROC SURVEYFREQ computes f as the ratio of the number of PSUs in the sample to the specified total.

If you specify stratum sampling rates with the RATE=*SAS-data-set* option, then PROC SURVEYFREQ computes stratum totals based on these stratum sampling rates and the number of sample PSUs in each stratum. The procedure sums the stratum totals to form the overall total, and computes f as the ratio of the number of sample PSUs to the overall total. Alternatively, if you specify stratum totals with the TOTAL=*SAS-data-set* option, then PROC SURVEYFREQ sums these totals to compute the overall total. The overall sampling fraction f is then computed as the ratio of the number of sample PSUs to the overall total.

Expected Weighted Frequency

If you specify the EXPECTED option in the TABLES statement, PROC SURVEYFREQ displays expected weighted frequencies for the table cells in two-way tables. The expected weighted frequencies are computed under the null hypothesis that the row and column variables are independent, as

$$E_{rc} \;=\; \widehat{N}_{r\cdot}\;\; \widehat{N}_{\cdot c} \;/\; \widehat{N}$$

where $\widehat{N}_{r\cdot}$ is the estimated total for row r, $\widehat{N}_{\cdot c}$ is the estimated total for column c, and $\widehat{N}$ is the estimated overall total. Equivalently,

$$E_{rc} \;=\; \widehat{P}_{r\cdot}\;\; \widehat{P}_{\cdot c}\;\; \widehat{N}$$

These expected values are used in the design-based chi-square tests of independence, as described in "Rao-Scott Chi-Square Test" and the section "Wald Chi-Square Test" on page 4221.

Rao-Scott Chi-Square Test

The Rao-Scott chi-square test is a design-adjusted version of the Pearson chi-square test, which involves differences between observed and expected frequencies. For two-way tables, the null hypothesis for this test is no association between the row and column variables. For one-way tables, the Rao-Scott chi-square tests the null hypothesis of equal proportions, or you can specify null proportions for one-way tables with the TESTP= option.

Two forms of the design correction are available for the Rao-Scott tests. One form of the design correction uses the proportion estimates, and you request the corresponding Rao-Scott chi-square test with the CHISQ option. The other form of the design correction uses the null hypothesis proportions. You request this test, called the Rao-Scott modified chi-square test, with the CHISQ1 option.

Refer to Lohr (1999), Thomas, Singh, and Roberts (1996), and Rao and Scott (1981, 1984, 1987) for details on design-adjusted chi-square tests.

Two-Way Tables

The Rao-Scott chi-square statistic is computed from the Pearson chi-square statistic and a design correction based on the design effects of the proportions. Under the null hypothesis of no association between the row and column variables, this statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. An F approximation is also given.

The Rao-Scott chi-square is computed as

$$Q_{RS} \;=\; Q_P \;/\; D$$

where D is the design correction described in the "Design Correction for Two-Way Tables" section on page 4217, and Q_P is the Pearson chi-square based on the estimated totals.

$$Q_P \;=\; \sum_r \sum_c \,(\widehat{N}_{rc} - E_{rc})^2 \;/\; E_{rc}$$

where $\widehat{N}_{rc}$ is the estimated total for table cell (r, c), and E_{rc} is the expected total for cell (r, c) under the null hypothesis of no association,

$$E_{rc} \;=\; \widehat{N}_{r\cdot}\, \widehat{N}_{\cdot c} \;/\; \widehat{N}$$

Under the null hypothesis of no association, the Rao-Scott chi-square Q_{RS} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. A better approximation may be obtained by the F statistic

$$F \;=\; Q_{RS} \,/\, (R-1)(C-1)$$

which has an F distribution with $(R-1)(C-1)$ and $(R-1)(C-1)\kappa$ degrees of freedom under the null hypothesis, where κ equals the number of clusters minus the number of strata, as described in the section “Degrees of Freedom” on page 4214.

Design Correction for Two-Way Tables

If you specify the CHISQ option, the design correction is computed from the estimated proportions, as

$$\begin{aligned} D \quad = \quad & \Big\{ \textstyle\sum_r \sum_c \, (1-\widehat{P}_{rc}) \ \mathrm{DEFF}(\widehat{P}_{rc}) - \sum_r \, (1-\widehat{P}_{r\cdot}) \ \mathrm{DEFF}(\widehat{P}_{r\cdot}) \\ & \quad - \textstyle\sum_c \, (1-\widehat{P}_{\cdot c}) \ \mathrm{DEFF}(\widehat{P}_{\cdot c}) \Big\} \ / \ (R-1)(C-1) \end{aligned}$$

where

$$\begin{aligned} \mathrm{DEFF}(\widehat{P}_{rc}) \quad & = \quad \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \ / \ \mathrm{Var}_{\mathrm{SRS}}(\widehat{P}_{rc}) \\ & = \quad \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \ / \ \Big\{ (\, 1-f \,) \ \widehat{P}_{rc} \ (\, 1-\widehat{P}_{rc} \,) \ / \ (n-1) \Big\} \end{aligned}$$

as described in the section “Design Effect” on page 4215. $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r,c), $\widehat{\mathrm{Var}}(\widehat{P}_{rc})$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample. $\mathrm{DEFF}(\widehat{P}_{r\cdot})$, the design effect for the estimate of the proportion in row r, and $\mathrm{DEFF}(\widehat{P}_{\cdot c})$, the design effect for the estimate of the proportion in row c, are computed similarly.

If you specify the CHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis cell proportions, computed as the product of the corresponding estimated row and cell proportions.

$$\begin{aligned} D_0 \quad = \quad & \Big\{ \textstyle\sum_r \sum_c \, (1-P^0_{rc}) \ \mathrm{DEFF}_0(\widehat{P}_{rc}) - \sum_r \, (1-\widehat{P}_{r\cdot}) \ \mathrm{DEFF}(\widehat{P}_{r\cdot}) \\ & \quad - \textstyle\sum_c \, (1-\widehat{P}_{\cdot c}) \ \mathrm{DEFF}(\widehat{P}_{\cdot c}) \Big\} \ / \ (R-1)(C-1) \end{aligned}$$

where

$$P^0_{rc} \quad = \quad \widehat{P}_{r\cdot} \ \cdot \ \widehat{P}_{\cdot c}$$

and

$$\begin{aligned} \mathrm{DEFF}_0(\widehat{P}_{rc}) \quad & = \quad \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \ / \ \mathrm{Var}_{\mathrm{SRS}}(P^0_{rc}) \\ & = \quad \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \ / \ \{ (\, 1-f \,) \ P^0_{rc} \ (\, 1-P^0_{rc} \,) \ / \ (\, n-1 \,) \} \end{aligned}$$

One-Way Tables

For one-way tables, the Rao-Scott chi-square statistic provides a design-based goodness-of-fit test for equal proportions. Or if you specify null proportions with the TESTP= option, PROC SURVEYFREQ computes the goodness-of-fit test for the specified proportions. Under the null hypothesis, the Rao-Scott chi-square statistic approximately follows a chi-square distribution with $(C-1)$ degrees of freedom for a table with C levels. PROC SURVEYFREQ also computes an F statistic that may provide a better approximation.

The Rao-Scott chi-square is computed as

$$Q_{RS} \quad = \quad Q_P \,/\, D$$

where D is the design correction described in the section "Design Correction for One-Way Tables" on page 4219, and Q_P is the Pearson chi-square based on the estimated totals,

$$Q_P \quad = \quad \sum_c \, (\widehat{N}_c - E_c)^2 \;/\; E_c$$

where E_c is the expected total for level c under the null hypothesis. For the null hypothesis of equal proportions,

$$E_c \quad = \quad \widehat{N} \,/\, C$$

For specified null proportions,

$$E_c \quad = \quad \widehat{N} \;\cdot\; P_c^0$$

where P_c^0 is the null proportion for level c.

Under the null hypothesis, the one-way Rao-Scott chi-square Q_{RS} approximately follows a chi-square distribution with $(C-1)$ degrees of freedom. A better approximation may be obtained by the F statistic

$$F \quad = \quad Q_{RS} \,/\, (C-1)$$

which has an F distribution with $(C-1)$ and $(C-1)\kappa$ degrees of freedom under the null hypothesis, where κ equals the number of clusters minus the number of strata, as described in the section "Degrees of Freedom" on page 4214.

Design Correction for One-Way Tables

If you specify the CHISQ option, the design correction is computed from the estimated proportions, as

$$D \;=\; \sum_c (1 - \widehat{P}_c)\,\mathrm{DEFF}(\widehat{P}_c) \;/\; (C-1)$$

where

$$\mathrm{DEFF}(\widehat{P}_c) \;=\; \widehat{\mathrm{Var}}(\widehat{P}_c) \;/\; \left\{ (\,1-f\,)\,\widehat{P}_c\,(\,1-\widehat{P}_c\,) \,/\, (n-1) \right\}$$

$\widehat{P}_c$ is the proportion estimate for table level c, $\widehat{\mathrm{Var}}(\widehat{P}_c)$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample.

If you specify the CHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis proportions – either equal proportions for all levels, or the proportions you specify with the TESTP= option.

$$D_0 \;=\; \sum_c (1 - P_c^0)\,\mathrm{DEFF}_0(\widehat{P}_c) \;/\; (C-1)$$

where

$$\mathrm{DEFF}_0(\widehat{P}_c) \;=\; \widehat{\mathrm{Var}}(\widehat{P}_c) \;/\; \{ (\,1-f\,) P_c^0\,(\,1-P_c^0\,) \,/\, (\,n-1\,) \}$$

and $P_c^0 \;=\; 1/C$ for equal proportions, or P_c^0 takes the value specified with the TESTP= option.

Rao-Scott Likelihood Ratio Chi-Square Test

The Rao-Scott likelihood ratio chi-square test is a design-adjusted version of the likelihood ratio test, which involves ratios between observed and expected frequencies and tests the null hypothesis of no association between the row and column variables in a two-way table. For a one-way tables the null hypothesis is equal proportions for the table levels, or you can specify other null proportions with the TESTP= option. Refer to Lohr (1999), Thomas, Singh, and Roberts (1996), and Rao and Scott (1981, 1984, 1987).

Two forms of the design correction are available for the Rao-Scott tests. One form of the design correction uses the proportion estimates, and you request the corresponding Rao-Scott likelihood ratio test with the LRCHISQ option. The other form of the design correction uses the null hypothesis proportions. You request this test, called the Rao-Scott modified likelihood ratio test, with the LRCHISQ1 option.

Two-Way Tables

The Rao-Scott likelihood ratio statistic is computed from the likelihood ratio chi-square statistic and a design correction based on the design effects of the proportions. Under the null hypothesis of no association, this statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. An F approximation is also given.

The Rao-Scott likelihood ratio chi-square is computed as

$$G^2_{RS} \quad = \quad G^2 \;/\; D$$

where G^2 is the likelihood ratio chi-square based on the estimated totals, and D is the design correction.

$$G^2 \quad = \quad 2 \sum_r \sum_c \widehat{N}_{rc} \, \ln \left(\widehat{N}_{rc} \;/\; E_{rc} \right)$$

where $\widehat{N}_{rc}$ is the estimated total for table cell (r,c), and E_{rc} is the expected total for cell (r,c) under the null hypothesis of no association,

$$E_{rc} \quad = \quad \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \;\; / \;\; \widehat{N}$$

The Rao-Scott likelihood ratio chi-square uses the same design correction D as the Rao-Scott (Pearson) chi-square uses, which is described in the section "Design Correction for Two-Way Tables" on page 4217. If you specify the LRCHISQ option, the design correction is computed from the estimated proportions. If you specify the LRCHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis cell proportions, computed as the product of the corresponding estimated row and column proportions.

Under the null hypothesis of no association, the Rao-Scott likelihood ratio chi-square G^2_{RS} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. A better approximation may be obtained by the F statistic

$$F \quad = \quad G^2_{RS} \;/\; (R-1)(C-1)$$

which has an F distribution with $(R-1)(C-1)$ and $(R-1)(C-1)\kappa$ degrees of freedom under the null hypothesis, where κ equals the number of clusters minus the number of strata, as described in the section "Degrees of Freedom" on page 4214.

One-Way Tables

For one-way tables, the Rao-Scott likelihood ratio chi-square statistic provides a design-based goodness-of-fit test for equal proportions. Or if you specify null proportions with the TESTP= option, PROC SURVEYFREQ computes the goodness-of-fit test for the specified proportions. Under the null hypothesis, the Rao-Scott likelihood ratio statistic approximately follows a chi-square distribution with $(C-1)$ degrees of freedom for a table with C levels. Am F approximation is also given.

The Rao-Scott likelihood ratio chi-square is computed as

$$G^2_{RS} \quad = \quad G^2 \;/\; D$$

where G^2 is the likelihood ratio chi-square based on the estimated totals, and D is the design correction.

$$G^2 \quad = \quad 2 \sum_c \widehat{N}_c \; \ln \left(\widehat{N}_c \; / \; E_c \right)$$

where E_c is the expected total for level c under the null hypothesis. For the null hypothesis of equal proportions,

$$E_c \quad = \quad \widehat{N} \; / \; C$$

For specified null proportions P_c^0,

$$E_c \quad = \quad \widehat{N} \cdot P_c^0$$

The Rao-Scott likelihood ratio chi-square uses the same design correction D as the Rao-Scott (Pearson) chi-square uses, which is described in the section "Design Correction for One-Way Tables" on page 4219. If you specify the LRCHISQ option, the design correction is computed from the estimated proportions. If you specify the LRCHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis cell proportions.

Under the null hypothesis of no association, the Rao-Scott likelihood ratio chi-square G^2_{RS} approximately follows a chi-square distribution with $(C-1)$ degrees of freedom. A better approximation may be obtained by the F statistic

$$F \quad = \quad G^2_{RS} \; / \; (C-1)$$

which has an F distribution with $(C-1)$ and $(C-1)\kappa$ degrees of freedom under the null hypothesis, where κ equals the number of clusters minus the number of strata, as described in the section "Degrees of Freedom" on page 4214.

Wald Chi-Square Test

PROC SURVEYFREQ provides two Wald chi-square tests for independence of the row and column variables in two-way tables: a Wald chi-square test based on the difference between observed and expected weighted cell frequencies, and a Wald log-linear chi-square test based on the log odds ratio. These statistics test for independence of the row and column variables in two-way tables, taking into account the complex survey design. Refer to Bedrick (1983), Koch, Freeman, and Freeman (1975), and Wald (1943) for information on Wald statistics and their applications to categorical data analysis.

For these two tests, PROC SURVEYFREQ computes the generalized Wald chi-square statistic, the corresponding Wald F statistic, and also an adjusted Wald F statistic for tables larger than 2×2. Under the null hypothesis of independence, the Wald chi-square statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for very large samples. However, it has been shown that this test may perform poorly in terms of actual significance level and power, especially for tables with a large number of cells or for samples with a relatively small number of clusters. Refer to Thomas and Rao (1984 and 1985) and Lohr (1999) for more information. Refer to Felligi (1980) and Hidiroglou, Fuller, and Hickman (1980) for information on the adjusted Wald F statistic. Thomas and Rao (1984) found that the adjusted Wald F statistic provides a more stable test than the chi-square statistic, although its power may be low when the number of sample clusters is not large. Refer also to Korn and Graubard (1990) and Thomas, Singh, and Roberts (1996).

If you specify the WCHISQ option in the TABLES statement, PROC SURVEYFREQ computes a Wald test for independence in the two-way table based on the differences between the observed (weighted) cell frequencies and the expected frequencies.

Under the null hypothesis of independence of the row and column variables, the expected cell frequencies are computed as

$$E_{rc} \quad = \quad \widehat{N}_{r\cdot} \; \widehat{N}_{\cdot c} \; / \; \widehat{N}$$

where $\widehat{N}_{r\cdot}$ is the estimated total for row r, $\widehat{N}_{\cdot c}$ is the estimated total for column c, and $\widehat{N}$ is the estimated overall total, as described in the section "Expected Weighted Frequency" on page 4215. And the null hypothesis that the population weighted frequencies equal the expected frequencies is

$$H_0\text{:} \quad Y_{rc} \quad = \quad N_{rc} \; - \; E_{rc} \quad = \quad 0$$

for all $r = 1, \ldots (R-1)$ and $c = 1, \ldots (C-1)$ This null hypothesis can be stated equivalently in terms of cell proportions, with the expected cell proportions computed as the products of the marginal row and column proportions.

The generalized Wald chi-square statistic is computed as

$$Q_{\text{Wald}} \quad = \quad \widehat{\mathbf{Y}}' \;\; (\mathbf{H} \;\; \widehat{\mathbf{V}}(\widehat{\mathbf{N}}) \;\; \mathbf{H}')^{-1} \;\; \widehat{\mathbf{Y}}$$

where $\widehat{\mathbf{Y}}$ is the $(R-1)(C-1)$ array of the differences between the observed and expected weighted frequencies $(\widehat{N}_{rc} - E_{rc})$, and $(\mathbf{H} \; \widehat{\mathbf{V}}(\widehat{\mathbf{N}}) \; \mathbf{H}')$ estimates the variance of $\widehat{\mathbf{Y}}$.

$\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$ is the covariance matrix of the estimates $\widehat{N}_{rc}$, and its computation is described in the section "Covariance of Totals" on page 4210.

$\mathbf{H}$ is an $(R-1)(C-1)$ by RC matrix containing the partial derivatives of the elements of $\widehat{\mathbf{Y}}$ with respect to the elements of $\widehat{\mathbf{N}}$. The elements of $\mathbf{H}$ are computed as follows, where a denotes a row different from row r, and b denotes a column different from column c.

$$\partial \widehat{Y}_{rc} / \partial \widehat{N}_{rc} \quad = \quad 1 \; - \; \left(\widehat{N}_{r\cdot} \; + \; \widehat{N}_{\cdot c} \; - \; \widehat{N}_{\cdot c} \, \widehat{N}_{r\cdot} \; / \; \widehat{N} \right) \; / \; \widehat{N}$$

$$\partial \widehat{Y}_{rc} / \partial \widehat{N}_{ac} \quad = \quad - \; \left(\widehat{N}_{r\cdot} \; - \; \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \; / \; \widehat{N} \right) \; / \; \widehat{N}$$

$$\partial \widehat{Y}_{rc} / \partial \widehat{N}_{rb} \quad = \quad - \; \left(\widehat{N}_{\cdot c} \; - \; \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \; / \; \widehat{N} \right) \; / \; \widehat{N}$$

$$\partial \widehat{Y}_{rc} / \partial \widehat{Y}_{ab} \quad = \quad \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \; / \; \widehat{N}^{\,2}$$

Under the null hypothesis of independence, the statistic Q_{Wald} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for very large samples.

PROC SURVEYFREQ computes the Wald F statistic as

$$F_{\text{Wald}} \quad = \quad Q_{\text{Wald}} \;/\; (R-1)(C-1)$$

Under the null hypothesis of independence, F_{Wald} approximately follows an F distribution with $(R-1)(C-1)$ numerator degrees of freedom. By default, PROC SURVEYFREQ computes the denominator degrees of freedom as the number of clusters minus the number of strata, as described in the section "Degrees of Freedom" on page 4214. Alternatively, you can specify the denominator degrees of freedom with the DDF= option in the TABLES statement.

For tables larger than 2×2, PROC SURVEYFREQ also computes the adjusted Wald F statistic

$$F_{\text{Adj Wald}} \quad = \quad \frac{s-k+1}{k\; s}\; Q_{\text{Wald}}$$

where $k = (R-1)(C-1)$, and s is the number of clusters minus the number of strata. Alternatively, you can specify the value of s with the DDF= option in the TABLES statement. Note that for 2×2 tables, $k = (R-1)(C-1) = 1$, so the the adjusted Wald F statistic equals the (unadjusted) Wald F statistic, with the same numerator and denominator degrees of freedom.

Under the null hypothesis, $F_{\text{Adj Wald}}$ approximately follows an F distribution with k numerator degrees of freedom and $(s-k+1)$ denominator degrees of freedom.

Wald Log-Linear Chi-Square Test

If you specify the WLLCHISQ option in the TABLES statement, PROC SURVEYFREQ computes a Wald test for independence based on the log odds ratios. See the section "Wald Chi-Square Test" on page 4221 for more information on Wald tests.

For a two-way table of R rows and C columns, the Wald log-linear test is based on the $(R-1)(C-1)$ array of

$$\widehat{Y}_{rc} \quad = \quad log\widehat{N}_{rc} \;-\; log\widehat{N}_{rC} \;-\; log\widehat{N}_{Rc} \;+\; log\widehat{N}_{RC}$$

where $\widehat{N}_{rc}$ is the estimated total for table cell (r,c). The null hypothesis of independence between the row and column variables is H_0: $Y_{rc} = 0$ for all $r = 1, \ldots (R-1)$ and $c = 1, \ldots (C-1)$. This null hypothesis can be stated equivalently in terms of cell proportions.

The generalized Wald log-linear chi-square statistic is computed as

$$Q_{\text{Wald LL}} \;=\; \widehat{\mathbf{Y}}' \ \widehat{\mathbf{V}}(\widehat{\mathbf{Y}})^{-1} \ \widehat{\mathbf{Y}}$$

where $\widehat{\mathbf{Y}}$ is the $(R-1)(C-1)$ array of the $\widehat{Y}_{rc}$, and $\widehat{\mathbf{V}}(\widehat{\mathbf{Y}})$ estimates the variance of $\widehat{\mathbf{Y}}$.

$$\widehat{\mathbf{V}}(\widehat{\mathbf{Y}}) \;=\; \mathbf{A} \ \mathbf{D}^{-1} \ \widehat{V}(\widehat{\mathbf{N}}) \ \mathbf{D}^{-1} \ \mathbf{A}'$$

where $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$ is the covariance matrix of the estimates $\widehat{N}_{rc}$, as described in the section "Covariance of Totals" on page 4210, $\mathbf{D}$ is a diagonal matrix with the estimated totals $\widehat{N}_{rc}$ on the diagonal, and $\mathbf{A}$ is the $(R-1)(C-1)$ by $RC \times RC$ linear contrast matrix.

Under the null hypothesis of independence, the statistic $Q_{\text{Wald LL}}$ approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for very large samples.

PROC SURVEYFREQ computes the Wald log-linear F statistic as

$$F_{\text{Wald LL}} \;=\; Q_{\text{Wald LL}} \,/\, (R-1)(C-1)$$

Under the null hypothesis of independence, $F_{\text{Wald LL}}$ approximately follows an F distribution with $(R-1)(C-1)$ numerator degrees of freedom. By default, PROC SURVEYFREQ computes the denominator degrees of freedom as the number of clusters minus the number of strata, as described in the section "Degrees of Freedom" on page 4214. Alternatively, you can specify the denominator degrees of freedom with the DDF= option in the TABLES statement.

For tables larger than 2 × 2, PROC SURVEYFREQ also computes the adjusted Wald log-linear F statistic

$$F_{\text{Adj Wald LL}} \;=\; \frac{s-k+1}{k \ s} \ Q_{\text{Wald LL}}$$

where $k = (R-1)(C-1)$, and s is the number of clusters minus the number of strata. Alternatively, you can specify the value of s with the DDF= option in the TABLES statement. Note that for 2 × 2 tables, $k = (R-1)(C-1) = 1$, so the the adjusted Wald F statistic equals the (unadjusted) Wald F statistic, with the same numerator and denominator degrees of freedom.

Under the null hypothesis, $F_{\text{Adj Wald LL}}$ approximately follows an F distribution with k numerator degrees of freedom and $(s-k+1)$ denominator degrees of freedom.

Displayed Output

Data and Sample Design Summary Table

The “Data Summary” table provides information on the input data set and the sample design. PROC SURVEYFREQ displays this table unless you specify the NOSUMMARY option in the PROC SURVEYFREQ statement.

The “Data Summary” table displays the total number of valid observations. To be considered *valid*, an observation must have a nonmissing, positive WEIGHT value if you specify a WEIGHT statement. If you do not specify the MISSING option, a valid observation must also have nonmissing values for all STRATA and CLUSTER variables. The number of valid observations may differ from the the number of nonmissing observations for an individual analysis variable, which the procedure displays in the frequency or crosstabulation tables. See the section “Missing Values” on page 4205 for more information.

PROC SURVEYFREQ displays the following information in the “Data Summary” table:

- Number of Strata, if you specify a STRATA statement
- Number of Clusters, if you specify a CLUSTER statement
- Number of Observations, which is the total number of valid observations
- Sum of Weights, which is the sum over all valid observations, if you specify a WEIGHT statement

Stratum Information Table

If you specify the LIST option in the STRATA statement, PROC SURVEYFREQ displays a “Stratum Information” table. This table provides the following information for each stratum.

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Number of Observations, which is the number of valid observations in the stratum
- Population Total for the stratum, if you specify the TOTAL= option
- Sampling Rate for the stratum, if you specify the TOTAL= option or the RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of valid observations in the stratum.
- Number of Clusters, which is the number of clusters in the stratum, if you specify a CLUSTER statement

One-Way Frequency Tables

PROC SURVEYFREQ displays one-way frequency tables for all one-way table requests in the TABLES statements, unless you specify the NOPRINT option in the TABLES statement. A one-way table shows the sample frequency distribution of a single variable, and provides estimates for its population distribution in terms of totals and proportions. For each level of the variable, PROC SURVEYFREQ displays the following information in the one-way table:

- Frequency count, giving the number of sample observations for the level
- Weighted Frequency total, estimating the total population frequency for the level
- Standard Deviation of Weighted Frequency
- Percent, estimating the population proportion for the level
- Standard Error of Percent

The one-way table displays weighted frequencies if your analysis includes a WEIGHT statement, or if you specify the WTFREQ option in the TABLES statement.

The one-way table also displays the Frequency Missing, or the number of observations with missing values.

You can suppress the frequency counts by specifying the NOFREQ option in the TABLES statement. Also, the NOWT option suppresses the weighted frequencies and their standard deviations. The NOPERCENT option suppresses the percentages and their standard errors. The NOSTD option suppresses the standard errors of the percentages and the standard deviations of the weighted frequencies. The NOTOTAL option suppresses the total row of the one-way table.

PROC SURVEYFREQ optionally displays the following information for a one-way table:

- Variance of Weighted Frequency, if you specify the VARWT option
- Confidence Limits for Weighted Frequency, if you specify the CLWT option
- Coefficient of Variation for Weighted Frequency, if you specify the CVWT option
- Test Percent, if you specify the TESTP= option
- Variance of Percent, if you specify the VAR option
- Confidence Limits for Percent, if you specify the CL option
- Coefficient of Variation for Percent, if you specify the CV option
- Design Effect for Percent, if you specify the DEFF option

Crosstabulation Tables

PROC SURVEYFREQ displays all multiway table requests in the TABLES statements, unless you specify the NOPRINT option in the TABLES statement. For two-way to multiway crosstabulation tables, the values of the last variable in the table request form the table columns. The values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables form one layer. PROC SURVEYFREQ produces a separate two-way crosstabulation table for each layer.

For each layer, the crosstabulation table displays the row and column variable names and values (or levels). Each two-way table lists levels of the column variable within each level of the row variable.

By default, the procedure displays all levels of the column variable within each level of the row variables, including any column variable levels with zero frequency for that row. For multiway tables, the procedure displays all levels of the row variable for each layer of the table by default, including any row levels with zero frequency for that layer. You can suppress the display of zero frequency levels by specifying the NOSPARSE option.

For each combination of variable levels, or table cell, the two-way table displays the following information:

- Frequency, giving the number of observations that have the indicated values of the two variables
- Weighted Frequency total, estimating the total population frequency for the table cell
- Standard Deviation of Weighted Frequency
- Percent, estimating the population proportion for the table cell
- Standard Error of Percent

The two-way table displays weighted frequencies if your analysis includes a WEIGHT statement, or if you specify the WTFREQ option in the TABLES statement.

The two-way table also displays the Frequency Missing, or the number of observations with missing values.

You can suppress the frequency counts by specifying the NOFREQ option in the TABLES statement. Also, the NOWT option suppresses the weighted frequencies and their standard deviations. The NOPERCENT option suppresses the percentages and their standard errors. The NOSTD option suppresses the standard errors of the percentages and the standard deviations of the weighted frequencies. The NOTOTAL option suppresses the row totals and column totals.

PROC SURVEYFREQ optionally displays the following information for a two-way table:

- Expected Weighted Frequency, if you specify the EXPECTED option
- Variance of Weighted Frequency, if you specify the VARWT option
- Confidence Limits for Weighted Frequency, if you specify the CLWT option
- Coefficient of Variation for Weighted Frequency, if you specify the CVWT option
- Variance of Percent, if you specify the VAR option
- Confidence Limits for Percent, if you specify the CL option
- Coefficient of Variation for Percent, if you specify the CV option
- Design Effect for Percent, if you specify the DEFF option
- Row Percent, estimating the cell's proportion of the population total for that cell's row, if you specify the ROW option
- Standard Error of Row Percent, if you specify the ROW option
- Variance of Row Percent, if you specify the VAR option and the ROW option
- Confidence Limits for Row Percent, if you specify the CL option and the ROW option
- Coefficient of Variation for Row Percent, if you specify the CV option and the ROW option
- Column Percent, estimating the cell's proportion of the population total for that cell's column, if you specify the COL option
- Standard Error of Column Percent, if you specify the COL option
- Variance of Column Percent, if you specify the VAR option and the COL option
- Confidence Limits for Column Percent, if you specify the CL option and the COL option
- Coefficient of Variation for Column Percent, if you specify the CV option and the COL option

If you specify the ROW option, the NOPERCENT option suppresses the row percentages and their standard errors. The NOSTD option suppresses the standard errors of the row percentages. Similarly, if you specify the COL option, the NOPERCENT option suppresses the column percentages and their standard errors. The NOSTD option suppresses the standard errors of the column percentages.

Statistical Tests

If you specify the CHISQ option for the Rao-Scott chi-square test, the CHISQ1 option for the modified test, the LRCHISQ option for the Rao-Scott likelihood ratio chi-square test, or the LRCHISQ1 option for the modified test, PROC SURVEYFREQ displays the following information:

- Pearson Chi-Square, if you specify the CHISQ or CHISQ1 option
- Likelihood Ratio Chi-Square, if you specify the LRCHISQ or LRCHISQ1 option
- Design Correction
- Rao-Scott Chi-Square, if you specify the CHISQ or CHISQ1 option
- Rao-Scott Likelihood Ratio Chi-Square, if you specify the LRCHISQ or LRCHISQ1 option
- DF, the degrees of freedom for the chi-square test
- Pr > ChiSq, the *p*-value for the chi-square test
- F Value
- Num DF, the numerator degrees of freedom for F
- Den DF, the denominator degrees of freedom for F
- Pr > F, the *p*-value for the F test

If you specify the WCHISQ option for the Wald chi-square test or the WLLCHISQ option for the Wald log-linear chi-square test, PROC SURVEYFREQ displays the following information:

- Wald Chi-Square, if you specify the WCHISQ option
- Wald Log-Linear Chi-Square, if you specify the WLLCHISQ option
- F Value
- Num DF, the numerator degrees of freedom for F
- Den DF, the denominator degrees of freedom for F
- Pr > F, the *p*-value for the F test
- Adjusted F Value, for tables larger than 2×2
- Num DF, the numerator degrees of freedom for Adjusted F
- Den DF, the denominator degrees of freedom for Adjusted F
- Pr > Adj F, the *p*-value for the Adjusted F test

ODS Table Names

PROC SURVEYFREQ assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 68.3. ODS Tables Produced in PROC SURVEYFREQ

ODS Table Name	Description	Statement	Option
ChiSq	Chi-square test	TABLES	CHISQ
ChiSq1	Modified chi-square test	TABLES	CHISQ1
CrossTabs	Crosstabulation table	TABLES	(n-way table request, $n > 1$)
LRChiSq	Likelihood ratio test	TABLES	LRCHISQ
LRChiSq1	Modified likelihood ratio test	TABLES	LRCHISQ1
OneWay	One-way frequency table	PROC	(with no TABLES stmt)
		or TABLES	(one-way table request)
StrataInfo	Stratum information	STRATA	LIST
Summary	Data summary	PROC	default
TableSummary	Table summary (not displayed)	TABLES	default
WChiSq	Wald chi-square test	TABLES	WCHISQ
WLLChiSq	Wald log-linear chi-square test	TABLES	WLLCHISQ

Examples

Example 68.1. Two-Way Tables

This example uses the SIS_Survey data set from the section "Getting Started" on page 4185. The data set contains results from a customer satisfaction survey for a student information system (SIS).

The following PROC SURVEYFREQ statements request a two-way table for the variables Department by Response and customize the crosstabulation table display.

```
proc surveyfreq data=SIS_Survey;
   tables  Department * Response /
           cv deff nowt nostd nototal;
   strata  State NewUser / list;
   cluster School;
   weight  SamplingWeight;
run;
```

The TABLES statement requests a two-way table of Department by Response. The CV option requests coefficients of variation for the percentage estimates. The DEFF option requests design effects for the percentage estimates. The NOWT option suppresses display of the weighted frequencies, and the NOSTD option suppresses display of standard errors for the estimates. The NOTOTAL option suppresses the row totals, column totals, and overall totals.

The WCHISQ option requests a Wald chi-square test of association between the variables Department and Response.

The STRATA, CLUSTER, and WEIGHT statements provide sample design information to the procedure, so that the analysis will be done according to the sample design used for the survey. The STRATA statement names the variables State and NewUser, which identify the first-stage strata. The LIST option in the STRATA statement requests a Stratum Information table. The CLUSTER statement identifies School as the cluster or first-stage sampling unit. The WEIGHT statement names the sampling weight variable.

Output 68.1.1 displays the Data Summary and Stratum Information tables produced by PROC SURVEYFREQ. The Stratum Information table lists the six strata in the survey and shows the number of clusters, or schools, and the number of observations in each stratum.

Output 68.1.1. Data Summary and Stratum Information

```
               School Information System Survey

                   The SURVEYFREQ Procedure

                         Data Summary

             Number of Strata                    6
             Number of Clusters                370
             Number of Observations           1850
             Sum of Weights             38899.6482


                     Stratum Information

Stratum                                 Number of    Number of
 Index     State    NewUser                   Obs     Clusters
--------------------------------------------------------------
   1       GA       Renewal Customer          315           63
   2       GA       New Customer              355           71
   3       NC       Renewal Customer          280           56
   4       NC       New Customer              420           84
   5       SC       Renewal Customer          210           42
   6       SC       New Customer              270           54
--------------------------------------------------------------
```

Output 68.1.2 displays the two-way table of Department by Response. According to the TABLES statement options specified, this two-way table includes coefficients of variation and design effects for the percentage estimates, and it does not show the weighted frequencies or the standard errors of the estimates.

Output 68.1.2. Two-Way Table of Department by Response

```
                        School Information System Survey

                            The SURVEYFREQ Procedure

                        Table of Department by Response

                                                                CV for      Design
   Department           Response     Frequency    Percent      Percent      Effect
--------------------------------------------------------------------------------
      Faculty   Very Unsatisfied           209    13.4987       0.0865      2.1586
                     Unsatisfied           203    13.0710       0.0868      2.0962
                         Neutral           346    22.4127       0.0629      2.1157
                       Satisfied           254    16.2006       0.0806      2.3232
                  Very Satisfied            98     6.2467       0.1362      2.2842
--------------------------------------------------------------------------------
Admin/Guidance   Very Unsatisfied            95     3.6690       0.1277      1.1477
                     Unsatisfied           123     4.6854       0.1060      1.0211
                         Neutral           235     9.1838       0.0700      0.9166
                       Satisfied           201     7.7305       0.0756      0.8848
                  Very Satisfied            86     3.3016       0.1252      0.9892
--------------------------------------------------------------------------------
```

The following PROC SURVEYFREQ statements request a two-way table of Department by Response that includes row percentages, and also a Wald chi-square test of association between the two table variables.

```
 proc surveyfreq data=SIS_Survey nosummary;
    tables Department * Response /
           row nowt wchisq;
    strata State NewUser;
    cluster School;
    weight SamplingWeight;
run;
```

Output 68.1.3 displays the two-way table. The row percentages show the distribution of Response for Department = ’Faculty’ and for Department = ’Admin/Guidance’.

Output 68.1.3. Table of Department by Response with Row Percentages

```
                              School Information System Survey

                                 The SURVEYFREQ Procedure

                              Table of Department by Response

                                                                  Std Err of         Row    Std Err of
   Department             Response     Frequency     Percent       Percent     Percent   Row Percent
   ---------------------------------------------------------------------------------------------------
      Faculty    Very Unsatisfied           209     13.4987        1.1675     18.8979        1.6326
                      Unsatisfied           203     13.0710        1.1350     18.2992        1.5897
                          Neutral           346     22.4127        1.4106     31.3773        1.9705
                        Satisfied           254     16.2006        1.3061     22.6805        1.8287
                   Very Satisfied            98      6.2467        0.8506      8.7452        1.1918

                            Total          1110     71.4297        0.1468     100.000
   ---------------------------------------------------------------------------------------------------
   Admin/Guidance  Very Unsatisfied          95      3.6690        0.4684     12.8419        1.6374
                      Unsatisfied           123      4.6854        0.4966     16.3995        1.7446
                          Neutral           235      9.1838        0.6430     32.1447        2.2300
                        Satisfied           201      7.7305        0.5842     27.0579        2.0406
                   Very Satisfied            86      3.3016        0.4133     11.5560        1.4466

                            Total           740     28.5703        0.1468     100.000
   ---------------------------------------------------------------------------------------------------
        Total    Very Unsatisfied           304     17.1676        1.2872
                      Unsatisfied           326     17.7564        1.2712
                          Neutral           581     31.5965        1.5795
                        Satisfied           455     23.9311        1.4761
                   Very Satisfied           184      9.5483        0.9523

                            Total          1850     100.000
   ---------------------------------------------------------------------------------------------------
```

Output 68.1.4 displays the Wald chi-square test for association between Department and Response. The Wald chi-square is 11.44, and the corresponding adjusted F value is 2.84 with a p-value of .0243. This indicates a significant association between department (faculty or admin/guidance) and satisfaction with the student information system.

Output 68.1.4. Wald Chi-Square Test

```
                    Table of Department by Response

                         Wald Chi-Square Test

                        Chi-Square      11.4454

                        F Value          2.8613
                        Num DF                4
                        Den DF              364
                        Pr > F           0.0234

                        Adj F Value      2.8378
                        Num DF                4
                        Den DF              361
                        Pr > Adj F       0.0243

                          Sample Size = 1850
```

Example 68.2. Multiway Tables

Continuing to use the SIS_Survey data set from the section "Getting Started" on page 4185, this example shows how to produce multiway tables. The following PROC SURVEYFREQ statements request a table of Department by SchoolType by Response for the student information system survey.

```
proc surveyfreq data=SIS_Survey;
   tables  Department * SchoolType * Response
           SchoolType * Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The TABLES statement requests a multiway table with SchoolType as the row variable, Response as the column variable, and Department as the layer variable. This request produces a separate two-way table of SchoolType by Response for each level of the variable Department. The TABLES statement also requests a two-way table of SchoolType by Response, which totals the multiway table over both levels of Department. As in the previous examples, the STRATA, CLUSTER, and WEIGHT statements provide sample design information, so that the analysis will be done according to the design used for this survey.

Output 68.2.1 displays the multiway table produced by PROC SURVEYFREQ, which includes a table of SchoolType by Response for Department = 'Faculty' and for Department = 'Admin/Guidance'.

Output 68.2.1. Multiway Table of Department by SchoolType by Response

```
                                  School Information System Survey

                                      The SURVEYFREQ Procedure

                                   Table of SchoolType by Response
                                 Controlling for Department=Faculty

                                                     Weighted    Std Dev of               Std Err of
   SchoolType            Response     Frequency     Frequency      Wgt Freq    Percent       Percent
----------------------------------------------------------------------------------------------------
Middle School    Very Unsatisfied            74          1846     301.22637     6.6443        1.0838
                      Unsatisfied            78          1929     283.11476     6.9428        1.0201
                          Neutral           130          3289     407.80855    11.8369        1.4652
                        Satisfied           113          2795     368.85087    10.0597        1.3288
                   Very Satisfied            55          1378     261.63311     4.9578        0.9411

                            Total           450         11237     714.97120    40.4415        2.5713
----------------------------------------------------------------------------------------------------
  High School    Very Unsatisfied           135          3405     389.42313    12.2536        1.3987
                      Unsatisfied           125          3155     384.56734    11.3563        1.3809
                          Neutral           216          5429     489.37826    19.5404        1.7564
                        Satisfied           141          3507     417.54773    12.6208        1.5040
                   Very Satisfied            43          1052     221.59367     3.7874        0.7984

                            Total           660         16549     719.61536    59.5585        2.5713
----------------------------------------------------------------------------------------------------
        Total    Very Unsatisfied           209          5251     454.82598    18.8979        1.6326
                      Unsatisfied           203          5085     442.39032    18.2992        1.5897
                          Neutral           346          8718     550.81735    31.3773        1.9705
                        Satisfied           254          6302     507.01711    22.6805        1.8287
                   Very Satisfied            98          2430     330.97602     8.7452        1.1918

                            Total          1110         27786     119.25529    100.000
----------------------------------------------------------------------------------------------------


                                   Table of SchoolType by Response
                              Controlling for Department=Admin/Guidance

                                                     Weighted    Std Dev of               Std Err of
   SchoolType            Response     Frequency     Frequency      Wgt Freq    Percent       Percent
----------------------------------------------------------------------------------------------------
Middle School    Very Unsatisfied            42     649.43427     133.06194     5.8435        1.1947
                      Unsatisfied            31     460.35557     100.80158     4.1422        0.9076
                          Neutral           104          1568     186.99946    14.1042        1.6804
                        Satisfied            84          1269     165.71127    11.4142        1.4896
                   Very Satisfied            39     574.93878     110.37243     5.1732        0.9942

                            Total           300          4521     287.86832    40.6774        2.5801
----------------------------------------------------------------------------------------------------
  High School    Very Unsatisfied            53     777.77725     136.41869     6.9983        1.2285
                      Unsatisfied            92          1362     175.40862    12.2573        1.5806
                          Neutral           131          2005     212.34804    18.0404        1.8990
                        Satisfied           117          1739     190.07798    15.6437        1.7118
                   Very Satisfied            47     709.37033     126.54394     6.3828        1.1371

                            Total           440          6593     288.92483    59.3226        2.5801
----------------------------------------------------------------------------------------------------
        Total    Very Unsatisfied            95          1427     182.28132    12.8419        1.6374
                      Unsatisfied           123          1823     193.43045    16.3995        1.7446
                          Neutral           235          3572     250.22739    32.1447        2.2300
                        Satisfied           201          3007     226.82311    27.0579        2.0406
                   Very Satisfied            86          1284     160.83434    11.5560        1.4466

                            Total           740         11114      60.78850    100.000
----------------------------------------------------------------------------------------------------
```

Example 68.3. Output Data Sets

PROC SURVEYFREQ uses the Output Delivery System (ODS) to create output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality. Using ODS, you can create a SAS data set from any piece of PROC SURVEYFREQ output. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

When selecting tables for ODS output data set, you reference tables by their ODS table names. Each table created by PROC SURVEYFREQ is assigned a name, and the section "ODS Table Names" on page 4230 lists the table names.

To save the one-way table of Response from Figure 68.3 in an output data set, use an ODS OUTPUT statement as follows:

```
proc surveyfreq data=SIS_Survey;
   tables  Response / cl nowt;
   ods output OneWay=ResponseTable;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

Output 68.3.1 displays the output data set ResponseTable, which contains the one-way table of Response. This data set has six observations, and each of these observations corresponds to a row of the one-way table. The first five observations correspond to the five levels of Response, as they are ordered in the display, and the last observation corresponds to the overall total, which is the last row of the table. The data set ResponseTable includes a variable corresponding to each column of the one-way table. For example, the variable Percent contains the percentage estimates, and the variables LowerCL and UpperCL contain the lower and upper confidence limits for the percentage estimates.

Output 68.3.1. ResponseTable Output Data Set

```
 Obs        Table              Response          Frequency    Percent     StdErr     LowerCL     UpperCL

   1     Table Response    Very Unsatisfied          304      17.1676     1.2872     14.6364     19.6989
   2     Table Response    Unsatisfied               326      17.7564     1.2712     15.2566     20.2562
   3     Table Response    Neutral                   581      31.5965     1.5795     28.4904     34.7026
   4     Table Response    Satisfied                 455      23.9311     1.4761     21.0285     26.8338
   5     Table Response    Very Satisfied            184       9.5483     0.9523      7.6756     11.4210
   6     Table Response                   .         1850      100.000      _           _           _
```

PROC SURVEYFREQ also creates a table summary that is not displayed. Some of the information in this table is similar to that contained in the "Data Summary" table, but the table summary describes data used to analyze the specified table, while the data summary describes the entire input data set. Due to missing values, for example, the number of observations (or strata or clusters) used to analyze a particular table may differ from the number of observations (or strata or clusters) reported for the input data set in the "Data Summary" table. See the section "Missing Values" on page

4205 for more details. If you request confidence limits, the “Table Summary” table also contains the degrees of freedom and the *t*-value used to compute the confidence limits.

The following statements store the nondisplayed “Table Summary” table in the output data set ResponseSummary.

```
proc surveyfreq data=SIS_Survey;
   tables  Response / cl nowt;
   ods output TableSummary=ResponseSummary;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

Output 68.3.2 displays the output data set ResponseSummary.

Output 68.3.2. ResponseSummary Output Data Set

```
                                              Number     Number       Degrees
                                Number of       of         of            of           t
 Obs      Table                Observations   Strata    Clusters     Freedom    Percentile

   1      Table Response            1850           6         370          364      1.966503
```

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Bedrick, E.J. (1983), “Adjusted Chi-Squared Tests for Cross-Classified Tables of Survey Data,” *Biometrika*, 70, 591–596.

Brick, J.M. and Kalton, G. (1996), “Handling Missing Data in Survey Research,” *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W.G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Fienberg, S.E. (1980), *The Analysis of Cross-Classified Data*, Second Edition, Cambridge, MA: MIT Press.

Felligi, I.P. (1980), “Approximate Tests of Independence and Goodness of Fit Based on Stratified Multistage Samples,” *Journal of the American Statistical Association*, 75, 261–268.

Fleiss, J.L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons, Inc.

Fuller, W.A. (1975), “Regression Analysis for Sample Survey,” *Sankhyā*, 37, Series C, Pt. 3, 117–132.

Fuller, W.A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H.J. (1989), *PC CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Hansen, M.H., Hurwitz, W.N., and Madow, W.G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons, Inc.

Hidiroglou, M.A., Fuller, W.A., and Hickman, R.D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications, Inc.

Kalton, G. and Kaspyzyk, D. (1986), “The Treatment of Missing Survey Data,” *Survey Methodology*, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Koch, G.G., Freeman, D.H., and Freeman, J.L. (1975), “Strategies in the Multivariate Analysis of Data from Complex Surveys,” *International Statistical Review*, 43, 59–78.

Koch, G.G., Landis, J.R., Freeman, D.H., Freeman, J.L., and Lehnen, R.G. (1977), “A General Methodology for the Analysis of Experiments with Repeated Measurement of Categorical Data,” *Biometrics*, 33, 133–158.

Korn, E.L. and Graubard, B.I. (1990), “Simultaneous Testing with Complex Survey Data: Use of Bonferroni *t*-Statistics,” *The American Statistician*, 44, 270–276.

Lee, E.S., Forthoffer, R.N., and Lorimor, R.J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications, Inc.

Levy, P. and Lemeshow, S. (1999), *Sampling of Populations, Methods and Applications*, Third Edition, New York: John Wiley & Sons, Inc.

Lohr, S.L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Nathan, G. (1975), “Tests for Independence in Contingency Tables from Stratified Samples,” *Sankhyā*, 37, Series C, 77–87.

Rao, J.N.K. and Scott, A.J. (1979), “Chi-Squared Tests for Analysis of Categorical Data from Complex Surveys,” *Proceedings of the Survey Research Methods Section, ASA*, 58–66.

Rao, J.N.K. and Scott, A.J. (1981), “The Analysis of Categorical Data from Complex Surveys: Chi-Squared Tests for Goodness of Fit and Independence in Two-Way Tables,” *Journal of the American Statistical Association*, 76, 221–230.

Rao, J.N.K. and Scott, A.J. (1984), “On Chi-Squared Tests for Multiway Contingency Tables with Cell Properties Estimated from Survey Data,” *The Annals of Statistics*, 12, 46–60.

Rao, J.N.K. and Scott, A.J. (1987), “On Simple Adjustments to Chi-Square Tests with Survey Data,” *The Annals of Statistics*, 15, 385–397.

Särndal, C.E., Swenson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag Inc.

Satterthwaite, F.E. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics*, 2, 110–114.

Shah, B.V., Barnwell, B.G., and Bieler, G.S. (1996), *SUDAAN User's Manual: Release 7.0*, Research Triangle Park, NC: Research Triangle Institute.

Thomas, D.R., and Rao, J.N.K. (1984), "A Monte Carlo Study of Exact Levels of Goodness-of-Fit Statistics Under Cluster Sampling," *Proceedings of the Survey Research Methods Section, ASA*, 207–211.

Thomas, D.R., and Rao, J.N.K. (1985), "On the Power of Some Goodness-of-Fit Tests Under Cluster Sampling," *Proceedings of the Survey Research Methods Section, ASA*, 291–296.

Thomas, D.R., Singh, A.C., and Roberts, G.R. (1996), "Tests of Independence on Two-Way Tables Under Cluster Sampling: An Evaluation," *International Statistical Review*, 64, 295–311.

Wald, A. (1943), "Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations is Large," *Transactions of the American Mathematical Society*, 54, 426–482.

Woodruff, R.S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 69
The SURVEYLOGISTIC Procedure

Chapter Contents

Chapter 69
The SURVEYLOGISTIC Procedure

Overview

Categorical responses arise extensively in survey research. Common examples of responses include

- binary: e.g., attended graduate school or not
- ordinal: e.g., mild, moderate, and severe pain
- nominal: e.g., ABC, NBC, CBS, FOX TV network viewed at a certain hour

Logistic regression analysis is often used to investigate the relationship between such discrete responses and a set of explanatory variables. See Binder (1981, 1983), Roberts, Rao, and Kumar (1987), Skinner, Holt, and Smith (1989), Morel (1989), and Lehtonen and Pahkinen (1995) for papers that describe logistic regression for sample survey data.

For binary response models, the response of a sampling unit can take a specified value or not (for example, attended graduate school or not). Suppose $\mathbf{x}$ is a row vector of explanatory variables and π is the response probability to be modeled. The linear logistic model has the form

$$\text{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \mathbf{x}\boldsymbol{\beta}$$

where α is the intercept parameter and $\boldsymbol{\beta}$ is the vector of slope parameters.

The logistic model shares a common feature with the more general class of generalized linear models, namely, that a function $g = g(\mu)$ of the expected value, μ, of the response variable is assumed to be linearly related to the explanatory variables. Since μ implicitly depends on the stochastic behavior of the response, and since the explanatory variables are assumed to be fixed, the function g provides the link between the random (stochastic) component and the systematic (deterministic) component of the response variable. For this reason, Nelder and Wedderburn (1972) refer to $g(\cdot)$ as a link function. One advantage of the logit function over other link functions is that differences on the logistic scale are interpretable regardless of whether the data are sampled prospectively or retrospectively (McCullagh and Nelder 1989, Chapter 4). Other link functions that are widely used in practice are the probit function and the complementary log-log function. The SURVEYLOGISTIC procedure enables you to choose one of these link functions, resulting in fitting a broad class of binary response models of the form

$$g(\pi) = \alpha + \mathbf{x}\boldsymbol{\beta}$$

For ordinal response models, the response Y of an individual or an experimental unit may be restricted to one of a usually small number of ordinal values, denoted for convenience by $1, \ldots, D, D+1$ $(D \geq 1)$. For example, the pain severity can be classified into three response categories as 1=mild, 2=moderate, and 3=severe. The SURVEYLOGISTIC procedure fits a common slopes cumulative model, which is a parallel lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$g(\Pr(Y \leq d \,|\, \mathbf{x})) = \alpha_d + \mathbf{x}\boldsymbol{\beta}, \quad 1 \leq d \leq D$$

where $\alpha_1, \ldots, \alpha_k$ are k intercept parameters and $\boldsymbol{\beta}$ is the vector of slope parameters. This model has been considered by many researchers. Aitchison and Silvey (1957) and Ashford (1959) employ a probit scale and provide a maximum likelihood analysis; Walker and Duncan (1967) and Cox and Snell (1989) discuss the use of the log-odds scale. For the log-odds scale, the cumulative logit model is often referred to as the *proportional odds* model.

For nominal response logistic models, where the $D+1$ possible responses have no natural ordering, the logit model can also be extended to a *generalized logit* model, which has the form

$$\log\left(\frac{\Pr(Y = i \,|\, \mathbf{x})}{\Pr(Y = D+1 \,|\, \mathbf{x})}\right) = \alpha_i + \boldsymbol{\beta}_i'\mathbf{x}, \quad i = 1, \ldots, D$$

where the $\alpha_1, \ldots, \alpha_D$ are D intercept parameters, and the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_D$ are D vectors of parameters. These models were introduced by McFadden (1974) as the *discrete choice* model, and they are also known as *multinomial* models.

The SURVEYLOGISTIC procedure fits linear logistic regression models for discrete response survey data by the method of maximum likelihood. For statistical inferences, PROC SURVEYLOGISTIC incorporates complex survey sample designs, including designs with stratification, clustering, and unequal weighting.

The maximum likelihood estimation is carried out with either the Fisher-scoring algorithm or the Newton-Raphson algorithm. You can specify starting values for the parameter estimates. The logit link function in the ordinal logistic regression models can be replaced by the probit function or the complementary log-log function.

Odds ratio estimates are displayed along with parameter estimates. You can also specify the change in the explanatory variables for which odds ratio estimates are desired. Variances of the regression parameters and odds ratios are computed using the Taylor expansion approximation; see Binder (1983).

The SURVEYLOGISTIC procedure enables you to specify categorical variables (also known as CLASS variables) as explanatory variables. It also enables you to specify interaction terms in the same way as in the LOGISTIC procedure.

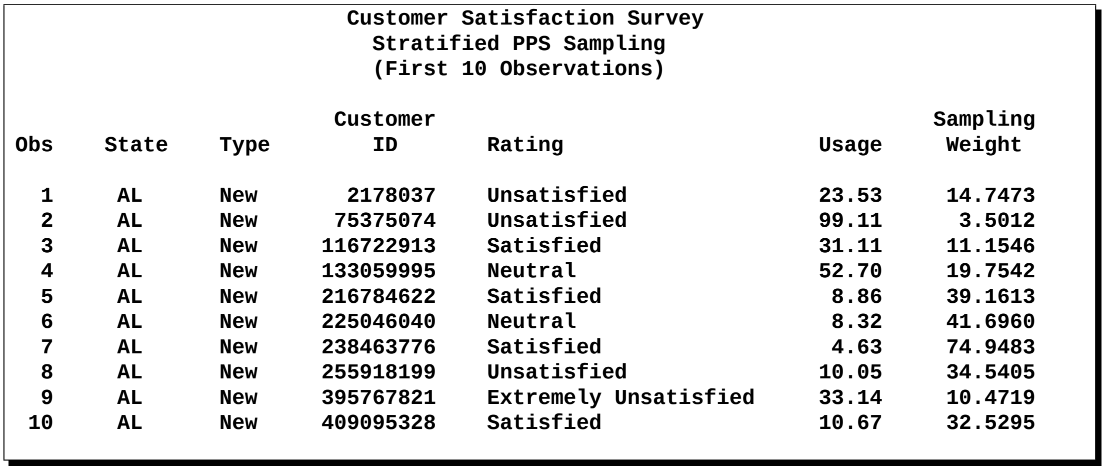
Like many procedures in SAS/STAT software that allow the specification of CLASS variables, the SURVEYLOGISTIC procedure provides a CONTRAST statement for specifying customized hypothesis tests concerning the model parameters. The CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful for obtaining odds ratio estimates for various levels of the CLASS variables.

Getting Started

The SURVEYLOGISTIC procedure is similar to the LOGISTIC procedure and other regression procedures in the SAS System. Please refer to Chapter 42, “The LOGISTIC Procedure,” for general information about how to perform logistic regression using SAS. PROC SURVEYLOGISTIC is designed to handle sample survey data, and thus it incorporates the sampling design information into the analysis.

The following example illustrates how to use PROC SURVEYLOGISTIC to perform logistic regression for sample survey data.

In the customer satisfaction survey example in the “Getting Started” section on page 4422 of Chapter 72, “The SURVEYSELECT Procedure,” an Internet service provider conducts a customer satisfaction survey. The survey population consists of the company’s current subscribers from four states: Alabama (AL), Florida (FL), Georgia (GA), and South Carolina (SC). The company plans to select a sample of customers from this population, interview the selected customers and ask their opinions on customer service, and then make inferences about the entire population of subscribers from the sample data. A stratified sample is selected using the probability proportional to size (PPS) method. The sample design divides the customers into strata depending on their types (‘Old’ or ‘New’) of their states (AL, FL, GA, SC). There are eight strata in all. Within each stratum, customers are selected and interviewed using the PPS with replacement method, where the size variable is Usage. The stratified PPS sample contains 192 customers. The data are stored in the SAS data set SampleStrata. Figure 69.1 displays the first 10 observations of this data set.



```
                          Customer Satisfaction Survey
                            Stratified PPS Sampling
                            (First 10 Observations)

                          Customer                                        Sampling
Obs     State    Type        ID       Rating                     Usage     Weight

  1      AL      New       2178037    Unsatisfied                23.53     14.7473
  2      AL      New      75375074    Unsatisfied                99.11      3.5012
  3      AL      New     116722913    Satisfied                  31.11     11.1546
  4      AL      New     133059995    Neutral                    52.70     19.7542
  5      AL      New     216784622    Satisfied                   8.86     39.1613
  6      AL      New     225046040    Neutral                     8.32     41.6960
  7      AL      New     238463776    Satisfied                   4.63     74.9483
  8      AL      New     255918199    Unsatisfied                10.05     34.5405
  9      AL      New     395767821    Extremely Unsatisfied      33.14     10.4719
 10      AL      New     409095328    Satisfied                  10.67     32.5295
```

Figure 69.1. Stratified PPS Sample (First 10 Observations)

In the SAS data set SampleSRS, the variable CustomerID uniquely identifies each customer. The variable State contains the state of the customer's address. The variable Type equals 'Old' if the customer has subscribed to the service for more than one year; otherwise, the variable Type equals 'New'. The variable Usage contains the customer's average monthly service usage, in hours. The variable Rating contains the customer's responses to the survey. The sample design uses an unequal probability sampling method, with the sampling weights stored in the variable SamplingWeight.

The following SAS statements fit a cumulative logistic model between the satisfaction levels and the Internet usage using the stratified PPS sample.

```
title 'Customer Satisfaction Survey';
proc surveylogistic data=SampleStrata;
strata state type/list;
model Rating (order=internal) = Usage;
weight SamplingWeight;
run;
```

The PROC statement invokes the SURVEYLOGISTIC procedure. The STRATA statement specifies the stratification variables State and Type that are used in the sample design. The LIST option requests a summary of the stratification. In the MODEL statement, Rating is the response variable and Usage is the explanatory variable. The ORDER=internal is used for the response variable Rating to ask the procedure to order the response levels using the internal numerical value (1-5) instead of the formatted character value. The WEIGHT statement specifies the variable SamplingWeight that contains the sampling weights.

The results of this analysis are shown in the following tables.

```
                          Customer Satisfaction Survey

                          The SURVEYLOGISTIC Procedure

                               Model Information

   Data Set                      WORK.SAMPLESTRATA
   Response Variable             Rating
   Number of Response Levels     5
   Stratum Variables             State
                                 Type
   Number of Strata              8
   Weight Variable               SamplingWeight              Sampling Weight
   Model                         Cumulative Logit
   Optimization Technique        Fisher's Scoring
   Variance Adjustment           Degrees of Freedom (DF)
```

Figure 69.2. Stratified PPS Sample, Model Information

PROC SURVEYLOGISTIC first lists the following model fitting information and sample design information in Figure 69.2:

- The link function is the logit of the cumulative of the lower response categories.

- The Fisher Scoring optimization technique is used to obtain the maximum likelihood estimates for the regression coefficients.
- The response variable is Rating, which has five response levels.
- The stratification variables are State and Type.
- There are eight strata in the sample.
- The weight variable is SamplingWeight.
- The variance adjustment method used for the regression coefficients is the default degrees of freedom adjustment.

```
                           Customer Satisfaction Survey

                     Number of Observations Read         192
                     Number of Observations Used         192
                     Sum of Weights Read            13262.74
                     Sum of Weights Used            13262.74
```

Figure 69.3. Stratified PPS Sample, Number of Observations

Figure 69.3 lists the number of observations in the data set and the number of observations used in the analysis. Since no missing value presents in this example, observations in the entire data set are used in the analysis. The sums of weights are also reported in this table.

```
                           Customer Satisfaction Survey

                                 Response Profile

         Ordered                                      Total            Total
           Value     Rating                       Frequency           Weight

               1     Extremely Unsatisfied               52        2067.1092
               2     Unsatisfied                         47        2148.7127
               3     Neutral                             47        3649.4869
               4     Satisfied                           38        2533.5379
               5     Extremely Satisfied                  8        2863.8888

        Probabilities modeled are cumulated over the lower Ordered Values.
```

Figure 69.4. Stratified PPS Sample, Response Profile

The “Response Profile” table in Figure 69.4 lists the five response levels, their ordered values, and their total frequencies and total weights for each category. Due to the ORDER=internal option for the response variable Rating, the category “Extremely Unsatisfied” has the Ordered Value 1, the category “Unsatisfied” has the Ordered Value 2, and so on.

```
                         Customer Satisfaction Survey

                             Stratum Information

                  Stratum
                   Index     State       Type           N Obs
                  ------------------------------------------
                     1       AL          New               22
                     2                   Old               24
                     3       FL          New               25
                     4                   Old               22
                     5       GA          New               25
                     6                   Old               25
                     7       SC          New               24
                     8                   Old               25
                  ------------------------------------------
```

Figure 69.5. Stratified PPS Sample, Stratification Summary

Figure 69.5 displays the output of the stratification summary. There are a total of eight strata, and each stratum is defined by the customer types within each state. The table also shows the number of customers within each stratum.

```
                         Customer Satisfaction Survey

                 Score Test for the Proportional Odds Assumption

                       Chi-Square       DF     Pr > ChiSq

                        3692.2558        3         <.0001
```

Figure 69.6. Stratified PPS Sample, Testing the Proportional Odds Assumption

Figure 69.6 shows the chi-square test for testing the proportional odds assumption. The test is highly significant, which indicates that the cumulative logit model may not adequately fit the data.

```
                         Customer Satisfaction Survey

                           Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.


                             Model Fit Statistics

                                                Intercept
                                  Intercept           and
                     Criterion         Only    Covariates

                     AIC          42099.954     41378.851
                     SC           42112.984     41395.139
                     -2 Log L     42091.954     41368.851
```

Figure 69.7. Stratified PPS Sample, Model Fitting Information

Figure 69.7 shows the iteration algorithm converged to obtain the MLE for this example. The "Model Fit Statistics" table contains the Akaike Information Criterion (AIC), the Schwarz Criterion (SC), and the negative of twice the log likelihood (-2 Log L) for the intercept-only model and the fitted model. AIC and SC can be used to compare different models, and the ones with smaller values are preferred.

```
                          Customer Satisfaction Survey

                    Testing Global Null Hypothesis: BETA=0

            Test                 Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio       723.1023        1         <.0001
            Score                  465.4939        1         <.0001
            Wald                     4.5212        1         0.0335
```

Figure 69.8. Stratified PPS Sample, Testing Global Null Hypothesis

The table "Testing Global Null Hypothesis: BETA=0" in Figure 69.8 shows the likelihood ratio test, the efficient score test, and the Wald test for testing the significance of the explanatory variable (Usage). All tests are significant.

```
                          Customer Satisfaction Survey

                   Analysis of Maximum Likelihood Estimates

                                                  Standard          Wald
Parameter                         DF    Estimate     Error    Chi-Square    Pr > ChiSq

Intercept Extremely Unsatisfied    1     -2.0168    0.3988       25.5769        <.0001
Intercept Unsatisfied              1     -1.0527    0.3543        8.8292        0.0030
Intercept Neutral                  1      0.1334    0.4189        0.1015        0.7501
Intercept Satisfied                1      1.0751    0.5794        3.4432        0.0635
Usage                              1      0.0377    0.0178        4.5212        0.0335
```

Figure 69.9. Stratified PPS Sample, Parameter Estimates

Figure 69.9 shows the parameter estimates of the logistic regression and their standard errors.

```
                          Customer Satisfaction Survey

                              Odds Ratio Estimates

                                Point          95% Wald
                   Effect    Estimate      Confidence Limits

                   Usage        1.038        1.003       1.075
```

Figure 69.10. Stratified PPS Sample, Odds Ratios

Figure 69.10 displays the odds ratio estimate and its standard error.

Syntax

The following statements are available in PROC SURVEYLOGISTIC:

PROC SURVEYLOGISTIC < *options* >**;**

BY *variables* **;**

CLASS *variable* <(*v-options*)> <*variable* <(*v-options*)>... >
< / *v-options* >**;**

CLUSTER *variables* **;**

CONTRAST *'label'* *effect values* <,... *effect values*>< /*options* >**;**

FREQ *variable* **;**

MODEL *events/trials* = < *effects* >< / *options* >**;**

MODEL *variable* < (*variable_options*) > = < *effects* >< / *options* >**;**

STRATA *variables* < / *options* > **;**

< *label:* > **TEST** *equation1* < , . . . , < *equationk* >>< /*option* >**;**

UNITS *independent1* = *list1* < . . . *independentk* = *listk* >< /*option* > **;**

WEIGHT *variable* </ *option* >**;**

The PROC SURVEYLOGISTIC and MODEL statements are required. The CLASS, CLUSTER, STRATA, and CONTRAST statements can appear multiple times. You should only use one MODEL statement and one WEIGHT statement. The CLASS statement (if used) must precede the MODEL statement, and the CONTRAST statement (if used) must follow the MODEL statement.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC SURVEYLOGISTIC statement. The remaining statements are covered in alphabetical order.

PROC SURVEYLOGISTIC Statement

PROC SURVEYLOGISTIC < *options* >**;**

The PROC SURVEYLOGISTIC statement invokes the SURVEYLOGISTIC procedure and optionally identifies input data sets and controls the ordering of the response levels.

ALPHA=α

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1 - \alpha)$% confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

INEST= *SAS-data-set*

names the SAS data set that contains initial estimates for all the parameters in the model. BY-group processing is allowed in setting up the INEST= data set. See the section "INEST= Data Set" on page 4280 for more information.

MISSING

requests that the procedure treat missing values as a valid category for all categorical variables, which include classification variables in the model, strata variables, and cluster variables.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOSORT

suppresses the internal sorting process to shorten the computation time if the data set is presorted by the STRATA and CLUSTER variables. By default, the procedure sorts the data by the STRATA variables if you use the STRATA statement; then the procedure sorts the data by the CLUSTER variables within strata. If your data are already stored by the order of STRATA and CLUSTER variables, then you can specify this option to omit this sorting process to reduce the usage of computing resources, especially when your data set is very large. However, if you specify this NOSORT option while your data is not presorted by STRATA and CLUSTER variables, then any changes in these variables creates a new stratum or cluster.

RATE=*value* | *SAS-data-set*
R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or specifies an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction (fpc) for variance estimation when the sample design is without replacement. If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of primary sampling units (PSUs) selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 4280 for more details.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYLOGISTIC will convert that number to a proportion. The procedure treats the value 1 as 100%.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

TOTAL=*value* | *SAS-data-set*
N=*value* | *SAS-data-set*

specifies the total number of primary sampling units (PSUs) in the study population as a positive *value*, or names an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for variance estimation.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Specification of Population Totals and Sampling Rates" on page 4280 for more details.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYLOGISTIC to obtain separate analyses on observations in groups defined by the BY variables.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty.

When a BY statement appears, the procedure expects the input data sets to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Use the BY statement options NOTSORTED or DESCENDING in the BY statement. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable <(v-options)> <variable <(v-options)>... >*
< / v-options >;

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. However, individual CLASS variable *v-options* override the global *v-options*.

CPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding dummy variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING
DESC

reverses the sorting order of the classification variable.

LPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding dummy variables.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST statement. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows how PROC SURVEYLOGISTIC interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter

on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=EFFECT. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used.

EFFECT	specifies effect coding
GLM	specifies less-than-full-rank, reference cell coding; this option can only be used as a global option
ORDINAL	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding
REFERENCE REF	specifies reference cell coding
ORTHEFFECT	orthogonalizes PARAM=EFFECT
ORTHORDINAL ORTHOTHERM	orthogonalizes PARAM=ORDINAL
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL
ORTHREF	orthogonalizes PARAM=REFERENCE

The EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization.

REF=*'level'* **|** *keyword*

specifies the reference level for PARAM=EFFECT or PARAM=REFERENCE. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference
LAST	designates the last ordered level as reference

CLUSTER Statement

CLUSTER | **CLUSTERS** *variables* **;**

The CLUSTER statement names variables that identify the clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement. See the section "Primary Sampling Units (PSUs)" on page 4281 for more information.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can use multiple CLUSTER statements to specify cluster variables. The procedure uses all variables from all CLUSTER statements to create clusters.

CONTRAST Statement

CONTRAST *'label' row-description* <*,..., row-description*>< / *options* >**;**

where a *row-description* is: *effect values* <*,...effect values*>

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST statement in PROC LOGISTIC and PROC GLM, depending on the coding schemes used with any classification variables involved.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\theta} = \mathbf{0}$, where $\boldsymbol{\theta}$ is the parameter vector. You must be familiar with the details of the model parameterization that PROC SURVEYLOGISTIC uses (for more information, see the PARAM= option in the section "CLASS Statement" on page 4253). Optionally, the CONTRAST statement enables you to estimate each row, $\boldsymbol{l}_i'\boldsymbol{\theta}$, of $\mathbf{L}\boldsymbol{\theta}$ and test the hypothesis $\boldsymbol{l}_i'\boldsymbol{\theta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. The name INTERCEPT can be used as an effect when one or more intercepts are included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the **L** matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The “Class Level Information” table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters.

The rows of **L** are specified in order and are separated by commas. Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*. For any of the full-rank parameterizations, if an effect is not specified in the CONTRAST statement, all of its coefficients in the **L** matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

When you use effect coding (by default or by specifying PARAM=EFFECT in the CLASS statement), all parameters are directly estimable (involve no other parameters). For example, suppose an effect coded CLASS variable A has four levels. Then there are three parameters ($\alpha_1, \alpha_2, \alpha_3$) representing the first three levels, and the fourth parameter is represented by

$$-\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\theta} = \mathbf{0}$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast ’1 vs. 4’ A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast ’1&2 vs. 3’ A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example,

```
contrast ’1 vs. 2    ’  A  1 -1  0;
contrast ’1&2 vs. 4  ’  A  3  3  2;
contrast ’1&2 vs. 3&4’  A  2  2  0;
contrast ’Main Effect’  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability. If PROC SURVEYLOGISTIC finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC SURVEYLOGISTIC handles missing level combinations of classification variables in the same manner as PROC LOGISTIC. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement. If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the LOGISTIC procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement, that is, the rank of **L**.

You can specify the following options after a slash (/).

ALPHA=α

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)$% confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

E

requests that the **L** matrix be displayed.

ESTIMATE=*keyword*

requests that each individual contrast (that is, each row, $l_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$) or exponentiated contrast ($\mathrm{e}^{l_i'\boldsymbol{\beta}}$) be estimated and tested. PROC SURVEYLOGISTIC displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the contrast or the exponentiated contrast ($\mathrm{e}^{l_i'\boldsymbol{\beta}}$), or both, by specifying one of the following *keywords*:

PARM	specifies that the contrast itself be estimated
EXP	specifies that the exponentiated contrast be estimated
BOTH	specifies that both the contrast and the exponentiated contrast be estimated

SINGULAR = *number*

tunes the estimability check. This option is ignored when the full-rank parameterization is used. If $\boldsymbol{v}$ is a vector, define ABS($\boldsymbol{v}$) to be the largest absolute value of the elements of $\boldsymbol{v}$. For a row vector $\boldsymbol{l}'$ of the contrast matrix $\mathbf{L}$, define c to be equal to ABS$(\boldsymbol{l})$ if ABS$(\boldsymbol{l})$ is greater than 0; otherwise, c equals 1. If ABS$(\boldsymbol{l}' - \boldsymbol{l}'\boldsymbol{T})$ is greater than $c * number$, then $\boldsymbol{l}$ is declared nonestimable. The $\boldsymbol{T}$ matrix is the Hermite form matrix $\boldsymbol{I}_0^- \boldsymbol{I}_0$, where $\boldsymbol{I}_0^-$ represents a generalized inverse of the information matrix $\boldsymbol{I}_0$ of the null model. The value for $number$ must be between 0 and 1; the default value is 1E$-$4.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC SURVEYLOGISTIC treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the model fitting. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

If you use the *events/trials* syntax in the MODEL statement, the FREQ statement is disallowed because the event and trial variables represent the frequencies in the data set.

MODEL Statement

MODEL *events/trials*= $<$ *effects* $><$ */ options* $>$**;**

MODEL *variable* $<$ *(variable_options)* $>$= $<$ *effects* $><$ */options* $>$**;**

The MODEL statement names the response variable and the explanatory effects, including covariates, main effects, interactions, and nested effects; see the section "Specification of Effects" on page 1784 of Chapter 32, "The GLM Procedure," for more information. If you omit the explanatory variables, the procedure fits an intercept-only model. Model options can be specified after a slash (/).

Two forms of the MODEL statement can be specified. The first form, referred to as *single-trial* syntax, is applicable to binary, ordinal, and nominal response data. The second form, referred to as *events/trials* syntax, is restricted to the case of binary response data. The *single-trial* syntax is used when each observation in the DATA= data set contains information on only a single trial, for instance, a single subject in an experiment. When each observation contains information on multiple binary-response trials, such as the counts of the number of subjects observed and the number responding, then *events/trials* syntax can be used.

In the *events/trials* syntax, you specify two variables that contain count data for a binomial experiment. These two variables are separated by a slash. The value of the first variable, *events*, is the number of positive responses (or events). The value of the second variable, *trials*, is the number of trials. The values of both *events* and ($trials - events$) must be nonnegative and the value of *trials* must be positive for the response to be valid.

In the *single-trial* syntax, you specify one variable (on the left side of the equal sign) as the response variable. This variable can be character or numeric. Options specific to the response variable can be specified immediately after the response variable with a pair of parentheses around them.

For both forms of the MODEL statement, explanatory *effects* follow the equal sign. Variables can be either continuous or classification variables. Classification variables can be character or numeric, and they must be declared in the CLASS statement. When an effect is a classification variable, the procedure enters a set of coded columns into the design matrix instead of directly entering a single column containing the values of the variable.

Response Variable Options

You specify the following options by enclosing them in a pair of parentheses after the response variable.

DESCENDING | DESC

reverses the order of response categories. If both the DESCENDING and ORDER= options are specified, PROC SURVEYLOGISTIC orders the response categories according to the ORDER= option and then reverses that order. See the "Response Level Ordering" section on page 4269 for more detail.

EVENT='*category*' | *keyword*

specifies the event category for the binary response model. PROC SURVEYLOGISTIC models the probability of the event category. The EVENT= option has no effect when there are more than two response categories. You can specify the value (formatted if a format is applied) of the event category in quotes or you can specify one of the following keywords. The default is EVENT=FIRST.

FIRST	designates the first ordered category as the event
LAST	designates the last ordered category as the event

One of the most common sets of response levels is {0,1}, with 1 representing the event for which the probability is to be modeled. Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. To specify the value 1 as the event category, use the model statement

```
model Y(event='1') = Exposure;
```

ORDER= DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the response variable. By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows the interpretation of the ORDER= values.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

REFERENCE='*category***' |** *keyword*
REF='*category***' |** *keyword*

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each nonreference category is contrasted with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes or you can specify one of the following keywords. The default is REF=LAST.

FIRST	designates the first ordered category as the reference
LAST	designates the last ordered category as the reference

Model Options

Model options can be specified after a slash (/). Table 69.1 summarizes the options available in the MODEL statement.

Table 69.1. Model Statement Options

Option	Description
Model Specification Options	
LINK=	Specifies link function
NOINT	Suppresses intercept(s)
OFFSET=	Specifies offset variable
Convergence Criterion Options	
ABSFCONV=	Specifies absolute function convergence criterion
FCONV=	Specifies relative function convergence criterion
GCONV=	Specifies relative gradient convergence criterion
XCONV=	Specifies relative parameter convergence criterion
MAXITER=	Specifies maximum number of iterations
NOCHECK	Suppresses checking for infinite parameters
RIDGING=	Specifies technique used to improve the log-likelihood function when its value is worse than that of the previous step
SINGULAR=	Specifies tolerance for testing singularity
TECHNIQUE=	Specifies iterative algorithm for maximization
Options for Adjustment to Variance Estimation	
VADJUST=	Choose variance estimation adjustment method
Options for Confidence Intervals	
ALPHA=	Specifies α for the $100(1-\alpha)\%$ confidence intervals
CLPARM	Computes confidence intervals for parameters
CLODDS	Computes confidence intervals for odds ratios
Options for Display of Details	
CORRB	Displays correlation matrix
COVB	Displays covariance matrix
EXPB	Displays exponentiated values of estimates
ITPRINT	Displays iteration history
NODUMMYPRINT	Suppresses “Class Level Information” table
PARMLABEL	Displays parameter labels
RSQUARE	Displays generalized R^2
STB	Displays standardized estimates

The following list describes these options.

ABSFCONV=*value*

specifies the absolute function convergence criterion. Convergence requires a small change in the log-likelihood function in subsequent iterations,

$$|l^{(i)} - l^{(i-1)}| < value$$

where $l^{(i)}$ is the value of the log-likelihood function at iteration i. See the section “Convergence Criteria” on page 4277.

ALPHA=α

sets the level of significance α for $100(1-\alpha)$% confidence intervals for regression parameters or odds ratios. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC SURVEYLOGISTIC statement, or $\alpha = 0.05$ if the option is not specified. This option has no effect unless confidence limits for the parameters or odds ratios are requested.

CLODDS

requests confidence intervals for the odds ratios. Computation of these confidence intervals is based on individual Wald tests. The confidence coefficient can be specified with the ALPHA= option. See the "Wald Confidence Intervals for Parameters" section on page 4288 for more information.

CLPARM

requests confidence intervals for the parameters. Computation of these confidence intervals is based on the individual Wald tests. The confidence coefficient can be specified with the ALPHA= option. See the "Wald Confidence Intervals for Parameters" section on page 4288 for more information.

CORRB

displays the correlation matrix of the parameter estimates.

COVB

displays the covariance matrix of the parameter estimates.

EXPB
EXPEST

displays the exponentiated values ($\mathrm{e}^{\hat{\theta}_i}$) of the parameter estimates $\hat{\theta}_i$ in the "Analysis of Maximum Likelihood Estimates" table for the logit model. These exponentiated values are the estimated odds ratios for the parameters corresponding to the continuous explanatory variables.

FCONV=*value*

specifies the relative function convergence criterion. Convergence requires a small relative change in the log-likelihood function in subsequent iterations,

$$\frac{|l^{(i)} - l^{(i-1)}|}{|l^{(i-1)}| + 1\mathrm{E}{-6}} < value$$

where $l^{(i)}$ is the value of the log-likelihood at iteration i. See the section "Convergence Criteria" on page 4277.

GCONV=*value*

specifies the relative gradient convergence criterion. Convergence requires that the normalized prediction function reduction is small,

$$\frac{\mathbf{g}'^{(i)}\mathbf{I}^{(i)}\mathbf{g}^{(i)}}{|l^{(i)}| + 1\mathrm{E}{-6}} < value$$

where $l^{(i)}$ is the value of the log-likelihood function, $\mathbf{g}^{(i)}$ is the gradient vector, and $\mathbf{I}^{(i)}$ the (expected) information matrix. All of these functions are evaluated at iteration

i. This is the default convergence criterion, and the default value is 1E$-$8. See the section "Convergence Criteria" on page 4277.

ITPRINT

displays the iteration history of the maximum-likelihood model fitting. The ITPRINT option also displays the last evaluation of the gradient vector and the final change in the -2 Log Likelihood.

LINK=*keyword*
L=*keyword*

specifies the link function linking the response probabilities to the linear predictors. You can specify one of the following keywords. The default is LINK=LOGIT.

CLOGLOG the complementary log-log function. PROC SURVEYLOGISTIC fits the binary complementary log-log model for binary response and fits the cumulative complementary log-log model when there are more than two response categories. Aliases: CCLOGLOG, CCLL, CUMCLOGLOG.

GLOGIT the generalized logit function. PROC SURVEYLOGISTIC fits the generalized logit model where each nonreference category is contrasted with the reference category. You can use the response variable option REF= to specify the reference category.

LOGIT the cumulative logit function. PROC SURVEYLOGISTIC fits the binary logit model when there are two response categories and fits the cumulative logit model when there are more than two response categories. Aliases: CLOGIT, CUMLOGIT.

PROBIT the inverse standard normal distribution function. PROC SURVEYLOGISTIC fits the binary probit model when there are two response categories and fits the cumulative probit model when there are more than two response categories. Aliases: NORMIT, CPROBIT, CUMPROBIT.

See the section "Link Functions and the Corresponding Distributions" on page 4273 for details.

MAXITER=*n*

specifies the maximum number of iterations to perform. By default, MAXITER=25. If convergence is not attained in n iterations, the displayed output created by the procedure contain results that are based on the last maximum likelihood iteration.

NOCHECK

disables the checking process to determine whether maximum likelihood estimates of the regression parameters exist. If you are sure that the estimates are finite, this option can reduce the execution time if the estimation takes more than eight iterations. For more information, see the "Existence of Maximum Likelihood Estimates" section on page 4277.

NODUMMYPRINT
NODESIGNPRINT
NODP

suppresses the “Class Level Information” table, which shows how the design matrix columns for the CLASS variables are coded.

NOINT

suppresses the intercept for the binary response model or the first intercept for the ordinal response model.

OFFSET= *name*

names the offset variable. The regression coefficient for this variable will be fixed at 1.

PARMLABEL

displays the labels of the parameters in the “Analysis of Maximum Likelihood Estimates” table.

RIDGING=ABSOLUTE | RELATIVE | NONE

specifies the technique used to improve the log-likelihood function when its value in the current iteration is less than that in the previous iteration. If you specify the RIDGING=ABSOLUTE option, the diagonal elements of the negative (expected) Hessian are inflated by adding the ridge value. If you specify the RIDGING=RELATIVE option, the diagonal elements are inflated by a factor of 1 plus the ridge value. If you specify the RIDGING=NONE option, the crude line search method of taking half a step is used instead of ridging. By default, RIDGING=RELATIVE.

RSQUARE
RSQ

requests a generalized R^2 measure for the fitted model. For more information, see the “Generalized Coefficient of Determination” section on page 4280.

SINGULAR=*value*

specifies the tolerance for testing the singularity of the Hessian matrix (Newton-Raphson algorithm) or the expected value of the Hessian matrix (Fisher-scoring algorithm). The Hessian matrix is the matrix of second partial derivatives of the log likelihood. The test requires that a pivot for sweeping this matrix be at least this number times a norm of the matrix. Values of the SINGULAR= option must be numeric. By default, SINGULAR=1E−12.

STB

displays the standardized estimates for the parameters for the continuous explanatory variables in the “Analysis of Maximum Likelihood Estimates” table. The standardized estimate of θ_i is given by $\hat{\theta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

For the intercept parameters and parameters associated with a CLASS variable, the standardized estimates are set to missing.

TECHNIQUE=FISHER | NEWTON
TECH=FISHER | NEWTON

specifies the optimization technique for estimating the regression parameters. NEWTON (or NR) is the Newton-Raphson algorithm and FISHER (or FS) is the Fisher-scoring algorithm. Both techniques yield the same estimates, but the estimated covariance matrices are slightly different except for the case when the LOGIT link is specified for binary response data. The default is TECHNIQUE=FISHER. See the section "Iterative Algorithms for Model-Fitting" on page 4275 for details.

VADJUST=DF | MOREL | NONE $<$ **(** *Morel-options* **)** $>$
VARADJ=DF | MOREL | NONE $<$ **(** *Morel-options* **)** $>$
VARADJUST=DF | MOREL | NONE $<$ **(** *Morel-options* **)** $>$

specifies an adjustment to the variance estimation (on page 4286) for the regression coefficients.

By default, PROC SURVEYLOGISTIC uses the degrees of freedom adjustment VADJUST=DF.

You can specify the VADJUST=MOREL option for the variance adjustment proposed by Morel (1989).

If you do not wish to use any variance adjustment, you can specify the VADJUST=NONE option.

You can specify the following *Morel-options* within parentheses after the VADJUST=MOREL option.

ADJBOUND=ϕ

sets the upper bound coefficient ϕ in the variance adjustment. This upper bound must be positive. By default, the procedure use $\phi = 0.5$. See the section "Adjustments to the Variance Estimation" on page 4286 for more details on how this upper bound is used in the variance estimation.

DEFFBOUND=δ

sets the lower bound of the estimated design effect in the variance adjustment. This lower bound must be positive. By default, the procedure use $\delta = 1$. See the section "Adjustments to the Variance Estimation" on page 4286 for more details on how this lower bound is used in the variance estimation.

XCONV=*value*

specifies the relative parameter convergence criterion. Convergence requires a small relative parameter change in subsequent iterations,

$$\max_j |\delta_j^{(i)}| < value$$

where

$$\delta_j^{(i)} = \begin{cases} \theta_j^{(i)} - \theta_j^{(i-1)} & |\theta_j^{(i-1)}| < 0.01 \\ \frac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}} & \text{otherwise} \end{cases}$$

and $\theta_j^{(i)}$ is the estimate of the jth parameter at iteration i. See the section "Iterative Algorithms for Model-Fitting" on page 4275.

STRATA Statement

STRATA | **STRATUM** *variables* < */ option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of levels of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Specification of Population Totals and Sampling Rates" on page 4280 for more information.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the STRATA variables determine the levels. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *SAS Procedures Guide*.

You can specify the following option in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which includes values of the STRATA variables and sampling rates for each stratum. This table also provides the number of observations and number of clusters for each stratum and analysis variable. See the section "Displayed Output" on page 4292 for more details.

TEST Statement

< *label:* > **TEST** *equation1* < *, . . . ,* < *equationk* >>< */option* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. The Wald test is used to jointly test the null hypotheses ($H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$) specified in a single TEST statement. When $\mathbf{c} = \mathbf{0}$ you should specify a CONTRAST statement instead.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\mathbf{c}$ vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output and should always be included. You can submit multiple TEST statements.

The form of an *equation* is as follows:

term < ±*term* **. . .** > < = ±*term* < ±*term* **. . .** >>

where *term* is a parameter of the model, or a constant, or a constant times a parameter. For a binary response model, the intercept parameter is named INTERCEPT;

for an ordinal response model, the intercept parameters are named INTERCEPT, INTERCEPT2, INTERCEPT3, and so on. When no equal sign appears, the expression is set to 0. The following code illustrates possible uses of the TEST statement:

```
proc surveylogistic;
   model y= a1 a2 a3 a4;
   test1: test intercept + .5 * a2 = 0;
   test2: test intercept + .5 * a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

You can specify the following option in the TEST statement after a slash(/).

PRINT

displays intermediate calculations in the testing of the null hypothesis $H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$. This includes $\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}'$ bordered by $(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$ and $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$, where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimator of $\boldsymbol{\theta}$ and $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\theta}}$.

For more information, see the "Testing Linear Hypotheses about the Regression Coefficients" section on page 4288.

UNITS Statement

UNITS *independent1 = list1 < . . . independentk = listk >< /option >* **;**

The UNITS statement enables you to specify units of change for the continuous explanatory variables so that customized odds ratios can be estimated. An estimate of the corresponding odds ratio is produced for each unit of change specified for an explanatory variable. The UNITS statement is ignored for CLASS variables. If the CLODDS option is specified in the MODEL statement, the corresponding confidence limits for the odds ratios are also displayed.

The term *independent* is the name of an explanatory variable and *list* represents a list of units of change, separated by spaces, that are of interest for that variable. Each unit of change in a list has one of the following forms:

- *number*
- SD or $-$SD
- *number* * SD

where *number* is any nonzero number, and SD is the sample standard deviation of the corresponding independent variable. For example, $X = -2$ requests an odds ratio that represents the change in the odds when the variable X is decreased by two units.

$X = 2{*}\mathrm{SD}$ requests an estimate of the change in the odds when X is increased by two sample standard deviations.

You can specify the following option in the UNITS statement after a slash(/).

DEFAULT= *list*

gives a list of units of change for all explanatory variables that are not specified in the UNITS statement. Each unit of change can be in any of the forms described previously. If the DEFAULT= option is not specified, PROC SURVEYLOGISTIC does not produce customized odds ratio estimates for any explanatory variable that is not listed in the UNITS statement.

For more information, see the "Odds Ratio Estimation" section on page 4288.

WEIGHT Statement

WEIGHT *variable* < */ option* >**;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric. If you do not specify a WEIGHT statement, PROC SURVEYLOGISTIC assigns all observations a weight of 1. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

Details

Missing Values

Any observation with missing values for the response, offset, or explanatory variables is excluded from the analysis. The estimated linear predictor, its standard error estimate, the fitted probabilities, and their confidence limits are not computed for any observation with missing offset or explanatory variable values.

An observation is also excluded if it has a missing value for any STRATA or CLUSTER variable, unless the MISSING option is used in the PROC SURVEYLOGISTIC statement.

Missing values in your survey data (such as nonresponse) can compromise the quality of your results. An observation without missing values is called a *complete respondent*, and an observation with missing values is called an *incomplete respondent*.

If the missing data are missing at random, then PROC SURVEYLOGISTIC produces unbiased results when it excludes observations with missing values. However, if the complete respondents are different from the incomplete respondents with regard to a survey effect or outcome, then excluding nonrespondents from the analysis may result in biased estimates that do not accurately represent the survey population.

When the missing data are not missing at random, you should use imputation to replace missing values with acceptable values and use sampling weight adjustments to compensate for nonresponse before you use PROC SURVEYLOGISTIC. Refer to Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more information.

Model Specification

Response Level Ordering

Response level ordering is important because, by default, PROC SURVEYLOGISTIC models the probabilities of response levels with lower *Ordered Value.* Ordered Values, displayed in the “Response Profile” table, are assigned to response levels in ascending sorted order. That is, the lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so on. For example, if your response variable Y takes values in $\{1, \ldots, D+1\}$, then the functions of the response probabilities modeled with the cumulative model are

$$\text{logit}(\Pr(Y \leq i|\mathbf{x})), i = 1, \ldots, D$$

and for the generalized logit model they are

$$\log\left(\frac{\Pr(Y = i|\mathbf{x})}{\Pr(Y = D+1|\mathbf{x})}\right), i = 1, \ldots, D$$

where the highest Ordered Value $Y = D+1$ is the reference level. You can change these default functions by specifying the EVENT=, the REF=, the DESCENDING, or the ORDER= response variable options in the MODEL statement.

For binary response data with event and nonevent categories, the procedure models the function

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

where π is the probability of the response level assigned Ordered Value 1 in the “Response Profiles” table. Since

$$\text{logit}(\pi) = -\text{logit}(1-\pi)$$

the effect of reversing the order of the two response levels is to change the signs of α and $\boldsymbol{\beta}$ in the model $\text{logit}(\pi) = \alpha + \boldsymbol{\beta}'\boldsymbol{x}$.

If your event category has a higher Ordered Value than the nonevent category, the procedure models the nonevent probability. You can use response variable options to model the event probability. For example, suppose the binary response variable Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. By default, the procedure assigns Ordered Value 1 to response level Y=0, and Ordered Value 2 to response level Y=1. Therefore, the procedure models the probability of the nonevent (Ordered Value=1) category. To model the event probability, you can do the following:

- explicitly state which response level is to be modeled using the response variable option EVENT= in the MODEL statement,

```
      model Y(event='1') = Exposure;
```

- specify the response variable option DESCENDING in the MODEL statement,

```
      model Y(descending)=Exposure;
```

- specify the response variable option REF= in the MODEL statement as the nonevent category for the response variable. This option is most useful when you are fitting a generalized logit model.

```
      model Y(ref='0') = Exposure;
```

- assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. For this example, Y=1 is assigned formatted value ‘event’ and Y=0 is assigned formatted value ‘nonevent’. Since ORDER=FORMATTED by default, Ordered Value 1 is assigned to response level Y=1 so the procedure models the event.

```
   proc format;
      value Disease 1='event' 0='nonevent';
   run;
   proc surveylogistic;
      format Y Disease.;
      model Y=Exposure;
   run;
```

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of -1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects using the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects using the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, $\ldots$ according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	**AOEFF1**	**AOEFF2**	**AOEFF3**
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	**AOORD1**	**AOORD2**	**AOORD3**
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	**AOPOLY1**	**AOPOLY2**	**AOPOLY5**
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	**AOREF1**	**AOREF2**	**AOREF3**
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Link Functions and the Corresponding Distributions

Four link functions are available in the SURVEYLOGISTIC procedure. The logit function is the default. To specify a different link function, use the LINK= option in the MODEL statement. The link functions and the corresponding distributions are as follows:

- The logit function

$$g(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

is the inverse of the cumulative logistic distribution function, which is

$$F(x) = \frac{1}{1+e^{-x}}$$

- The probit (or normit) function

$$g(\pi) = \Phi^{-1}(\pi)$$

is the inverse of the cumulative standard normal distribution function, which is

$$F(x) = \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}z^2} dz$$

Traditionally, the probit function includes an additive constant 5, but throughout PROC SURVEYLOGISTIC, the terms probit and normit are used interchangeably, defined as $g(p)$ above.

- The complementary log-log function

$$g(\pi) = \log(-\log(1-\pi))$$

is the inverse of the cumulative extreme-value function (also called the Gompertz distribution), which is

$$F(x) = 1 - e^{-e^{x}}$$

- The generalized logit function extends the binary logit link to a vector of levels $(\pi_1, \ldots, \pi_{k+1})$ by contrasting each level with a fixed level

$$g(\pi_i) = \log\left(\frac{\pi_i}{\pi_{k+1}}\right) \quad i = 1, \ldots, k$$

The variances of the normal, logistic, and extreme-value distributions are not the same. Their respective means and variances are

Distribution	**Mean**	**Variance**
Normal	0	1
Logistic	0	$\pi^2/3$
Extreme-value	$-\gamma$	$\pi^2/6$

where γ is the Euler constant. In comparing parameter estimates using different link functions, you need to take into account the different scalings of the corresponding distributions and, for the complementary log-log function, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates using the logit link function should be about $\pi/\sqrt{3} \approx 1.8$ larger than the estimates from the probit link function.

Model Fitting

Determining Observations for Likelihood Contributions

If you use *events/trials* syntax, each observation is split into two observations. One has the response value 1 with a frequency equal to the frequency of the original observation (which is 1 if the FREQ statement is not used) times the value of the *events* variable. The other observation has the response value 2 and a frequency equal to the frequency of the original observation times the value of (*trials* − *events*). These two observations have the same explanatory variable values and the same FREQ and WEIGHT values as the original observation.

For either *single-trial* or *events/trials* syntax, let j index all observations. In other words, for *single-trial* syntax, j indexes the actual observations. And, for *events/trials* syntax, j indexes the observations after splitting (as described previously). If your data set has 30 observations and you use *single-trial* syntax, j has values from 1 to 30; if you use *events/trials* syntax, j has values from 1 to 60.

Suppose the response variable in a cumulative response model can take on the ordered values $1, \ldots, k, k{+}1$ where k is an integer ≥ 1. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector (row vectors) $\mathbf{x}_j$ is given by

$$L_j = \left\{ \begin{array}{ll} F(\alpha_1 + \mathbf{x}_j\boldsymbol{\beta}) & y_j = 1 \\ F(\alpha_i + \mathbf{x}_j\boldsymbol{\beta}) - F(\alpha_{i-1} + \mathbf{x}_j\boldsymbol{\beta}) & 1 < y_j = i \leq k \\ 1 - F(\alpha_k + \mathbf{x}_j\boldsymbol{\beta}) & y_j = k+1 \end{array} \right.$$

where $F(.)$ is the logistic, normal, or extreme-value distribution function, $\alpha_1, \ldots, \alpha_k$ are ordered intercept parameters, and $\boldsymbol{\beta}$ is the slope parameter vector.

For the generalized logit model, letting the $k+1$st level be the reference level, the intercepts $\alpha_1, \ldots, \alpha_k$ are unordered and the slope vector $\boldsymbol{\beta}_i$ varies with each logit. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector $\mathbf{x}_j$ (row vectors) is given by

$$\begin{array}{rcl} L_j & = & \Pr(Y = y_j | \mathbf{x}_j) \\ & = & \left\{ \begin{array}{ll} \dfrac{e^{\alpha_i + \mathbf{x}_j\boldsymbol{\beta}_i}}{1 + \sum_{i=1}^k e^{\alpha_i + \mathbf{x}_j\boldsymbol{\beta}_i}} & 1 \leq y_j = i \leq k \\ & \\ \dfrac{1}{1 + \sum_{i=1}^k e^{\alpha_i + \mathbf{x}_j\boldsymbol{\beta}_i}} & y_j = k+1 \end{array} \right. \end{array}$$

Iterative Algorithms for Model-Fitting

Two iterative maximum likelihood algorithms are available in PROC SURVEYLOGISTIC to obtain the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ of the model parameter $\boldsymbol{\theta}$. The default is the Fisher-scoring method, which is equivalent to fitting by iteratively reweighted least squares. The alternative algorithm is the

Newton-Raphson method. Both algorithms give the same parameter estimates; The covariance matrix of $\widehat{\boldsymbol{\theta}}$ is estimated in the section "Variance Estimation for Sample Survey Data" on page 4282. For a generalized logit model, only the Newton-Raphson technique is available. You can use the TECHNIQUE= option to select a fitting algorithm.

Iteratively Reweighted Least-Squares Algorithm (Fisher Scoring)

Let Y be the response variable which takes values $1, \ldots, k, k+1$ $(k \geq 1)$. Let j index all observations and Y_j be the value of response for the jth observation. Consider the multinomial variable $\mathbf{Z}_j = (Z_{1j}, \ldots, Z_{kj})'$ such that

$$Z_{ij} = \begin{cases} 1 & \text{if } Y_j = i \\ 0 & \text{otherwise} \end{cases}$$

and $Z_{(k+1)j} = 1 - \sum_{i=1}^{k} Z_{ij}$. With π_{ij} denoting the probability that the *j*th observation has response value *i*, the expected value of $\mathbf{Z}_j$ is $\boldsymbol{\pi}_j = (\pi_{1j}, \ldots, \pi_{kj})'$, and $\pi_{(k+1)j} = 1 - \sum_{i=1}^{k} \pi_{ij}$. The covariance matrix of $\mathbf{Z}_j$ is $\mathbf{V}_j$, which is the covariance matrix of a multinomial random variable for one trial with parameter vector $\boldsymbol{\pi}_j$. Let $\boldsymbol{\theta}$ be the vector of regression parameters; for example, $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}')'$ for cumulative logit model. Let $\mathbf{D}_j$ be the matrix of partial derivatives of $\boldsymbol{\pi}_j$ with respect to $\boldsymbol{\theta}$. The estimating equation for the regression parameters is

$$\sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z_j} - \boldsymbol{\pi}_\mathbf{j}) = \mathbf{0}$$

where $\mathbf{W}_j = w_j f_j \mathbf{V}_j^{-1}$, w_j and f_j are the WEIGHT and FREQ values of the jth observation.

With a starting value of $\boldsymbol{\theta}^{(0)}$, the maximum likelihood estimate of $\boldsymbol{\theta}$ is obtained iteratively as

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + (\sum_j \mathbf{D}_j' \mathbf{W_j} \mathbf{D}_j)^{-1} \sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z_j} - \boldsymbol{\pi}_\mathbf{j})$$

where $\mathbf{D}_j$, $\mathbf{W}_j$, and $\boldsymbol{\pi}_j$ are evaluated at the ith iteration $\boldsymbol{\theta}^{(i)}$. The expression after the plus sign is the step size. If the log-likelihood evaluated at $\boldsymbol{\theta}^{(i+1)}$ is less than that evaluated at $\boldsymbol{\theta}^{(i)}$, then $\boldsymbol{\theta}^{(i+1)}$ is recomputed by step-halving or ridging. The iterative scheme continues until convergence is obtained, that is, until $\boldsymbol{\theta}^{(i+1)}$ is sufficiently close to $\boldsymbol{\theta}^{(i)}$. Then the maximum likelihood estimate of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i+1)}$.

By default, starting values are zero for the slope parameters, and for the intercept parameters, starting values are the observed cumulative logits (that is, logits of the observed cumulative proportions of response). Alternatively, the starting values may be specified with the INEST= option.

Newton-Raphson Algorithm

Let

$$\begin{aligned}\mathbf{g} &= \sum_j w_j f_j \frac{\partial l_j}{\partial \boldsymbol{\theta}} \\ \mathbf{H} &= \sum_j -w_j f_j \frac{\partial^2 l_j}{\partial \boldsymbol{\theta}^2}\end{aligned}$$

be the gradient vector and the Hessian matrix, where $l_j = \log L_j$ is the log likelihood for the jth observation. With a starting value of $\boldsymbol{\theta}^{(0)}$, the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ is obtained iteratively until convergence is obtained:

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \mathbf{H}^{-1}\mathbf{g}$$

where $\mathbf{H}$ and $\mathbf{g}$ are evaluated at the ith iteration $\boldsymbol{\theta}^{(i)}$. If the log likelihood evaluated at $\boldsymbol{\theta}^{(i+1)}$ is less than that evaluated at $\boldsymbol{\theta}^{(i}$, then $\boldsymbol{\theta}^{(i+1)}$ is recomputed by step-halving or ridging. The iterative scheme continues until convergence is obtained, that is, until $\boldsymbol{\theta}^{(i+1)}$ is sufficiently close to $\boldsymbol{\theta}^{(i)}$. Then the maximum likelihood estimate of $\boldsymbol{\theta}$ is $\widehat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i+1)}$.

Convergence Criteria

Four convergence criteria are allowed, namely, ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E$-$8.

Existence of Maximum Likelihood Estimates

The likelihood equation for a logistic regression model does not always have a finite solution. Sometimes there is a nonunique maximum on the boundary of the parameter space, at infinity. The existence, finiteness, and uniqueness of maximum likelihood estimates for the logistic regression model depend on the patterns of data points in the observation space (Albert and Anderson 1984; Santner and Duffy 1986).

Consider a binary response model. Let Y_j be the response of the ith subject and let $\boldsymbol{x}_j$ be the vector of explanatory variables (including the constant 1 associated with the intercept). There are three mutually exclusive and exhaustive types of data configurations: complete separation, quasi-complete separation, and overlap.

Complete Separation

There is a complete separation of data points if there exists a vector $\mathbf{b}$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \mathbf{b}'\boldsymbol{x}_j > 0 & Y_j = 1 \\ \mathbf{b}'\boldsymbol{x}_j < 0 & Y_j = 2 \end{cases}$$

This configuration gives nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the log likelihood diminishes to zero, and the dispersion matrix becomes unbounded.

Quasi-Complete Separation

The data are not completely separable but there is a vector $\mathbf{b}$ such that

$$\begin{cases} \mathbf{b}'\boldsymbol{x}_j \geq 0 & Y_j = 1 \\ \mathbf{b}'\boldsymbol{x}_j \leq 0 & Y_j = 2 \end{cases}$$

and equality holds for at least one subject in each response group. This configuration also yields nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the dispersion matrix becomes unbounded and the log likelihood diminishes to a nonzero constant.

Overlap

If neither complete nor quasi-complete separation exists in the sample points, there is an overlap of sample points. In this configuration, the maximum likelihood estimates exist and are unique.

Complete separation and quasi-complete separation are problems typically encountered with small data sets. Although complete separation can occur with any type of data, quasi-complete separation is not likely with truly continuous explanatory variables.

The SURVEYLOGISTIC procedure uses a simple empirical approach to recognize the data configurations that lead to infinite parameter estimates. The basis of this approach is that any convergence method of maximizing the log likelihood must yield a solution giving complete separation, if such a solution exists. In maximizing the log likelihood, there is no checking for complete or quasi-complete separation if convergence is attained in eight or fewer iterations. Subsequent to the eighth iteration, the probability of the observed response is computed for each observation. If the probability of the observed response is one for all observations, there is a complete separation of data points and the iteration process is stopped. If the complete separation of data has not been determined and an observation is identified to have an extremely large probability ($\geq$0.95) of the observed response, there are two possible situations. First, there is overlap in the data set, and the observation is an atypical observation of its own group. The iterative process, if allowed to continue, will stop when a maximum is reached. Second, there is quasi-complete separation in the data set, and the asymptotic dispersion matrix is unbounded. If any of the diagonal elements of the dispersion matrix for the standardized observations vectors (all explanatory variables standardized to zero mean and unit variance) exceeds 5,000, quasi-complete separation is declared and the iterative process is stopped. If either complete separation or quasi-complete separation is detected, a warning message is displayed in the procedure output.

Checking for quasi-complete separation is less foolproof than checking for complete separation. The NOCHECK option in the MODEL statement turns off the process of checking for infinite parameter estimates. In cases of complete or quasi-complete separation, turning off the checking process typically results in the procedure failing to converge.

Model Fitting Statistics

Suppose the model contains s explanatory effects. For the jth observation, let $\hat{\pi}_j$ be the estimated probability of the observed response. The three criteria displayed by the SURVEYLOGISTIC procedure are calculated as follows:

- -2 Log Likelihood:

  $$-2 \text{ Log L} = -2 \sum_j w_j f_j \log(\hat{\pi}_j)$$

  where w_j and f_j are the weight and frequency values of the jth observation. For binary response models using *events/trials* syntax, this is equivalent to

  $$-2 \text{ Log L} = -2 \sum_j w_j f_j \{r_j \log(\hat{\pi}_j) + (n_j - r_j) \log(1 - \hat{\pi}_j)\}$$

  where r_j is the number of events, n_j is the number of trials, and $\hat{\pi}_j$ is the estimated event probability.

- Akaike Information Criterion:

  $$\text{AIC} = -2 \text{ Log L} + 2p$$

  where p is the number of parameters in the model. For cumulative response models, $p = k + s$ where k is the total number of response levels minus one, and s is the number of explanatory effects. For the generalized logit model, $p = k(s + 1)$.

- Schwarz Criterion:

  $$\text{SC} = -2 \text{ Log L} + p \log(\sum_j f_j)$$

  where p is as defined previously.

The -2 Log Likelihood statistic has a chi-square distribution under the null hypothesis (that all the explanatory effects in the model are zero) and the procedure produces a p-value for this statistic. The AIC and SC statistics give two different ways of adjusting the -2 Log Likelihood statistic for the number of terms in the model and the number of observations used.

Generalized Coefficient of Determination

Cox and Snell (1989, pp. 208–209) propose the following generalization of the coefficient of determination to a more general linear model:

$$R^2 = 1 - \left\{ \frac{L(\mathbf{0})}{L(\widehat{\boldsymbol{\theta}})} \right\}^{\frac{2}{n}}$$

where $L(\mathbf{0})$ is the likelihood of the intercept-only model, $L(\widehat{\boldsymbol{\theta}})$ is the likelihood of the specified model, and n is the sample size. The quantity R^2 achieves a maximum of less than 1 for discrete models, where the maximum is given by

$$R^2_{\max} = 1 - \{L(\mathbf{0})\}^{\frac{2}{n}}$$

Nagelkerke (1991) proposes the following adjusted coefficient, which can achieve a maximum value of 1:

$$\tilde{R}^2 = \frac{R^2}{R^2_{\max}}$$

Properties and interpretation of R^2 and $\tilde{R}^2$ are provided in Nagelkerke (1991). In the “Testing Global Null Hypothesis: BETA=0” table, R^2 is labeled as “RSquare” and $\tilde{R}^2$ is labeled as “Max-rescaled RSquare.” Use the RSQUARE option to request R^2 and $\tilde{R}^2$.

INEST= Data Set

You can specify starting values for the iterative algorithm in the INEST= data set.

The INEST= data set contains one observation for each BY group. The INEST= data set must contain the intercept variables (named Intercept for binary response models and Intercept, Intercept2, Intercept3, and so forth, for ordinal response models) and all explanatory variables in the MODEL statement. If BY processing is used, the INEST= data set should also include the BY variables, and there must be one observation for each BY group. If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value ’PARMS’ are used as starting values.

Survey Design Information

Specification of Population Totals and Sampling Rates

Variance estimates in survey samples involve a finite population correction (*fpc*) for sampling without replacement. For small sampling fractions or sampling with replacement, it is appropriate to ignore this correction (Cochran 1977; Kish 1965), and PROC SURVEYLOGISTIC does so by default. If your analysis requires an *fpc*, specify the sampling fraction using either the RATE= option or the TOTAL= option on the PROC SURVEYLOGISTIC statement. If you do not specify one of these options, the procedure does not use the *fpc* when computing variance estimates.

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of *primary sampling units* (PSUs) in the sample to the total number of PSUs in the study population. If you are using the TOTAL= option for a multistage design, you should specify the total number of PSUs in the study population. See the section "Primary Sampling Units (PSUs)" on page 4281 for more details.

For a nonstratified sample design, or for a stratified sample design for which all the strata have the same sampling rate or population total, you should specify the rate or total as the value of the RATE=*value* option or the TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in the strata, then you should use the RATE=*SAS-data-set* option or the TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement, as well as all the variables in the BY statement if any of these statements are specified. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. If you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be positive numbers. You can specify *value* as a number between 0 and 1; or you can specify *value* in percentage form as a number between 1 and 100, in which case PROC SURVEYLOGISTIC will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, this *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or *primary sampling units* (PSUs), in your sample design, the procedure estimates the variance based on the variation among PSUs. Use the CLUSTER statement to identify the first-stage clusters in your design. PROC SURVEYLOGISTIC assumes that each cluster represents a PSU in the sample and that each observation is an element of a PSU. If you do not specify a CLUSTER statement, the procedure treats each observation as a PSU.

Variance Estimation for Sample Survey Data

Due to the variability of characteristics among items in the population, researchers apply scientific sample designs in the sample selection process to reduce the risk of a distorted view of the population, and they make inferences about the population based on the information from the sample survey data. In order to make statistically valid inferences for the population, they must incorporate the sample design in the data analysis.

The SURVEYLOGISTIC procedure fits linear logistic regression models for discrete response survey data using the maximum likelihood method. In the variance estimation, the procedure incorporates complex survey sample designs, including designs with stratification, clustering, and unequal weighting.

The procedure uses the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself. See Binder (1981, 1983), Roberts, Rao, and Kumar (1987), Skinner, Holt, and Smith (1989), Morel (1989), and Lehtonen and Pahkinen (1995) for papers that describe logistic regression for sample survey data. When there are clusters, or primary sampling units (PSUs), in the sample design, the procedure estimates variance from the variation among PSUs. When the design is stratified, the procedure pools stratum variance estimates to compute the overall variance estimate. For *t* tests of the estimates, the degrees of freedom equals the number of clusters minus the number of strata in the sample design. Statistical analyses, such as hypothesis tests and confident limits, will depend on these variance estimates.

Notation

Let Y be the response variable with categories $1, 2, \ldots, D, D+1$. The p covariates are denoted by a p-dimension row vector $\mathbf{x}$.

For a stratified clustered sample design, each observation is represented by a row vector,

$$(w_{hij}, \mathbf{y}'_{hij}, y_{hij(D+1)}, \mathbf{x}_{hij})$$

where

- $h = 1, 2, \ldots, H$ is the stratum number with a total of H strata
- $i = 1, 2, \ldots, n_h$ is the cluster number within stratum h, with a total of n_h clusters
- $j = 1, 2, \ldots, m_{hi}$ is the unit number within cluster i of stratum h, with a total of m_{hi} units
- w_{hij} denotes the sampling weight
- $\mathbf{y}_{hij}$ is a D-dimensional column vector whose elements are indicator variables for the first D categories for variable Y. If the response of the jth member of the ith cluster in stratum h falls in category d, the dth row of the vector is one, and the remaining elements of the vector are zero, where $d = 1, 2, \ldots, D$

- $y_{hij(D+1)}$ is the indicator variable for the $(D+1)$ category of variable Y
- $\mathbf{x}_{hij}$ denotes the k-dimensional row vector of explanatory variables for the jth member of the ith cluster in stratum h. If there is an intercept, then $x_{hij1} \equiv 1$.
- $\tilde{n} = \sum_{h=1}^{H} n_h$ is the total number of clusters in the entire sample
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total sample size

The following notations are also used in the following sections:

- f_h denotes the sampling rate for stratum h
- $\boldsymbol{\pi}_{hij}$ is the expected vector of the response variable

$$\begin{aligned}
\boldsymbol{\pi}_{hij} &= E(\mathbf{y}_{hij}|\mathbf{x}_{hij}) \\
&= (\pi_{hij1}, \pi_{hij2}, \ldots, \pi_{hijD})' \\
\pi_{hij(D+1)} &= E(y_{hij(D+1)}|\mathbf{x}_{hij})
\end{aligned}$$

Note that $\pi_{hij(D+1)} = 1 - \mathbf{1}'\boldsymbol{\pi}_{hij}$ where $\mathbf{1}$ is a D-dimensional column vector whose elements are 1.

Likelihood Function

Let $\mathbf{f}(\cdot)$ be a link function such that

$$\boldsymbol{\pi} = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is a p-dimensional column vector for regression coefficients. The pseudo log likelihood is

$$l(\boldsymbol{\theta}) = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij} \left((\log(\boldsymbol{\pi}_{hij}))'\mathbf{y}_{hij} + \log(\pi_{hij(D+1)})y_{hij(D+1)} \right)$$

Denote the maximum likelihood estimator as $\hat{\boldsymbol{\theta}}$, which is a solution to the estimating equations:

$$\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij} \mathbf{D}'_{hij} \left(\text{diag}(\boldsymbol{\pi}_{hij}) - \boldsymbol{\pi}_{hij}\boldsymbol{\pi}'_{hij} \right)^{-1} (\mathbf{y}_{hij} - \boldsymbol{\pi}_{hij}) = \mathbf{0}$$

where $\mathbf{D}_{hij}$ is the matrix of partial derivatives of the link function f with respect to $\boldsymbol{\theta}$.

To obtain the maximum likelihood estimator $\hat{\boldsymbol{\theta}}$, the procedure uses iterations with a starting value $\boldsymbol{\theta}^{(0)}$ for $\boldsymbol{\theta}$. See the section "Iterative Algorithms for Model-Fitting" on page 4275 for detail.

Generalized Logistic Model

Formulation of the generalized logit models for nominal response variables can be found in Agresti (1990). Without loss of generality, let the last category, $D+1$, be the reference category for the response variable Y. The link function for the generalized logistic model is defined as

$$\pi_{hijd} = \frac{e^{\mathbf{x}_{hij}\boldsymbol{\beta}_d}}{1 + \sum_{r=1}^{D} e^{\mathbf{x}_{hij}\boldsymbol{\beta}_r}}$$

and the model parameters are:

$$\begin{aligned}\boldsymbol{\beta}_d &= (\beta_{d1}, \beta_{d2}, \ldots, \beta_{dk})' \\ \boldsymbol{\theta} &= (\boldsymbol{\beta}_1', \boldsymbol{\beta}_2', \ldots, \boldsymbol{\beta}_D')'\end{aligned}$$

for $d = 1, 2, \ldots, D$.

Cumulative Logit Model

Details of the cumulative logit model (or proportional odds model) can be found in McCullagh and Nelder (1989). Denote the cumulative sum of the expected proportions for the first d categories of variable Y by

$$F_{hijd} = \sum_{r=1}^{d} \pi_{hijr}$$

for $d = 1, 2, \ldots, D$. Then the link function for the proportional odds model is

$$\log\left(\frac{F_{hijd}}{1 - F_{hijd}}\right) = \alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta}$$

with the model parameters:

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \;\; \alpha_1 < \alpha_2 < \cdots < \alpha_D \\ \boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'\end{aligned}$$

Complementary log-log Model

Use the notations in the previous section, the link function for the complementary log-log is

$$\log(-\log(1 - F_{hijd})) = \alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta}$$

with the model parameters:

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \;\; \alpha_1 < \alpha_2 < \cdots < \alpha_D \\ \boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'\end{aligned}$$

Probit Model

Another commonly used model for ordinal responses is the probit model with the link function

$$F_{hijd} = \Phi(\alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta})$$

where

$$\Phi(z_0) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{z_0} e^{-\frac{1}{2}z^2} dz$$

is the cumulative distribution function of the standard normal distribution. The model parameters are:

$$\begin{aligned}
\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\
\boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \quad \alpha_1 < \alpha_2 < \cdots < \alpha_D \\
\boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'
\end{aligned}$$

Estimated Variances

Using Taylor approximation, the estimated covariance matrix of $\hat{\boldsymbol{\theta}}$ is

$$\widehat{V}(\hat{\boldsymbol{\theta}}) = \widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\widehat{\mathbf{Q}}^{-1}$$

where

$$\begin{aligned}
\widehat{\mathbf{Q}} &= \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\widehat{\mathbf{D}}_{hij}\left(\mathrm{diag}(\hat{\boldsymbol{\pi}}_{hij}) - \hat{\boldsymbol{\pi}}_{hij}\hat{\boldsymbol{\pi}}_{hij}{}'\right)^{-1}\widehat{\mathbf{D}}'_{hij} \\
\widehat{\mathbf{G}} &= \frac{n-1}{n-p}\sum_{h=1}^{H}\frac{n_h(1-f_h)}{n_h-1}\sum_{i=1}^{n_h}(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})' \\
\mathbf{e}_{hi\cdot} &= \sum_{j=1}^{m_{hi}} w_{hij}\widehat{\mathbf{D}}_{hij}\left(\mathrm{diag}(\hat{\boldsymbol{\pi}}_{hij}) - \hat{\boldsymbol{\pi}}_{hij}\hat{\boldsymbol{\pi}}_{hij}{}'\right)^{-1}(\mathbf{y}_{hij} - \hat{\boldsymbol{\pi}}_{hij}) \\
\bar{\mathbf{e}}_{h\cdot\cdot} &= \frac{1}{n_h}\sum_{i=1}^{n_h}\mathbf{e}_{hi\cdot}
\end{aligned}$$

If you use the Newton-Raphson algorithm by using the TECHNIQUE=NEWTON option in the MODEL statement, the matrix $\widehat{\mathbf{Q}}$ is replaced by the negative (expected) Hessian matrix,

The matrices of partial derivatives $\widehat{\mathbf{D}}_{hij}$ and the response probabilities $\hat{\boldsymbol{\pi}}_{hij}$ are evaluated at $\hat{\boldsymbol{\theta}}$.

Adjustments to the Variance Estimation

The factor $(n-1)/(n-p)$ in the computation of the matrix $\widehat{\mathbf{G}}$ should reduce the small sample bias associated with using the estimated function to calculate deviations (Morel 1989; Hidiroglou, Fuller, and Hickman 1980). For simple random sampling, this factor contributes to the degrees of freedom correction applied to the residual mean square for ordinary least squares in which p parameter are estimated. By default, the procedure will use this adjustment in variance estimation. It is equivalent to specify the VADJUST=DF option in the MODEL statement. If you do not wish to use this multiplier in the variance estimation, you can specify the VADJUST=NONE option in the MODEL statement to suppress this factor.

In addition, you can specify the VADJUST=MOREL option to compute a further adjustment to the variance estimator for the regression coefficients $\hat{\boldsymbol{\theta}}$, introduced by Morel (1989):

$$\widehat{V}(\hat{\boldsymbol{\theta}}) = \widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\widehat{\mathbf{Q}}^{-1} + \kappa\lambda\widehat{\mathbf{Q}}^{-1}$$

where for given nonnegative constants δ and ϕ,

$$\begin{aligned} \kappa &= \max\left(\delta,\ p^{-1}\mathrm{tr}\left(\widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\right)\right) \\ \lambda &= \min\left(\phi,\ \frac{p}{\tilde{n}-p}\right) \end{aligned}$$

The adjustment $\kappa\lambda\widehat{\mathbf{Q}}^{-1}$ will

- reduce the small sample bias reflected in inflated Type 1 error rates
- guarantee a positive definite estimated covariance matrix provided that $\widehat{\mathbf{Q}}^{-1}$ exists
- be close to zero when the sample size becomes large

In this adjustment, κ is an estimate of the design effect, which has been bounded below by the positive constant δ. You can use DEFFBOUND=δ in the VADJUST=MOREL option in the MODEL statement to specify this lower bound; by default, the procedure uses $\delta = 1$. The factor λ converges to zero when the sample size becomes large, and λ has an upper bound ϕ. You can use ADJBOUND=ϕ in the VADJUST=MOREL option in the MODEL statement to specify this upper bound; by default, the procedure uses $\phi = 0.5$.

Hypothesis Testing and Estimation

Score Statistics and Tests

To understand the general form of the score statistics, let $\mathbf{g}(\boldsymbol{\theta})$ be the vector of first partial derivatives of the log likelihood with respect to the parameter vector $\boldsymbol{\theta}$, and let $\mathbf{g}(\boldsymbol{\theta})$ be the matrix of second partial derivatives of the log likelihood with respect to $\boldsymbol{\theta}$. That is, $\mathbf{g}(\boldsymbol{\theta})$ is the gradient vector, and $\mathbf{H}(\boldsymbol{\theta})$ is the Hessian matrix. Let $\mathbf{I}(\boldsymbol{\theta})$ be either $-\mathbf{H}(\boldsymbol{\theta})$ or the expected value of $-\mathbf{H}(\boldsymbol{\theta})$. Consider a null hypothesis H_0. Let $\hat{\boldsymbol{\theta}}_0$ be the MLE of $\boldsymbol{\theta}$ under H_0. The chi-square score statistic for testing H_0 is defined by

$$\mathbf{g}'(\hat{\boldsymbol{\theta}}_0)\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_0)\mathbf{g}(\hat{\boldsymbol{\theta}}_0)$$

and it has an asymptotic χ^2 distribution with r degrees of freedom under H_0, where r is the number of restrictions imposed on $\boldsymbol{\theta}$ by H_0.

Testing the Parallel Lines Assumption

For an ordinal response, PROC SURVEYLOGISTIC performs a test of the parallel lines assumption. In the displayed output, this test is labeled “Score Test for the Equal Slopes Assumption” when the LINK= option is NORMIT or CLOGLOG. When LINK=LOGIT, the test is labeled as “Score Test for the Proportional Odds Assumption” in the output. This section describes the methods used to calculate the test.

For this test the number of response levels, $D+1$, is assumed to be strictly greater than 2. Let Y be the response variable taking values $1,\ldots,D,D+1$. Suppose there are k explanatory variables. Consider the general cumulative model without making the parallel lines assumption

$$g(\Pr(Y\le d\mid \mathbf{x})) = (1,\mathbf{x})\boldsymbol{\theta}_d, \quad 1\le d\le D$$

where $g(\cdot)$ is the link function, and $\boldsymbol{\theta}_d=(\alpha_d,\beta_{d1},\ldots,\beta_{dk})'$ is a vector of unknown parameters consisting of an intercept α_d and k slope parameters $\beta_{k1},\ldots,\beta_{kd}$. The parameter vector for this general cumulative model is

$$\boldsymbol{\theta}=(\boldsymbol{\theta}_1',\ldots,\boldsymbol{\theta}_D')'$$

Under the null hypothesis of parallelism $H_0{:}\,\beta_{1i}=\beta_{2i}=\cdots=\beta_{Di}, 1\le i\le k$, there is a single common slope parameter for each of the s explanatory variables. Let $\beta_1,\ldots,\beta_k$ be the common slope parameters. Let $\hat{\alpha}_1,\ldots,\hat{\alpha}_D$ and $\hat{\beta}_1,\ldots,\hat{\beta}_D$ be the MLEs of the intercept parameters and the common slope parameters. Then, under H_0, the MLE of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}}_0=(\hat{\boldsymbol{\theta}}_1',\ldots,\hat{\boldsymbol{\theta}}_D')' \quad \text{with} \quad \hat{\boldsymbol{\theta}}_d=(\hat{\alpha}_d,\hat{\beta}_1,\ldots,\hat{\beta}_k)' \quad 1\le d\le D$$

and the chi-squared score statistic $\mathbf{g}'(\hat{\boldsymbol{\theta}}_0)\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}}_0)\mathbf{g}(\hat{\boldsymbol{\theta}}_0)$ has an asymptotic chi-square distribution with $k(D-1)$ degrees of freedom. This tests the parallel lines assumption by testing the equality of separate slope parameters simultaneously for all explanatory variables.

Wald Confidence Intervals for Parameters

Wald confidence intervals are sometimes called the normal confidence intervals. They are based on the asymptotic normality of the parameter estimators. The $100(1-\alpha)\%$ Wald confidence interval for θ_j is given by

$$\widehat{\theta}_j \pm z_{1-\alpha/2}\widehat{\sigma}_j$$

where z_p is the $100p$th percentile of the standard normal distribution, $\widehat{\theta}_j$ is the maximum likelihood estimate of θ_j, and $\widehat{\sigma}_j$ is the standard error estimate of $\widehat{\theta}_j$ in the section "Variance Estimation for Sample Survey Data" on page 4282.

Testing Linear Hypotheses about the Regression Coefficients

Linear hypotheses for $\boldsymbol{\theta}$ are expressed in matrix form as

$$H_0\colon \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\mathbf{c}$ is a vector of constants. The vector of regression coefficients $\boldsymbol{\theta}$ includes slope parameters as well as intercept parameters. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = (\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})'[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\theta}} - \mathbf{c})$$

where $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\theta}})$ is the estimated covariance matrix in the section "Variance Estimation for Sample Survey Data" on page 4282. Under H_0, χ^2_W has an asymptotic chi-square distribution with r degrees of freedom, where r is the rank of $\mathbf{L}$.

Odds Ratio Estimation

Consider a dichotomous response variable with outcomes *event* and *nonevent*. Consider a dichotomous risk factor variable X that takes the value 1 if the risk factor is present and 0 if the risk factor is absent. According to the logistic model, the log odds function, $g(X)$, is given by

$$g(X) \equiv \log\left(\frac{\Pr(\textit{event} \mid X)}{\Pr(\textit{nonevent} \mid X)}\right) = \beta_0 + \beta_1 X$$

The odds ratio ψ is defined as the ratio of the odds for those with the risk factor ($X = 1$) to the odds for those without the risk factor ($X = 0$). The log of the odds ratio is given by

$$\log(\psi) \equiv \log(\psi(X = 1, X = 0)) = g(X = 1) - g(X = 0) = \beta_1$$

The parameter, β_1, associated with X represents the change in the log odds from $X = 0$ to $X = 1$. So, the odds ratio is obtained by simply exponentiating the value of the parameter associated with the risk factor. The odds ratio indicates how the odds of *event* change as you change X from 0 to 1. For instance, $\psi = 2$ means that the odds of an event when $X = 1$ are twice the odds of an event when $X = 0$.

Suppose the values of the dichotomous risk factor are coded as constants a and b instead of 0 and 1. The odds when $X = a$ become $\exp(\beta_0 + a\beta_1)$, and the odds when $X = b$ become $\exp(\beta_0 + b\beta_1)$. The odds ratio corresponding to an increase in X from a to b is

$$\psi = \exp[(b-a)\beta_1] = [\exp(\beta_1)]^{b-a} \equiv [\exp(\beta_1)]^c$$

Note that for any a and b such that $c = b - a = 1, \psi = \exp(\beta_1)$. So the odds ratio can be interpreted as the change in the odds for any increase of one unit in the corresponding risk factor. However, the change in odds for some amount other than one unit is often of greater interest. For example, a change of one pound in body weight may be too small to be considered important, while a change of 10 pounds may be more meaningful. The odds ratio for a change in X from a to b is estimated by raising the odds ratio estimate for a unit change in X to the power of $c = b - a$ as shown previously.

For a polytomous risk factor, the computation of odds ratios depends on how the risk factor is parameterized. For illustration, suppose that Race is a risk factor with four categories: White, Black, Hispanic, and Other.

For the effect parameterization scheme (PARAM=EFFECT) with White as the reference group, the design variables for Race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	-1	-1	-1

The log odds for Black is

$$\begin{aligned} g(\text{Black}) &= \beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) \\ &= \beta_0 + \beta_1 \end{aligned}$$

The log odds for White is

$$\begin{aligned} g(\text{White}) &= \beta_0 + \beta_1(X_1 = -1) + \beta_2(X_2 = -1) + \beta_3(X_3 = -1)) \\ &= \beta_0 - \beta_1 - \beta_2 - \beta_3 \end{aligned}$$

Therefore, the log odds ratio of Black versus White becomes

$$\begin{aligned}\log(\psi(\text{Black}, \text{White})) &= g(\text{Black}) - g(\text{White}) \\ &= 2\beta_1 + \beta_2 + \beta_3\end{aligned}$$

For the reference cell parameterization scheme (PARAM=REF) with White as the reference cell, the design variables for race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	0	0	0

The log odds ratio of Black versus White is given by

$$\begin{aligned}&\log(\psi(\text{Black}, \text{White})) \\ &\quad= g(\text{Black}) - g(\text{White}) \\ &\quad= (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0)) + \beta_3(X_3 = 0)) - \\ &\qquad (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0)) \\ &\quad= \beta_1\end{aligned}$$

For the GLM parameterization scheme (PARAM=GLM), the design variables are as follows.

	Design Variables			
Race	X_1	X_2	X_3	X_4
Black	1	0	0	0
Hispanic	0	1	0	0
Other	0	0	1	0
White	0	0	0	1

The log odds ratio of Black versus White is

$$\begin{aligned}&\log(\psi(\text{Black}, \text{White})) \\ &\quad= g(\text{Black}) - g(\text{White}) \\ &\quad= (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 0)) - \\ &\qquad (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 1)) \\ &\quad= \beta_1 - \beta_4\end{aligned}$$

Consider the hypothetical example of heart disease among race in Hosmer and Lemeshow (2000, p. 51). The entries in the following contingency table represent counts.

	Race			
Disease Status	White	Black	Hispanic	Other
Present	5	20	15	10
Absent	20	10	10	10

The computation of odds ratio of Black versus White for various parameterization schemes is tabulated in the following table.

Odds Ratio of Heart Disease Comparing Black to White					
	Parameter Estimates				
PARAM	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	Odds Ratio Estimation
EFFECT	0.7651	0.4774	0.0719		$\exp(2 \times 0.7651 + 0.4774 + 0.0719) = 8$
REF	2.0794	1.7917	1.3863		$\exp(2.0794) = 8$
GLM	2.0794	1.7917	1.3863	0.0000	$\exp(2.0794) = 8$

Since the log odds ratio ($\log(\psi)$) is a linear function of the parameters, the Wald confidence interval for $\log(\psi)$ can be derived from the parameter estimates and the estimated covariance matrix. Confidence intervals for the odds ratios are obtained by exponentiating the corresponding confidence intervals for the log odd ratios. In the displayed output of PROC SURVEYLOGISTIC, the “Odds Ratio Estimates” table contains the odds ratio estimates and the corresponding 95% Wald confidence intervals computed using the covariance matrix in the section “Variance Estimation for Sample Survey Data” on page 4282. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.

To customize odds ratios for specific units of change for a continuous risk factor, you can use the UNITS statement to specify a list of relevant units for each explanatory variable in the model. Estimates of these customized odds ratios are given in a separate table. Let (L_j, U_j) be a confidence interval for $\log(\psi)$. The corresponding lower and upper confidence limits for the customized odds ratio $\exp(c\beta_j)$ are $\exp(cL_j)$ and $\exp(cU_j)$, respectively (for $c > 0$), or $\exp(cU_j)$ and $\exp(cL_j)$, respectively (for $c < 0$). You use the CLODDS= option to request the confidence intervals for the odds ratios.

For a generalized logit model, odds ratios are computed similarly, except D odds ratios are computed for each effect, corresponding to the D logits in the model.

Rank Correlation of Observed Responses and Predicted Probabilities

The predicted mean score of an observation is the sum of the Ordered Values (shown in the Response Profile table) minus one, weighted by the corresponding predicted probabilities for that observation; that is, the predicted means score= $\sum_{d=1}^{D+1}(d-1)\hat{\pi}_d$, where $D+1$ is the number of response levels and $\hat{\pi}_d$ is the predicted probability of the dth (ordered) response.

A pair of observations with different observed responses is said to be *concordant* if the observation with the lower ordered response value has a lower predicted mean score than the observation with the higher ordered response value. If the observation with the lower ordered response value has a higher predicted mean score than the observation with the higher ordered response value, then the pair is *discordant*. If the pair is neither concordant nor discordant, it is a *tie*. Enumeration of the total numbers of concordant and discordant pairs is carried out by categorizing the predicted mean score into intervals of length $D/500$ and accumulating the corresponding frequencies of observations.

Let N be the sum of observation frequencies in the data. Suppose there is a total of t pairs with different responses, n_c of them are concordant, n_d of them are discordant, and $t-n_c-n_d$ of them are tied. PROC SURVEYLOGISTIC computes the following four indices of rank correlation for assessing the predictive ability of a model:

$$\begin{aligned}
\textit{c} &= (n_c + 0.5(t - n_c - n_d))/t \\
\text{Somers' } \textit{D} &= (n_c - n_d)/t \\
\text{Goodman-Kruskal Gamma} &= (n_c - n_d)/(n_c + n_d) \\
\text{Kendall's Tau-}\textit{a} &= (n_c - n_d)/(0.5N(N-1))
\end{aligned}$$

Note that c also gives an estimate of the area under the receiver operating characteristic (ROC) curve when the response is binary (Hanley and McNeil 1982).

For binary responses, the predicted mean score is equal to the predicted probability for Ordered Value 2. As such, the preceding definition of concordance is consistent with the definition used in previous releases for the binary response model.

Output

Displayed Output

The displayed output of the SURVEYLOGISTIC procedure includes the following:

- name of the input Data Set
- name and label of the Response Variable if the *single-trial* syntax is used
- number of Response Levels
- name of the Events Variable if the *events/trials* syntax is used
- name of the Trials Variable if the *events/trials* syntax is used
- Number of Observations read from the input data set

- Number of Observations used in the analysis
- name of the Frequency Variable if the FREQ statement is specified
- Sum of Frequencies of all the observations read from the input data set
- Sum of Frequencies of all the observations used in the analysis
- name of the Weight Variable if the WEIGHT statement is specified
- Sum of Weights of all the observations read from the input data set
- Sum of Weights of all the observations used in the analysis
- name of the Offset Variable if the OFFSET= option is specified
- name(s) of the stratification variable(s) if the STRATA statement is specified
- total number of strata if the STRATA statement is specified
- name(s) of the cluster variable(s) if the CLUSTER statement is specified
- total number of clusters if the CLUSTER statement is specified
- Sum of Weights of all the observations used in the analysis
- Link Function
- variance adjustment method
- parameters used in the VADJUST=MOREL option if this option is specified
- "Response Profile" table, which gives, for each response level, the ordered value (an integer between one and the number of response levels, inclusive); the value of the response variable if the *single-trial* syntax is used or the values "EVENT" and "NO EVENT" if the *events/trials* syntax is used; the count or frequency; and the sum of weights if the WEIGHT statement is specified
- "Class Level Information" table, which gives the level and the design variables for each CLASS explanatory variable
- "Maximum Likelihood Iterative Phase" table, which gives the iteration number, the step size (in the scale of 1.0, .5, .25, and so on) or the ridge value, -2 log likelihood, and parameter estimates for each iteration. Also displayed are the last evaluation of the gradient vector and the last change in the -2 log likelihood. You need to use the ITPRINT option in the MODEL statement to obtain this table
- score test result for testing the parallel lines assumption, if an ordinal response model is fitted. If LINK=CLOGLOG or LINK=PROBIT, this test is labeled "Score Test for the Parallel Slopes Assumption." The proportion odds assumption is a special case of the parallel lines assumption when LINK=LOGIT. In this case, the test is labeled "Score Test for the Proportional Odds Assumption"
- "Model Fit Statistics" and "Testing Global Null Hypothesis: BETA=0" tables, which give the various criteria (-2 Log L, AIC, SC) based on the likelihood for fitting a model with intercepts only and for fitting a model with intercepts and explanatory variables. If you specify the NOINT option, these statistics are calculated without considering the intercept parameters. The third column of the table gives the chi-square statistics and p-values for the -2 Log L statistic and for the Score statistic. These test the joint effect of the explanatory

variables included in the model. The Score criterion is always missing for the models identified by the first two columns of the table. Note also that the first two rows of the Chi-Square column are always missing, since tests cannot be performed for AIC and SC

- generalized R^2 measures for the fitted model if you specify the RSQUARE option in the MODEL statement
- “Type III Analysis of Effects” table if the model contains an effect involving a CLASS variable. This table gives the Wald Chi-square statistic, the degrees of freedom, and the p-value for each effect in the model
- “Analysis of Maximum Likelihood Estimates” table, which includes
  - maximum likelihood estimate of the parameter
  - estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
  - Wald chi-square statistic, computed by squaring the ratio of the parameter estimate divided by its standard error estimate
  - p-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom
  - standardized estimate for the slope parameter, given by $\hat{\beta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

    $$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

    You need to specify the STB option in the MODEL statement to obtain these estimates. Standardized estimates of the intercept parameters are set to missing.
  - value of ($e^{\hat{\beta}_i}$) for each slope parameter β_i if you specify the EXPB option in the MODEL statement. For continuous variables, this is equivalent to the estimated odds ratio for a 1 unit change.
  - label of the variable (if space permits) if you specify the PARMLABEL option in the MODEL statement. Due to constraints on the line size, the variable label may be suppressed in order to display the table in one panel. Use the SAS system option LINESIZE= to specify a larger line size to accommodate variable labels. A shorter line size can break the table into two panels allowing labels to be displayed.
- “Odds Ratio Estimates” table, which contains the odds ratio estimates and the corresponding 95% Wald confidence intervals. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.
- measures of association between predicted probabilities and observed responses, which include a breakdown of the number of pairs with different responses, and four rank correlation indexes: Somers’ *D*, Goodman-Kruskal Gamma, and Kendall’s Tau-*a*, and *c*

- confidence intervals for all the parameters if you use the CLPARM option in the MODEL statement
- confidence intervals for all the odds ratios if you use the CLODDS option in the MODEL statement
- "Analysis of Effects not in the Model" table, which gives the score chi-square statistic for testing the significance of each variable not in the model after adjusting for the variables already in the model, and the p-value of the chi-square statistic with respect to a chi-square distribution with one degree of freedom. You specify the DETAILS option in the MODEL statement to obtain this table.
- estimated covariance matrix of the parameter estimates if you use the COVB option in the MODEL statement
- estimated correlation matrix of the parameter estimates if you use the CORRB option in the MODEL statement
- "Linear Hypothesis Testing" table, which gives the result of the Wald test for each TEST statement (if specified)

ODS Table Names

PROC SURVEYLOGISTIC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 69.2. ODS Tables Produced in PROC SURVEYLOGISTIC

ODS Table Name	**Description**	**Statement**	**Option**
ClassLevelInfo	CLASS variable levels and design variables	MODEL	default (with CLASS vars)
CLOdds	Wald's confidence limits for odds ratios	MODEL	CLODDS
CLparmWald	Wald's confidence limits for parameters	MODEL	CLPARM
ContrastCoeff	L matrix from CONTRAST	CONTRAST	E
ContrastEstimate	Estimates from CONTRAST	CONTRAST	ESTIMATE=
ContrastTest	Wald test for CONTRAST	CONTRAST	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB
CumulativeModelTest	Test of the cumulative model assumption	MODEL	(ordinal response)
DesignSummary	Design summary	STRATA | CLUSTER	default
FitStatistics	Model fit statistics	MODEL	default
GlobalTests	Test for global null hypothesis	MODEL	default
IterHistory	Iteration history	MODEL	ITPRINT

Table 69.2. (continued)

ODS Table Name	Description	Statement	Option
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
LogLikeChange	Final change in the log likelihood	MODEL	ITPRINT
ModelInfo	Model information	PROC	default
NObs	Number of observations	PROC	default
OddsRatios	Odds ratios	MODEL	default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
RSquare	R-square	MODEL	RSQUARE
ResponseProfile	Response profile	PROC	default
SimpleStatistics	Summary statistics for explanatory variables	PROC	SIMPLE
StrataInfo	Stratum information	STRATA	LIST
TestPrint1	**L**[cov(**b**)]**L**’ and **Lb-c**	TEST	PRINT
TestPrint2	Ginv(**L**[cov(**b**)]**L**’) and Ginv(**L**[cov(**b**)]**L**’)(**Lb-c**)	TEST	PRINT
TestStmts	Linear hypotheses testing results	TEST	default
TypeIII	Type III tests of effects	MODEL	default (with CLASS variables)

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

Examples

Example 69.1. Logistic Regression with Different Link Functions for Stratified Cluster Sampling

A market research firm conducts a survey among undergraduate students at a certain university to evaluate three new Web designs for a commercial Web site targeting undergraduate students at the university.

The sample design is a stratified sample where strata are students’ classes. Within each class, 300 students are randomly selected using simple random sampling without replacement. The total number of students in each class in the fall semester of 2001 is shown in the following table:

Class	Enrollment
1 - Freshman	3,734
2 - Sophomore	3,565
3 - Junior	3,903
4 - Senior	4,196

This total enrollment information is saved in the SAS data set Enrollment using the following SAS statements:

```
proc format ;
   value Class  1='Freshman' 2='Sophomore'
                3='Junior'   4='Senior';
run;
data Enrollment;
   format Class Class.;
   input Class _TOTAL_;
datalines;
1 3734
2 3565
3 3903
4 4196
;
```

In the data set Enrollment, the variable _TOTAL_ contains the enrollment figures for all classes. They are also the population size for each stratum in this example.

Each student selected in the sample evaluates one randomly selected Web design using the following scale:

1	dislike very much
2	dislike
3	neutral
4	like
5	like very much

The survey results are collected and shown in the following table, with the three different Web designs coded as A, B, and C.

Evaluation of New Web Designs

		Rating Counts				
Strata	Design	**1**	**2**	**3**	**4**	**5**
Freshman	A	10	34	35	16	15
	B	5	6	24	30	25
	C	11	14	20	34	21
Sophomore	A	19	12	26	18	25
	B	10	18	32	23	26
	C	15	22	34	9	20
Junior	A	8	21	23	26	22
	B	1	4	15	33	47
	C	16	19	30	23	12
Senior	A	11	14	24	33	18
	B	8	15	25	30	22
	C	2	34	30	18	16

The survey results are stored in a SAS data set WebSurvey using the following SAS statements.

```
proc format ;
   value Design 1='A' 2='B' 3='C';
   value Rating 1='dislike very much'
                2='dislike'
                3='neutral'
                4='like'
                5='like very much';
run;
data WebSurvey;
   format Class Class. Design Design. Rating Rating. ;
   do Class=1 to 4;
      do Design=1 to 3;
         do Rating=1 to 5;
           input Count @@;
           output;
         end;
      end;
   end;
datalines;
10 34 35 16 15   8 21 23 26 22   5 10 24 30 21
 1 14 25 23 37  11 14 20 34 21  16 19 30 23 12
19 12 26 18 25  11 14 24 33 18  10 18 32 23 17
 8 15 35 30 12  15 22 34  9 20   2 34 30 18 16
;
data WebSurvey; set WebSurvey;
   if Class=1 then Weight=3734/300;
   if Class=2 then Weight=3565/300;
   if Class=3 then Weight=3903/300;
   if Class=4 then Weight=4196/300;
```

The data set WebSurvey contains the variables Class, Design, Rating, Count, and Weight. The variable class is the stratum variable, with four strata: freshman, sophomore, junior, and senior. The variable Design specifies the three new Web designs: A, B, and C. The variable Rating contains students' evaluations for the new Web designs. The variable counts gives the frequency with which each Web design received each rating within each stratum. The variable weight contains the sampling weights, which are the reciprocals of selection probabilities in this example.

Output 69.1.1. Web Design Survey Sample (First 20 Observation)

```
Obs      Class       Design    Rating               Count      Weight

  1     Freshman       A       dislike very much      10      12.4467
  2     Freshman       A       dislike                34      12.4467
  3     Freshman       A       neutral                35      12.4467
  4     Freshman       A       like                   16      12.4467
  5     Freshman       A       like very much         15      12.4467
  6     Freshman       B       dislike very much       8      12.4467
  7     Freshman       B       dislike                21      12.4467
  8     Freshman       B       neutral                23      12.4467
  9     Freshman       B       like                   26      12.4467
 10     Freshman       B       like very much         22      12.4467
 11     Freshman       C       dislike very much       5      12.4467
 12     Freshman       C       dislike                10      12.4467
 13     Freshman       C       neutral                24      12.4467
 14     Freshman       C       like                   30      12.4467
 15     Freshman       C       like very much         21      12.4467
 16     Sophomore      A       dislike very much       1      11.8833
 17     Sophomore      A       dislike                14      11.8833
 18     Sophomore      A       neutral                25      11.8833
 19     Sophomore      A       like                   23      11.8833
 20     Sophomore      A       like very much         37      11.8833
```

Output 69.1.1 shows the first 20 observations of the data set.

The following SAS statements perform the logistic regression.

```
proc surveylogistic data=WebSurvey total=Enrollment;
   stratum Class;
   freq Count;
   class Design;
   model Rating (order=internal) = design ;
   weight Weight;
run;
```

The PROC statement invokes PROC SURVEYLOGISTIC. The TOTAL= option specifies the data set Enrollment, which contains the population totals in the strata. The population totals are used to calculate the finite population correction factor in the variance estimates. The response variable Rating is in the ordinal scale. A cumulative logit model is used to investigate the responses to the Web designs. In the MODEL statement, rating is the response variable, and Design is the effect in the regression model. The ORDER=INTERNAL option is used for the response variable Rating to sort the ordinal response levels of Rating by its internal (numerical) values rather than by the formatted values (e.g., “like very much”). Because the sample design involves stratified simple random sampling, the STRATA statement is used to specify the stratification variable Class. The WEIGHT statement specifies the variable Weight for sampling weights.

Output 69.1.2. Web Design Survey, Model Information

```
                          The SURVEYLOGISTIC Procedure

                               Model Information

            Data Set                              WORK.WEBSURVEY
            Response Variable                     Rating
            Number of Response Levels             5
            Frequency Variable                    Count
            Stratum Variable                      Class
            Number of Strata                      4
            Weight Variable                       Weight
            Model                                 Cumulative Logit
            Optimization Technique                Fisher's Scoring
            Variance Adjustment                   Degrees of Freedom (DF)
            Finite Population Correction          Used


                                Response Profile

          Ordered                                Total             Total
            Value     Rating                 Frequency            Weight

                1     dislike very much            116         1489.0733
                2     dislike                      227         2933.0433
                3     neutral                      338         4363.3767
                4     like                         283         3606.8067
                5     like very much               236         3005.7000

       Probabilities modeled are cumulated over the lower Ordered Values.
```

The sample and analysis summary is shown in Output 69.1.2. There are five response levels for the Rating with 'dislike very much' as the lowest ordered value. The regression model is modeling lower cumulative probabilities using logit as the link function. Because the TOTAL= option is used, the finite population correction is included in the variance estimation. The sampling weight is also used in the analysis.

Output 69.1.3. Web Design Survey, Testing the Proportional Odds Assumption

```
                  Score Test for the Proportional Odds Assumption

                        Chi-Square       DF     Pr > ChiSq

                           98.1957        6         <.0001
```

In Output 69.1.3, the score chi-square for testing the proportional odds assumption is 98.1957, which is highly significant. This indicates that the cumulative logit model may not adequately fit the data.

An alternative model is to use the generalized logit model with the LINK=GLOGIT option as shown in the following SAS statements:

```
proc surveylogistic data=WebSurvey total=Enrollment;
   stratum Class;
```

```
      freq Count;
      class Design;
      model Rating (ref=’neutral’) = Design /link=glogit;
      weight Weight;
   run;
```

The REF=’neutral’ option is used for the response variable Rating to indicate that all other response levels are referenced to the level ‘neutral.’ The option LINK=GLOGIT option requests the procedure to fit a generalized logit model.

Output 69.1.4. Web Design Survey, Model Information

```
                          The SURVEYLOGISTIC Procedure

                               Model Information

             Data Set                            WORK.WEBSURVEY
             Response Variable                   Rating
             Number of Response Levels           5
             Frequency Variable                  Count
             Stratum Variable                    Class
             Number of Strata                    4
             Weight Variable                     Weight
             Model                               Generalized Logit
             Optimization Technique              Fisher’s Scoring
             Variance Adjustment                 Degrees of Freedom (DF)
             Finite Population Correction        Used


                                Response Profile

           Ordered                              Total            Total
             Value     Rating                Frequency           Weight

                 1     dislike                     227        2933.0433
                 2     dislike very much           116        1489.0733
                 3     like                        283        3606.8067
                 4     like very much              236        3005.7000
                 5     neutral                     338        4363.3767

          Logits modeled use Rating=’neutral’ as the reference category.
```

The summary of the analysis is shown in Output 69.1.4, which indicates that the generalized logit model is used in the analysis.

Output 69.1.5. Web Design Survey, Class Level Information

```
                            Class Level Information

                                              Design
                         Class      Value     Variables

                         Design     A           1      0
                                    B           0      1
                                    C          -1     -1
```

Output 69.1.5 shows the parameterization for the main effect Design.

Output 69.1.6. Web Design Survey, Parameter and Odds Ratio Estimates

```
                     Analysis of Maximum Likelihood Estimates

                                                  Standard         Wald
Parameter    Rating              DF   Estimate       Error   Chi-Square   Pr > ChiSq

Intercept    dislike              1    -0.3964      0.0832      22.7100       <.0001
Intercept    dislike very much    1    -1.0826      0.1045     107.3889       <.0001
Intercept    like                 1    -0.1892      0.0780       5.8888       0.0152
Intercept    like very much       1    -0.3767      0.0824      20.9223       <.0001
Design    A  dislike              1    -0.0942      0.1166       0.6518       0.4195
Design    A  dislike very much    1    -0.0647      0.1469       0.1940       0.6596
Design    A  like                 1    -0.1370      0.1104       1.5400       0.2146
Design    A  like very much       1     0.0446      0.1130       0.1555       0.6933
Design    B  dislike              1     0.0391      0.1201       0.1057       0.7451
Design    B  dislike very much    1     0.2721      0.1448       3.5294       0.0603
Design    B  like                 1     0.1669      0.1102       2.2954       0.1298
Design    B  like very much       1     0.1420      0.1174       1.4641       0.2263


                               Odds Ratio Estimates

                                                Point          95% Wald
    Effect           Rating                  Estimate      Confidence Limits

    Design A vs C    dislike                    0.861       0.583       1.272
    Design A vs C    dislike very much          1.153       0.692       1.923
    Design A vs C    like                       0.899       0.618       1.306
    Design A vs C    like very much             1.260       0.851       1.865
    Design B vs C    dislike                    0.984       0.659       1.471
    Design B vs C    dislike very much          1.615       0.975       2.675
    Design B vs C    like                       1.218       0.838       1.768
    Design B vs C    like very much             1.389       0.925       2.086
```

The parameter and odds ratio estimates are are shown in Output 69.1.6. For each odds ratio estimate, its 95% confidence limits shown in the table contain the value 1.0. Therefore, no conclusion can be made based on this survey about which Web design is preferred.

Example 69.2. The Household Component of the Medical Expenditure Panel Survey (MEPS)

The Household Component of the Medical Expenditure Panel Survey (MEPS-HC) is designed to produce national and regional estimates of the health care use, expenditures, sources of payment, and insurance coverage of the U.S. civilian noninstitutionalized population (MEPS Fact Sheet, 2001). The sample design of the survey includes stratification, clustering, multiple stages of selection, and disproportionate sampling. Furthermore, the MEPS sampling weights reflect adjustments for survey nonresponse and adjustments to population control totals from the Current Population Survey (Computing Standard Errors for MEPS Estimates, 2003).

In this example, the 1999 full-year consolidated data file HC-038 (PUF Data Files, 2002) from the MEPS is used to investigate the relationship between medical insurance coverage and the demographic variables. The data can be downloaded directly from the Agency for Healthcare Research and Quality (AHRQ) Web site (http://www.meps.ahrq.gov/Puf/PufDetail.asp?ID=93) in either ASCII format or SAS transport format. The Web site includes a detailed description of the data as well as the SAS program code used to access and to format it.

For this example, the SAS transport format data file for HC-038 is downloaded to 'C:H38.ssp' on a Windows-based PC. The instructions on the Web site lead to the following SAS statements for creating a SAS data set named MEPS, which contains only the sample design variables and other variables necessary for this analysis.

```
proc format;
   value racex
      -9 = 'NOT ASCERTAINED'
      -8 = 'DK'
      -7 = 'REFUSED'
      -1 = 'INAPPLICABLE'
      1 = 'AMERICAN INDIAN'
      2 = 'ALEUT, ESKIMO'
      3 = 'ASIAN OR PACIFIC ISLANDER'
      4 = 'BLACK'
      5 = 'WHITE'
      91 = 'OTHER'
      ;
   value sex
      -9 = 'NOT ASCERTAINED'
      -8 = 'DK'
      -7 = 'REFUSED'
      -1 = 'INAPPLICABLE'
      1 = 'MALE'
      2 = 'FEMALE'
      ;
   value povcat9h
      1 = 'NEGATIVE OR POOR'
      2 = 'NEAR POOR'
      3 = 'LOW INCOME'
      4 = 'MIDDLE INCOME'
      5 = 'HIGH INCOME'
      ;
   value inscov9f
      1 = 'ANY PRIVATE'
      2 = 'PUBLIC ONLY'
      3 = 'UNINSURED'
      ;
   run;

libname puflib 'C:';
filename in1 'C:H38.ssp';
proc xcopy in=in1 out=puflib import;
run;

data meps; set puflib.H38;
   label racex= sex= inscov99= povcat99=
      varstr99= varpsu99= perwt99f= totexp99=;
   format racex racex. sex sex.
      povcat99 povcat9h. inscov99 inscov9f.;
   keep inscov99 sex racex povcat99 varstr99
      varpsu99 perwt99f totexp99;
run;
```

There are a total of 24,618 observations in this SAS data set. Each observation corresponds to a person in the survey. The stratification variable is VARSTR99, which identifies the 143 strata in the sample. The variable VARPSU99 identifies the 460 PSUs in the sample. The sampling weights are stored in the variable PERWT99F. The response variable is the health insurance coverage indicator variable, INSCOV99, which has three values:

1	the person had any private insurance coverage any time during 1999
2	the person had only public insurance coverage during 1999
3	the person was uninsured during all of 1999

The demographic variables include gender (SEX), race (RACEX), and family income level as a percent of the poverty line (POVCAT99). The variable RACEX has five categories:

1	American Indian
2	Aleut, Eskimo
3	Asian or Pacific Islander
4	Black
5	White

The variable POVCAT99 is constructed by dividing family income by the applicable poverty line (based on family size and composition), with the resulting percentages grouped into five categories:

1	negative or poor (less than 100%)
2	near poor (100% to less than 125%)
3	low income (125% to less than 200%)
4	middle income (200% to less than 400%)
5	high income (greater than or equal to 400%)

The data set also contains the total health care expenditure in 1999, TOTEXP99, which is used as a covariate in the analysis.

Output 69.2.1. 1999 Full-year MEPS (First 30 Observations)

```
                           P               I          T         P   V    V
                           O               N          O         E   A    A
                           V               S          T         R   R    R
               R           C               C          E         W   S    P
               A           A               O          X         T   T    S
O      S       C           T               V          P         9   R    U
b      E       E           9               9          9         9   9    9
s      X       X           9               9          9         F   9    9

  1  MALE    WHITE  MIDDLE INCOME     PUBLIC ONLY  2735  14137.86  131    2
  2  FEMALE  WHITE  MIDDLE INCOME     ANY PRIVATE  6687  17050.99  131    2
  3  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE    60  35737.55  131    2
  4  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE    60  35862.67  131    2
  5  FEMALE  WHITE  MIDDLE INCOME     ANY PRIVATE   786  19407.11  131    2
  6  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE   345  18499.83  131    2
  7  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE   680  18499.83  131    2
  8  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE  3226  22394.53  136    1
  9  FEMALE  WHITE  MIDDLE INCOME     ANY PRIVATE  2852  27008.96  136    1
 10  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE   112  25108.71  136    1
 11  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE  3179  17569.81  136    1
 12  MALE    WHITE  MIDDLE INCOME     ANY PRIVATE   168  21478.06  136    1
 13  FEMALE  WHITE  MIDDLE INCOME     ANY PRIVATE  1066  21415.68  136    1
 14  MALE    WHITE  NEGATIVE OR POOR  PUBLIC ONLY     0  12254.66  125    1
 15  MALE    WHITE  NEGATIVE OR POOR  ANY PRIVATE     0  17699.75  125    1
 16  FEMALE  WHITE  NEGATIVE OR POOR  UNINSURED       0  18083.15  125    1
 17  MALE    BLACK  NEGATIVE OR POOR  PUBLIC ONLY   230   6537.97   78   10
 18  MALE    WHITE  LOW INCOME        UNINSURED     408   8951.36   95    2
 19  FEMALE  WHITE  LOW INCOME        UNINSURED       0  11833.00   95    2
 20  MALE    WHITE  LOW INCOME        UNINSURED      40  12754.07   95    2
 21  FEMALE  WHITE  LOW INCOME        UNINSURED      51  14698.57   95    2
 22  MALE    WHITE  LOW INCOME        UNINSURED       0   3890.20   92   19
 23  FEMALE  WHITE  LOW INCOME        UNINSURED     610   5882.29   92   19
 24  MALE    WHITE  LOW INCOME        PUBLIC ONLY    24   8610.47   92   19
 25  FEMALE  BLACK  MIDDLE INCOME     UNINSURED    1758      0.00   64    1
 26  MALE    BLACK  MIDDLE INCOME     PUBLIC ONLY   551   7049.70   64    1
 27  MALE    BLACK  MIDDLE INCOME     ANY PRIVATE    65  34067.03   64    1
 28  FEMALE  BLACK  NEGATIVE OR POOR  PUBLIC ONLY     0   9313.84   73   12
 29  FEMALE  BLACK  NEGATIVE OR POOR  PUBLIC ONLY    10  14697.03   73   12
 30  MALE    BLACK  NEGATIVE OR POOR  PUBLIC ONLY     0   4574.73   73   12
```

Output 69.2.1 displays the first 30 observations of this data set.

The following SAS statements fit a generalized logit model for the 1999 full-year consolidated MEPS data.

```
proc surveylogistic data=meps;
   stratum VARSTR99;
   cluster VARPSU99;
   weight PERWT99F;
   class SEX RACEX POVCAT99;
   model INSCOV99 = TOTEXP99 SEX RACEX POVCAT99 / link=glogit;
run;
```

The STRATUM statement specifies the stratification variable VARSTR99. The CLUSTER statement specifies the PSU variable VARPSU99. The WEIGHT statement specifies the sample weight variable PERWT99F. The demographic variables

SEX, RACEX, and POVCAT99 are listed in the CLASS statement to indicate that they are categorical independent variables in the MODEL statement. In the MODEL statement, the response variable is INSCOV99, and the independent variables are TOTEXP99 along with the selected demographic variables. The LINK= option requests the procedure to fit the generalized logit model because the response variable INSCOV99 has nominal responses.

The results of this analysis are shown in the following tables.

Output 69.2.2. MEPS, Model Information

```
                          The SURVEYLOGISTIC Procedure

                               Model Information

              Data Set                      WORK.MEPS
              Response Variable             INSCOV99
              Number of Response Levels     3
              Stratum Variable              VARSTR99
              Number of Strata              143
              Cluster Variable              VARPSU99
              Number of Clusters            460
              Weight Variable               PERWT99F
              Model                         Generalized Logit
              Optimization Technique        Fisher's Scoring
              Variance Adjustment           Degrees of Freedom (DF)
```

PROC SURVEYLOGISTIC lists the model fitting information and sample design information in Output 69.2.2:

Output 69.2.3. MEPS, Number of Observations

```
                     Number of Observations Read       24618
                     Number of Observations Used       23565
                     Sum of Weights Read            2.7641E8
                     Sum of Weights Used            2.7641E8
```

Output 69.2.3 displays the number of observations and the total of sampling weights both in the data set and used in the analysis. Only the observations with positive person-level weight are used in the analysis. Therefore, 1,053 observations with zero person-level weights were deleted.

Output 69.2.4. MEPS, Response Profile

```
                              Response Profile

         Ordered                                 Total            Total
           Value      INSCOV99               Frequency           Weight

               1      ANY PRIVATE                16130        204403997
               2      PUBLIC ONLY                 4241         41809572
               3      UNINSURED                   3194         30197198

       Logits modeled use INSCOV99='UNINSURED' as the reference category.
```

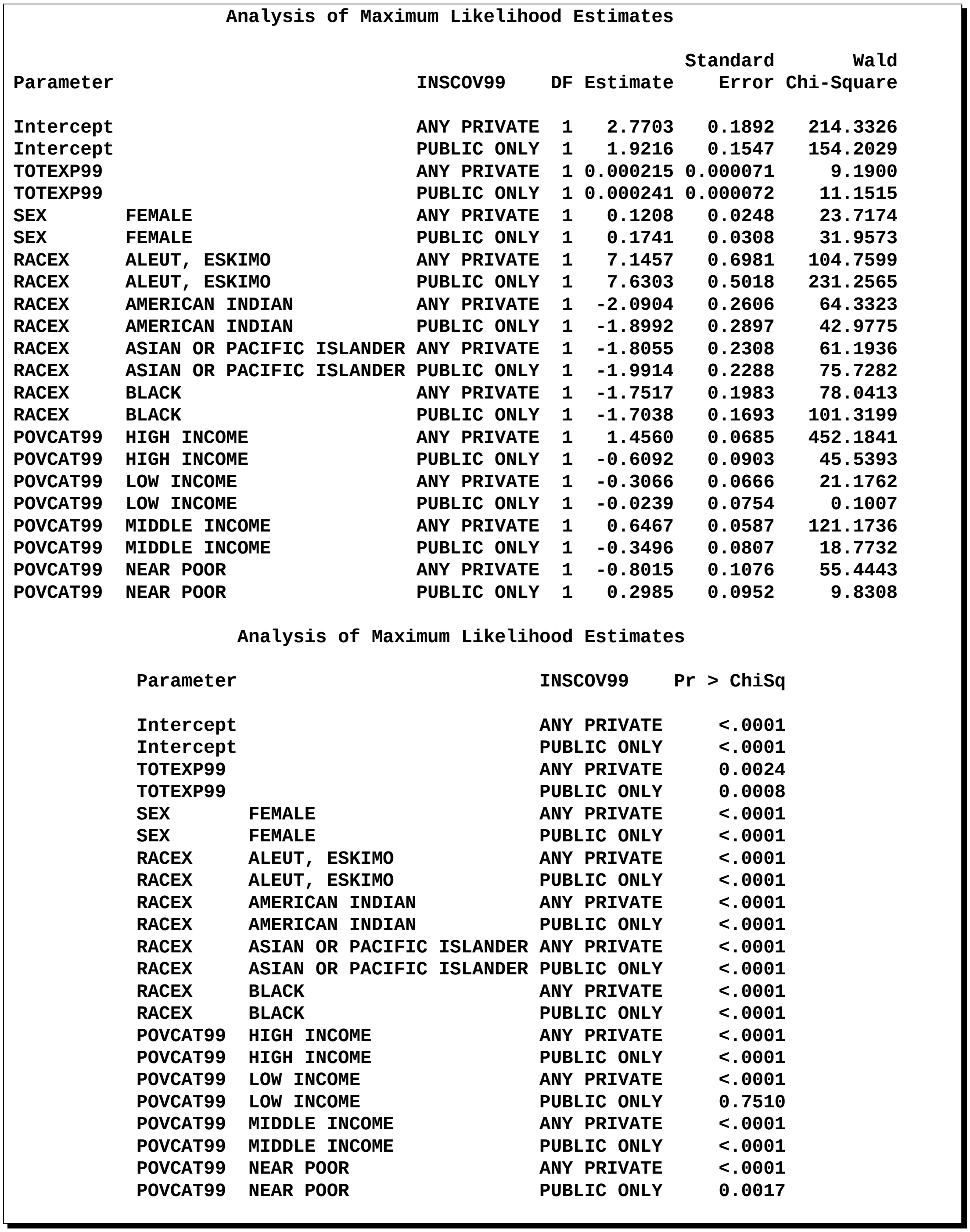
Output 69.2.4 lists the three insurance coverage levels for the response variable INSCOV99. The “UNINSURED” category is used as the reference category in the model.

Output 69.2.5. MEPS, Classification Levels

```
                          Class Level Information

       Class         Value                            Design Variables

       SEX           FEMALE                          1
                     MALE                           -1

       RACEX         ALEUT, ESKIMO                   1      0      0      0
                     AMERICAN INDIAN                 0      1      0      0
                     ASIAN OR PACIFIC ISLANDER       0      0      1      0
                     BLACK                           0      0      0      1
                     WHITE                          -1     -1     -1     -1

       POVCAT99      HIGH INCOME                     1      0      0      0
                     LOW INCOME                      0      1      0      0
                     MIDDLE INCOME                   0      0      1      0
                     NEAR POOR                       0      0      0      1
                     NEGATIVE OR POOR               -1     -1     -1     -1
```

Output 69.2.5 shows the parameterization in the regression model for each categorical independent variable.

Output 69.2.6. MEPS, Parameter Estimates



```
                   Analysis of Maximum Likelihood Estimates

                                                            Standard       Wald
Parameter                           INSCOV99    DF Estimate    Error Chi-Square

Intercept                           ANY PRIVATE  1   2.7703   0.1892   214.3326
Intercept                           PUBLIC ONLY  1   1.9216   0.1547   154.2029
TOTEXP99                            ANY PRIVATE  1 0.000215 0.000071     9.1900
TOTEXP99                            PUBLIC ONLY  1 0.000241 0.000072    11.1515
SEX       FEMALE                    ANY PRIVATE  1   0.1208   0.0248    23.7174
SEX       FEMALE                    PUBLIC ONLY  1   0.1741   0.0308    31.9573
RACEX     ALEUT, ESKIMO             ANY PRIVATE  1   7.1457   0.6981   104.7599
RACEX     ALEUT, ESKIMO             PUBLIC ONLY  1   7.6303   0.5018   231.2565
RACEX     AMERICAN INDIAN           ANY PRIVATE  1  -2.0904   0.2606    64.3323
RACEX     AMERICAN INDIAN           PUBLIC ONLY  1  -1.8992   0.2897    42.9775
RACEX     ASIAN OR PACIFIC ISLANDER ANY PRIVATE  1  -1.8055   0.2308    61.1936
RACEX     ASIAN OR PACIFIC ISLANDER PUBLIC ONLY  1  -1.9914   0.2288    75.7282
RACEX     BLACK                     ANY PRIVATE  1  -1.7517   0.1983    78.0413
RACEX     BLACK                     PUBLIC ONLY  1  -1.7038   0.1693   101.3199
POVCAT99  HIGH INCOME               ANY PRIVATE  1   1.4560   0.0685   452.1841
POVCAT99  HIGH INCOME               PUBLIC ONLY  1  -0.6092   0.0903    45.5393
POVCAT99  LOW INCOME                ANY PRIVATE  1  -0.3066   0.0666    21.1762
POVCAT99  LOW INCOME                PUBLIC ONLY  1  -0.0239   0.0754     0.1007
POVCAT99  MIDDLE INCOME             ANY PRIVATE  1   0.6467   0.0587   121.1736
POVCAT99  MIDDLE INCOME             PUBLIC ONLY  1  -0.3496   0.0807    18.7732
POVCAT99  NEAR POOR                 ANY PRIVATE  1  -0.8015   0.1076    55.4443
POVCAT99  NEAR POOR                 PUBLIC ONLY  1   0.2985   0.0952     9.8308

                    Analysis of Maximum Likelihood Estimates

           Parameter                           INSCOV99    Pr > ChiSq

           Intercept                           ANY PRIVATE      <.0001
           Intercept                           PUBLIC ONLY      <.0001
           TOTEXP99                            ANY PRIVATE      0.0024
           TOTEXP99                            PUBLIC ONLY      0.0008
           SEX       FEMALE                    ANY PRIVATE      <.0001
           SEX       FEMALE                    PUBLIC ONLY      <.0001
           RACEX     ALEUT, ESKIMO             ANY PRIVATE      <.0001
           RACEX     ALEUT, ESKIMO             PUBLIC ONLY      <.0001
           RACEX     AMERICAN INDIAN           ANY PRIVATE      <.0001
           RACEX     AMERICAN INDIAN           PUBLIC ONLY      <.0001
           RACEX     ASIAN OR PACIFIC ISLANDER ANY PRIVATE      <.0001
           RACEX     ASIAN OR PACIFIC ISLANDER PUBLIC ONLY      <.0001
           RACEX     BLACK                     ANY PRIVATE      <.0001
           RACEX     BLACK                     PUBLIC ONLY      <.0001
           POVCAT99  HIGH INCOME               ANY PRIVATE      <.0001
           POVCAT99  HIGH INCOME               PUBLIC ONLY      <.0001
           POVCAT99  LOW INCOME                ANY PRIVATE      <.0001
           POVCAT99  LOW INCOME                PUBLIC ONLY      0.7510
           POVCAT99  MIDDLE INCOME             ANY PRIVATE      <.0001
           POVCAT99  MIDDLE INCOME             PUBLIC ONLY      <.0001
           POVCAT99  NEAR POOR                 ANY PRIVATE      <.0001
           POVCAT99  NEAR POOR                 PUBLIC ONLY      0.0017
```

Output 69.2.6 displays the parameter estimates and their standard errors.

Output 69.2.7. MEPS, Odds Ratios

```
                              Odds Ratio Estimates

Effect                                                          INSCOV99

TOTEXP99                                                        ANY PRIVATE
TOTEXP99                                                        PUBLIC ONLY
SEX      FEMALE vs MALE                                         ANY PRIVATE
SEX      FEMALE vs MALE                                         PUBLIC ONLY
RACEX    ALEUT, ESKIMO             vs WHITE                     ANY PRIVATE
RACEX    ALEUT, ESKIMO             vs WHITE                     PUBLIC ONLY
RACEX    AMERICAN INDIAN           vs WHITE                     ANY PRIVATE
RACEX    AMERICAN INDIAN           vs WHITE                     PUBLIC ONLY
RACEX    ASIAN OR PACIFIC ISLANDER vs WHITE                     ANY PRIVATE
RACEX    ASIAN OR PACIFIC ISLANDER vs WHITE                     PUBLIC ONLY
RACEX    BLACK                     vs WHITE                     ANY PRIVATE
RACEX    BLACK                     vs WHITE                     PUBLIC ONLY
POVCAT99 HIGH INCOME      vs NEGATIVE OR POOR                   ANY PRIVATE
POVCAT99 HIGH INCOME      vs NEGATIVE OR POOR                   PUBLIC ONLY
POVCAT99 LOW INCOME       vs NEGATIVE OR POOR                   ANY PRIVATE
POVCAT99 LOW INCOME       vs NEGATIVE OR POOR                   PUBLIC ONLY
POVCAT99 MIDDLE INCOME    vs NEGATIVE OR POOR                   ANY PRIVATE
POVCAT99 MIDDLE INCOME    vs NEGATIVE OR POOR                   PUBLIC ONLY
POVCAT99 NEAR POOR        vs NEGATIVE OR POOR                   ANY PRIVATE
POVCAT99 NEAR POOR        vs NEGATIVE OR POOR                   PUBLIC ONLY

                              Odds Ratio Estimates

                           Point          95% Wald
                        Estimate      Confidence Limits

                           1.000       1.000       1.000
                           1.000       1.000       1.000
                           1.273       1.155       1.403
                           1.417       1.255       1.598
                        >999.999    >999.999    >999.999
                        >999.999    >999.999    >999.999
                           0.553       0.340       0.901
                           1.146       0.603       2.179
                           0.735       0.500       1.082
                           1.045       0.656       1.665
                           0.776       0.639       0.943
                           1.394       1.132       1.717
                          11.595       9.301      14.455
                           0.274       0.213       0.353
                           1.990       1.607       2.464
                           0.492       0.395       0.614
                           5.162       4.200       6.343
                           0.356       0.280       0.451
                           1.213       0.903       1.630
                           0.680       0.527       0.877
```

Output 69.2.7 displays the odds ratio estimates and their standard errors.

For example, after adjusting for the effects of sex, race, and total health care expenditures, a person with high income is estimated to be 11.595 times more likely than a poor person to choose private health care insurance over no insurance, but only 0.274 times as likely to choose public health insurance over no insurance.

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons, Inc.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons, Inc.

Aitchison, J. and Silvey, S. D. (1957), "The Generalization of Probit Analysis to the Case of Multiple Responses," *Biometrika*, 44, 131–40.

Albert, A. and Anderson, J. A. (1984), "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 71, 1–10.

Ashford, J. R. (1959), "An Approach to the Analysis of Data for Semi-Quantal Responses in Biology Response," *Biometrics*, 15, 573–81.

Binder, D. A. (1981), "On the Variances of Asymptotically Normal Estimators from Complex Surveys," *Survey Methodology*, 7, 157–170.

Binder, D. A. (1983), "On the Variances of Asymptotically Normal Estimators from Complex Surveys," *International Statistical Review*, 51, 279–292.

Binder, D. A. and Roberts, G. R. (2003), "Design-based and Model-based Methods for Estimating Model Parameters," in *Analysis of Survey Data*, ed. C. Skinner and R. Chambers, New York: John Wiley & Sons, Inc.

Brick, J. M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215 –238.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman and Hall.

Computing Standard Errors for MEPS Estimates, January 2003, Agency for Healthcare Research and Quality, Rockville, MD, [http://www.meps.ahrq.gov/factsheets/FS_StandardErrors.htm].

Cox, D. R. and Snell, E. J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman and Hall.

Fleiss, J. L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons, Inc.

Freeman, D. H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker, Inc.

Hanley, J. A. and McNeil, B. J. (1982), "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, 143 29–36.

Hidiroglou, M. A., Fuller, W. A., and Hickman, R. D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Hosmer, D. W, Jr. and Lemeshow, S. (2000), *Applied Logistic Regression*, Second Edition, New York: John Wiley & Sons, Inc.

Kalton, G., and Kaspyzyk, D. (1986), “The Treatment of Missing Survey Data,” *Survey Methodology*, 12, 1 –16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Korn, E. and Graubard B. (1999), *Analysis of Health Survey*, New York: John Wiley & Sons, Inc.

Lancaster, H. O., (1961), “Significance Tests in Discrete Distributions,” *JASA*, 56, 223–234.

Lehtonen, R. and Pahkinen E. (1995), *Practical Methods for Design and Analysis of Complex Surveys*, Chichester: John Wiley & Sons, Inc.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, London: Chapman Hall.

McFadden, D. (1974), “Conditional Logit Analysis of Qualitative Choice Behaviour” in *Frontiers in Econometrics*, ed. by P. Zarembka, New York: Academic Press.

MEPS Fact Sheet, February 2001, Agency for Healthcare Research and Quality, Rockville, MD, [http://www.meps.ahrq.gov/whatismeps/bulletin.htm].

Morel, G. (1989) “Logistic Regression under Complex Survey Designs,” *Survey Methodology*, 15, 203–223.

Nagelkerke, N. J. D. (1991), “A Note on a General Definition of the Coefficient of Determination,” *Biometrika*, 78, 691–692.

Nelder, J. A. and Wedderburn, R. W. M. (1972), “Generalized Linear Models,” *Journal of the Royal Statistical Society, Series A*, 135, 761–768.

PUF Data Files, October 2002, Agency for Healthcare Research and Quality, Rockville, MD, [http://www.meps.ahrq.gov/Puf/PufDetail.asp?ID=93].

Roberts, G., Rao, J. N. K., and Kumar, S. (1987), “Logistic Regression Analysis of Sample Survey Data,” *Biometrika*, 74, 1–12.

Santner, T. J. and Duffy, E. D. (1986), “A Note on A. Albert and J. A. Anderson’s Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models,” *Biometrika*, 73, 755–758.

Skinner, C. J., Holt, D., and Smith, T. M. F. (1989), *Analysis of Complex Surveys*, New York: John Wiley & Sons, Inc.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Walker, S. H. and Duncan, D. B. (1967), “Estimation of the Probability of an Event as a Function of Several Independent Variables,” *Biometrika*, 54, 167–179.

Chapter 70
The SURVEYMEANS Procedure

Chapter Contents

Chapter 70
The SURVEYMEANS Procedure

Overview

The SURVEYMEANS procedure produces estimates of survey population means and totals from sample survey data. The procedure also produces variance estimates, confidence limits, and other descriptive statistics. When computing these estimates, the procedure takes into account the sample design used to select the survey sample. The sample design can be a complex survey sample design with stratification, clustering, and unequal weighting.

PROC SURVEYMEANS uses the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971, Fuller 1975). When there are clusters, or primary sampling units (PSUs), in the sample design, the procedure estimates variance from the variation among PSUs. When the design is stratified, the procedure pools stratum variance estimates to compute the overall variance estimate.

PROC SURVEYMEANS uses the Output Delivery System (ODS) to place results in output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality.

Getting Started

This section demonstrates how you can use the SURVEYMEANS procedure to produce descriptive statistics from sample survey data. For a complete description of PROC SURVEYMEANS, please refer to the "Syntax" section on page 4322. The "Examples" section on page 4350 provides more complicated examples to illustrate the applications of PROC SURVEYMEANS.

Simple Random Sampling

This example illustrates how you can use PROC SURVEYMEANS to estimate population means and proportions from sample survey data. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on average, and what percentage of students spend at least $10 weekly for ice cream.

To answer these questions, 40 students were selected from the entire student population using simple random sampling (SRS). Selection by simple random sampling means that all students have an equal chance of being selected, and no student can be selected more than once. Each student selected for the sample was asked how much he spends for ice cream per week, on average. The SAS data set named IceCream saves the responses of the 40 students:

```
data IceCream;
   input Grade Spending @@;
   if (Spending < 10) then Group='less';
     else Group='more';
   datalines;
7 7  7  7  8 12  9 10  7  1  7 10  7  3  8 20  8 19  7 2
7 2  9 15  8 16  7  6  7  6  7  6  9 15  8 17  8 14  9 8
9 8  9  7  7  3  7 12  7  4  9 14  8 18  9  9  7  2  7 1
7 4  7 11  9  8  8 10  8 13  7  2  9  6  9 11  7  2  7 9
;
```

The variable Grade contains a student's grade. The variable Spending contains a student's response on how much he spends per week for ice cream, in dollars. The variable Group is created to indicate whether a student spends at least $10 weekly for ice cream: Group='more' if a student spends at least $10, or Group='less' if a student spends less than $10.

You can use PROC SURVEYMEANS to produce estimates for the entire student population, based on this random sample of 40 students:

```
title1 'Analysis of Ice Cream Spending';
title2 'Simple Random Sample Design';
proc surveymeans data=IceCream total=4000;
   var Spending Group;
   run;
```

The PROC SURVEYMEANS statement invokes the procedure. The TOTAL=4000 option specifies the total number of students in the study population, or school. The procedure uses this total to adjust variance estimates for the effects of sampling from a finite population. The VAR statement names the variables to analyze, Spending and Group.

Figure 70.1 displays the results from this analysis. There are a total of 40 observations used in the analysis. The "Class Level Information" table lists the two levels of the variable Group. This variable is a character variable, and so PROC SURVEYMEANS provides a categorical analysis for it, estimating the relative frequency or proportion for each level. If you want a categorical analysis for a numeric variable, you can name that variable in the CLASS statement.

```
                      Analysis of Ice Cream Spending
                       Simple Random Sample Design

                        The SURVEYMEANS Procedure

                               Data Summary

                   Number of Observations            40


                         Class Level Information

                   Class
                   Variable      Levels    Values

                   Group              2    less more


                                Statistics

                                                         Std Error       Lower 95%
Variable    Level                 N             Mean       of Mean     CL for Mean
----------------------------------------------------------------------------------
Spending                         40         8.750000      0.845139        7.040545
Group       less                 23         0.575000      0.078761        0.415690
            more                 17         0.425000      0.078761        0.265690
----------------------------------------------------------------------------------


                                Statistics

                                      Upper 95%
                    Variable    Level     CL for Mean
                    ---------------------------------
                    Spending               10.459455
                    Group       less        0.734310
                                more        0.584310
                    ---------------------------------
```

Figure 70.1. Analysis of Ice Cream Spending, Simple Random Sample Design

The "Statistics" table displays the estimates for each analysis variable. By default, PROC SURVEYMEANS displays the number of observations, the estimate of the mean, its standard error, and 95% confidence limits for the mean. You can obtain other statistics by specifying the corresponding statistic-keywords in the PROC SURVEYMEANS statement.

The estimate of the average weekly ice cream expense is $8.75 for students in this school. The standard error of this estimate if $0.85, and the 95% confidence interval for weekly ice cream expense is from $7.04 to $10.46.

The analysis variable Group is a character variable, and so PROC SURVEYMEANS analyzes it as categorical, estimating the relative frequency or proportion for each level or category. These estimates are displayed in the Mean column of the "Statistics" table. It is estimated that 57.5% of all students spend less than $10 weekly on ice cream, while 42.5% of the students spend at least $10 weekly. The standard error of each estimate is 7.9%.

Stratified Sampling

Suppose that the sample of students described in the previous section was actually selected using stratified random sampling. In stratified sampling, the study population is divided into nonoverlapping strata, and samples are selected from each stratum independently.

The list of students in this junior high school was stratified by grade, yielding three strata: grades 7, 8, and 9. A simple random sample of students was selected from each grade. Table 70.1 shows the total number of students in each grade.

Table 70.1. Number of Students by Grade

Grade	Number of Students
7	1,824
8	1,025
9	1,151
Total	4,000

To analyze this stratified sample, you need to provide the population totals for each stratum to PROC SURVEYMEANS. The SAS data set named StudentTotals contains the information from Table 70.1:

```
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

The variable Grade is the stratum identification variable, and the variable _TOTAL_ contains the total number of students for each stratum. PROC SURVEYMEANS requires you to use the variable name _TOTAL_ for the stratum population totals.

The procedure uses the stratum population totals to adjust variance estimates for the effects of sampling from a finite population. If you do not provide population totals or sampling rates, then the procedure assumes that the proportion of the population in the sample is very small, and the computation does not involve a finite population correction.

In a stratified sample design, when the sampling rates in the strata are unequal, you need to use sampling weights to reflect this information in order to produce an unbiased mean estimator. In this example, the appropriate sampling weights are reciprocals of the probabilities of selection. You can use the following data step to create the sampling weights:

```
data IceCream;
   set IceCream;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

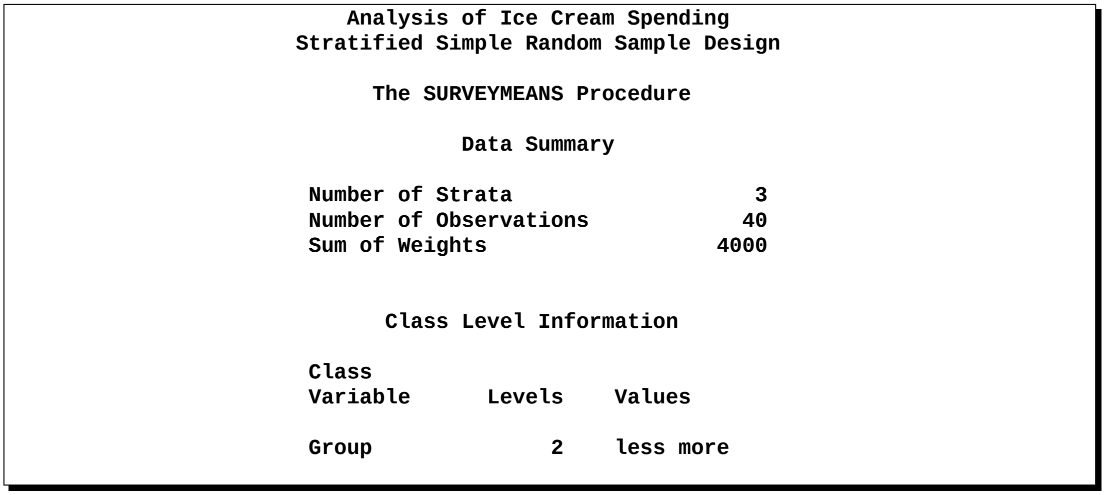
If you use PROC SURVEYSELECT to select your sample, PROC SURVEYSELECT creates these sampling weights for you.

The following SAS statements perform the stratified analysis of the survey data:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
run;
```

The PROC SURVEYMEANS statement invokes the procedure. The DATA= option names the SAS data set IceCream as the input data set to be analyzed. The TOTAL= option names the data set StudentTotals as the input data set containing the stratum population totals. Comparing this to the analysis in the "Simple Random Sampling" section on page 4315, notice that the TOTAL=StudentTotals option is used here instead of the TOTAL=4000 option. In this stratified sample design, the population totals are different for different strata, and so you need to provide them to PROC SURVEYMEANS in a SAS data set.

The STRATA statement identifies the stratification variable Grade. The LIST option in the STRATA statement requests that the procedure display stratum information. The WEIGHT statement tells the procedure that the variable Weight contains the sampling weights.



```
                 Analysis of Ice Cream Spending
             Stratified Simple Random Sample Design

                   The SURVEYMEANS Procedure

                          Data Summary

              Number of Strata                     3
              Number of Observations              40
              Sum of Weights                    4000


                    Class Level Information

              Class
              Variable      Levels    Values

              Group              2    less more
```

Figure 70.2. Data Summary

Figure 70.2 displays information on the input data set. There are three strata in the design, and 40 observations in the sample. The categorical variable Group has two levels, 'less' and 'more'.

```
                    Analysis of Ice Cream Spending
                Stratified Simple Random Sample Design

                      The SURVEYMEANS Procedure

                         Stratum Information

 Stratum            Population  Sampling
  Index      Grade       Total      Rate     N Obs  Variable  Level         N
 ----------------------------------------------------------------------------
    1            7        1824     1.10%        20  Spending               20
                                                    Group     less         17
                                                              more          3
    2            8        1025     0.88%         9  Spending                9
                                                    Group     less          0
                                                              more          9
    3            9        1151     0.96%        11  Spending               11
                                                    Group     less          6
                                                              more          5
 ----------------------------------------------------------------------------
```

Figure 70.3. Stratum Information

Figure 70.3 displays information for each stratum. The table displays a Stratum Index and the values of the STRATA variable. The Stratum Index identifies each stratum by a sequentially assigned number. For each stratum, the table gives the population total (total number of students), the sampling rate, and the sample size. The stratum sampling rate is the ratio of the number of students in the sample to the number of students in the population for that stratum. The table also lists each analysis variable and the number of stratum observations for that variable. For categorical variables, the table lists each level and the number of sample observations in that level.

```
                      Analysis of Ice Cream Spending
                  Stratified Simple Random Sample Design

                        The SURVEYMEANS Procedure

                               Statistics

                                                      Std Error       Lower 95%
Variable    Level               N            Mean       of Mean     CL for Mean
-------------------------------------------------------------------------------
Spending                       40        9.141298      0.531799        8.063771
Group       less               23        0.544555      0.058424        0.426177
            more               17        0.455445      0.058424        0.337068
-------------------------------------------------------------------------------

                               Statistics

                                          Upper 95%
                        Variable    Level     CL for Mean
                        ---------------------------------
                        Spending               10.218825
                        Group       less        0.662932
                                    more        0.573823
                        ---------------------------------
```

Figure 70.4. Analysis of Ice Cream Spending, Stratified SRS Design

Figure 70.4 shows that

- the estimate of average weekly ice cream expense is $9.14 for students in this school, with a standard error of $0.53, and a 95% confidence interval from $8.06 to $10.22.
- an estimate of 54.5% of all students spend less than $10 weekly on ice cream, and 45.5% spend more, with a standard error of 5.8%.

Output Data Set

PROC SURVEYMEANS uses the Output Delivery System (ODS) to create output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

For example, to save the "Statistics" table shown in Figure 70.4 in the previous section in an output data set, you use the ODS OUTPUT statement as follows:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
   ods output Statistics=MyStat;
run;
```

The statement

```
ods output Statistics=MyStat;
```

requests that the "Statistics" table that appears in Figure 70.4 be placed in a SAS data set named MyStat.

The PRINT procedure displays observations of the data set MyStat:

```
proc print data=MyStat;
run;
```

Figure 70.5 displays the data set MyStat.

```
                         Analysis of Ice Cream Spending
                     Stratified Simple Random Sample Design

                                                                  L            U
                                                                  o            p
                                                                  w            p
            V                                                     e            e
     V      a                                                     r            r
     a      r                                        S            C            C
     r      L                                        t            L            L
     N      e                           M            d            M            M
O    a      v                           e            E            e            e
B    m      e                           a            r            a            a
S    e      l              N            n            r            n            n

1 Spending                40     9.141298     0.531799     8.063771    10.218825
2 Group    less           23     0.544555     0.058424     0.426177     0.662932
3 Group    more           17     0.455445     0.058424     0.337068     0.573823
```

Figure 70.5. The Data Set MyStat

The section "ODS Table Names" on page 4349 gives the complete list of the tables produced by PROC SURVEYMEANS.

Syntax

The following statements are available in PROC SURVEYMEANS.

PROC SURVEYMEANS < *options* > < *statistic-keywords* > **;**
  BY *variables* **;**
  CLASS *variables* **;**
  CLUSTER *variables* **;**
  DOMAIN *variables* < *variable*∗*variable*
    variable∗*variable*∗*variable* . . . > **;**
  RATIO < *'label'* > *variables* / *variables* **;**
  STRATA *variables* < / *option* > **;**
  VAR *variables* **;**
  WEIGHT *variable* **;**

The PROC SURVEYMEANS statement invokes the procedure. It optionally names the input data sets and specifies statistics for the procedure to compute. The PROC SURVEYMEANS statement is required.

The VAR statement identifies the variables to be analyzed. The CLASS statement identifies those numeric variables that are to be analyzed as categorical variables. The STRATA statement lists the variables that form the strata in a stratified sample design. The CLUSTER statement specifies cluster identification variables in a clustered sample design. The DOMAIN statement lists the variables that define domains for subpopulation analysis. The RATIO statement requests ratio analysis for means or proportions of analysis variables. The WEIGHT statement names the sampling weight variable. You can use a BY statement with PROC SURVEYMEANS to obtain separate analyses for groups defined by the BY variables.

All statements can appear multiple times except the PROC SURVEYMEANS statement and the WEIGHT statement, which can appear only once.

The rest of this section gives detailed syntax information for the BY, CLASS, CLUSTER, DOMAIN, RATIO, STRATA, VAR, and WEIGHT statements in alphabetical order after the description of the PROC SURVEYMEANS statement.

PROC SURVEYMEANS Statement

PROC SURVEYMEANS < *options* > < *statistic-keywords* > **;**

The PROC SURVEYMEANS statement invokes the procedure. In this statement, you identify the data set to be analyzed and specify sample design information. The DATA= option names the input data set to be analyzed. If your analysis includes a finite population correction factor, you can input either the sampling rate or the population total using the RATE= or TOTAL= option. If your design is stratified, with different sampling rates or totals for different strata, then you can input these stratum rates or totals in a SAS data set containing the stratification variables.

In the PROC SURVEYMEANS statement, you also can use statistic-keywords to specify statistics for the procedure to compute. Available statistics include the population mean and population total, together with their variance estimates and confidence limits. You can also request data set summary information and sample design information.

You can specify the following options in the PROC SURVEYMEANS statement:

ALPHA=α

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)$% confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed by PROC SURVEYMEANS. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MISSING

requests that the procedure treat missing values as a valid category for all categorical variables, which include categorical analysis variables, strata variables, cluster variables, and domain variables.

ORDER=DATA | FORMATTED | INTERNAL

specifies the order in which the values of the categorical variables are to be reported. The following shows how PROC SURVEYMEANS interprets values of the ORDER= option:

DATA	orders values according to their order in the input data set.

FORMATTED orders values by their formatted values. This order is operating environment dependent. By default, the order is ascending.

INTERNAL orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is operating environment dependent.

By default, ORDER=FORMATTED.

The ORDER= option applies to all the categorical variables. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

RATE=*value* | *SAS-data-set*
R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or names an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction for variance estimation. If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 4334 for more details.

The sampling rate *value* must be a nonnegative number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYMEANS will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

STACKING

requests the procedure to produce the output data sets using a stacking table structure, which was the default in releases prior to Version 9. The new default is to produce a rectangular table structure in the output data sets.

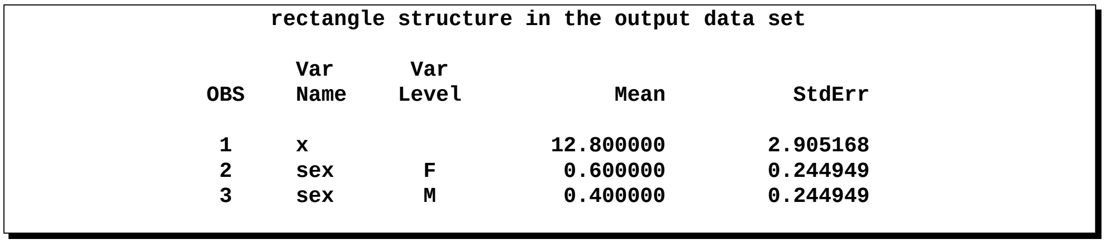
The STACKING option affects the following tables:

- Domain
- Ratio
- Statistics
- StrataInfo

When you use the ODS statement to create SAS data sets for these tables in the output, the data set structure can be either stacking or rectangular. A rectangular structure creates one observation for each analysis variable in the data set. However, if you use the STACKING option in Version 9, the procedure creates only one observation in the output data set for all analysis variables. The following example shows these two structures in output data sets.

```
data new;
  input sex$ x;
  datalines;
M 12
F 5
M 13
F 23
F 11
;

proc surveymeans data=new mean;
  ods output statistics=rectangle;
  run;

proc print data=rectangle;
  run;

proc surveymeans data=new mean stacking;
  ods output statistics=stacking;
  run;

proc print data=stacking;
  run;
```

Figure 70.6 shows the rectangular structure of the output data set for the statistics table.



```
                    rectangle structure in the output data set

                           Var      Var
                   OBS     Name     Level            Mean           StdErr

                     1     x                    12.800000         2.905168
                     2     sex        F          0.600000         0.244949
                     3     sex        M          0.400000         0.244949
```

Figure 70.6. Rectangular Structure in the Output Data Set

Figure 70.7 shows the stacking structure of the output data set for the statistics table.

```
                  stacking structure in the output data set

        OBS    x          x_Mean        x_StdErr    sex_F      sex_F_Mean

         1     x       12.800000        2.905168    sex=F        0.600000

        OBS    sex_F_StdErr    sex_M      sex_M_Mean    sex_M_StdErr

         1         0.244949    sex=M        0.400000        0.244949
```

Figure 70.7. Stacking Structure in the Output Data Set

TOTAL=*value* | *SAS-data-set*
N=*value* | *SAS-data-set*

specifies the total number of primary sampling units (PSUs) in the study population as a positive *value*, or names an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for variance estimation.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Specification of Population Totals and Sampling Rates" on page 4334 for more details.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

statistic-keywords

specifies the statistics for the procedure to compute. If you do not specify any statistic-keywords, PROC SURVEYMEANS computes the NOBS, MEAN, STDERR, and CLM statistics by default.

The statistics produced depend on the type of the analysis variable. If you name a numeric variable in the CLASS statement, then the procedure analyzes that variable as a categorical variable. The procedure always analyzes character variables as categorical. See the section "CLASS Statement" on page 4329 for more information.

PROC SURVEYMEANS computes MIN, MAX, and RANGE for numeric variables but not for categorical variables. For numeric variables, the keyword MEAN produces the mean, but for categorical variables it produces the proportion in each category or level. Also for categorical variables, the keyword NOBS produces the number of observations for each variable level, and the keyword NMISS produces the number of missing observations for each level. If you request the keyword NCLUSTER for a categorical variable, PROC SURVEYMEANS displays for each level the number of clusters with observations in that level. PROC SURVEYMEANS computes SUMWGT in the same way for both categorical and numeric variables, as the sum of the weights over all nonmissing observations.

PROC SURVEYMEANS performs univariate analysis, analyzing each variable separately. Thus the number of nonmissing and missing observations may not be the same for all analysis variables. See the section "Missing Values" on page 4333 for more information.

If you use the keyword RATIO without the keyword MEAN, the keyword MEAN is implied.

Other available statistics computed for a ratio are N, NCLU, SUMWGT, RATIO, STDERR, DF, T, PROBT, and CLM, as listed below. If no statistics are requested, the procedure will compute the ratio and its standard error by default for a RATIO statement.

The valid statistic-keywords are as follows:

ALL	all statistics listed
CLM	$100(1-\alpha)\%$ two-sided confidence limits for the MEAN, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$
CLSUM	$100(1-\alpha)\%$ two-sided confidence limits for the SUM, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$
CV	coefficient of variation for MEAN
CVSUM	coefficient of variation for SUM
DF	degrees of freedom for the *t* test
LCLM	$100(1-\alpha)\%$ one-sided lower confidence limit of the MEAN, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$
LCLMSUM	$100(1-\alpha)\%$ one-sided lower confidence limit of the SUM, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$
MAX	maximum value
MEAN	mean for a numeric variable, or the proportion in each category for a categorical variable
MIN	minimum value
NCLUSTER	number of clusters
NMISS	number of missing observations
NOBS	number of nonmissing observations
RANGE	range, MAX$-$MIN
RATIO	ratio of means or proportions
STD	standard deviation of the SUM. When you request SUM, the procedure computes STD by default.
STDERR	standard error of the MEAN or RATIO. When you request MEAN or RATIO, the procedure computes STDERR by default.

SUM weighted sum, $\sum w_i y_i$, or estimated population total when the appropriate sampling weights are used

SUMWGT sum of the weights, $\sum w_i$

T *t*-value and its corresponding *p*-value with DF degrees of freedom for

$$H_0 : \theta = 0$$

where θ is the population mean or the population ratio

UCLM $100(1 - \alpha)\%$ one-sided upper confidence limit of the MEAN, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$

UCLMSUM $100(1 - \alpha)\%$ one-sided upper confidence limit of the SUM, where α is determined by the ALPHA= option described on page 4323, and the default is $\alpha = 0.05$

VAR variance of the MEAN or RATIO

VARSUM variance of the SUM

See the section “Statistical Computations” on page 4336 for details on how PROC SURVEYMEANS computes these statistics.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYMEANS to obtain separate analyses on observations in groups defined by the BY variables.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. You should use the DOMAIN statement to obtain domain analysis.

When a BY statement appears, the procedure expects the input data sets to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Use the BY statement options NOTSORTED or DESCENDING in the BY statement. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS | **CLASSES** *variables* **;**

The CLASS statement names variables to be analyzed as categorical variables. For categorical variables, PROC SURVEYMEANS estimates the proportion in each category or level, instead of the overall mean. PROC SURVEYMEANS always analyzes character variables as categorical. If you want categorical analysis for a numeric variable, you must include that variable in the CLASS statement.

The CLASS *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLASS variables determine the categorical variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can use multiple CLASS statements to specify categorical variables.

When you specify class variables, you may use the SAS system option SUMSIZE= to limit (or to specify) the amount of memory that is available for data analysis. Refer to the chapter on SAS System options in *SAS Language Reference: Dictionary* for a description of the SUMSIZE= option.

CLUSTER Statement

CLUSTER | **CLUSTERS** *variables* **;**

The CLUSTER statement names variables that identify the clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement. See the section “Primary Sampling Units (PSUs)” on page 4335 for more information.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can use multiple CLUSTER statements to specify cluster variables. The procedure uses variables from all CLUSTER statements to create clusters.

DOMAIN Statement

DOMAIN | SUBGROUP *variables* < *variable*∗*variable* *variable*∗*variable*∗*variable* . . . > **;**

The DOMAIN statement requests analysis for subpopulations, or domains, in addition to analysis for the entire study population. The DOMAIN statement names the variables that identify domains, which are called domain variables.

It is common practice to compute statistics for domains. The formation of these domains may be unrelated to the sample design. Therefore, the sample sizes for the domains are random variables. In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement.

Note that a DOMAIN statement is different from a BY statement. In a BY statement, you treat the sample sizes as fixed in each subpopulation, and you perform analysis within each BY group independently. See the section "Domain Statistics" on page 4342 for more details.

A domain variable can be either character or numeric. However, the procedure treats domain variables as categorical variables. If a variable appears by itself in a DOMAIN statement, each level of this variable determines a domain in the study population. If two or more variables are joined by asterisks (*), then every possible combination of levels of the variables determines a domain. The procedure performs a descriptive analysis within each domain defined by the domain variables.

The formatted values of the domain variables determine the categorical variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

RATIO Statement

RATIO < *'label'* > *variables / variables* **;**

The RATIO statement requests ratio analysis for means or proportions of analysis variables. A ratio statement names the variables whose means will be used as numerators or denominators in a ratio. Variables appearing before the slash (/), called *numerator variables*, are used for numerators. Variables appearing after the slash (/), called *denominator variables*, are used for denominators. These *variables* can be any number of analysis variables, either continuous or categorical, in the input data set.

You can optionally specify a label for each RATIO statement to identify the ratios in the output. Labels must be enclosed in single quotes.

If a RATIO statement does not have any numerator variable or denominator variable specified, the RATIO statement will be ignored.

A numerator or denominator variable must be an analysis variable. That is, if there is a VAR statement, then a numerator or denominator variable must appear in the VAR statement. If there is no VAR statement, a numerator or denominator variable must be on the default analysis variable list (see the section "VAR Statement" on page 4332). If a numerator or denominator variable is not an analysis variable, it is ignored.

The computation of ratios depends on whether the numerator and denominator variables are continuous or categorical.

For continuous variables, ratios are calculated with the mean of the variables. For example, for continuous variables X, Y, Z, and T, the following RATIO statement requests the procedure to analyze the ratios $\bar{x}/\bar{z}$, $\bar{x}/\bar{t}$, $\bar{y}/\bar{z}$, and $\bar{y}/\bar{t}$:

```
ratio x y / z t;
```

If a continuous variable appears as both a numerator and a denominator variable, the ratio of this variable itself is ignored.

For categorical variables, ratios are calculated with the proportions for the categories of a categorical variable. For example, if categorical variable Gender has values "Male" and "Female," with proportions p_m = Pr(Gender="Male") and p_f = Pr(Gender="Female"), and Y is a continuous variable, then the following RATIO statement requests the procedure to analyze the ratios p_m/p_f, p_f/p_m, $\bar{y}/p_m$, and $\bar{y}/p_f$:

```
ratio Gender y / Gender;
```

If a categorical variable appears as both a numerator and a denominator variable, then the ratios of the proportions for all categories are computed, except the ratio of each category with itself.

You may have more than one RATIO statement. Each RATIO statement produces ratios independently using its own numerator and denominator variables. Each RATIO statement also produces its own ratio analysis table.

Available statistics for a ratio are

- N, number of observations used to compute the ratio
- NCLU, number of clusters
- SUMWGT, sum of weights
- RATIO, ratio
- STDERR, standard error of ratio
- VAR, variance of ratio
- T, *t*-value of ratio
- PROBT, *p*-value of *t*
- DF, degrees of freedom of *t*
- CLM, two-sided confidence limits of ratio
- UCLM, one-sided upper confidence limit of ratio

- LCLM, one-sided lower confidence limit of ratio

The procedure will calculate these statistics based on the statistic-keywords described on page 4326 which you specified in the PROC statement. If a statistic-keyword is not appropriate for RATIO statement, that statistic-keyword is ignored. If no valid statistics are requested for a RATIO statement, the procedure will compute the ratio and its standard error by default.

Note that ratios within a domain are currently not available.

When calculating the means or proportions for the numerator and denominator variables in a ratio, an observation is excluded if it has a missing value in either the continuous numerator variable or the denominator variable. An observation with missing values is also excluded for the categorical numerator or denominator variables, unless the MISSING option is used.

STRATA Statement

STRATA | **STRATUM** *variables* < */ option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of categories of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Specification of Population Totals and Sampling Rates" on page 4334 for more information.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the STRATA variables determine the levels. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *SAS Procedures Guide*.

You can specify the following option in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which includes values of the STRATA variables and sampling rates for each stratum. This table also provides the number of observations and number of clusters for each stratum and analysis variable. See the section "Displayed Output" on page 4345 for more details.

VAR Statement

VAR *variables* **;**

The VAR statement names the variables to be analyzed.

If you want a categorical analysis for a numeric variable, you must also name that variable in the CLASS statement. For categorical variables, PROC SURVEYMEANS estimates the proportion in each category or level, instead of the overall mean. Character variables are always analyzed as categorical variables. See the section "CLASS Statement" on page 4329 for more information.

If you do not specify a VAR statement, then PROC SURVEYMEANS analyzes all variables in the DATA= input data set, except those named in the BY, CLUSTER, STRATA, and WEIGHT statements.

WEIGHT Statement

WEIGHT | **WGT** *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric. If you do not specify a WEIGHT statement, PROC SURVEYMEANS assigns all observations a weight of 1. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

Details

Missing Values

When computing statistics for an analysis variable, PROC SURVEYMEANS omits observations with missing values for that variable. The procedure bases statistics for each variable only on observations that have nonmissing values for that variable. If you specify the MISSING option described on page 4323 in the PROC SURVEYMEANS statement, the procedure treats missing values of a categorical variable as a valid category.

An observation is also excluded if it has a missing value for any STRATA or CLUSTER variable, unless the MISSING option is used.

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then PROC SURVEYMEANS excludes that observation from the analysis.

The procedure performs univariate analysis and analyzes each VAR variable separately. Thus, the number of missing observations may be different for different variables. You can specify the keyword NMISS in the PROC SURVEYMEANS statement to display the number of missing values for each analysis variable in the "Statistics" table.

If you have missing values in your survey data for any reason (such as nonresponse), this can compromise the quality of your survey results. An observation without missing values is called a complete respondent, and an observation with missing values is called an incomplete respondent. If the complete respondents are different from the incomplete respondents with regard to a survey effect or outcome, then the survey estimates will be biased and will not accurately represent the survey population. A variety of techniques in sample design and survey operations can reduce nonresponse. Once data collection is complete, you can use imputation to replace missing values with acceptable values, and you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYMEANS. Refer to Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more details.

PROC SURVEYMEANS assumes that missing data are missing at random, because the patterns of missing data are unknown. Therefore, PROC SURVEYMEANS excludes those observations with missing values.

If there is evidence indicating that the missing data are not at random, for example, if complete respondents are different from incomplete respondents for your study, you can use the DOMAIN statement to compute the descriptive statistics among complete respondents from your survey data without imputation on incomplete respondents. See Example 70.4 on page 4358.

If missing values result in empty strata in the sample, then they will have an impact on the statistical computation, which uses the total number of strata. If all the observations in a stratum have missing weights or missing values for the current analysis variable, this stratum is an *empty stratum*. For example,

```
data new;
   input stratum y z w;
   datalines;
1 . 13 40
1 2  9  .
1 .  5 25
2 5 10 20
2 8 60 15
;
proc surveymeans df mean nobs nmiss;
   strata stratum;
   var y z;
   weight w;
run;
```

You analyze variable Y and Z, with weight variable W and stratum variable STRATUM. For variable Y, all observations have missing values or missing weights in STRATUM=1, therefore, the analysis for variable Y uses only observations in STRATUM=2. Thus, for variable Y, STRATUM=1 is an empty stratum and STRATUM=2 is a non-empty stratum. Note, however, that STRATUM=1 is a non-empty stratum for variable Z.

If your sample design contains stratification, PROC SURVEYMEANS analyzes only the data in non-empty strata. Therefore, the total number of strata for an analysis variable means the total number of *non-empty* strata. In this example, the total number of strata for Y and Z is one and two, respectively.

Survey Data Analysis

Specification of Population Totals and Sampling Rates

If your analysis should include a finite population correction (*fpc*), you can input either the sampling rate or the population total using the RATE= option or the TOTAL= option. (You cannot specify both of these options in the same PROC SURVEYMEANS statement.) If you do not specify one of these options, the procedure does not use the *fpc* when computing variance estimates. For fairly small sampling fractions, it is appropriate to ignore this correction. Refer to Cochran (1977) and Kish (1965).

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population. See the section "Primary Sampling Units (PSUs)" on page 4335 for more details. For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you should use the RATE=*value* option or the TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in the strata, then you can use the RATE=*SAS-data-set* option or the TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Or if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYMEANS will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or primary sampling units (PSUs), in your sample design, the procedure estimates variance from the variation among PSUs. See the section "Variance and Standard Error of the Mean" on page 4338 and the section "Variance and Standard Deviation of the Total" on page 4341. You can use the CLUSTER statement to identify the first stage clusters in your design. PROC SURVEYMEANS assumes that each cluster represents a PSU in the sample and that each observation is an element of a PSU. If you do not specify a CLUSTER statement, the procedure treats each observation as a PSU.

Domain Analysis

It is common practice to compute statistics for subpopulations, or domains, in addition to computing statistics for the entire study population. Analysis for domains using the entire sample is called *domain analysis* (subgroup analysis, subpopulation analysis, subdomain analysis). The formation of these subpopulations of interest may be unrelated to the sample design. Therefore, the sample sizes for the subpopulations may actually be random variables.

In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement. Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. For more detailed information about domain analysis, refer to Kish (1965).

Statistical Computations

The SURVEYMEANS procedure uses the Taylor expansion method to estimate sampling errors of estimators based on complex sample designs. This method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971, Fuller 1975). When there are clusters, or PSUs, in the sample design, the procedure estimates variance from the variation among PSUs. When the design is stratified, the procedure pools stratum variance estimates to compute the overall variance estimate. For *t* tests of the estimates, the degrees of freedom equals the number of clusters minus the number of strata in the sample design.

For a multistage sample design, the variance estimation method depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling. This variance estimation method assumes that the first-stage sampling fraction is small, or the first-stage sample is drawn with replacement, as it often is in practice.

Quite often in complex surveys, respondents have unequal weights, which reflect unequal selection probabilities and adjustments for nonresponse. In such surveys, the appropriate sampling weights must be used to obtain valid estimates for the study population.

For more information on the analysis of sample survey data, refer to Lee, Forthoffer, and Lorimor (1989), Cochran (1977), Kish (1965), and Hansen, Hurwitz, and Madow (1953).

Definition and Notation

For a stratified clustered sample design, together with the sampling weights, the sample can be represented by an $n \times (P+1)$ matrix

$$\begin{aligned}(\mathbf{w}, \mathbf{Y}) &= (w_{hij}, \mathbf{y}_{hij}) \\ &= \left(w_{hij}, y_{hij}^{(1)}, y_{hij}^{(2)}, \ldots, y_{hij}^{(P)}\right)\end{aligned}$$

where

- $h = 1, 2, \ldots, H$ is the stratum number, with a total of H strata
- $i = 1, 2, \ldots, n_h$ is the cluster number within stratum h, with a total of n_h clusters
- $j = 1, 2, \ldots, m_{hi}$ is the unit number within cluster i of stratum h, with a total of m_{hi} units
- $p = 1, 2, \ldots, P$ is the analysis variable number, with a total of P variables
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total number of observations in the sample
- w_{hij} denotes the sampling weight for observation j in cluster i of stratum h
- $\mathbf{y}_{hij} = \left(y_{hij}^{(1}), y_{hij}^{(2)}, \ldots, y_{hij}^{(P)} \right)$ are the observed values of the analysis variables for observation j in cluster i of stratum h, including both the values of numerical variables and the values of indicator variables for levels of categorical variables.

For a categorical variable C, let l denote the number of levels of C, and denote the level values as $c_1, c_2, \ldots, c_l$. Then there are l indicator variables associated with these levels. That is, for level $C = c_k$ $(k = 1, 2, \ldots, l)$, a $y^{(q)}$ $(q \in \{1, 2, \ldots, P\})$ contains the values of the indicator variable for the category $C = c_k$, with the value of observation j in cluster i of stratum h:

$$y_{hij}^{(q)} \quad = \quad I_{\{C=c_k\}}(h,i,j) = \begin{cases} 1 & \text{if } C_{hij} = c_k \\ 0 & \text{otherwise} \end{cases}$$

Therefore, the total number of analysis variables, P, is the total number of numerical variables plus the total number of levels of all categorical variables.

Also, f_h denotes the sampling rate for stratum h. You can use the TOTAL= option or the RATE= option to input population totals or sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 4334 for details. If you input stratum totals, PROC SURVEYMEANS computes f_h as the ratio of the stratum sample size to the stratum total. If you input stratum sampling rates, PROC SURVEYMEANS uses these values directly for f_h. If you do not specify the TOTAL= option or the RATE= option, then the procedure assumes that the stratum sampling rates f_h are negligible, and a finite population correction is not used when computing variances.

This notation is also applicable to other sample designs. For example, for a sample design without stratification, you can let $H = 1$; for a sample design without clusters, you can let $m_{hi} = 1$ for every h and i.

Mean

When you specify the keyword MEAN, the procedure computes the estimate of the mean (mean per element) from the survey data. Also, the procedure computes the mean by default if you do not specify any statistic-keywords in the PROC SURVEYMEANS statement.

PROC SURVEYMEANS computes the estimate of the mean as

$$\widehat{\bar{Y}} \quad = \quad \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij} \, y_{hij} \right) / \; w_{\cdots}$$

where

$$w_{\cdots} \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij}$$

is the sum of the weights over all observations in the sample.

Variance and Standard Error of the Mean

When you specify the keyword STDERR, the procedure computes the standard error of the mean. Also, the procedure computes the standard error by default if you specify the keyword MEAN, or if you do not specify any statistic-keywords in the PROC SURVEYMEANS statement. The keyword VAR requests the variance of the mean.

PROC SURVEYMEANS uses the Taylor series expansion theory to estimate the variance of the mean $\widehat{\bar{Y}}$. The procedure computes the estimated variance as

$$\widehat{V}(\widehat{\bar{Y}}) \quad = \quad \sum_{h=1}^{H} \widehat{V_h}(\widehat{\bar{Y}})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{\bar{Y}}) &= \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (e_{hi\cdot} - \bar{e}_{h\cdot\cdot})^2 \\
e_{hi\cdot} &= \left(\sum_{j=1}^{m_{hi}} w_{hij} \, (y_{hij} - \widehat{\bar{Y}}) \right) / \; w_{\cdots} \\
\bar{e}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} e_{hi\cdot} \right) / \; n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{\bar{Y}}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard error of the mean is the square root of the estimated variance.

$$\text{StdErr}(\widehat{\bar{Y}}) = \sqrt{\widehat{V}(\widehat{\bar{Y}})}$$

Ratio

When you use a RATIO statement, the procedure produces statistics requested by the statistics-keywords in the PROC SURVEYMEANS statement.

Suppose that you want to calculate the ratio of variable Y over variable X. Let x_{hij} be the value of variable X for the jth member in cluster i in the hth stratum.

The ratio of Y over X is

$$\widehat{R} \;=\; \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} \; w_{hij}\; y_{hij}}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} \; w_{hij}\; x_{hij}}$$

PROC SURVEYMEANS uses the Taylor series expansion method to estimate the variance of the ratio $\widehat{R}$ as

$$\widehat{V}(\widehat{R}) \;=\; \sum_{h=1}^{H} \widehat{V_h}(\widehat{R})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{R}) &= \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (g_{hi\cdot} - \bar{g}_{h\cdot\cdot})^2 \\
g_{hi\cdot} &= \frac{\sum_{j=1}^{m_{hi}} w_{hij}\; (y_{hij} - x_{hij}\widehat{R})}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} \; w_{hij}\; x_{hij}} \\
\bar{g}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} g_{hi\cdot} \right) \;/\; n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{R}) \;=\; \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard error of the ratio is the square root of the estimated variance.

$$\text{StdErr}(\widehat{R}) = \sqrt{\widehat{V}(\widehat{R})}$$

t Test for the Mean

If you specify the keyword T, PROC SURVEYMEANS computes the *t*-value for testing that the population mean equals zero, $H_0 : \bar{Y} = 0$. The test statistic equals

$$t(\widehat{\bar{Y}}) \;=\; \widehat{\bar{Y}} \;/\; \text{StdErr}(\widehat{\bar{Y}})$$

The two-sided *p*-value for this test is

$$\text{Prob}(\; |T| > |t(\widehat{\bar{Y}})| \;)$$

where T is a random variable with the *t* distribution with *df* degrees of freedom.

PROC SURVEYMEANS calculates the degrees of freedom for the *t* test as the number of clusters minus the number of strata. If there are no clusters, then *df* equals the number of observations minus the number of strata. If the design is not stratified, then *df* equals the number of clusters minus one. The procedure displays *df* for the *t* test if you specify the keyword DF in the PROC SURVEYMEANS statement.

If missing values or missing weights are present in your data, the number of strata, the number of observations, and the number of clusters are counted based on the observations in non-empty strata. See the section "Missing Values" on page 4333 for details. For degrees of freedom in domain analysis, see the section "Domain Statistics" on page 4342.

Confidence Limits for the Mean

If you specify the keyword CLM, the procedure computes two-sided confidence limits for the mean. Also, the procedure includes the confidence limits by default if you do not specify any statistic-keywords in the PROC SURVEYMEANS statement.

The confidence coefficient is determined by the value of the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. The confidence limits are computed as

$$\widehat{\bar{Y}} \quad \pm \quad \text{StdErr}(\widehat{\bar{Y}}) \ \cdot \ t_{df,\ \alpha/2}$$

where $\widehat{\bar{Y}}$ is the estimate of the mean, $\text{StdErr}(\widehat{\bar{Y}})$ is the standard error of the mean, and $t_{df,\ \alpha/2}$ is the $100(1-\alpha/2)$ percentile of the *t* distribution with *df* calculated as described in the section "*t* Test for the Mean" on page 4339.

If you specify the keyword UCLM, the procedure computes the one-sided upper $100(1-\alpha)$ confidence limit for the mean:

$$\widehat{\bar{Y}} \quad + \quad \text{StdErr}(\widehat{\bar{Y}}) \ \cdot \ t_{df,\ \alpha}$$

If you specify the keyword LCLM, the procedure computes the one-sided lower $100(1-\alpha)$ confidence limit for the mean:

$$\widehat{\bar{Y}} \quad - \quad \text{StdErr}(\widehat{\bar{Y}}) \ \cdot \ t_{df,\ \alpha}$$

Coefficient of Variation

If you specify the keyword CV, PROC SURVEYMEANS computes the coefficient of variation, which is the ratio of the standard error of the mean to the estimated mean.

$$cv(\bar{Y}) \quad = \quad \text{StdErr}(\widehat{\bar{Y}}) \ / \ \widehat{\bar{Y}}$$

If you specify the keyword CVSUM, PROC SURVEYMEANS computes the coefficient of variation for the estimated total, which is the ratio of the standard deviation of the sum to the estimated total.

$$cv(Y) \quad = \quad \text{Std}(\widehat{Y}) \ / \ \widehat{Y}$$

Proportions

If you specify the keyword MEAN for a categorical variable, PROC SURVEYMEANS estimates the proportion, or relative frequency, for each level of the categorical variable. If you do not specify any statistic-keywords in the PROC SURVEYMEANS statement, the procedure estimates the proportions for levels of the categorical variables, together with their standard errors and confidence limits.

The procedure estimates the proportion in level c_k for variable C as

$$\hat{p} = \frac{\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \; w_{hij} \; y_{hij}^{(q)}}{\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \; w_{hij}}$$

where $y_{hij}^{(q)}$ is the value of the indicator function for level $C = c_k$, defined in the section "Definition and Notation" on page 4336, and $y_{hij}^{(q)}$ equals 1 if the observed value of variable C equals c_k, and $y_{hij}^{(q)}$ equals 0 otherwise. Since the proportion estimator is actually an estimator of the mean for an indicator variable, the procedure computes its variance and standard error according to the method outlined in the section "Variance and Standard Error of the Mean" on page 4338. Similarly, the procedure computes confidence limits for proportions as described in the section "Confidence Limits for the Mean" on page 4340.

Total

If you specify the keyword SUM, the procedure computes the estimate of the population total from the survey data. The estimate of the total is the weighted sum over the sample.

$$\widehat{Y} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \; w_{hij} \; y_{hij}$$

For a categorical variable level, $\widehat{Y}$ estimates its total frequency in the population.

Variance and Standard Deviation of the Total

When you specify the keyword STD or the keyword SUM, the procedure estimates the standard deviation of the total. The keyword VARSUM requests the variance of the total.

PROC SURVEYMEANS estimates the variance of the total as

$$\widehat{V}(\widehat{Y}) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{Y})$$

where if $n_h > 1$,

$$\widehat{V_h}(\widehat{Y}) \quad = \quad \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (y_{hi\cdot} - \bar{y}_{h\cdot\cdot})^2$$

$$\begin{aligned} y_{hi\cdot} &= \sum_{j=1}^{m_{hi}} w_{hij}\, y_{hij} \\ \bar{y}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} y_{hi\cdot} \right) / \; n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{Y}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

The standard deviation of the total equals

$$\text{Std}(\widehat{Y}) = \sqrt{\widehat{V}(\widehat{Y})}$$

Confidence Limits of a Total

If you specify the keyword CLSUM, the procedure computes confidence limits for the total. The confidence coefficient is determined by the value of the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. The confidence limits are computed as

$$\widehat{Y} \quad \pm \quad \text{Std}(\widehat{Y}) \; \cdot \; t_{df,\; \alpha/2}$$

where $\widehat{Y}$ is the estimate of the total, Std$(\widehat{Y})$ is the estimated standard deviation, and $t_{df,\; \alpha/2}$ is the $100(1 - \alpha/2)$ percentile of the *t* distribution with *df* calculated as described in the section "*t* Test for the Mean" on page 4339.

If you specify the keyword UCLSUM, the procedure computes the one-sided upper $100(1 - \alpha)$ confidence limit for the sum:

$$\widehat{Y} \quad + \quad \text{Std}(\widehat{Y}) \; \cdot \; t_{df,\; \alpha}$$

If you specify the keyword LCLSUM, the procedure computes the one-sided lower $100(1 - \alpha)$ confidence limit for the sum:

$$\widehat{Y} \quad - \quad \text{Std}(\widehat{Y}) \; \cdot \; t_{df,\; \alpha}$$

Domain Statistics

When you use a DOMAIN statement to request a domain analysis, the procedure computes the requested statistics for each domain.

For a domain D, let I_D be the corresponding indicator variable:

$$I_D(h, i, j) \quad = \begin{cases} 1 & \text{if observation } (h, i, j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

Let

$$z_{hij} \quad = \quad y_{hij} I_D(h,i,j) \quad = \begin{cases} y_{hij} & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

The requested statistics for variable y in domain D are computed based on the values of z.

Domain Mean The estimated mean of y in the domain D is

$$\widehat{\bar{Y}_D} \quad = \quad \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij} \; z_{hij} \right) / \; v_{\cdots}$$

where

$$\begin{aligned} v_{hij} \quad &= \quad w_{hij} I_D(h,i,j) = \begin{cases} w_{hij} & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases} \\ v_{\cdots} \quad &= \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij} \end{aligned}$$

The variance of $\widehat{\bar{Y}_D}$ is estimated by

$$\widehat{V}(\widehat{\bar{Y}_D}) \quad = \quad \sum_{h=1}^{H} \widehat{V_h}(\widehat{\bar{Y}_D})$$

where if $n_h > 1$,

$$\begin{aligned} \widehat{V_h}(\widehat{\bar{Y}_D}) \quad &= \quad \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (r_{hi\cdot} - \bar{r}_{h\cdot\cdot})^2 \\ r_{hi\cdot} \quad &= \quad \left(\sum_{j=1}^{m_{hi}} v_{hij} \; (z_{hij} - \widehat{\bar{Y}_D}) \right) / \; v_{\cdots} \\ \bar{r}_{h\cdot\cdot} \quad &= \quad \left(\sum_{i=1}^{n_h} r_{hi\cdot} \right) / \; n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{\bar{Y}_D}) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

Domain Total The estimated total in domain D is

$$\widehat{Y}_D \quad = \quad \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij} \; z_{hij}$$

and its estimated variance is

$$\widehat{V}(\widehat{Y}_D) = \sum_{h=1}^{H} \widehat{V}_h(\widehat{Y}_D)$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V}_h(\widehat{Y}_D) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (z_{hi\cdot} - \bar{z}_{h\cdot\cdot})^2 \\
z_{hi\cdot} &= \sum_{j=1}^{m_{hi}} v_{hij}\, z_{hij} \\
\bar{z}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} z_{hi\cdot} \right) / \, n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V}_h(\widehat{Y}_D) \quad = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 < h' < H \end{cases}$$

Degrees of Freedom For domain analysis, PROC SURVEYMEANS computes the degrees of freedom for *t* tests as the number of clusters in the non-empty strata minus the number of non-empty strata. When the sample design has no clusters, the degrees of freedom equals the number of observations in non-empty strata minus the number of non-empty strata. As discussed in the section "Missing Values" on page 4333, missing values and missing weights can result in empty strata. In domain analysis, an empty stratum can also occur when the stratum contains no observations in the specified domain. If no observations in a whole stratum belong to a domain, then this stratum is called an empty stratum for that domain.

For example,

```
data new;
   input str clu y w d;
   datalines;
1 1 . 40 9
1 2 2  . 9
1 3 . 25 9
2 4 5 20 9
2 5 8 15 9
3 6 5 30 7
3 7 9 89 7
3 8 6 23 7
;
proc surveymeans df nobs nclu nmiss;
   strata str;
   cluster clu;
   var y;
   weight w;
   domain d;
run;
```

Table 70.2. Calculations of df for Y

	Domain D=7	**Domain** D=9
Non Empty Strata	STR=3	STR=2
Clusters Used in the Analysis	CLU=6, CLU=7, and CLU=8	CLU=4 and CLU=5
df	$3 - 1 = 2$	$2 - 1 = 1$

Although there are three strata in the data set, STR=1 is an empty stratum for variable Y because of missing values and missing weights. In addition, no observations in stratum STR=3 belong to domain D=9. Therefore, STR=3 becomes an empty stratum as well for variable Y in domain D=9. As a result, the total number of non-empty strata for domain D=9 is one. The non-empty stratum for domain D=9 and variable Y is stratum STR=2. The total number of clusters for domain D=9 is two, which belong to stratum STR=2. Thus, for variable Y in domain D=9, the degrees of freedom for the t tests of the domain mean is $df = 2 - 1 = 1$. Similarly, for domain D=7, strata STR=1 and STR=2 are both empty strata, so the total number of strata is one (STR=3), and the total number of clusters is three (CLU=6, CLU=7, and CLU=8). Table 70.2 illustrates how domains affect the total number of clusters and total number of strata in the df calculation. Figure 70.8 shows the df computed by the procedure.

```
                              The SURVEYMEANS Procedure

                                Domain Analysis: d

       d     Variable                 N           N Miss          Clusters         DF
  ------------------------------------------------------------------------------------
       7     y                        3                0                 3          6
       9     y                        2                2                 2          4
  ------------------------------------------------------------------------------------
```

Figure 70.8. Degrees of Freedoms in Domain Analysis

Output

Output Data Sets

Output data sets from PROC SURVEYMEANS are produced using ODS (Output Delivery System). ODS encompasses more than just the production of output data sets. For example, you can use ODS to manipulate the format of your output, the headers and titles of the tables, and the order of the columns in a table. For a more detailed description on using ODS, see Chapter 14, “Using the Output Delivery System.”

Displayed Output

By default PROC SURVEYMEANS displays a “Data Summary” table and a “Statistics” table. If you specify CLASS variables, or if you specify any character variables in the VAR statement, then the procedure displays a “Class Level Information” table. If you specify the LIST option in the STRATA statement, then the procedure displays a “Stratum Information” table. If you have a DOMAIN statement, the procedure displays a “Domain Analysis” table. If you have a RATIO statement, the procedure displays a “Ratio Analysis” table.

Data and Sample Design Summary

The “Data Summary” table provides information on the input data set and the sample design. This table displays the total number of valid observations, where an observation is considered *valid* if it has nonmissing values for all procedure variables other than the analysis variables; that is, for all specified STRATA, CLUSTER, and WEIGHT variables. This number may differ from the number of nonmissing observations for an individual analysis variable, which the procedure displays in the “Statistics” table. See the section “Missing Values” on page 4333 for more information.

PROC SURVEYMEANS displays the following information in the “Data Summary” table:

- Number of Strata, if you specify a STRATA statement
- Number of Clusters, if you specify a CLUSTER statement
- Number of Observations, which is the total number of valid observations
- Sum of Weights, which is the sum over all valid observations, if you specify a WEIGHT statement

Class Level Information

If you use a CLASS statement to name classification variables for categorical analysis, or if you list any character variables in the VAR statement, then PROC SURVEYMEANS displays a “Class Level Information” table. This table contains the following information for each classification variable:

- Class Variable, which lists each CLASS variable name
- Levels, which is the number of values or levels of the classification variable
- Values, which lists the values of the classification variable. The values are separated by a white space character; therefore, to avoid confusion, you should not include a white space character within a classification variable value.

Stratum Information

If you specify the LIST option in the STRATA statement, PROC SURVEYMEANS displays a “Stratum Information” table. This table displays the number of valid observations in each stratum, as well as the number of nonmissing stratum observations for each analysis variable. The “Stratum Information” table provides the following for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum

- Population Total, if you specify the TOTAL= option
- Sampling Rate, if you specify the TOTAL= option or the RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of valid observations in the stratum.
- N Obs, which is the number of valid observations
- Variable, which lists each analysis variable name
- Levels, which identifies each level for categorical variables
- N, which is the number of nonmissing observations for the analysis variable
- Clusters, which is the number of clusters, if you specify a CLUSTER statement

Statistics

The "Statistics" table displays all of the statistics that you request with statistic-keywords described on page 4326 in the PROC SURVEYMEANS statement. If you do not specify any statistic-keywords, then by default this table displays the following information for each analysis variable: the sample size, the mean, the standard error of the mean, and the confidence limits for the mean. The "Statistics" table may contain the following information for each analysis variable, depending on which statistic-keywords you request:

- Variable name
- Level, which identifies each level for categorical variables
- N, which is the number of nonmissing observations
- N Miss, which is the number of missing observations
- Minimum
- Maximum
- Range
- Number of Clusters
- Sum of Weights
- DF, which is the degrees of freedom for the t test
- Mean
- Std Error of Mean, which is the standard error of the mean
- Var of Mean, which is the variance of the mean
- t Value, for testing H_0 : population MEAN $= 0$
- Pr $>$ $|\ t\ |$, which is the two-sided p-value for the t test
- $100(1-\alpha)$% CL for Mean, which are two-sided confidence limits for the mean
- $100(1-\alpha)$% Upper CL for Mean, which are one-sided upper confidence limits for the mean
- $100(1-\alpha)$% Lower CL for Mean, which are one-sided lower confidence limits for the mean

- Coeff of Variation, which is the coefficients of variation for the mean and the sum
- Sum
- Std Dev, which is the standard deviation of the sum
- Var of Sum, which is the variance of the sum
- $100(1-\alpha)$% CL for Sum, which are two-sided confidence limits for the sum
- $100(1-\alpha)$% Upper CL for Sum, which are one-sided upper confidence limits for the sum
- $100(1-\alpha)$% Lower CL for Sum, which are one-sided lower confidence limits for the Sum

Domain Analysis

If you use a DOMAIN statement, the procedure displays statistics in each domain in a "Domain Analysis" table. A "Domain Analysis" table contains all the columns in the "Statistics" table, plus columns of domain variable values.

Note that depending on how you define the domains with domain variables, the procedure may produce more than one "Domain Analysis" table. For example, in the following DOMAIN statement

```
domain A B*C*D A*C C;
```

you use four definitions to define domains:

- A: all the levels of A
- C: all the levels of C
- A*C: all the interactive levels of A and C
- B*C*D: all the interactive levels of B, C, and D

The procedure displays four "Domain Analysis" tables, one for each domain definition. However, if you use ODS output statement to create an output data set for domain analysis, the output data set contains a variable Domain whose values are these domain definitions.

Ratio Analysis

The "Ratio Analysis" table displays all of the statistics that you request with statistic-keywords in the PROC statement described on page 4326. If you do not specify any statistic-keywords, then by default this table displays the ratio and its standard error. The "Ratio Analysis" table may contain the following information for each ratio, depending on which statistic-keywords you request:

- Numerator, which identifies the numerator variable of the ratio
- Denominator, which identifies the denominator variable of the ratio

- N, which is the number of observations used in the ratio analysis
- number of Clusters
- Sum of Weights
- DF, which is the degrees of freedom for the t test
- Ratio
- Std Error of Ratio, which is the standard error of the ratio
- Var of Ratio, which is the variance of the ratio
- t Value, for testing H_0 : population RATIO $= 0$
- Pr $>$ $\mid t \mid$, which is the two-sided p-value for the t test
- $100(1-\alpha)$% CL for Ratio, which are two-sided confidence limits for the Ratio
- Upper $100(1-\alpha)$% CL for Ratio, which are one-sided upper confidence limits for the Ratio
- Lower $100(1-\alpha)$% CL for Ratio, which are one-sided lower confidence limits for the Ratio

When you use the ODS output statement to create an output data set, if you use labels for your RATIO statement, these labels are saved in a variable Ratio Statement in the output data set.

ODS Table Names

PROC SURVEYMEANS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 70.3. ODS Tables Produced in PROC SURVEYMEANS

ODS Table Name	Description	Statement	Option
ClassVarInfo	Class level information	CLASS	default
Domain	Statistics in domains	DOMAIN	default
Ratio	Statistics for ratios	RATIO	default
Statistics	Statistics	PROC	default
StrataInfo	Stratum information	STRATA	LIST
Summary	Data summary	PROC	default

For example, the following statements create an output data set named MyStrata, which contains the "StrataInfo" table, and an output data set named MyStat, which contains the "Statistics" table for the ice cream study discussed in the section "Stratified Sampling" on page 4318:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   strata Grade / list;
   var Spending Group;
   weight Weight;
   ods output StrataInfo = MyStrata
              Statistics = MyStat;
run;
```

Examples

The “Getting Started” section on page 4315 contains examples of analyzing data from simple random sampling and stratified simple random sample designs. This section provides more examples that illustrate how to use PROC SURVEYMEANS.

Example 70.1. Stratified Cluster Sample Design

Consider the example in the section “Stratified Sampling” on page 4318. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on the average, and what percentage of students spend at least $10 weekly for ice cream.

The example in the section “Stratified Sampling” on page 4318 assumes that the sample of students was selected using a stratified simple random sample design. This example shows analysis based on a more complex sample design.

Suppose that every student belongs to a study group and that study groups are formed within each grade level. Each study group contains between two and four students. Table 70.4 shows the total number of study groups for each grade.

Table 70.4. Study Groups and Students by Grade

Grade	Number of Study Groups	Number of Students
7	608	1,824
8	252	1,025
9	403	1,151
Total	617	4,000

It is quicker and more convenient to collect data from students in the same study group than to collect data from students individually. Therefore, this study uses a stratified clustered sample design. The primary sampling units, or clusters, are study groups. The list of all study groups in the school is stratified by grade level. From each grade level, a sample of study groups is randomly selected, and all students in each selected study group are interviewed. The sample consists of eight study groups from the 7th grade, three groups from the 8th grade, and five groups from the 9th grade.

The SAS data set named IceCreamStudy saves the responses of the selected students:

```
data IceCreamStudy;
   input Grade StudyGroup Spending @@;
   if (Spending < 10) then Group='less';
     else Group='more';
   datalines;
7  34  7     7  34  7    7 412  4     9  27 14
7  34  2     9 230 15    9  27 15     7 501  2
9 230  8     9 230  7    7 501  3     8  59 20
7 403  4     7 403 11    8  59 13     8  59 17
8 143 12     8 143 16    8  59 18     9 235  9
8 143 10     9 312  8    9 235  6     9 235 11
9 312 10     7 321  6    8 156 19     8 156 14
7 321  3     7 321 12    7 489  2     7 489  9
7  78  1     7  78 10    7 489  2     7 156  1
7  78  6     7 412  6    7 156  2     9 301  8
;
```

In the data set IceCreamStudy, the variable Grade contain a student's grade. The variable StudyGroup identifies a student's study group. It is possible for students from different grades to have the same study group number because study groups are sequentially numbered within each grade. The variable Spending contains a student's response to how much he spends per week for ice cream, in dollars. The variable GROUP indicates whether a student spends at least $10 weekly for ice cream. It is not necessary to store the data in order of grade and study group.

The SAS data set StudyGroup is created to provide PROC SURVEYMEANS with the sample design information shown in Table 70.4:

```
data StudyGroups;
   input Grade _total_; datalines;
7 608
8 252
9 403
;
```

The variable Grade identifies the strata, and the variable _TOTAL_ contains the total number of study groups in each stratum. As discussed in the section "Specification of Population Totals and Sampling Rates" on page 4334, the population totals stored in the variable _TOTAL_ should be expressed in terms of the primary sampling units (PSUs), which are study groups in this example. Therefore, the variable _TOTAL_ contains the total number of study groups for each grade, rather than the total number of students.

In order to obtain unbiased estimates, you create sampling weights using the following SAS statements:

```
data IceCreamStudy;
   set IceCreamStudy;
   if Grade=7 then Prob=8/608;
   if Grade=8 then Prob=3/252;
   if Grade=9 then Prob=5/403;
   Weight=1/Prob;
```

The sampling weights are the reciprocals of the probabilities of selections. The variable Weight contains the sampling weights. Because the sampling design is clustered, and all students from each selected cluster are interviewed, the sampling weights equal the inverse of the cluster (or study group) selection probabilities.

The following SAS statements perform the analysis for this sample design:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Clustered Sample Design';
proc surveymeans data=IceCreamStudy total=StudyGroups;
   strata Grade / list;
   cluster StudyGroup;
   var Spending Group;
   weight Weight;
run;
```

Output 70.1.1. Data Summary and Class Information

```
   Analysis of Ice Cream Spending
 Stratified Clustered Sample Design

     The SURVEYMEANS Procedure

            Data Summary

Number of Strata                   3
Number of Clusters                16
Number of Observations            40
Sum of Weights                3162.6


      Class Level Information

Class
Variable      Levels    Values

Group              2    less more
```

Output 70.1.1 provides information on the sample design and the input data set. There are 3 strata in the sample design, and the sample contains 16 clusters and 40 observations. The variable Group has two levels, 'less' and 'more'.

Output 70.1.2. Stratum Information

```
                       Analysis of Ice Cream Spending
                     Stratified Clustered Sample Design

                         The SURVEYMEANS Procedure

                            Stratum Information

Stratum            Population  Sampling
 Index      Grade       Total      Rate     N Obs  Variable  Level         N
----------------------------------------------------------------------------
   1            7         608     1.32%        20  Spending               20
                                                   Group     less         17
                                                             more          3
   2            8         252     1.19%         9  Spending                9
                                                   Group     less          0
                                                             more          9
   3            9         403     1.24%        11  Spending               11
                                                   Group     less          6
                                                             more          5
----------------------------------------------------------------------------

                            Stratum Information

Stratum            Population  Sampling
 Index      Grade       Total      Rate     N Obs  Variable  Level  Clusters
----------------------------------------------------------------------------
   1            7         608     1.32%        20  Spending                8
                                                   Group     less          8
                                                             more          3
   2            8         252     1.19%         9  Spending                3
                                                   Group     less          0
                                                             more          3
   3            9         403     1.24%        11  Spending                5
                                                   Group     less          4
                                                             more          4
----------------------------------------------------------------------------
```

Output 70.1.2 displays information for each stratum. Since the primary sampling units in this design are study groups, the population totals shown in Output 70.1.2 are the total numbers of study groups for each stratum or grade. This differs from Figure 70.3 on page 4320, which provides the population totals in terms of students since students were the primary sampling units for that design. Output 70.1.2 also displays the number of clusters for each stratum and analysis variable.

Output 70.1.3. Statistics

```
                          Analysis of Ice Cream Spending
                        Stratified Clustered Sample Design

                            The SURVEYMEANS Procedure

                                   Statistics

                                                         Std Error       Lower 95%
Variable     Level                N            Mean        of Mean     CL for Mean
---------------------------------------------------------------------------------
Spending                         40        8.923860       0.650859        7.517764
Group        less                23        0.561437       0.056368        0.439661
             more                17        0.438563       0.056368        0.316787
---------------------------------------------------------------------------------

                                   Statistics

                                          Upper 95%
                        Variable     Level     CL for Mean
                        ---------------------------------
                        Spending                 10.329957
                        Group        less         0.683213
                                     more         0.560339
                        ---------------------------------
```

Output 70.1.3 displays the estimates of the average weekly ice cream expense and the percentage of students spending at least $10 weekly for ice cream.

Example 70.2. Domain Analysis

Suppose that you are studying profiles of the 800 top-performing companies to provide information on their impact on the economy. You are also interested in the company profiles within each market type. A sample of 66 companies is selected with unequal probability across market types. However, market type is not included in the sample design. Thus, the number of companies within each market type is a random variable in your sample. To obtain statistics within each market type, you should use domain analysis. The data of the 66 companies are saved in the following data set:

```
data Company;
   length Type $14;
   input Type$ Asset Sale Value Profit Employee Weight;
   datalines;
Other             2764.0  1828.0  1850.3   144.0   18.7   9.6
Energy           13246.2  4633.5  4387.7   462.9   24.3  42.6
Finance           3597.7   377.8    93.0    14.0    1.1  12.2
Transportation    6646.1  6414.2  2377.5   348.2   47.1  21.8
HiTech            1068.4  1689.8  1430.2    72.9    4.6   4.3
Manufacturing     1125.0  1719.4  1057.5    98.1   20.4   4.5
Other             1459.0  1241.4   452.7    24.5   20.1   5.5
Finance           2672.3   262.5   296.2    23.1    2.2   9.3
Finance            311.0   566.2   932.0    52.8    2.7   1.9
Energy            1148.6  1014.6   485.1    60.6    4.0   4.5
```

```
Finance          5327.0   572.4   372.9    25.2    4.2  17.7
Energy           1602.7   678.4   653.0    75.6    2.8   6.0
Energy           5808.8  1288.4  2007.0   318.8    5.9  19.2
Medical           268.8   204.4   820.9    45.6    3.7   1.8
Transportation   5222.6  2627.8  1910.0   245.6   22.8  17.4
Other             872.7  1419.4   939.3    69.7   12.2   3.7
Retail           4461.7  8946.8  4662.7   289.0  132.1  15.0
HiTech           6719.2  6942.0  8240.2   381.3   85.8  22.1
Retail            833.4  1538.8  1090.3    64.9   15.4   3.5
Finance           415.9   167.3  1126.8    56.8    0.7   2.2
HiTech            442.4  1139.9  1039.9    57.6   22.7   2.3
Other             801.5  1157.0   664.2    56.9   15.5   3.4
Finance          4954.8   468.8   366.4    41.7    3.0  16.5
Finance          2661.9   257.9   181.1    21.2    2.1   9.3
Finance          5345.8   530.1   337.4    36.4    4.3  17.8
Energy           3334.3  1644.7  1407.8   157.6    6.4  11.4
Manufacturing    1826.6  2671.7   483.2    71.3   25.3   6.7
Retail            618.8  2354.7   767.7    58.6   19.0   2.9
Retail           1529.1  6534.0   826.3    58.3   65.8   5.7
Manufacturing    4458.4  4824.5  3132.1    28.9   67.0  15.0
HiTech           5831.7  6611.1  9464.7   459.6   86.7  19.3
Medical          6468.3  4199.2  3170.4   270.1   59.5  21.3
Energy           1720.7   473.1   811.1    86.6    1.6   6.3
Energy           1679.7  1379.9   721.1    91.8    4.5   6.2
Retail           4018.2 16823.4  2038.3   178.1  162.0  13.6
Other             227.1   575.8  1083.8    62.6    1.9   1.6
Finance          3872.8   362.0   209.3    27.6    2.4  13.1
Retail           3359.3  4844.7  2651.4   224.1   75.6  11.5
Energy           1295.6   356.9   180.8   162.3    0.6   5.0
Energy           1658.0   626.6   688.0   126.0    3.5   6.1
Finance         12156.7  1345.5   680.7   106.6    9.4  39.2
HiTech           3982.6  4196.0  3946.8   313.9   64.3  13.5
Finance          8760.7   886.4  1006.9    90.0    7.5  28.5
Manufacturing    2362.2  3153.3  1080.0   137.0   25.2   8.4
Transportation   2499.9  3419.0   992.6    47.2   25.3   8.8
Energy           1430.4  1610.0   664.3    77.7    3.5   5.4
Energy          13666.5 15465.4  2736.7   411.4   26.6  43.9
Manufacturing    4069.3  4174.7  2907.6   289.2   38.2  13.7
Energy           2924.7   711.9  1067.8   146.7    3.4  10.1
Transportation   1262.1  1716.0   364.3    71.2   14.5   4.9
Medical           684.4   672.9   287.4    61.8    6.0   3.1
Energy           3069.3  1719.0  1439.0   196.4    4.9  10.6
Medical           246.5   318.8   924.1    43.8    3.1   1.7
Finance         11562.2  1128.5   580.4    64.2    6.7  37.3
Finance          9316.0  1059.4   816.5    95.9    8.0  30.2
Retail           1094.3  3848.0   563.3    29.4   44.7   4.4
Retail           1102.1  4878.3   932.4    65.2   47.3   4.4
HiTech            466.4   675.8   845.7    64.5    5.2   2.4
Manufacturing   10839.4  5468.7  1895.4   232.8   47.8  35.0
Manufacturing     733.5  2135.3    96.6    10.9    2.7   3.2
Manufacturing   10354.2 14477.4  5607.2   321.9  188.5  33.5
Energy           1902.1  2697.9   329.3    34.2    2.2   6.9
Other            2245.2  2132.2  2230.4   198.9    8.0   8.0
Transportation    949.4  1248.3   298.9    35.4   10.4   3.9
```

```
Retail          2834.4  2884.6   458.2    41.2   49.8   9.8
Retail          2621.1  6173.8  1992.7   183.7  115.1   9.2
;
```

For each company in your sample,

- the variable Type identifies the type of market for the company.
- the variable Asset contains the company's assets in millions of dollars.
- the variable Sale contains sales in millions of dollars.
- the variable Value contains the market value of the company in millions of dollars.
- the variable Profit contains the profit in millions of dollars.
- the variable Employee stores the number of employees in thousands.
- the variable Weight contains the sampling weight.

The following SAS statements use PROC SURVEYMEANS to perform the domain analysis, estimating means and other statistics for the overall population and also for the subpopulations (or domain) defined by market type. The DOMAIN statement specifies Type as the domain variable:

```
title1 'Top Companies Profile Study';
proc surveymeans data=Company total=800 mean sum;
   var Asset Sale Value Profit Employee;
   weight Weight;
   domain Type;
run;
```

Output 70.2.1. Company Profile Study

```
                      Top Companies Profile Study

                       The SURVEYMEANS Procedure

                              Data Summary

                  Number of Observations            66
                  Sum of Weights                 799.8


                               Statistics

                               Std Error
Variable            Mean         of Mean             Sum          Std Dev
------------------------------------------------------------------------
Asset        6523.488510      720.557075         5217486          1073829
Sale         4215.995799      839.132506         3371953           847885
Value        2145.935121      342.531720         1716319           359609
Profit        188.788210       25.057876          150993            30144
Employee       36.874869        7.787857           29493      7148.003298
------------------------------------------------------------------------
```

Output 70.2.1 shows that there are 66 observations in the sample. The sum of the sampling weights equals 799.8, which is close to the total number of companies in the study population.

The “Statistics” table in Output 70.2.1 displays the estimates of the mean and total for all analysis variables for the entire 800 companies, while Output 70.2.2 shows the mean and total estimates for each company type.

Output 70.2.2. Domain Analysis for Company Profile Study

```
                         Top Companies Profile Study

                          The SURVEYMEANS Procedure

                           Domain Analysis: Type

                                              Std Error
Type            Variable           Mean         of Mean             Sum          Std Dev
-------------------------------------------------------------------------------
Energy          Asset       7868.302932     1941.699163         1449341           785962
                Sale        5419.679099     2416.214417          998305           673373
                Value       2249.297177      520.295162          414321           213580
                Profit       289.564658       52.512141           53338            25927
                Employee      14.151194        3.974697     2606.650000      1481.777769
Finance         Asset       7890.190264     1057.185336         1855773           704506
                Sale         829.210502      115.762531          195030            74436
                Value        565.068197       76.964547          132904            48156
                Profit        63.716837       10.099341           14986      5801.108513
                Employee       5.806293        0.811555     1365.640000       519.658410
HiTech          Asset       5031.959781      732.436967          321542           183302
                Sale        5464.292019      731.296997          349168           196013
                Value       6707.828482     1194.160584          428630           249154
                Profit       346.407042       42.299004           22135            12223
                Employee      70.766980        8.683595     4522.010000      2524.778281
Manufacturing   Asset       7403.004250     1454.921083          888361           492577
                Sale        7207.638833     2112.444703          864917           501679
                Value       2986.442750      799.121544          358373           196979
                Profit       211.933583       39.993255           25432            13322
                Employee      83.314333       31.089019     9997.720000      6294.309490
Medical         Asset       5046.570609     1218.444638          140799           131942
                Sale        3313.219713      758.216303           92439            85655
                Value       2561.614695      530.802245           71469            64663
                Profit       218.682796       44.051447     6101.250000      5509.560969
                Employee      46.518996       11.135955     1297.880000      1213.651734
Other           Asset       1850.250000      338.128984           58838            31375
                Sale        1620.784906      168.686773           51541            24593
                Value       1432.820755      297.869828           45564            24204
                Profit       115.089937       27.970560     3659.860000      2018.201371
                Employee      14.306604        2.313733      454.950000       216.327710
Retail          Asset       2939.845750      393.692369          235188            94605
                Sale        7395.453500     1746.187580          591636           263263
                Value       2103.863125      529.756409          168309            78304
                Profit       157.171875       31.734253           12574      5478.281027
                Employee      93.624000       15.726743     7489.920000      3093.832061
Transportation  Asset       4712.047359      888.954411          267644           163516
                Sale        4030.233275     1015.555708          228917           142669
                Value       1703.330282      313.841326           96749            58947
                Profit       224.762324       56.168925           12767      8287.585418
                Employee      30.946303        6.786270     1757.750000      1066.586615
-------------------------------------------------------------------------------
```

Example 70.3. Ratio Analysis

Suppose you are interested in the profit per employee and the sale per employee among the 800 top-performing companies in the data in the previous example. The following SAS statements illustrate how you can use PROC SURVEYMEANS to estimate these ratios:

```
title1 'Ratio Analysis in Top Companies Profile Study';
proc surveymeans data=Company total=800 ratio;
   var Profit Sale Employee;
   weight Weight;
   ratio Profit Sale / Employee;
run;
```

The RATIO statement requests the ratio of the profit and the sale to the number of employees.

Output 70.3.1. Estimate Ratios

```
                 Ratio Analysis in Top Companies Profile Study

                          The SURVEYMEANS Procedure

                               Ratio Analysis

              Numerator Denominator        Ratio         Std Err
              --------------------------------------------------
              Sale      Employee      114.332497       20.502742
              Profit    Employee        5.119698        1.058939
              --------------------------------------------------
```

Output 70.3.1 shows the estimated ratios and their standard errors. Because the profit and the sale figures are in millions of dollars, and the employee numbers in thousands, the profit per employee is estimated as $5,120 with a standard error of $1,059, and the sale per employee is $114,333 with a standard error of $20,503.

Example 70.4. Analyzing Survey Data with Missing Values

As described in the section "Missing Values" on page 4333, the SURVEYMEANS procedure excludes an observation from the analysis if it has a missing value for the analysis variable or a nonpositive value for the WEIGHT variable.

However, if there is evidence indicating that the nonrespondents are different from the respondents for your study, you can use the DOMAIN statement to compute descriptive statistics among respondents from your survey data without imputation for nonrespondents. Note that although the variance estimation for respondents takes into account the assumption that the study population consists of distinct groups of respondents and nonrespondents, the degrees of freedom will not adjust for the nonrespondents because they are deleted from the computation. As a result, there are fewer degrees of freedom and wider confidence limits in comparison to counting those nonrespondents for degrees of freedom. When the sample size and the number of respondents are large, the difference maybe ignored.

Consider the ice cream example in the section "Stratified Sampling" on page 4318. Suppose that some of the students failed to provide the amounts spent on ice cream, as shown in the following data set IceCream:

```
data IceCream;
   input Grade Spending @@; datalines;
7 7   7  7   8  .   9 10   7  .   7 10   7  3   8 20   8 19   7 2
7 .   9 15   8 16   7  6   7  6   7  6   9 15   8 17   8 14   9 .
9 8   9  7   7  3   7 12   7  4   9 14   8 18   9  9   7  2   7 1
7 4   7 11   9  8   8  .   8 13   7  .   9  .   9 11   7  2   7 9
;
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

Considering the possibility that those students who didn't respond spend differently than those students who did respond, you can create an indicator variable to identify the respondents and non-respondents with the following SAS DATA step statements:

```
data IceCream;
   set IceCream;
   if Spending=. then Indicator='Nonrespondent';
   else do;
      Indicator='Respondent';
      if (Spending < 10) then Group='less';
         else Group='more';
      end;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

The variable Indicator identifies a student in the data set as either a respondent or a nonrespondent. The variable Group specifies whether a student spent more than $10 among the respondents.

The following SAS statements produce the desired analysis:

```
title1 'Analysis of Ice Cream Spending';
proc surveymeans data=IceCream total=StudentTotals mean sum;
   strata Grade / list;
   var Spending Group;
   weight Weight;
   domain Indicator;
run;
```

Output 70.4.1. Analysis of Incomplete Ice Cream Data Excluding Observations with Missing Values

```
                          Analysis of Ice Cream Spending

                            The SURVEYMEANS Procedure

                                   Data Summary

                       Number of Strata                   3
                       Number of Observations            40
                       Sum of Weights                  4000


                                   Statistics

                                        Std Error
Variable    Level                Mean     of Mean             Sum         Std Dev
---------------------------------------------------------------------------------
Spending                     9.770542    0.541381           32139     1780.792065
Group       less             0.515404    0.067092     1695.345455      220.690305
            more             0.484596    0.067092     1594.004040      220.690305
---------------------------------------------------------------------------------
```

Output 70.4.2 shows the mean and total estimates excluding those students who failed to provide the spending amount on ice cream.

Output 70.4.2. Analysis of Incomplete Ice Cream Data Treating Respondents as a Domain

```
                          Analysis of Ice Cream Spending

                            The SURVEYMEANS Procedure

                            Domain Analysis: Indicator

                                                         Std Error
Indicator        Variable    Level            Mean         of Mean             Sum
----------------------------------------------------------------------------------
Nonrespondent    Spending                        .               .               .
                 Group       less                .               .               .
                             more                .               .               .
Respondent       Spending                 9.770542        0.652347           32139
                 Group       less         0.515404        0.067092     1695.345455
                             more         0.484596        0.067092     1594.004040
----------------------------------------------------------------------------------

                            Domain Analysis: Indicator

                Indicator        Variable    Level         Std Dev
                --------------------------------------------------
                Nonrespondent    Spending                        .
                                 Group       less                .
                                             more                .
                Respondent       Spending              3515.126876
                                 Group       less       220.690305
                                             more       220.690305
                --------------------------------------------------
```

Output 70.4.1 shows the mean and total estimates treating respondents as a domain in the student population. Compared to the estimates in Output 70.4.1, the point estimates are the same, but the variance estimations are slightly higher.

References

Brick, J.M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215 –238.

Cochran, W.G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Foreman, E.K. (1991), *Survey Sampling Principles*, New York: Marcel Dekker, Inc.

Fuller, W.A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37, Series C, Pt. 3, 117 –132.

Fuller, W.A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H.J. (1989), *PC CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Hansen, M.H., Hurwitz, W.N., and Madow, W.G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons, Inc.

Hidiroglou, M.A., Fuller, W.A., and Hickman, R.D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications, Inc.

Kalton, G., and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodology*, 12, 1 –16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Lee, E.S., Forthoffer, R.N., and Lorimor, R.J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications, Inc.

Levy, P. and Lemeshow, S. (1999), *Sampling of Populations, Methods and Applications*, Third Edition, New York: John Wiley & Sons, Inc.

Pringle, R.M. and Raynor, A.A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Woodruff, R.S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 71
The SURVEYREG Procedure

Chapter Contents

Chapter 71

The SURVEYREG Procedure

Overview

The SURVEYREG procedure performs regression analysis for sample survey data. This procedure can handle complex survey sample designs, including designs with stratification, clustering, and unequal weighting. The procedure fits linear models for survey data and computes regression coefficients and their variance-covariance matrix. The procedure also provides significance tests for the model effects and for any specified estimable linear functions of the model parameters. Using the regression model, the procedure can compute predicted values for the sample survey data.

PROC SURVEYREG computes the regression coefficient estimators by generalized least-squares estimation using element-wise regression. The procedure assumes that the regression coefficients are the same across strata and primary sampling units (PSUs). To estimate the variance-covariance matrix for the regression coefficients, PROC SURVEYREG uses the Taylor expansion theory for estimating sampling errors of estimators based on complex sample designs (Woodruff 1971; Fuller 1975; Särndal, Swenson, and Wretman 1992, Chapter 5 and Chapter 13). This method obtains a linear approximation for the estimator and then uses the variance estimator for this approximation to estimate the variance of the estimator itself.

PROC SURVEYREG uses the ODS (Output Delivery System) to place results in output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality.

Getting Started

This section demonstrates how you can use PROC SURVEYREG to perform a regression analysis for sample survey data. For a complete description of the usage of PROC SURVEYREG, see the section "Syntax" on page 4373. The "Examples" section on page 4395 provides more detailed examples that illustrate the applications of PROC SURVEYREG.

Simple Random Sampling

Suppose that, in a junior high school, there are a total of 4,000 students in grades 7, 8, and 9. You want to know how household income and the number of children in a household affect students' average weekly spending for ice cream.

In order to answer this question, you draw a sample using simple random sampling from the student population in the junior high school. You randomly select 40 students and ask them their average weekly expenditure for ice cream, their household income, and the number of children in their household. The answers from the 40 students are saved as a SAS data set:

```
data IceCream;
   input Grade Spending Income Kids @@;
   datalines;
 7   7  39  2   7   7  38  1   8  12  47  1
 9  10  47  4   7   1  34  4   7  10  43  2
 7   3  44  4   8  20  60  3   8  19  57  4
 7   2  35  2   7   2  36  1   9  15  51  1
 8  16  53  1   7   6  37  4   7   6  41  2
 7   6  39  2   9  15  50  4   8  17  57  3
 8  14  46  2   9   8  41  2   9   8  41  1
 9   7  47  3   7   3  39  3   7  12  50  2
 7   4  43  4   9  14  46  3   8  18  58  4
 9   9  44  3   7   2  37  1   7   1  37  2
 7   4  44  2   7  11  42  2   9   8  41  2
 8  10  42  2   8  13  46  1   7   2  40  3
 9   6  45  1   9  11  45  4   7   2  36  1
 7   9  46  1
 ;
```

In the data set IceCream, the variable Grade indicates a student's grade. The variable Spending contains the dollar amount of each student's average weekly spending for ice cream. The variable Income specifies the household income, in thousands of dollars. The variable Kids indicates how many children are in a student's family.

The following PROC SURVEYREG statements request a regression analysis:

```
title1 'Ice Cream Spending Analysis';
title2 'Simple Random Sample Design';
proc surveyreg data=IceCream total=4000;
   class Kids;
   model Spending = Income Kids / solution anova;
run;
```

The PROC SURVEYREG statement invokes the procedure. The TOTAL=4000 option specifies the total in the population from which the sample is drawn. The CLASS statement requests that the procedure use the variable Kids as a classification variable in the analysis. The MODEL statement describes the linear model that you want to fit, with Spending as the dependent variable and Income and Kids as the independent variables. The SOLUTION option in the MODEL statement requests that the procedure output the regression coefficient estimates. The ANOVA option requests that the procedure output the ANOVA table.

```
                     Ice Cream Spending Analysis
                     Simple Random Sample Design

                       The SURVEYREG Procedure

         Regression Analysis for Dependent Variable Spending

                            Data Summary

                Number of Observations             40
                Mean of Spending              8.75000
                Sum of Spending             350.00000


                           Fit Statistics

                     R-square            0.8132
                     Root MSE            2.4506
                     Denominator DF          39


                       Class Level Information

                  Class
                  Variable      Levels    Values

                  Kids               4    1 2 3 4
```

Figure 71.1. Summary of Data

Figure 71.1 displays the summary of the data, the summary of the fit, and the levels of the classification variable Kids. The “Fit Statistics” table displays the denominator degrees of freedom, which are used in *F* tests and *t* tests in the regression analysis.

```
                     Ice Cream Spending Analysis
                     Simple Random Sample Design

                       The SURVEYREG Procedure

         Regression Analysis for Dependent Variable Spending

                       Tests of Model Effects

              Effect       Num DF    F Value    Pr > F

              Model             4     119.15    <.0001
              Intercept         1     153.32    <.0001
              Income            1     324.45    <.0001
              Kids              3       0.92    0.4385

     NOTE: The denominator degrees of freedom for the F tests is 39.
```

Figure 71.2. Testing Effects in the Regression

Figure 71.2 displays the ANOVA table for the regression and the tests for model effects. The effect Income is significant in the linear regression model, while the effect Kids is not significant at the 5% level.

```
                          Ice Cream Spending Analysis
                          Simple Random Sample Design

                            The SURVEYREG Procedure

                Regression Analysis for Dependent Variable Spending

                         Estimated Regression Coefficients

                                           Standard
          Parameter      Estimate             Error    t Value    Pr > |t|

          Intercept    -26.084677        2.46720403     -10.57      <.0001
          Income         0.775330        0.04304415      18.01      <.0001
          Kids 1         0.897655        1.12352876       0.80      0.4292
          Kids 2         1.494032        1.24705263       1.20      0.2381
          Kids 3        -0.513181        1.33454891      -0.38      0.7027
          Kids 4         0.000000        0.00000000        .           .

NOTE: The denominator degrees of freedom for the t tests is 39.
      Matrix X'X is singular and a generalized inverse was used to solve the
      normal equations.  Estimates are not unique.
```

Figure 71.3. Regression Coefficients

The regression coefficient estimates and their standard errors and associated *t* tests are displayed in Figure 71.3.

Stratified Sampling

Suppose that the previous student sample is actually drawn using a stratified sample design. The strata are grades in the junior high school: 7, 8, and 9. Within strata, simple random samples are selected. Table 71.1 provides the number of students in each grade.

Table 71.1. Students in Grades

Grade	Number of Students
7	1,824
8	1,025
9	1,151
Total	4,000

In order to analyze this sample using PROC SURVEYREG, you need to input the stratification information by creating a SAS data set with the information in Table 71.1. The following SAS statements create such a data set called StudentTotals:

```
data StudentTotals;
   input Grade _TOTAL_;
   datalines;
7 1824
```

```
8 1025
9 1151
;
```

The variable Grade is the stratification variable, and the variable _TOTAL_ contains the total numbers of students in each stratum in the survey population. PROC SURVEYREG requires you to use the keyword _TOTAL_ as the name of the variable that contains the population total information.

In a stratified sample design, when the sampling rates in the strata are unequal, you need to use sampling weights to reflect this information. For this example, the appropriate sampling weights are the reciprocals of the probabilities of selection. You can use the following data step to create the sampling weights:

```
data IceCream;
   set IceCream;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

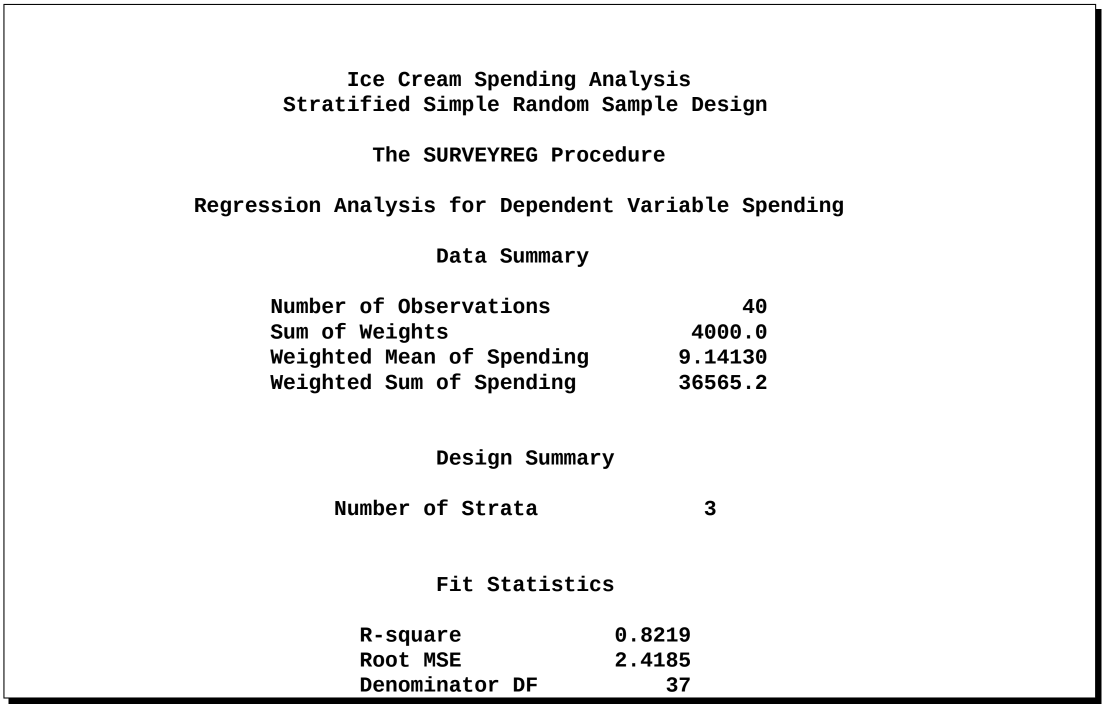
If you use PROC SURVEYSELECT to select your sample, PROC SURVEYSELECT creates these sampling weights for you.

The following statements demonstrate how you can fit a linear model while incorporating the sample design information (stratification):

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution anova;
   weight Weight;
run;
```

Comparing these statements to those in the section "Simple Random Sampling" on page 4365, you can see how the TOTAL=StudentTotals option replaces the previous TOTAL=4000 option.

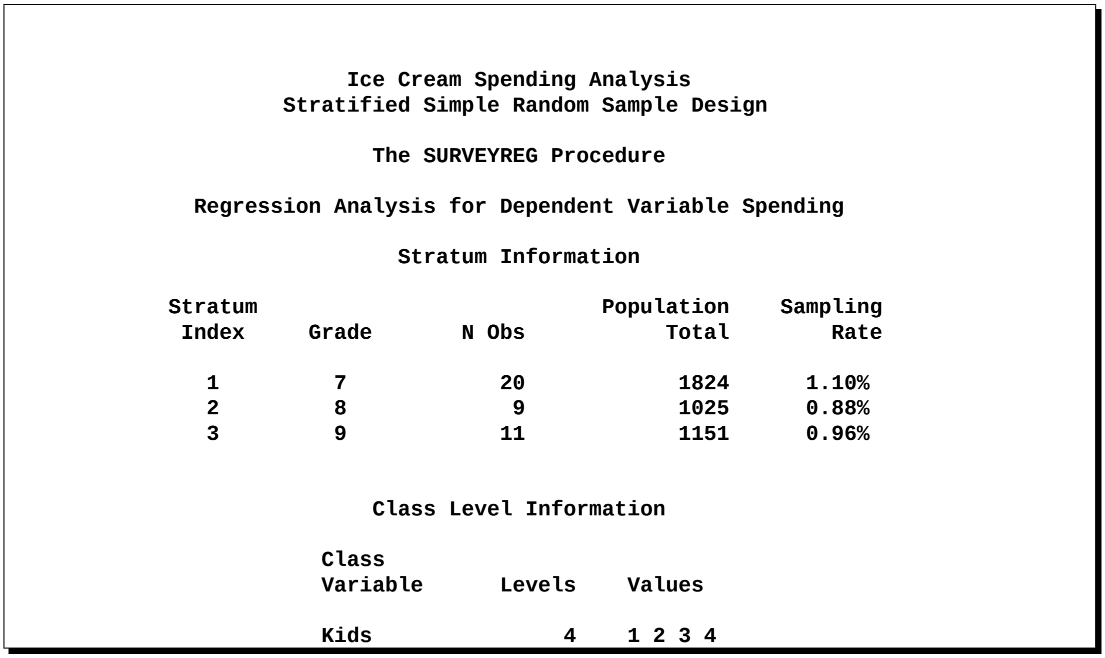
The STRATA statement specifies the stratification variable Grade. The LIST option in the STRATA statement requests that the stratification information be included in the output. The WEIGHT statement specifies the weight variable.



```
                  Ice Cream Spending Analysis
             Stratified Simple Random Sample Design

                    The SURVEYREG Procedure

        Regression Analysis for Dependent Variable Spending

                         Data Summary

              Number of Observations              40
              Sum of Weights                  4000.0
              Weighted Mean of Spending      9.14130
              Weighted Sum of Spending       36565.2


                         Design Summary

                   Number of Strata             3


                         Fit Statistics

                     R-square            0.8219
                     Root MSE            2.4185
                     Denominator DF          37
```

Figure 71.4. Summary of the Regression

Figure 71.4 summarizes the data information, the sample design information, and the fit information. Note that, due to the stratification, the denominator degrees of freedom for *F* tests and *t* tests is 37, which is different from the analysis in Figure 71.1.



```
                  Ice Cream Spending Analysis
             Stratified Simple Random Sample Design

                    The SURVEYREG Procedure

        Regression Analysis for Dependent Variable Spending

                      Stratum Information

     Stratum                           Population    Sampling
      Index     Grade       N Obs           Total        Rate

        1         7           20             1824       1.10%
        2         8            9             1025       0.88%
        3         9           11             1151       0.96%


                    Class Level Information

                  Class
                  Variable      Levels    Values

                  Kids               4    1 2 3 4
```

Figure 71.5. Stratification and Classification Information

For each stratum, Figure 71.5 displays the value of identifying variables, the number of observations (sample size), the total population size, and the calculated sampling rate or fraction.

```
                          Ice Cream Spending Analysis
                     Stratified Simple Random Sample Design

                            The SURVEYREG Procedure

              Regression Analysis for Dependent Variable Spending

                             Tests of Model Effects

                    Effect       Num DF    F Value    Pr > F

                    Model             4     124.85    <.0001
                    Intercept         1     150.95    <.0001
                    Income            1     326.89    <.0001
                    Kids              3       0.99    0.4081

        NOTE: The denominator degrees of freedom for the F tests is 37.
```

Figure 71.6. Testing Effects

Figure 71.6 displays the ANOVA table for the regression and tests for the significance of model effects under the stratified sample design. The Income effect is strongly significant, while the Kids effect is not significant at the 5% level.

```
                          Ice Cream Spending Analysis
                     Stratified Simple Random Sample Design

                            The SURVEYREG Procedure

              Regression Analysis for Dependent Variable Spending

                       Estimated Regression Coefficients

                                       Standard
          Parameter      Estimate         Error    t Value    Pr > |t|

          Intercept    -26.086882    2.44108058     -10.69      <.0001
          Income         0.776699    0.04295904      18.08      <.0001
          Kids 1         0.888631    1.07000634       0.83      0.4116
          Kids 2         1.545726    1.20815863       1.28      0.2087
          Kids 3        -0.526817    1.32748011      -0.40      0.6938
          Kids 4         0.000000    0.00000000        .          .

NOTE: The denominator degrees of freedom for the t tests is 37.
      Matrix X'WX is singular and a generalized inverse was used to solve the
      normal equations.  Estimates are not unique.
```

Figure 71.7. Regression Coefficients

The regression coefficient estimates for the stratified sample, along with their standard errors and associated *t* tests, are displayed in Figure 71.7.

You can request other statistics and tests using PROC SURVEYREG. You can also analyze data from a more complex sample design. The remainder of this chapter provides more detailed information.

Output Data Set

PROC SURVEYREG uses the Output Delivery System (ODS) to create output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

For example, to save the “ParameterEstimates” table (Figure 71.7) in the previous section in an output data set, you use the ODS OUTPUT statement as follows:

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution;
   weight Weight;
   ods output ParameterEstimates = MyParmEst;
   run;
```

The statement

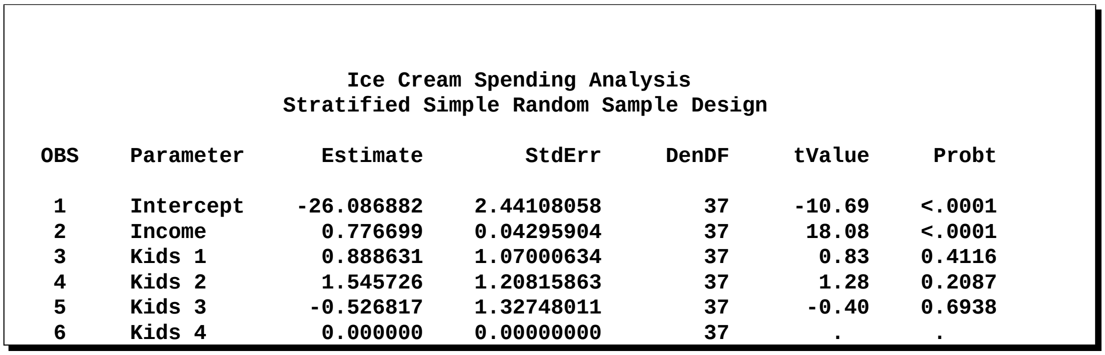
```
ods output ParameterEstimates = MyParmEst;
```

requests that the “ParameterEstimates” table that appears in Figure 71.7 be placed in a SAS data set named MyParmEst.

The PRINT procedure displays observations of the data set MyParmEst:

```
proc print data=MyParmEst;
run;
```

Figure 71.8 displays the observations in the data set MyParmEst.



```
                      Ice Cream Spending Analysis
                  Stratified Simple Random Sample Design

OBS     Parameter       Estimate          StdErr     DenDF      tValue      Probt

 1      Intercept     -26.086882      2.44108058        37      -10.69     <.0001
 2      Income          0.776699      0.04295904        37       18.08     <.0001
 3      Kids 1          0.888631      1.07000634        37        0.83     0.4116
 4      Kids 2          1.545726      1.20815863        37        1.28     0.2087
 5      Kids 3         -0.526817      1.32748011        37       -0.40     0.6938
 6      Kids 4          0.000000      0.00000000        37         .          .
```

Figure 71.8. The Data Set MyParmEst

The section "ODS Table Names" on page 4394 gives the complete list of the tables produced by PROC SURVEYREG.

Syntax

The following statements are available in PROC SURVEYREG:

PROC SURVEYREG < *options* > **;**
  BY *variables* **;**
  CLASS *variables* **;**
  CLUSTER *variables* **;**
  CONTRAST *'label' effect values*
        < . . . *effect values* > < / *options* > **;**
  ESTIMATE *'label' effect values*
        < . . . *effect values* > < / *options* > **;**
  MODEL *dependent* = < *effects* > < / *options* > **;**
  STRATA *variables* < / *options* > **;**
  WEIGHT *variable* **;**

The PROC SURVEYREG and MODEL statements are required. If your model contains classification effects, you must list the classification variables in a CLASS statement, and the CLASS statement must precede the MODEL statement. If you use a CONTRAST statement or an ESTIMATE statement, the MODEL statement must precede the CONTRAST or ESTIMATE statement.

The CLASS, CLUSTER, STRATA, CONTRAST, and ESTIMATE statements can appear multiple times. You should only use one MODEL statement and one WEIGHT statement.

PROC SURVEYREG Statement

PROC SURVEYREG < *options* >**;**

The PROC SURVEYREG statement invokes the procedure. You can specify the following options in the PROC SURVEYREG statement:

ALPHA=α

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)$% confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed by PROC SURVEYREG. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

RATE=*value* | *SAS-data-set*
R=*value* | *SAS-data-set*

specifies the sampling rate as a non-negative *value*, or specifies an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction for variance estimation. If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a non-negative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section Specification of Population Totals and Sampling Rates on page 4382 for more details.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be non-negative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYREG will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

TOTAL=*value* | *SAS-data-set*
N=*value* | *SAS-data-set*

specifies the total number of primary sampling units in the study population as a positive *value*, or specifies an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for variance estimation.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section Specification of Population Totals and Sampling Rates on page 4382 for more details.

If you do not specify the TOTAL= option or the RATE= option, then the variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option.

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of the CLASS, STRATA, and CLUSTER variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYREG to obtain separate analyses on observations in groups defined by the BY variables.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. For more information on subpopulation analysis for sample survey data, refer to Cochran (1977).

When a BY statement appears, the procedure expects the input data sets to be sorted in order of the BY variables. If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the SURVEYREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

CLASS Statement

CLASS | **CLASSES** *variables* **;**

The CLASS statement specifies the classification variables to be used in the model. Typical class variables are TREATMENT, GENDER, RACE, GROUP, and REPLICATION. If you specify the CLASS statement, it must appear before the MODEL statement.

Classification variables can be either character or numeric. Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Concepts*. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to Version 9, class levels were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the PROC SURVEYREG statement.

You can use multiple CLASS statements to specify classification variables.

CLUSTER Statement

CLUSTER | **CLUSTERS** *variables* **;**

The CLUSTER statement specifies variables that identify clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. By default, clusters are determined from the entire formatted values of the CLUSTER variables. Note that this represents a slight change from previous releases in the way in which clusters are determined. In releases prior to Version 9, clusters were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the PROC SURVEYREG statement.

You can use multiple CLUSTER statements to specify cluster variables. The procedure uses variables from all CLUSTER statements to create clusters.

CONTRAST Statement

CONTRAST *'label' effect values* < */ options* > **;**

CONTRAST *'label' effect values* < . . . *effect values* > < */ options* > **;**

The CONTRAST statement provides custom hypothesis tests for linear combinations of the regression parameters $H_0\colon \mathbf{L}\boldsymbol{\beta} = 0$, where **L** is the vector or matrix you specify and $\boldsymbol{\beta}$ is the vector of regression parameters. Thus, to use this feature, you must be familiar with the details of the model parameterization used by PROC SURVEYREG. For information on the parameterization, see the section "Parameterization of PROC GLM Models" on page 1787 in Chapter 32, "The GLM Procedure."

Each term in the MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or a special notation using variable names and operators. For more details on how to specify an effect, see the section "Specification of Effects" on page 1784 in Chapter 32, "The GLM Procedure."

For each CONTRAST statement, PROC SURVEYREG computes Wald's *F* test. The procedure displays this value with the degrees of freedom, and identifies it with the contrast label. The numerator degrees of freedom for Wald's *F* test equals rank(**L**). The denominator degrees of freedom equals the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of strata. Alternatively, you can use the DF= option in the MODEL statement to specify the denominator degrees of freedom.

You can specify any number of CONTRAST statements, but they must appear after the MODEL statement.

In the CONTRAST statement,

label identifies the contrast in the output. A label is required for every contrast specified. Labels must be enclosed in single quotes.

effect identifies an effect that appears in the MODEL statement. You can use the INTERCEPT keyword as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of **L** associated with the effect.

You can specify the following options in the CONTRAST statement after a slash (/):

E

displays the entire coefficient **L** vector or matrix.

NOFILL

requests no filling in higher-order effects. When you specify only certain portions of **L**, by default PROC SURVEYREG constructs the remaining elements from the context. (For more information, see the section "Specification of ESTIMATE Expressions" on page 1801 in Chapter 32, "The GLM Procedure.")

When you specify the NOFILL option, PROC SURVEYREG does not construct the remaining portions and treats the vector or matrix **L** as it is defined in the CONTRAST statement.

SINGULAR=*value*

specifies the sensitivity for checking estimability. If **v** is a vector, define ABS(**v**) to be the largest absolute value of the elements of **v**. Say **H** is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS(**L**) except for elements of **L** that equal 0, and then C is 1. If ABS$(\mathbf{L} - \mathbf{LH}) > C \cdot$*value*, then **L** is declared nonestimable. The SINGULAR=*value* must be between 0 and 1, and the default is 10^{-4}.

As stated previously, the CONTRAST statement enables you to perform hypothesis tests $H_0\colon \mathbf{L}\boldsymbol{\beta} = 0$.

If the **L** matrix contains more than one contrast, then you can separate the rows of the **L** matrix with commas. For example, for the model

```
proc surveyreg;
   class A B;
   model Y=A B;
run;
```

with A at 5 levels and B at 2 levels, the parameter vector is

$$(\mu \ \ \alpha_1 \ \ \alpha_2 \ \ \alpha_3 \ \ \alpha_4 \ \ \alpha_5 \ \ \beta_1 \ \ \beta_2)$$

To test the hypothesis that the pooled A linear and A quadratic effect is zero, you can use the following **L** matrix:

$$\mathbf{L} = \left[\begin{array}{rrrrrrrr} 0 & -2 & -1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & -1 & -2 & -1 & 2 & 0 & 0 \end{array} \right]$$

The corresponding CONTRAST statement is

```
contrast ’A Linear & Quadratic’
         a -2 -1  0  1  2,
         a  2 -1 -2 -1  2;
```

ESTIMATE Statement

ESTIMATE *’label’ effect values* < */ options* > **;**

ESTIMATE *’label’ effect values* < . . . *effect values* > < */ options* > **;**

You can use an ESTIMATE statement to estimate a linear function of the regression parameters by multiplying a row vector **L** by the parameter estimate vector $\widehat{\boldsymbol{\beta}}$.

Each term in the MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or with a special notation using variable names and operators. For more details on how to specify an effect, see the section “Specification of Effects” on page 1784 in Chapter 32, “The GLM Procedure.”

PROC SURVEYREG checks the linear function for estimability. (See the SINGULAR= option described on page 4379).

The procedure displays the estimate $\mathbf{L}\widehat{\beta}$ along with its standard error and *t* test. If you specify the CLPARM option in the MODEL statement, PROC SURVEYREG also displays confidence limits for the linear function. By default, the degrees of freedom for the *t* test equals the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of strata. Alternatively, you can specify the degrees of freedom with the DF= option in the MODEL statement.

You can specify any number of ESTIMATE statements, but they must appear after the MODEL statement.

In the ESTIMATE statement,

label identifies the linear function **L** in the output. A label is required for every function specified. Labels must be enclosed in single quotes.

effect identifies an effect that appears in the MODEL statement. You can use the INTERCEPT keyword as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values values are constants that are elements of the vector $\mathbf{L}$ associated with the effect. For example, the following code forms an estimate that is the difference between the parameters estimated for the first and second levels of the CLASS variable A.

```
estimate 'A1 vs A2' A  1 -1;
```

You can specify the following options in the ESTIMATE statement after a slash (/):

DIVISOR=*value*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integers. For example, you can use

```
estimate '1/3(A1+A2) - 2/3A3' a 1 1 -2 / divisor=3;
```

instead of

```
estimate '1/3(A1+A2) - 2/3A3' a 0.33333 0.33333 -0.66667;
```

E

displays the entire coefficient vector $\mathbf{L}$.

NOFILL

requests no filling in higher-order effects. When you specify only certain portions of the vector $\mathbf{L}$, by default PROC SURVEYREG constructs the remaining elements from the context. (See the section "Specification of ESTIMATE Expressions" on page 1801 in Chapter 32, "The GLM Procedure.") When you specify the NOFILL option, PROC SURVEYREG does not construct the remaining portions and treats the vector $\mathbf{L}$ as it is defined in the ESTIMATE statement.

SINGULAR=*value*

specifies the sensitivity for checking estimability. If $\mathbf{v}$ is a vector, define ABS($\mathbf{v}$) to be the largest absolute value of the elements of $\mathbf{v}$. Say $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS($\mathbf{L}$) except for elements of $\mathbf{L}$ that equal 0, and then C is 1. If ABS($\mathbf{L} - \mathbf{LH}$) $> C\times$*value*, then $\mathbf{L}$ is declared nonestimable. The SINGULAR=*value* must be between 0 and 1, and the default is 10^{-4}.

MODEL Statement

MODEL *dependent* = < *effects* >< / *options* >**;**

The MODEL statement specifies the dependent (response) variable and the independent (regressor) variables or effects. Each term in a MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or with special notation using variable names and operators. For more information on how to specify an effect, see the section "Specification of Effects" on page 1784 in Chapter 32, "The GLM Procedure." The dependent variable must be numeric. Only one MODEL statement is allowed for each PROC SURVEYREG statement. If you specify more than one MODEL statement, the procedure uses the first model and ignores the rest.

You can specify the following options in the MODEL statement after a slash (/):

ADJRSQ

requests the procedure to compute the adjusted multiple R-square.

ANOVA

requests the ANOVA table to be produced in the output. By default, the ANOVA table will not be printed in the output.

CLPARM

requests confidence limits for the parameter estimates. The SURVEYREG procedure determines the confidence coefficient using the ALPHA= option, which by default equals 0.05 and produces 95% confidence bounds. The CLPARM option also requests confidence limits for all the estimable linear functions of regression parameters in the ESTIMATE statements.

Note that when there is a CLASS statement, you need to use the SOLUTION option with the CLPARM option to obtain the parameter estimates and their confidence limits.

COVB

displays the estimated covariance matrix of the estimated regression estimates.

DEFF

displays design effects for the regression coefficient estimates.

DF=*value*

specifies the denominator degrees of freedom for the *F* tests and the degrees of freedom for the *t* tests. The default is the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of actual strata. The number of actual strata equals the number of strata in the data before collapsing minus the number of strata collapsed plus 1. See the section "Stratum Collapse" on page 4388 for details on "collapsing of strata."

I
INVERSE

displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{X}$ matrix. When there is a WEIGHT variable, the procedure displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{W}\mathbf{X}$ matrix, where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values.

NOINT

omits the intercept from the model.

SOLUTION

displays a solution to the normal equations, which are the parameter estimates. The SOLUTION option is useful only when you use a CLASS statement. If you do not specify a CLASS statement, PROC SURVEYREG displays parameter estimates by default. But if you specify a CLASS statement, PROC SURVEYREG does not display parameter estimates unless you also specify the SOLUTION option.

VADJUST=DF | NONE
VARADJ=DF | NONE
VARADJUST=DF | NONE

specifies if the you want to use degrees of freedom adjustment $(n-1)/(n-p)$ in the computation of the matrix $\mathbf{G}$ for the variance estimation on page 4385. If you do not specify the VADJUST= option, by default, PROC SURVEYREG uses the degrees of freedom adjustment, that is equivalent to the VARADJ=DF option. If you do not wish to use this variance adjustment, you can specify the VADJUST=NONE option.

X
XPX

displays the $\mathbf{X'X}$ matrix, or the $\mathbf{X'WX}$ matrix when there is a WEIGHT variable, where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values. The X option also displays the crossproducts vector $\mathbf{X'y}$, or $\mathbf{X'Wy}$.

STRATA Statement

STRATA | **STRATUM** *variables* < */ options* > **;**

The STRATA statement specifies variables that form the strata in a stratified sample design. The combinations of categories of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section “Specification of Population Totals and Sampling Rates” on page 4382 for more information.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. By default, strata are determined from the entire formatted values of the STRATA variables. Note that this represents a slight change from previous releases in the way in which strata are determined. In releases prior to Version 9, strata were determined using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option in the PROC SURVEYREG statement.

Thus, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *SAS Procedures Guide*.

You can use multiple STRATA statements to specify stratum variables.

You can specify the following options in the STRATA statement after a slash (/):

LIST

displays a “Stratum Information” table, which includes values of the STRATA variables, and the number of observations, number of clusters, population total, and sampling rate for each stratum. This table also displays stratum collapse information.

NOCOLLAPSE

prevents the procedure from collapsing, or combining, strata that have only one sampling unit. By default, the procedure collapses strata that contain only one sampling unit. See the section “Stratum Collapse” on page 4388 for details.

WEIGHT Statement

WEIGHT | **WGT** *variable* **;**

The WEIGHT statement specifies the variable that contains the sampling weights. This variable must be numeric. If you do not specify a WEIGHT statement, PROC SURVEYREG assigns all observations a weight of 1. Sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

Details

Missing Values

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then PROC SURVEYREG excludes that observation from the analysis. An observation is also excluded if it has a missing value for any STRATA variable, CLUSTER variable, dependent variable, or any variable used in the independent effects. The analysis includes all observations in the data set that have nonmissing values for all these design and analysis variables.

If you have missing values in your survey data for any reason (such as nonresponse), this can compromise the quality of your survey results. If the respondents are different from the nonrespondents with regard to a survey effect or outcome, then survey estimates will be biased and will not accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. Once data collection is complete, you can use imputation to replace missing values with acceptable values, and you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYREG. Refer to Cochran (1977) for more details.

Survey Design Information

Specification of Population Totals and Sampling Rates

If your analysis should include a finite population correction (*fpc*), you can input either the sampling rate or the population total using the RATE= option or the TOTAL= option. You cannot specify both of these options in the same PROC SURVEYREG statement. If you do not specify one of these options, the procedure does not use the *fpc* when computing variance estimates. For fairly small sampling fractions, it is appropriate to ignore this correction. Refer to Cochran (1977) and Kish (1965).

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you should use the RATE=*value* option or the TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in the strata, then you can use the RATE=*SAS-data-set* option or the TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Or if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates.

The secondary data set must contain all BY and STRATA groups that occur in the primary data set. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option, or the values of _RATE_ in the secondary data set, must be non-negative numbers. You can specify a sampling rate as a number between 0 and 1. Or you can specify a sampling rate in percentage form as a number between 1 and 100, and PROC SURVEYREG will convert that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or primary sampling units (PSUs), in your sample design, the procedure estimates variance from the variation among PSUs. For more information, see the section "Variance Estimation" on page 4385. You can use the CLUSTER statement to identify the first stage clusters in your design. PROC SURVEYREG assumes that each cluster represents a PSU in the sample and that each observation is an element of a PSU. If you do not specify a CLUSTER statement, the procedure treats each observation as a PSU.

Computational Details

Notation

For a stratified clustered sample design, observations are represented by an $n \times (p+2)$ matrix

$$(\mathbf{w}, \mathbf{y}, \mathbf{X}) = (w_{hij}, y_{hij}, \mathbf{x}_{hij})$$

where

- w denotes the sampling weight vector
- y denotes the dependent variable
- $\mathbf{X}$ denotes the design matrix. (When an effect contains only classification variables, the columns of $\mathbf{X}$ corresponding to this effect contain only 0s and 1s; no reparameterization is made.)
- $h = 1, 2, \ldots, H$ is the stratum number with a total of H strata
- $i = 1, 2, \ldots, n_h$ is the cluster number within stratum h, with a total of n_h clusters
- $j = 1, 2, \ldots, m_{hi}$ is the unit number within cluster i of stratum h, with a total of m_{hi} units
- p is the total number of parameters (including an intercept if the INTERCEPT effect is included in the MODEL statement)
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total number of observations in the sample

Also, f_h denotes the sampling rate for stratum h. You can use the TOTAL= option or the RATE= option to input population totals or sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 4382 for details. If you input stratum totals, PROC SURVEYREG computes f_h as the ratio of the stratum sample size to the stratum total. If you input stratum sampling rates, PROC SURVEYREG uses these values directly for f_h. If you do not specify the TOTAL= option or the RATE= option, then the procedure assumes that the stratum sampling rates f_h are negligible, and a finite population correction is not used when computing variances.

Regression Coefficients

PROC SURVEYREG solves the normal equations $\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{y}$ using a modified sweep routine that produces a generalized (g2) inverse $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}$ and a solution (Pringle and Raynor 1971)

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{X}'\mathbf{W}\mathbf{y}$$

where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values.

For models with class variables, there are more design matrix columns than there are degrees of freedom (DF) for the effect. Thus, there are linear dependencies among the columns. In this case, the parameters are not estimable; there is an infinite number of least-squares solutions. PROC SURVEYREG uses a generalized (g2) inverse to obtain values for the estimates. The solution values are not displayed unless you specify the SOLUTION option in the MODEL statement. The solution has the characteristic that estimates are 0 whenever the design column for that parameter is a linear combination of previous columns. (Strictly termed, the solution values should not be called estimates.) With this full parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Variance Estimation

PROC SURVEYREG uses the Taylor series expansion theory to estimate the covariance-variance matrix of the estimated regression coefficients (Fuller 1975). Let

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$$

where the (h, i, j)th element is r_{hij}. Compute $1 \times p$ row vectors

$$\begin{aligned} \mathbf{e}_{hij} &= w_{hij} r_{hij} \mathbf{x}_{hij} \\ \mathbf{e}_{hi\cdot} &= \sum_{j=1}^{m_{hi}} \mathbf{e}_{hij} \\ \bar{\mathbf{e}}_{h\cdot\cdot} &= \frac{1}{n_h} \sum_{i=1}^{n_h} \mathbf{e}_{hi\cdot} \end{aligned}$$

and calculate the $p \times p$ matrix

$$\mathbf{G} = \frac{n-1}{n-p} \sum_{h=1}^{H} \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})'(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})$$

PROC SURVEYREG computes the covariance matrix of $\boldsymbol{\beta}$ as

$$\widehat{\mathbf{V}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{G}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}$$

The factor $(n-1)/(n-p)$ in the computation of the matrix $\widehat{\mathbf{G}}$ should reduce the small sample bias associated with using the estimated function to calculate deviations (Hidiroglou *et al.* (1980)). For simple random sampling, this factor contributes to the degrees of freedom correction applied to the residual mean square for ordinary least squares in which p parameter are estimated. By default, the procedure will use this adjustment in the variance estimation. It is equivalent to specify the VADJUST=DF option in the MODEL statement. If you do not wish to use this multiplier in the variance estimation, you can specify the VADJUST=NONE option in the MODEL statement to suppress this factor.

Degrees of Freedom

PROC SURVEYREG produces tests for the significance of model effects, regression parameters, estimable functions specified in the ESTIMATE statement, and contrasts specified in the CONTRAST statement. It computes all these tests taking into account the sample design. The degrees of freedom for these tests differ from the degrees of freedom for the ANOVA table, which does not consider the sample design.

Denominator Degrees of Freedom

The denominator DF refers to the denominator degrees of freedom for *F* tests and to the degrees of freedom for *t* tests in the analysis. By default, the denominator DF equals the number of clusters minus the actual number of strata. If there are no clusters, the denominator DF equals the number of observations minus the actual number of strata. The *actual number of strata* equals

- one, if there is no STRATA statement
- the number of strata in the sample, if there is a STRATA statement but the procedure does not collapse any strata
- the number of strata in the sample after collapsing, if there is a STRATA statement and the procedure collapses strata that have only one sampling unit

Alternatively, you can specify the denominator DF using the DF= option on page 4380 in the MODEL statement.

Numerator Degrees of Freedom

The numerator DF refers to the numerator degrees of freedom for the Wald *F* statistic associated with an effect or with a contrast. The procedure computes the Wald *F* statistic for an effect as a Type III test; that is, the test has the following properties:

- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- The hypotheses to be tested are invariant to the ordering of effects in the model.

See the section "Testing Effects" on page 4386 for more information. The numerator DF for the Wald *F* statistic for a contrast is the rank of the **L** matrix that defines the contrast.

Testing Effects

For each effect in the model, PROC SURVEYREG computes an **L** matrix such that every element of $\mathbf{L}\boldsymbol{\beta}$ is estimable; the **L** matrix has the maximum possible rank associated with the effect. To test the effect, the procedure uses the Wald *F* statistic for the hypothesis $H_0\colon \mathbf{L}\boldsymbol{\beta} = 0$. The Wald *F* statistic equals

$$F_{\text{Wald}} = \frac{(\mathbf{L}\widehat{\boldsymbol{\beta}})'(\mathbf{L}'\widehat{\mathbf{V}}\mathbf{L})^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}})}{\text{rank}(\mathbf{L})}$$

with numerator degrees of freedom equal to $\text{rank}(\mathbf{L})$ and denominator degrees of freedom equal to the number of clusters minus the number of strata (unless you have specified the denominator degrees of freedom with the DF= option in the MODEL statement; see the section "Denominator Degrees of Freedom" on page 4386). It is possible that the **L** matrix cannot be constructed for an effect, in which case that effect is not testable. For more information on how the matrix **L** is constructed, see the discussion in Chapter 11, "The Four Types of Estimable Functions."

Analysis of Variance (ANOVA)

PROC SURVEYREG produces an analysis of variance table for the model specified in the MODEL statement. This table is identical to the one produced by the GLM procedure for the model. PROC SURVEYREG computes ANOVA table entries using the sampling weights, but not the sample design information on stratification and clustering.

The degrees of freedom (DF) displayed in the ANOVA table are the same as those in the ANOVA table produced by PROC GLM. The Total DF is the total degrees of freedom used to obtain the regression coefficient estimates. The Total DF equals the total number of observations minus 1 if the model includes an intercept. If the model does not include an intercept, the Total DF equals the total number of observations. The Model DF equals the degrees of freedom for the effects in the MODEL statement, not including the intercept. The Error DF equals the total DF minus the model DF.

Multiple R-square

PROC SURVEYREG computes a multiple R-square for the weighted regression as

$$R^2 = 1 - \frac{\mathrm{SS}_{\mathrm{error}}}{\mathrm{SS}_{\mathrm{total}}}$$

where $\mathrm{SS}_{\mathrm{error}}$ is the error sum of squares in the ANOVA table

$$\mathrm{SS}_{\mathrm{error}} = \mathbf{r}'\mathbf{W}\mathbf{r}$$

and $\mathrm{SS}_{\mathrm{total}}$ is the total sum of squares

$$\mathrm{SS}_{\mathrm{total}} = \begin{cases} \mathbf{y}'\mathbf{W}\mathbf{y} & \text{if no intercept} \\ \mathbf{y}'\mathbf{W}\mathbf{y} - \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij} y_{hij} \right)^2 / \, w_{\cdots} & \text{otherwise} \end{cases}$$

where $w_{\cdots}$ is the sum of the sampling weights over all observations.

Adjusted R-square

If you specify the option ADJRSQ in the MODEL statement, PROC SURVEYREG computes an multiple R-square adjusted as the weighted regression as

$$\mathrm{ADJRSQ} = \begin{cases} 1 - \dfrac{n(1-R^2)}{n-p} & \text{if no intercept} \\[2ex] 1 - \dfrac{(n-1)(1-R^2)}{n-p} & \text{otherwise} \end{cases}$$

where R^2 is the multiple R-square.

Root Mean Square Errors

PROC SURVEYREG computes the square root of mean square errors as

$$\sqrt{\mathrm{MSE}} = \sqrt{n\ \mathrm{SS}_{\mathrm{error}}\ /\ (n-p)\ w_{\cdots}}$$

where $w_{\cdots}$ is the sum of the sampling weights over all observations.

Design Effect

If you specify the DEFF option in the MODEL statement, PROC SURVEYREG calculates the design effects for the regression coefficients. The design effect of an estimate is the ratio of the actual variance to the variance computed under the assumption of simple random sampling.

$$\mathrm{DEFF} = \frac{\text{Variance under the Sample Design}}{\text{Variance under Simple Random Sampling}}$$

Refer to Kish (1965, p. 258). PROC SURVEYREG computes the numerator as described in the section "Variance Estimation" on page 4385. And the denominator is computed under the assumption that the sample design is simple random sampling, with no stratification and no clustering.

To compute the variance under the assumption of simple random sampling, PROC SURVEYREG calculates the sampling rate as follows. If you specify both sampling weights and sampling rates (or population totals) for the analysis, then the sampling rate under simple random sampling is calculated as

$$f_{\text{SRS}} = n\ /\ w_{\cdots}$$

where n is the sample size and $w_{\cdots}$ (the sum of the weights over all observations) estimates the population size. If the sum of the weights is less than the sample size, f_{SRS} is set to zero. If you specify sampling rates for the analysis but not sampling weights, then PROC SURVEYREG computes the sampling rate under simple random sampling as the average of the stratum sampling rates.

$$f_{\text{SRS}} = \frac{1}{H}\sum_{h=1}^{H} f_h$$

If you do not specify sampling rates (or population totals) for the analysis, then the sampling rate under simple random sampling is assumed to be zero.

$$f_{\text{SRS}} = 0$$

Stratum Collapse

If there is only one sampling unit in a stratum, then PROC SURVEYREG cannot estimate the variance for this stratum. To estimate stratum variances, by default the procedure collapses, or combines, those strata that contain only one sampling unit. If you specify the NOCOLLAPSE option in the STRATA statement, PROC SURVEYREG does not collapse strata and uses a variance estimate of 0 for any stratum that contains only one sampling unit.

If you do not specify the NOCOLLAPSE option, PROC SURVEYREG collapses strata according to the following rules. If there are multiple strata that each contain only one sampling unit, then the procedure collapses, or combines, all these strata into a new pooled stratum. If there is only one stratum with a single sampling unit, then PROC SURVEYREG collapses that stratum with the preceding stratum, where strata are ordered by the STRATA variable values. If the stratum with one sampling unit is the first stratum, then the procedure combines it with the following stratum.

If you specify stratum sampling rates using the RATE=*SAS-data-set* option, PROC SURVEYREG computes the sampling rate for the new pooled stratum as the weighted average of the sampling rates for the collapsed strata. See the section "Computational Details" on page 4384 for details. If the specified sampling rate equals 0 for any of the collapsed strata, then the pooled stratum is assigned a sampling rate of 0. If you specify stratum totals using the TOTAL=*SAS-data-set* option, PROC SURVEYREG combines the totals for the collapsed strata to compute the sampling rate for the new pooled stratum.

Sampling Rate of the Pooled Stratum from Collapse

Assuming that PROC SURVEYREG collapses single-unit strata $h_1, h_2, \ldots, h_c$ into the pooled stratum, the procedure calculates the sampling rate for the pooled stratum as

$$f_{\text{Pooled Stratum}} = \begin{cases} 0 & \text{if any of } f_{h_l} = 0 \text{ where } l = 1, 2, \ldots, c \\ \left(\sum_{l=1}^{c} n_{h_l} f_{h_l}^{-1} \right)^{-1} \sum_{l=1}^{c} n_{h_l} & \text{otherwise} \end{cases}$$

Contrasts

You can use the CONTRAST statement to perform custom hypothesis tests. If the hypothesis is testable in the univariate case, the Wald *F* statistic for $H_0 : \mathbf{L}\boldsymbol{\beta} = 0$ is computed as

$$F_{\text{Wald}} = \frac{(\mathbf{L}_{\text{Full}}\widehat{\boldsymbol{\beta}})'(\mathbf{L}_{\text{Full}}{}'\widehat{\mathbf{V}}\mathbf{L}_{\text{Full}})^{-1}(\mathbf{L}_{\text{Full}}\widehat{\boldsymbol{\beta}})}{\text{rank}(\mathbf{L})}$$

where $\mathbf{L}$ is the contrast vector or matrix you specify, $\boldsymbol{\beta}$ is the vector of regression parameters, $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{X}'\mathbf{W}\mathbf{Y}$, $\widehat{\mathbf{V}}$ is the estimated covariance matrix of $\widehat{\boldsymbol{\beta}}$, rank($\mathbf{L}$) is the rank of $\mathbf{L}$, and $\mathbf{L}_{\text{Full}}$ is a matrix such that

- $\mathbf{L}_{\text{Full}}$ has the same number of columns as $\mathbf{L}$
- $\mathbf{L}_{\text{Full}}$ has full row rank
- the rank of $\mathbf{L}_{\text{Full}}$ equals the rank of the $\mathbf{L}$ matrix
- all rows of $\mathbf{L}_{\text{Full}}$ are estimable functions
- the Wald *F* statistic computed using the $\mathbf{L}_{\text{Full}}$ matrix is equivalent to the Wald *F* statistic computed using the $\mathbf{L}$ matrix with any row deleted that is a linear combination of previous rows

If $\mathbf{L}$ is a full-rank matrix, and all rows of $\mathbf{L}$ are estimable functions, then $\mathbf{L}_{\mathrm{Full}}$ is the same as $\mathbf{L}$. It is possible that $\mathbf{L}_{\mathrm{Full}}$ matrix cannot be constructed for contrasts in a CONTRAST statement, in which case the contrasts are not testable.

Output

Displayed Output

The SURVEYREG procedure produces the following output.

Data Summary

By default, PROC SURVEYREG displays the following information in the "Data Summary" table:

- Number of Observations, which is the total number of observations used in the analysis, excluding observations with missing values
- Sum of Weights, if you specify a WEIGHT statement
- Mean of the dependent variable in the MODEL statement, or Weighted Mean if you specify a WEIGHT statement
- Sum of the dependent variable in the MODEL statement, or Weighted Sum if you specify a WEIGHT statement

Design Summary

When you specify a CLUSTER statement or a STRATA statement, the procedure displays a "Design Summary" table, which provides the following sample design information:

- Number of Strata, if you specify a STRATA statement
- Number of Strata Collapsed, if the procedure collapses strata
- Number of Clusters, if you specify a CLUSTER statement
- Overall Sampling Rate used to calculate the design effect, if you specify the DEFF option in the MODEL statement

Fit Statistics

By default, PROC SURVEYREG displays the following regression statistics in the "Fit Statistics" table:

- R-square for the regression
- Root MSE, which is the square root of the mean square error
- Denominator DF, which is the denominator degrees of freedom for the *F* tests and also the degrees of freedom for the *t* tests produced by the procedure

Stratum Information

When you specify the LIST option in the STRATA statement, PROC SURVEYREG displays a “Stratum Information” table, which provides the following information for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Population Total, if you specify the TOTAL= option
- Sampling Rate, if you specify the TOTAL= option or the RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of nonmissing observations in the stratum.
- N Obs, which is the number of observations
- number of Clusters, if you specify a CLUSTER statement
- Collapsed, which has the value ‘Yes’ if the stratum is collapsed with another stratum before analysis

If PROC SURVEYREG collapses strata, the “Stratum Information” table also displays stratum information for the new, collapsed stratum. The new stratum has a Stratum Index of 0 and is labeled ‘Pooled’.

Class Level Information

If you use a CLASS statement to name classification variables, PROC SURVEYREG displays a “Class Level Information” table. This table contains the following information for each classification variable:

- Class Variable, which lists each CLASS variable name
- Levels, which is the number of values or levels of the classification variable
- Values, which lists the values of the classification variable. The values are separated by a white space character; therefore, to avoid confusion, you should not include a white space character within a classification variable value.

$\mathbf{X'X}$ Matrix

If you specify the XPX option in the MODEL statement, PROC SURVEYREG displays the $\mathbf{X'X}$ matrix, or the $\mathbf{X'WX}$ matrix when there is a WEIGHT variable. This option also displays the crossproducts vector $\mathbf{X'y}$ or $\mathbf{X'Wy}$, where $\mathbf{y}$ is the response vector (dependent variable).

Inverse Matrix of $\mathbf{X'X}$

If you specify the INV option in the MODEL statement, PROC SURVEYREG displays the inverse or the generalized inverse of the $\mathbf{X'X}$ matrix. When there is a WEIGHT variable, the procedure displays the inverse or the generalized inverse of the $\mathbf{X'WX}$ matrix.

ANOVA for Dependent Variable

If you specify the ANOVA option in the model statement, PROC SURVEYREG displays an analysis of variance table for the dependent variable. This table is identical to the ANOVA table displayed by the GLM procedure.

Tests of Model Effects

By default, PROC SURVEYREG displays a "Tests of Model Effects" table, which provides Wald's *F* test for each effect in the model. The table contains the following information for each effect:

- Effect, which is the effect name
- Num DF, which is the numerator degrees of freedom for Wald's *F* test
- F Value, which is Wald's *F* statistic
- Pr > F, which is the significance probability corresponding to the F Value

A footnote displays the denominator degrees of freedom, which is the same for all effects.

Estimated Regression Coefficients

PROC SURVEYREG displays the "Estimated Regression Coefficients" table by default when there is no CLASS statement. Also, the procedure displays this table when you specify a CLASS statement and also specify the SOLUTIONS option in the MODEL statement. This table contains the following information for each regression parameter:

- Parameter, which identifies the effect or regressor variable
- Estimate, which is the estimate of the regression coefficient
- Standard Error, which is the standard error of the estimate
- t Value, which is the *t* statistic for testing $H_0\colon \text{Parameter} = 0$
- Pr > | t |, which is the two-sided significance probability corresponding to the t Value

Covariance of Estimated Regression Coefficients

When you specify the COVB option in the MODEL statement, PROC SURVEYREG displays the "Covariance of Estimated Regression Coefficients" matrix.

Coefficients of Contrast

When you specify the E option in a CONTRAST statement, PROC SURVEYREG displays a "Coefficients of Contrast" table for the contrast. You can use this table to check the coefficients you specified in the CONTRAST statement. Also, this table gives a note for a nonestimable contrast.

Analysis of Contrasts

If you specify a CONTRAST statement, PROC SURVEYREG produces an "Analysis of Contrasts" table, which displays Wald's *F* test for the contrast. If you use more than one CONTRAST statement, the procedure displays all results in the same table. The "Analysis of Contrasts" table contains the following information for each contrast:

- Contrast, which is the label of the contrast
- Num DF, which is the numerator degrees of freedom for Wald's *F* test
- F Value, which is Wald's *F* statistic for testing $H_0\colon \text{Contrast} = 0$
- Pr > F, which is the significance probability corresponding to the F Value

Coefficients of Estimate

When you specify the E option in an ESTIMATE statement, PROC SURVEYREG displays a "Coefficients of Estimate" table for the linear function of the regression parameters in the ESTIMATE statement. You can use this table to check the coefficients you specified in the ESTIMATE statement. Also, this table gives a note for a nonestimable function.

Analysis of Estimable Functions

If you specify an ESTIMATE statement, PROC SURVEYREG checks the function for estimability. If the function is estimable, PROC SURVEYREG produces an "Analysis of Estimable Functions" table, which displays the estimate and the corresponding *t* test. If you use more than one ESTIMATE statement, the procedure displays all results in the same table. The table contains the following information for each estimable function:

- Parameter, which is the label of the function
- Estimate, which is the estimate of the estimable liner function
- Standard Error, which is the standard error of the estimate
- t Value, which is the *t* statistic for testing $H_0\colon \text{Estimable Function} = 0$
- Pr > | t |, which is the two-sided significance probability corresponding to the t Value

Output Data Sets

Output data sets from PROC SURVEYREG are produced using ODS (Output Delivery System). ODS encompasses more than just the production of output data sets. For example, you can use ODS to manipulate the format of your output, the headers and titles of the tables, the order of the columns in a table. For a more detailed description on using ODS, see Chapter 14, "Using the Output Delivery System."

ODS Table Names

PROC SURVEYREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 71.2. ODS Tables Produced in PROC SURVEYREG

ODS Table Name	Description	Statement	Option
ANOVA	ANOVA for dependent variable	MODEL	ANOVA
ClassVarInfo	Class level information	CLASS	default
ContrastCoef	Coefficients of contrast	CONTRAST	E
Contrasts	Analysis of contrasts	CONTRAST	default
CovB	Covariance of estimated regression coefficients	MODEL	COVB
DataSummary	Data summary	MODEL	default
DesignSummary	Design summary	STRATA | CLUSTER	default
Effects	Tests of model effects	MODEL	
EstimateCoef	Coefficients of estimate	ESTIMATE	E
Estimates	Analysis of estimable functions	ESTIMATE	default
FitStatistics	Fit Statistics	MODEL	default
InvXPX	Inverse matrix of $\mathbf{X}'\mathbf{X}$	MODEL	INV
ParameterEstimates	Estimated regression coefficients	MODEL	default
StrataInfo	Stratum information	STRATA	LIST
XPX	$\mathbf{X}'\mathbf{X}$ matrix	MODEL	XPX

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

For example, the following statements create an output data set named MyStrata, which contains the “StrataInfo” table, an output data set named MyParmEst, which contains the “ParameterEstimates” table, and an output data set named Cov, which contains the “CovB” table for the ice cream study discussed in the section “Stratified Sampling” on page 4368:

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution covb;
   weight Weight;
   ods output StrataInfo = MyStrata
              ParameterEstimates = MyParmEst
              CovB = Cov;
run;
```

Note that the option CovB is specified in the MODEL statement in order to produce the covariance matrix table.

Examples

Example 71.1. Simple Random Sampling

This example investigates the relationship between the labor force participation rate (LFPR) of women in 1968 and 1972 in large cities in the United States. A simple random sample of 19 cities is drawn from a total of 200 cities. For each selected city, the LFPRs are recorded and saved in a SAS data set named Labor. The LFPR in 1972 is contained in the variable LFPR1972, and the LFPR in 1968 is identified by the variable LFPR1968:

```
data Labor;
   input City $ 1-16 LFPR1972 LFPR1968;
   datalines;
New York        .45     .42
Los Angeles     .50     .50
Chicago         .52     .52
Philadelphia    .45     .45
Detroit         .46     .43
San Francisco   .55     .55
Boston          .60     .45
Pittsburgh      .49     .34
St. Louis       .35     .45
Connecticut     .55     .54
Washington D.C. .52     .42
Cincinnati      .53     .51
Baltimore       .57     .49
Newark          .53     .54
Minn/St. Paul   .59     .50
Buffalo         .64     .58
Houston         .50     .49
Patterson       .57     .56
Dallas          .64     .63
;
```

Assume that the LFPRs in 1968 and 1972 have a linear relationship, as shown in the following model:

$$\text{LFPR1972} = \beta_0 + \beta_1 * \text{LFPR1968} + \text{error}$$

You can use PROC SURVEYREG to obtain the estimated regression coefficients and estimated standard errors of the regression coefficients. The following statements perform the regression analysis:

```
title 'Study of Labor Force Participation Rates of Women';
proc surveyreg data=Labor total=200;
   model LFPR1972 = LFPR1968;
run;
```

Here, the TOTAL=200 option specifies the finite population total from which the simple random sample of 19 cities is drawn. You can specify the same information by using the sampling rate option RATE=0.095 (19/200=.095).

Output 71.1.1. Summary of Regression Using Simple Random Sampling

```
                Study of Labor Force Participation Rates of Women

                             The SURVEYREG Procedure

               Regression Analysis for Dependent Variable LFPR1972

                                  Data Summary

                       Number of Observations            19
                       Mean of LFPR1972             0.52684
                       Sum of LFPR1972             10.01000


                                 Fit Statistics

                            R-square            0.3970
                            Root MSE           0.05657
                            Denominator DF          18
```

Output 71.1.1 summarizes the data information, the fit information.

Output 71.1.2. Regression Coefficient Estimates

```
                Study of Labor Force Participation Rates of Women

                             The SURVEYREG Procedure

               Regression Analysis for Dependent Variable LFPR1972

                             Tests of Model Effects

                     Effect       Num DF    F Value    Pr > F

                     Model             1      13.84    0.0016
                     Intercept         1       4.63    0.0452
                     LFPR1968          1      13.84    0.0016

         NOTE: The denominator degrees of freedom for the F tests is 18.


                        Estimated Regression Coefficients

                                               Standard
          Parameter      Estimate         Error    t Value    Pr > |t|

          Intercept    0.20331056    0.09444296       2.15      0.0452
          LFPR1968     0.65604048    0.17635810       3.72      0.0016

         NOTE: The denominator degrees of freedom for the t tests is 18.
```

Output 71.1.2 presents the significance tests for the model effects and estimated regression coefficients. The *F* tests and *t* tests for the effects in the model are also presented in these tables.

From the regression performed by PROC SURVEYREG, you obtain a positive estimated slope for the linear relationship between the LFPR in 1968 and the LFPR in 1972. The regression coefficients are all significant at the 5% level. Effects Intercept and LFPR1968 are significant in the model at the 5% level. In this example, the *F* test for the overall model without intercept is the same as the effect LFPR1968.

Example 71.2. Simple Random Cluster Sampling

This example illustrates the use of regression analysis in a simple random cluster sample design. The data are from Särndal, Swenson, and Wretman (1992, p. 652).

A total of 284 Swedish municipalities are grouped into 50 clusters of neighboring municipalities. Five clusters with a total of 32 municipalities are randomly selected. The results from the regression analysis in which clusters are used in the sample design are compared to the results of a regression analysis that ignores the clusters. The linear relationship between the population in 1975 and in 1985 is investigated.

The 32 selected municipalities in the sample are saved in the data set Municipalities:

```
data Municipalities;
   input Municipality Cluster Population85 Population75;
   datalines;
   205   37    5    5
   206   37   11   11
   207   37   13   13
   208   37    8    8
   209   37   17   19
     6    2   16   15
     7    2   70   62
     8    2   66   54
     9    2   12   12
    10    2   60   50
    94   17    7    7
    95   17   16   16
    96   17   13   11
    97   17   12   11
    98   17   70   67
    99   17   20   20
   100   17   31   28
   101   17   49   48
   276   50    6    7
   277   50    9   10
   278   50   24   26
   279   50   10    9
   280   50   67   64
   281   50   39   35
   282   50   29   27
   283   50   10    9
   284   50   27   31
```

```
      52   10    7    6
      53   10    9    8
      54   10   28   27
      55   10   12   11
      56   10  107  108
   ;
```

The variable Municipality identifies the municipalities in the sample; the variable Cluster indicates the cluster to which a municipality belongs; and the variables Population85 and Population75 contain the municipality populations in 1985 and in 1975 (in thousands), respectively. A regression analysis is performed by PROC SURVEYREG with a CLUSTER statement:

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Cluster Simple Random Sampling';
proc surveyreg data=Municipalities total=50;
   cluster Cluster;
   model Population85=Population75;
run;
```

The TOTAL=50 option specifies the total number of clusters in the sampling frame.

Output 71.2.1. Regression Analysis for Simple Random Cluster Sampling

```
                 Regression Analysis for Swedish Municipalities
                         Cluster Simple Random Sampling

                            The SURVEYREG Procedure

            Regression Analysis for Dependent Variable Population85

                                  Data Summary

                     Number of Observations            32
                     Mean of Population85        27.50000
                     Sum of Population85        880.00000


                                 Design Summary

                       Number of Clusters             5


                                 Fit Statistics

                          R-square            0.9860
                          Root MSE            3.0488
                          Denominator DF           4


                        Estimated Regression Coefficients

                                        Standard
   Parameter         Estimate          Error    t Value    Pr > |t|

   Intercept       -0.0191292     0.89204053      -0.02      0.9839
   Population75     1.0546253     0.05167565      20.41      <.0001

    NOTE: The denominator degrees of freedom for the t tests is 4.
```

Output 71.2.1 displays the data summary, design summary, fit statistics, and regression coefficient estimates. Since the sample design includes clusters, the procedure displays the total number of clusters in the sample in the “Design Summary” table. In the “Estimated Regression Coefficients” table, the estimated slope for the linear relationship is 1.05, which is significant at the 5% level; but the intercept is not significant. This suggests that a regression line crossing the original can be established between populations in 1975 and in 1985.

The CLUSTER statement is necessary in PROC SURVEYREG in order to incorporate the sample design. If you do not specify a CLUSTER statement in the regression analysis, the standard deviation of the regression coefficients will be incorrectly estimated:

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Simple Random Sampling';
proc surveyreg data=Municipalities total=284;
   model Population85=Population75;
run;
```

The analysis ignores the clusters in the sample, assuming that the sample design is a simple random sampling. Therefore, the TOTAL= option specifies the total number of municipalities, which is 284.

Output 71.2.2. Regression Analysis for Simple Random Sampling

```
         Regression Analysis for Swedish Municipalities
                     Simple Random Sampling

                    The SURVEYREG Procedure

    Regression Analysis for Dependent Variable Population85

                          Data Summary

              Number of Observations            32
              Mean of Population85        27.50000
              Sum of Population85        880.00000


                         Fit Statistics

                   R-square            0.9860
                   Root MSE            3.0488
                   Denominator DF          31


               Estimated Regression Coefficients

                                Standard
Parameter         Estimate         Error    t Value    Pr > |t|

Intercept       -0.0191292    0.67417606      -0.03      0.9775
Population75     1.0546253    0.03668414      28.75      <.0001

NOTE: The denominator degrees of freedom for the t tests is 31.
```

Output 71.2.2 displays the regression results ignoring the clusters. Compared to the results in Output 71.2.1 on page 4398, the regression coefficient estimates are the same. However, without using clusters, the regression coefficients have a smaller variance estimate in Output 71.2.2. Using clusters in the analysis, the estimated regression coeffiecient for effect Population75 is 1.05, with the estimated standard error 0.05, as displayed in Output 71.2.1; without using the clusters, the estimate is 1.05, but with the estimated standard error 0.04, as displayed in Output 71.2.2. To estimated the variance of the regression coefficients correctly, you should include the clustering information in the regression analysis.

Example 71.3. Regression Estimator for Simple Random Sample

Using auxiliary information, you can construct the regression estimators to provide more accurate estimates of the population characteristics that are of interest. With ESTIMATE statements in PROC SURVEYREG, you can specify a regression estimator as a linear function of the regression parameters to estimate the population total. This example illustrates this application, using the data in the previous example.

In this sample, a linear model between the Swedish populations in 1975 and in 1985 is established:

$$\text{Population85} = \alpha + \beta * \text{Population75} + \text{error}$$

Assuming that the total population in 1975 is known to be 8200 (in thousands), you can use the ESTIMATE statement to predict the 1985 total population using the following statements:

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Estimate Total Population';
proc surveyreg data=Municipalities total=50;
   cluster Cluster;
   model Population85=Population75;
   estimate '1985 population' Intercept 284 Population75 8200;
run;
```

Since each observation in the sample is a municipality, and there is a total of 284 municipalities in Sweden, the coefficient for Intercept (α) in the ESTIMATE statement is 284, and the coefficient for Population75 (β) is the total population in 1975 (8.2 million).

Output 71.3.1. Use the Regression Estimator to Estimate the Population Total

```
                 Regression Analysis for Swedish Municipalities
                           Estimate Total Population

                            The SURVEYREG Procedure

            Regression Analysis for Dependent Variable Population85

                        Analysis of Estimable Functions

                                          Standard
      Parameter             Estimate         Error     t Value    Pr > |t|

      1985 population    8642.49485    258.558613      33.43      <.0001

        NOTE: The denominator degrees of freedom for the t tests is 4.
```

Output 71.3.1 displays the regression results and the estimation of the total population. Using the linear model, you can predict the total population in 1985 to be 8.64 million, with a standard error of 0.26 million.

Example 71.4. Stratified Sampling

This example illustrates using the SURVEYREG procedure to perform a regression in a stratified sample design. Consider a population of 235 farms producing corn in Nebraska and Iowa. You are interested in the relationship between corn yield (CornYield) and the total farm size (FarmArea).

Each state is divided into several regions, and each region is used as a stratum. Within each stratum, a simple random sample with replacement is drawn. A total of 19 farms is selected using a stratified simple random sample. The sample size and population size within each stratum are displayed in Table 71.3.

Table 71.3. Number of Farms in Each Stratum

			Number of Farms	
Stratum	**State**	**Region**	**Population**	**Sample**
1	Iowa	1	100	3
2		2	50	5
3		3	15	3
4	Nebraska	1	30	6
5		2	40	2
	Total		235	19

Three models for the data are considered:

- Model I — Common intercept and slope:

$$\text{Corn Yield} = \alpha + \beta * \text{Farm Area}$$

- Model II — Common intercept, different slope:

$$\text{Corn Yield} = \begin{cases} \alpha + \beta_{\text{Iowa}} * \text{Farm Area} & \text{if the farm is in Iowa} \\ \alpha + \beta_{\text{Nebraska}} * \text{Farm Area} & \text{if the farm is in Nebraska} \end{cases}$$

- Model III — Different intercept and different slope:

$$\text{Corn Yield} = \begin{cases} \alpha_{\text{Iowa}} + \beta_{\text{Iowa}} * \text{Farm Area} & \text{if the farm is in Iowa} \\ \alpha_{\text{Nebraska}} + \beta_{\text{Nebraska}} * \text{Farm Area} & \text{if the farm is in Nebraska} \end{cases}$$

Data from the stratified sample are saved in the SAS data set Farms. In the data set Farms, the variable Weight represents the sampling weight. In this example, the sampling weights are reciprocal of selection probabilities:

```
data Farms;
   input State $ Region FarmArea CornYield Weight;
   datalines;
Iowa     1 100  54 33.333
Iowa     1  83  25 33.333
Iowa     1  25  10 33.333
Iowa     2 120  83 10.000
Iowa     2  50  35 10.000
Iowa     2 110  65 10.000
Iowa     2  60  35 10.000
Iowa     2  45  20 10.000
Iowa     3  23   5  5.000
Iowa     3  10   8  5.000
Iowa     3 350 125  5.000
Nebraska 1 130  20  5.000
Nebraska 1 245  25  5.000
Nebraska 1 150  33  5.000
Nebraska 1 263  50  5.000
Nebraska 1 320  47  5.000
Nebraska 1 204  25  5.000
Nebraska 2  80  11 20.000
Nebraska 2  48   8 20.000
;
```

The information on population size in each stratum is saved in the SAS data set StratumTotals:

```
data StratumTotals;
   input State $ Region _TOTAL_;
   datalines;
Iowa     1 100
Iowa     2  50
Iowa     3  15
Nebraska 1  30
Nebraska 2  40
;
```

Using the sample data from the data set Farms and the control information data from the data set StratumTotals, you can fit Model I using PROC SURVEYREG with the following statements:

```
title1 'Analysis of Farm Area and Corn Yield';
title2 'Model I: Same Intercept and Slope';
proc surveyreg data=Farms total=StratumTotals;
   strata State Region / list;
   model CornYield = FarmArea;
   weight Weight;
run;
```

Output 71.4.1. Data Summary and Stratum Information Fitting Model I

```
                       Analysis of Farm Area and Corn Yield
                        Model I: Same Intercept and Slope

                             The SURVEYREG Procedure

               Regression Analysis for Dependent Variable CornYield

                                  Data Summary

                    Number of Observations                 19
                    Sum of Weights                  234.99900
                    Weighted Mean of CornYield       31.56029
                    Weighted Sum of CornYield          7416.6


                                 Design Summary

                         Number of Strata             5


                                 Fit Statistics

                           R-square            0.3882
                           Root MSE           20.6422
                           Denominator DF          14


                               Stratum Information

    Stratum                                               Population    Sampling
     Index        State        Region         N Obs            Total        Rate

         1       Iowa            1               3              100       3.00%
         2                       2               5               50       10.0%
         3                       3               3               15       20.0%
         4       Nebraska        1               6               30       20.0%
         5                       2               2               40       5.00%
```

Output 71.4.1 displays the data summary and stratification information fitting Model I. The sampling rates are automatically computed by the procedure based on the sample sizes and the population totals in strata.

Output 71.4.2. Estimated Regression Coefficients and the Estimated Covariance Matrix

```
                    Analysis of Farm Area and Corn Yield
                     Model I: Same Intercept and Slope

                          The SURVEYREG Procedure

              Regression Analysis for Dependent Variable CornYield

                          Tests of Model Effects

                  Effect       Num DF    F Value    Pr > F

                  Model             1      21.74    0.0004
                  Intercept         1       4.93    0.0433
                  FarmArea          1      21.74    0.0004

        NOTE: The denominator degrees of freedom for the F tests is 14.


                     Estimated Regression Coefficients

                                            Standard
          Parameter      Estimate          Error    t Value    Pr > |t|

          Intercept    11.8162978     5.31981027       2.22      0.0433
          FarmArea      0.2126576     0.04560949       4.66      0.0004

        NOTE: The denominator degrees of freedom for the t tests is 14.
```

Output 71.4.2 displays tests of model effects and the estimated regression coefficients.

Alternatively, you can assume that the linear relationship between corn yield (CornYield) and farm area (FarmArea) is different among the states (Model II). In order to analyze the data using this model, you create auxiliary variables FarmAreaNE and FarmAreaIA to represent farm area in different states:

$$\text{FarmAreaNE} = \begin{cases} 0 & \text{if the farm is in Iowa} \\ \text{FarmArea} & \text{if the farm is in Nebraska} \end{cases}$$

$$\text{FarmAreaIA} = \begin{cases} \text{FarmArea} & \text{if the farm is in Iowa} \\ 0 & \text{if the farm is in Nebraska} \end{cases}$$

The following statements create these variables in a new data set called FarmsByState and use PROC SURVEYREG to fit Model II:

```
title1 'Analysis of Farm Area and Corn Yield';
title2 'Model II: Same Intercept, Different Slopes';
data FarmsByState; set Farms;
   if State='Iowa' then do;
      FarmAreaIA=FarmArea ; FarmAreaNE=0; end;
   else do;
      FarmAreaIA=0 ; FarmAreaNE=FarmArea; end;
run;
```

The following statements perform the regression using the new data set FarmsByState. The analysis uses the auxilary variables FarmAreaIA and FarmAreaNE as the regressors:

```
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   model CornYield = FarmAreaIA FarmAreaNE;
   weight Weight;
run;
```

Output 71.4.3. Regression Results from Fitting Model II

```
                   Analysis of Farm Area and Corn Yield
                 Model II: Same Intercept, Different Slopes

                          The SURVEYREG Procedure

            Regression Analysis for Dependent Variable CornYield

                               Data Summary

                 Number of Observations                 19
                 Sum of Weights                  234.99900
                 Weighted Mean of CornYield       31.56029
                 Weighted Sum of CornYield          7416.6


                              Design Summary

                      Number of Strata             5


                              Fit Statistics

                       R-square            0.8158
                       Root MSE           11.6759
                       Denominator DF          14


                    Estimated Regression Coefficients

                                     Standard
       Parameter       Estimate         Error    t Value    Pr > |t|

       Intercept     4.04234816    3.80934848       1.06      0.3066
       FarmAreaIA    0.41696069    0.05971129       6.98      <.0001
       FarmAreaNE    0.12851012    0.02495495       5.15      0.0001

      NOTE: The denominator degrees of freedom for the t tests is 14.
```

Output 71.4.3 displays the data summary, design information, fit statistics, and parameter estimates. The estimated slope parameters for each state are quite different from the estimated slope in Model I. The results from the regression show that Model II fits these data better than Model I.

For Model III, different intercepts are used for the linear relationship in two states. The following statements illustrate the use of the NOINT option in the MODEL statement associated with the CLASS statement to fit Model III:

```
title2 'Model III: Different Intercepts and Slopes';
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   class State;
   model CornYield = State FarmAreaIA FarmAreaNE / noint covb solution;
   weight Weight;
run;
```

The model statement includes the classification effect State as a regressor. Therefore, the parameter estimates for effect State will presents the intercepts in two states.

Output 71.4.4. Regression Results for Fitting Model III

```
                      Analysis of Farm Area and Corn Yield
                   Model III: Different Intercepts and Slopes

                            The SURVEYREG Procedure

              Regression Analysis for Dependent Variable CornYield

                                 Data Summary

                    Number of Observations              19
                    Sum of Weights               234.99900
                    Weighted Mean of CornYield    31.56029
                    Weighted Sum of CornYield       7416.6


                                Design Summary

                         Number of Strata             5


                                Fit Statistics

                           R-square            0.9300
                           Root MSE           11.9810
                           Denominator DF          14


                       Estimated Regression Coefficients

                                                Standard
       Parameter            Estimate           Error    t Value    Pr > |t|

       State Iowa         5.27797099      5.27170400       1.00      0.3337
       State Nebraska     0.65275201      1.70031616       0.38      0.7068
       FarmAreaIA         0.40680971      0.06458426       6.30      <.0001
       FarmAreaNE         0.14630563      0.01997085       7.33      <.0001

        NOTE: The denominator degrees of freedom for the t tests is 14.


                  Covariance of Estimated Regression Coefficients

                                               State
                      State Iowa            Nebraska         FarmAreaIA        FarmAreaNE

 State Iowa         27.790863033                   0       -0.205517205                 0
 State Nebraska                0        2.8910750385                  0      -0.027354011
 FarmAreaIA         -0.205517205                   0       0.0041711265                 0
 FarmAreaNE                    0        -0.027354011                  0      0.0003988349
```

Output 71.4.4 displays the regression results for fitting Model III, including the data summary, parameter estimates, and covariance matrix of the regression coefficients. The estimated covariance matrix shows a lack of correlation between the regression coefficients from different states. This suggests that Model III might be the best choice for building a model for farm area and corn yield in these two states.

However, some statistics remain the same under different regression models, for example, Weighted Mean of CornYield. These estimators do not rely on the particular model you use.

Example 71.5. Regression Estimator for Stratified Sample

This example uses the corn yield data from the previous example to illustrate how to construct a regression estimator for a stratified sample design.

Similar to Example 71.3 on page 4400, by incorporating auxilary information into a regression estimator, the procedure can produce more accurate estimates of the population characteristics that are of interest. In this example, the sample design is a stratified sample design. The auxilary information is the total farm areas in regions of each state, as displayed in Table 71.4. You want to estimate the total corn yield using this information under the three linear models given in Example 71.4.

Table 71.4. Information for Each Stratum

			Number of Farms in		
Stratum	**State**	**Region**	**Population**	**Sample**	**Total Farm Area**
1	Iowa	1	100	3	13,200
2		2	50	5	
3		3	15	3	
4	Nebraska	1	30	6	8,750
5		2	40	2	
	Total		235	19	21,950

The regression estimator to estimate the total corn yield under Model I can be obtained by using PROC SURVEYREG with an ESTIMATE statement:

```
title1 'Estimate Corn Yield from Farm Size';
title2 'Model I: Same Intercept and Slope';
proc surveyreg data=Farms total=StratumTotals;
   strata State Region / list;
   class  State Region;
   model  CornYield = FarmArea State*Region /solution;
   weight Weight;
   estimate 'Estimate of CornYield under Model I'
           INTERCEPT 235 FarmArea 21950
           State*Region 100 50 15 30 40 /e;
run;
```

To apply the contraint in each stratum that the weighted total number of farms equals to the total number of farms in the stratum, you can include the strata as an effect in the MODEL statement, effect State*Region. Thus, the CLASS statement must list the STRATA variables, State and Region, as classification variables. The following

ESTIMATE statement specifies the regression estimator, which is a linear function of the regression parameters:

```
      estimate 'Estimate of CornYield under Model I'
              INTERCEPT 235 FarmArea 21950
              State*Region 100 50 15 30 40 /e;
```

This linear function contains the total for each explanatory variable in the model. Because the sampling units are farms in this example, the coefficient for Intercept in the ESTIMATE statement is the total number of farms (235); the coefficient for FarmArea is the total farm area listed in Table 71.4 (21950); and the coefficients for effect State*Region are the total number of farms in each strata (as displayed in Table 71.4).

Output 71.5.1. Regression Estimator for the Total of CornYield under Model I

```
                      Estimate Corn Yield from Farm Size
                      Model I: Same Intercept and Slope

                          The SURVEYREG Procedure

             Regression Analysis for Dependent Variable CornYield

                       Analysis of Estimable Functions

                                                       Standard
Parameter                               Estimate          Error  t Value  Pr > |t|

Estimate of CornYield under Model I   7463.52329   926.841541     8.05    <.0001

       NOTE: The denominator degrees of freedom for the t tests is 14.
```

Output 71.5.1 displays the results of the ESTIMATE statement. The regression estimator for the total of CornYield in Iowa and Nebraska is 7464 under Model I, with a standard error of 927.

Under Model II, a regression estimator for totals can be obtained using the following statements:

```
   title1 'Estimate Corn Yield from Farm Size';
   title2 'Model II: Same Intercept, Different Slopes';
   proc surveyreg data=FarmsByState total=StratumTotals;
      strata State Region;
      class  State Region;
      model  CornYield = FarmAreaIA FarmAreaNE
                         state*region /solution;
      weight Weight;
      estimate 'Total of CornYield under Model II'
              INTERCEPT 235 FarmAreaIA 13200 FarmAreaNE 8750
              State*Region 100 50 15 30 40 /e;
   run;
```

In this model, you also need to include strata as a fixed effect in the MODEL statement. Other regressors are the auxiliary variables FarmAreaIA and FarmAreaNE (defined in Example 71.4). In the following ESTIMATE statement, the coefficient for Intercept is still the total number of farms; and the coefficients for FarmAreaIA and FarmAreaNE are the total farm area in Iowa and Nebraska, respectively, as displayed in Table 71.4. The total number of farms in each strata are the coefficients for the strata effect:

```
estimate 'Total of CornYield under Model II'
        INTERCEPT 235 FarmAreaIA 13200 FarmAreaNE 8750
        State*Region 100 50 15 30 40 /e;
```

Output 71.5.2. Regression Estimator for the Total of CornYield under Model II

```
                          Estimate Corn Yield from Farm Size
                       Model II: Same Intercept, Different Slopes

                              The SURVEYREG Procedure

                    Regression Analysis for Dependent Variable CornYield

                           Analysis of Estimable Functions

                                                         Standard
Parameter                             Estimate            Error    t Value   Pr > |t|

Total of CornYield under Model II   7580.48657      859.180439      8.82     <.0001

        NOTE: The denominator degrees of freedom for the t tests is 14.
```

Output 71.5.2 displays that the results of the regression estimator for the total of corn yield in two states under Model II is 7580 with a standard error of 859. The regression estimator under Model II has a slightly smaller standard error than under Model I.

Finally, you can apply Model III to the data and estimate the total corn yield. Under Model III, you can also obtain the regression estimators for the total corn yield for each state. Three ESTIMATE statements are used in the following statements to create the three regression estimators:

```
title1 'Estimate Corn Yield from Farm Size';
title2 'Model III: Different Intercepts and Slopes';
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   class  State Region;
   model  CornYield = state FarmAreaIA FarmAreaNE
      State*Region /noint solution;
   weight Weight;
   estimate 'Total CornYield in Iowa under Model III'
            State 165 0 FarmAreaIA 13200 FarmAreaNE  0
            State*region 100 50 15  0  0 /e;
   estimate 'Total CornYield in Nebraska under Model III'
```

```
                State 0 70 FarmAreaIA 0 FarmAreaNE 8750
                State*Region 0 0 0 30 40 /e;
      estimate 'Total CornYield in both states under Model III'
                State 165 70 FarmAreaIA 13200 FarmAreaNE 8750
                State*Region 100 50 15 30 40 /e;
   run;
```

The fixed effect State is added to the MODEL statement to obtain different intercepts in different states, using the NOINT option. Among the ESTIMATE statements, the coefficients for explanatory variables are different depending on which regression estimator is estimated. For example, in the ESTIMATE statement

```
      estimate 'Total CornYield in Iowa under Model III'
                State 165 0 FarmAreaIA 13200 FarmAreaNE  0
                State*region 100 50 15  0  0 /e;
```

the coefficients for the effect State are 165 and 0, respectively. This indicates that the total number of farms in Iowa is 165 and the total number of farms in Nebraska is 0, because the estimation is the total corn yield in Iowa only. Similarly, the total numbers of farms in three regions in Iowa are used for the coefficients of the strata effect State*Region, as displayed in Table 71.4.

Output 71.5.3. Regression Estimator for the Total of CornYield under Model III

```
                        Estimate Corn Yield from Farm Size
                     Model III: Different Intercepts and Slopes

                              The SURVEYREG Procedure

                 Regression Analysis for Dependent Variable CornYield

                           Analysis of Estimable Functions

                                                               Standard
Parameter                                          Estimate       Error  t Value

Total CornYield in Iowa under Model III          6246.10697  851.272372     7.34
Total CornYield in Nebraska under Model III      1334.37961  116.302948    11.47
Total CornYield in both states under Model III   7580.48657  859.180439     8.82

                           Analysis of Estimable Functions

            Parameter                                        Pr > |t|

            Total CornYield in Iowa under Model III            <.0001
            Total CornYield in Nebraska under Model III        <.0001
            Total CornYield in both states under Model III     <.0001

         NOTE: The denominator degrees of freedom for the t tests is 14.
```

Output 71.5.3 displays the results from the three regression estimators using Model III. Since the estimations are independent in each state, the total corn yield from both states is equal to the sum of the estimated total of corn yield in Iowa and Nebraska,

$6246 + 1334 = 7580$. This regression estimator is the same as the one under Model II. The variance of regression estimator of the total corn yield in both states is the sum of variances of regression estimators for total corn yield in each state. Therefore, it is not necessary to use Model III to obtain the regression estimator for the total corn yield unless you need to estimate the total corn yield for each individual state.

Example 71.6. Stratum Collapse

In a stratified sample, it is possible that some strata will have only one sampling unit. When this happens, PROC SURVEYREG collapses the strata that contain a single sampling unit into a pooled stratum. For more detailed information on stratum collapse, see the section "Stratum Collapse" on page 4388.

Suppose that you have the following data:

```
data Sample;
   input Stratum X Y W;
   datalines;
10 0 0 5
10 1 1 5
11 1 1 10
11 1 2 10
12 3 3 16
33 4 4 45
14 6 7 50
12 3 4 16
;
```

The variable Stratum is again the stratification variable, the variable X is the independent variable, and the variable Y is the dependent variable. You want to regress Y on X. In the data set Sample, both Stratum=33 and Stratum=14 contain one observation. By default, PROC SURVEYREG collapses these strata into one pooled stratum in the regression analysis.

To input the finite population correction information, you create the SAS data set StratumTotals:

```
data StratumTotals;
   input Stratum _TOTAL_;
   datalines;
10 10
11 20
12 32
33 40
33 45
14 50
15  .
66 70
;
```

The variable Stratum is the stratification variable, and the variable _TOTAL_ contains the stratum totals. The data set StratumTotals contains more strata than the data set Sample. Also in the data set StratumTotals, more than one observation contains the stratum totals for Stratum=33:

```
33 40
33 45
```

PROC SURVEYREG allows this type of input. The procedure simply ignores strata that are not present in the data set Sample; for the multiple entries of a stratum, the procedure uses the first observation. In this example, Stratum=33 has the stratum total _TOTAL_=40.

The following SAS statements perform the regression analysis:

```
title1 'Stratified Sample with Single Sampling Unit in Strata';
title2 'With Stratum Collapse';
proc SURVEYREG data=Sample total=StratumTotals;
   strata Stratum/list;
   model Y=X;
   weight W;
run;
```

Output 71.6.1. Summary of Data and Regression

```
           Stratified Sample with Single Sampling Unit in Strata
                           With Stratum Collapse

                          The SURVEYREG Procedure

                Regression Analysis for Dependent Variable Y

                               Data Summary

                 Number of Observations             8
                 Sum of Weights             157.00000
                 Weighted Mean of Y           4.31210
                 Weighted Sum of Y          677.00000


                              Design Summary

               Number of Strata                        5
               Number of Strata Collapsed              2


                              Fit Statistics

                      R-square            0.9564
                      Root MSE            0.5111
                      Denominator DF           4
```

Output 71.6.1 displays that there are a total of five strata in the input data set, and two strata are collapsed into a pooled stratum. The denominator degrees of freedom is 4, due to the collapse (see the section “Denominator Degrees of Freedom” on page 4386).

Output 71.6.2. Stratification Information

```
             Stratified Sample with Single Sampling Unit in Strata
                             With Stratum Collapse

                            The SURVEYREG Procedure

                  Regression Analysis for Dependent Variable Y

                              Stratum Information

    Stratum                                          Population    Sampling
     Index     Collapsed    Stratum       N Obs           Total        Rate

       1                      10              2              10      20.0%
       2                      11              2              20      10.0%
       3                      12              2              32      6.25%
       4        Yes           14              1              50      2.00%
       5        Yes           33              1              40      2.50%

       0        Pooled                        2              90      2.22%

NOTE: Strata with only one observation are collapsed into the stratum with
      Stratum Index "0".
```

Output 71.6.2 displays the stratification information, including stratum collapse. Under the column Collapsed, the fourth stratum (Stratum=14) and the fifth (Stratum=33) are marked as ‘Yes’, which indicates that these two strata are collapsed into the pooled stratum (Stratum Index=0). The sampling rate for the pooled stratum is 2% (see the section “Sampling Rate of the Pooled Stratum from Collapse” on page 4389).

Output 71.6.3. Parameter Estimates and Effect Tests

```
           Stratified Sample with Single Sampling Unit in Strata
                          With Stratum Collapse

                         The SURVEYREG Procedure

                Regression Analysis for Dependent Variable Y

                           Tests of Model Effects

                  Effect       Num DF    F Value    Pr > F

                  Model             1     173.01    0.0002
                  Intercept         1       0.00    0.9961
                  X                 1     173.01    0.0002

       NOTE: The denominator degrees of freedom for the F tests is 4.


                     Estimated Regression Coefficients

                                          Standard
        Parameter      Estimate          Error    t Value    Pr > |t|

        Intercept    0.00179469     0.34306373       0.01      0.9961
        X            1.12598708     0.08560466      13.15      0.0002

       NOTE: The denominator degrees of freedom for the t tests is 4.
```

Output 71.6.3 displays the parameter estimates and the tests of the significance of the model effects.

Alternatively, if you prefer not to collapse strata with a single sampling unit, you can specify the NOCOLLAPSE option in the STRATA statement:

```
title1 'Stratified Sample with Single Sampling Unit in Strata';
title2 'Without Stratum Collapse';
proc SURVEYREG data=Sample total=StratumTotals;
   strata Stratum/list nocollapse;
model Y = X;
weight W;
run;
```

Output 71.6.4. Summary of Data and Regression

```
              Stratified Sample with Single Sampling Unit in Strata
                             Without Stratum Collapse

                             The SURVEYREG Procedure

                   Regression Analysis for Dependent Variable Y

                                  Data Summary

                     Number of Observations              8
                     Sum of Weights              157.00000
                     Weighted Mean of Y            4.31210
                     Weighted Sum of Y           677.00000


                                 Design Summary

                        Number of Strata             5


                                 Fit Statistics

                          R-square            0.9564
                          Root MSE            0.5111
                          Denominator DF           3
```

Output 71.6.4 does not contain the stratum collapse information displayed in Output 71.6.1, and the denominator degrees of freedom is 3 instead of 4.

Output 71.6.5. Stratification Information

```
              Stratified Sample with Single Sampling Unit in Strata
                             Without Stratum Collapse

                             The SURVEYREG Procedure

                   Regression Analysis for Dependent Variable Y

                               Stratum Information

           Stratum                                   Population    Sampling
            Index     Stratum        N Obs                Total        Rate

              1          10              2                   10       20.0%
              2          11              2                   20       10.0%
              3          12              2                   32       6.25%
              4          14              1                   50       2.00%
              5          33              1                   40       2.50%
```

In Output 71.6.5, although the fourth stratum and the fifth stratum contain only one observation, no stratum collapse occurs.

Output 71.6.6. Parameter Estimates and Effect Tests

```
          Stratified Sample with Single Sampling Unit in Strata
                       Without Stratum Collapse

                       The SURVEYREG Procedure

               Regression Analysis for Dependent Variable Y

                        Tests of Model Effects

                 Effect       Num DF    F Value    Pr > F

                 Model             1     347.27    0.0003
                 Intercept         1       0.00    0.9962
                 X                 1     347.27    0.0003

      NOTE: The denominator degrees of freedom for the F tests is 3.


                    Estimated Regression Coefficients

                                          Standard
       Parameter      Estimate               Error    t Value    Pr > |t|

       Intercept    0.00179469      0.34302581       0.01      0.9962
       X            1.12598708      0.06042241      18.64      0.0003

      NOTE: The denominator degrees of freedom for the t tests is 3.
```

As a result of not collapsing strata, the standard error estimates of the parameters are different from those in Output 71.6.3, as are the tests of the significance of model effects are.

References

Cochran, W.G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Foreman, E.K. (1991), *Survey Sampling Principles*, New York: Marcel Dekker, Inc.

Fuller, W.A. (1975), “Regression Analysis for Sample Survey,” *Sankhyā*, 37 (3), Series C, 117–132.

Fuller, W.A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H.J. (1989), *PC CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Hidiroglou, M.A., Fuller, W.A., and Hickman, R.D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Pringle, R.M. and Raynor, A.A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Särndal, C.E., Swenson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag Inc.

Woodruff, R.S. (1971), “A Simple Method for Approximating the Variance of a Complicated Estimate,” *Journal of the American Statistical Association*, 66, 411–414.

Chapter 72
The SURVEYSELECT Procedure

Chapter Contents

Chapter 72

The SURVEYSELECT Procedure

Overview

The SURVEYSELECT procedure provides a variety of methods for selecting probability-based random samples. The procedure can select a simple random sample or can sample according to a complex multistage sample design that includes stratification, clustering, and unequal probabilities of selection. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population.

To select a sample with PROC SURVEYSELECT, you input a SAS data set that contains the sampling frame or list of units from which the sample is to be selected. You also specify the selection method, the desired sample size or sampling rate, and other selection parameters. The SURVEYSELECT procedure selects the sample, producing an output data set that contains the selected units, their selection probabilities, and sampling weights. When you are selecting a sample in multiple stages, you invoke the procedure separately for each stage of selection, inputting the frame and selection parameters for each current stage.

The SURVEYSELECT procedure provides methods for both equal probability sampling and probability proportional to size (PPS) sampling. In equal probability sampling, each unit in the sampling frame, or in a stratum, has the same probability of being selected for the sample. In PPS sampling, a unit's selection probability is proportional to its size measure. For details on probability sampling methods, refer to Lohr (1999), Kish (1965, 1987), Kalton (1983), and Cochran (1977).

The SURVEYSELECT procedure provides the following equal probability sampling methods:

- simple random sampling
- unrestricted random sampling (with replacement)
- systematic random sampling
- sequential random sampling

This procedure also provides the following probability proportional to size (PPS) methods:

- PPS sampling without replacement
- PPS sampling with replacement
- PPS systematic sampling
- PPS algorithms for selecting two units per stratum
- sequential PPS sampling with minimum replacement

The procedure uses fast, efficient algorithms for these sample selection methods. Thus, it performs well even for large input data sets or sampling frames, which may occur in practice for large-scale sample surveys.

The SURVEYSELECT procedure can perform stratified sampling, selecting samples independently within the specified strata, or nonoverlapping subgroups of the survey population. Stratification controls the distribution of the sample size in the strata. It is widely used in practice toward meeting a variety of survey objectives. For example, with stratification you can ensure adequate sample sizes for subgroups of interest, including small subgroups, or you can use stratification toward improving the precision of the overall estimates. When you are using a systematic or sequential selection method, the SURVEYSELECT procedure also can sort by control variables within strata for the additional control of implicit stratification.

The SURVEYSELECT procedure provides replicated sampling, where the total sample is composed of a set of replicates, each selected in the same way. You can use replicated sampling to study variable nonsampling errors, such as variability in the results obtained by different interviewers. You can also use replication to compute standard errors for the combined sample estimates.

Getting Started

In this example, an Internet service provider conducts a customer satisfaction survey. The survey population consists of the company's current subscribers. The company plans to select a sample of customers from this population, interview the selected customers, and then make inferences about the entire survey population from the sample data.

The SAS data set Customers contains the sampling frame, which is the list of units in the survey population. The sample of customers will be selected from this sampling frame. The data set Customers is constructed from the company's customer database. It contains one observation for each customer, with a total of 13,471 observations. Figure 72.1 displays the first 10 observations of the data set Customers.

```
                 Internet Service Provider Customers
                       (First 10 Observations)

            Obs     CustomerID     State     Type      Usage

              1    416-87-4322      AL       New         839
              2    288-13-9763      GA       Old         224
              3    339-00-8654      GA       Old        2451
              4    118-98-0542      GA       New         349
              5    421-67-0342      FL       New         562
              6    623-18-9201      SC       New          68
              7    324-55-0324      FL       Old         137
              8    832-90-2397      AL       Old        1563
              9    586-45-0178      GA       New         615
             10    801-24-5317      SC       New         728
```

Figure 72.1. Customers Data Set (First 10 Observations)

In the SAS data set Customers, the variable CustomerID uniquely identifies each customer. The variable State contains the state of the customer's address. The company has customers in the following four states: Georgia (GA), Alabama (AL), Florida (FL), and South Carolina (SC). The variable Type equals 'Old' if the customer has subscribed to the service for more than one year; otherwise, the variable Type equals 'New'. The variable Usage contains the customer's average monthly service usage, in minutes.

The following sections illustrate the use of PROC SURVEYSELECT for probability sampling with three different designs for the customer satisfaction survey. All three designs are one stage, with customers as the sampling units. The first design is simple random sampling without stratification. In the second design, customers are stratified by state and type, and the sample is selected by simple random sampling within strata. In the third design, customers are sorted within strata by usage, and the sample is selected by systematic random sampling within strata.

Simple Random Sampling

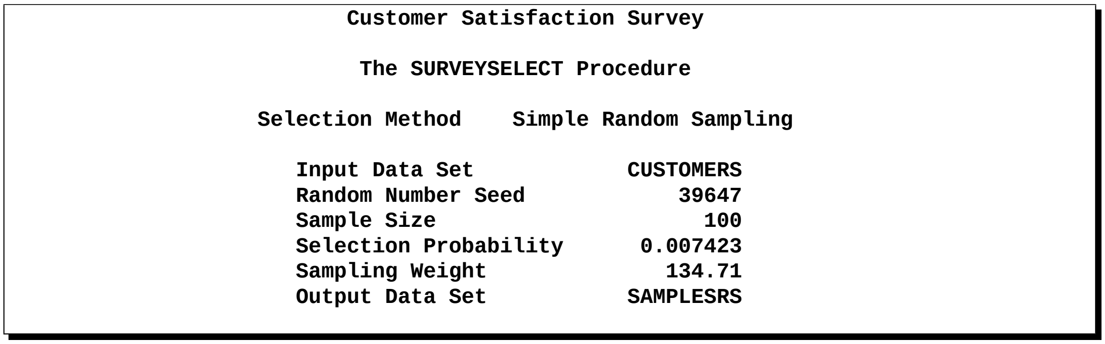
The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set using simple random sampling:

```
title 'Customer Satisfaction Survey';
proc surveyselect data=Customers
   method=srs n=100
   out=SampleSRS;
run;
```

The PROC SURVEYSELECT statement invokes the procedure. The DATA= option names the SAS data set Customers as the input data set from which to select the sample. The METHOD=SRS option specifies simple random sampling as the sample selection method. In simple random sampling, each unit has an equal probability of selection, and sampling is without replacement. Without-replacement sampling means that a unit cannot be selected more than once. The N=100 option specifies a sample size of 100 customers. The OUT= option stores the sample in the SAS data set named SampleSRS.

Figure 72.2 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A sample of 100 customers is selected from the data set Customers by simple random sampling. With simple random sampling and no stratification in the sample design, the selection probability is the same for all units in the sample. In this sample, the selection probability for each customer equals 0.007423, which is the sample size (100) divided by the population size (13,471). The sampling weight equals 134.71 for each customer in the sample, where the weight is the inverse of the selection probability. If you specify the STATS option, PROC SURVEYSELECT includes the selection probabilities and sampling weights in the output data set. (This information is always included in the output data set for more complex designs.)

The random number seed is 39647. PROC SURVEYSELECT uses this number as the initial seed for random number generation. Since the SEED= option is not specified in the PROC SURVEYSELECT statement, the seed value is obtained using the time of day from the computer's clock. You can specify SEED=39647 to reproduce this sample.



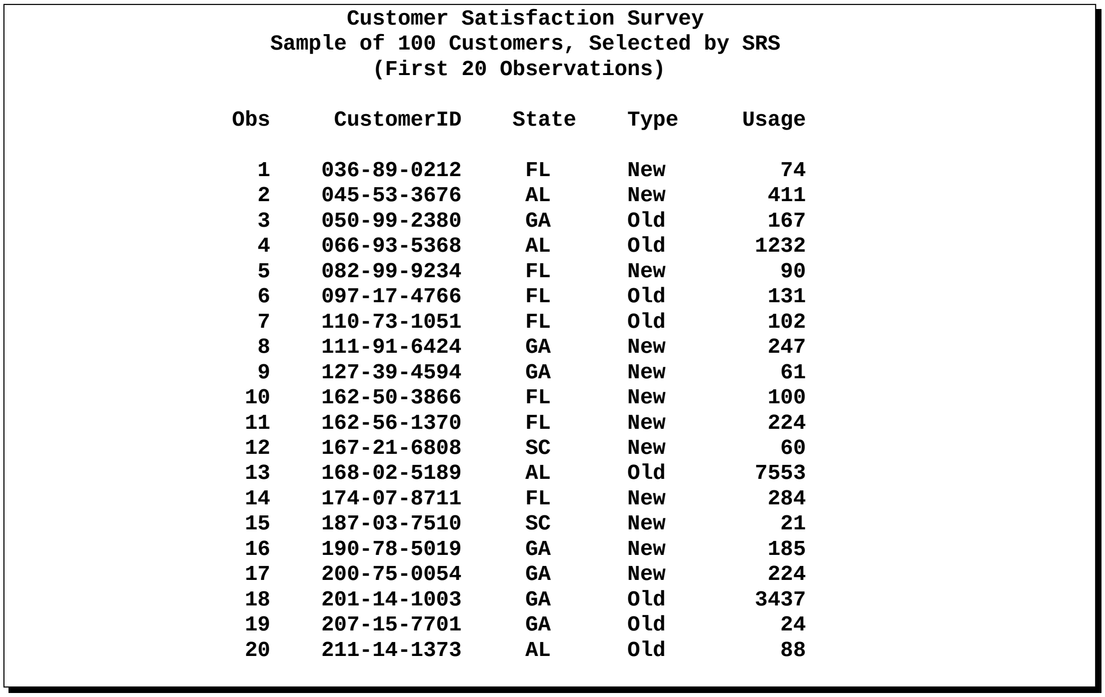
```
                          Customer Satisfaction Survey

                            The SURVEYSELECT Procedure

                    Selection Method    Simple Random Sampling

                       Input Data Set            CUSTOMERS
                       Random Number Seed            39647
                       Sample Size                     100
                       Selection Probability      0.007423
                       Sampling Weight              134.71
                       Output Data Set            SAMPLESRS
```

Figure 72.2. Sample Selection Summary

The sample of 100 customers is stored in the SAS data set SampleSRS. PROC SURVEYSELECT does not display this output data set. The following PROC PRINT statements display the first 20 observations of SampleSRS:

```
title1 'Customer Satisfaction Survey';
title2 'Sample of 100 Customers, Selected by SRS';
title3 '(First 20 Observations)';
proc print data=SampleSRS(obs=20);
run;
```

Figure 72.3 displays the first 20 observations of the output data set SampleSRS, which contains the sample of customers. This data set includes all the variables from the DATA= input data set Customers. If you do not want to include all variables, you can use the ID statement to specify which variables to copy from the input data set to the output (sample) data set.



```
                    Customer Satisfaction Survey
              Sample of 100 Customers, Selected by SRS
                      (First 20 Observations)

          Obs     CustomerID     State     Type     Usage

            1    036-89-0212      FL       New         74
            2    045-53-3676      AL       New        411
            3    050-99-2380      GA       Old        167
            4    066-93-5368      AL       Old       1232
            5    082-99-9234      FL       New         90
            6    097-17-4766      FL       Old        131
            7    110-73-1051      FL       Old        102
            8    111-91-6424      GA       New        247
            9    127-39-4594      GA       New         61
           10    162-50-3866      FL       New        100
           11    162-56-1370      FL       New        224
           12    167-21-6808      SC       New         60
           13    168-02-5189      AL       Old       7553
           14    174-07-8711      FL       New        284
           15    187-03-7510      SC       New         21
           16    190-78-5019      GA       New        185
           17    200-75-0054      GA       New        224
           18    201-14-1003      GA       Old       3437
           19    207-15-7701      GA       Old         24
           20    211-14-1373      AL       Old         88
```

Figure 72.3. Customer Sample (First 20 Observations)

Stratified Sampling

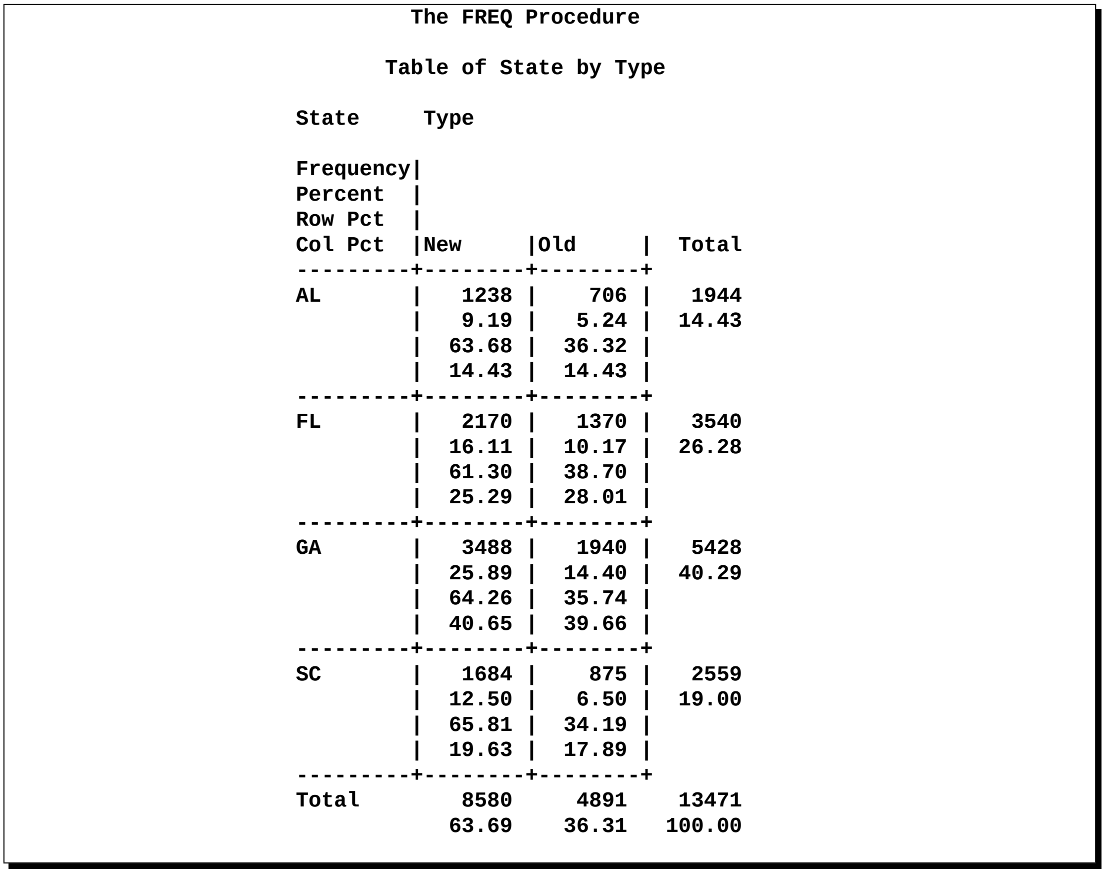
In this section, stratification is added to the sample design for the customer satisfaction survey. The sampling frame, or list of all customers, is stratified by State and Type. This divides the sampling frame into nonoverlapping subgroups formed from the values of the State and Type variables. Samples are then selected independently within the strata.

PROC SURVEYSELECT requires that the input data set be sorted by the STRATA variables. The following PROC SORT statements sort the Customers data set by the stratification variables State and Type:

```
proc sort data=Customers;
   by State Type;
run;
```

The following PROC FREQ statements display the crosstabulation of the Customers data set by State and Type:

```
proc freq data=Customers;
   tables State*Type;
run;
```



```
                              The FREQ Procedure

                            Table of State by Type

                      State     Type

                      Frequency|
                      Percent  |
                      Row Pct  |
                      Col Pct  |New     |Old     |  Total
                      ---------+--------+--------+
                      AL       |   1238 |    706 |   1944
                               |   9.19 |   5.24 |  14.43
                               |  63.68 |  36.32 |
                               |  14.43 |  14.43 |
                      ---------+--------+--------+
                      FL       |   2170 |   1370 |   3540
                               |  16.11 |  10.17 |  26.28
                               |  61.30 |  38.70 |
                               |  25.29 |  28.01 |
                      ---------+--------+--------+
                      GA       |   3488 |   1940 |   5428
                               |  25.89 |  14.40 |  40.29
                               |  64.26 |  35.74 |
                               |  40.65 |  39.66 |
                      ---------+--------+--------+
                      SC       |   1684 |    875 |   2559
                               |  12.50 |   6.50 |  19.00
                               |  65.81 |  34.19 |
                               |  19.63 |  17.89 |
                      ---------+--------+--------+
                      Total        8580     4891    13471
                                  63.69    36.31   100.00
```

Figure 72.4. Stratification of Customers by State and Type

Figure 72.4 presents the table of State by Type for the 13,471 customers. There are four states and two levels of Type, forming a total of eight strata.

The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set according to the stratified sample design:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling';
proc surveyselect data=Customers
      method=srs n=15
      seed=1953 out=SampleStrata;
   strata State Type;
run;
```

The STRATA statement names the stratification variables State and Type. In the PROC SURVEYSELECT statement, the METHOD=SRS option specifies simple random sampling. The N=15 option specifies a sample size of 15 customers for each stratum. If you want to specify different sample sizes for different strata, you can use the N=*SAS-data-set* option to name a secondary data set that contains the stratum sample sizes. The SEED=1953 option specifies '1953' as the initial seed for random number generation.

```
                    Customer Satisfaction Survey
                        Stratified Sampling

                     The SURVEYSELECT Procedure

              Selection Method    Simple Random Sampling
              Strata Variables    State
                                  Type

                 Input Data Set            CUSTOMERS
                 Random Number Seed             1953
                 Stratum Sample Size              15
                 Number of Strata                  8
                 Total Sample Size               120
                 Output Data Set         SAMPLESTRATA
```

Figure 72.5. Sample Selection Summary

Figure 72.5 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A total of 120 customers are selected.

The following PROC PRINT statements display the first 30 observations of the output data set SampleStrata:

```
title1 'Customer Satisfaction Survey';
title2 'Sample Selected by Stratified Design';
title3 '(First 30 Observations)';
proc print data=SampleStrata(obs=30);
run;
```

Figure 72.6 displays the first 30 observations of the output data set SampleStrata, which contains the sample of 120 customers, 15 customers from each of the eight strata. The variable SelectionProb contains the selection probability for each customer in the sample. Since customers are selected with equal probability within strata in this design, the selection probability equals the stratum sample size (15) divided by the stratum population size. The selection probabilities differ from stratum to stratum since the population sizes differ. The selection probability for each customer in the first stratum (State='AL' and Type='New') is 0.012116, and the selection probability is 0.021246 for customers in the second stratum. The variable SamplingWeight contains the sampling weights, which are computed as inverse selection probabilities.

```
                          Customer Satisfaction Survey
                      Sample Selected by Stratified Design
                            (First 30 Observations)

                                                      Selection    Sampling
    Obs    State    Type      CustomerID     Usage       Prob       Weight

      1     AL      New      002-26-1498      1189     0.012116     82.5333
      2     AL      New      070-86-8494       106     0.012116     82.5333
      3     AL      New      121-28-6895        76     0.012116     82.5333
      4     AL      New      131-79-7630       265     0.012116     82.5333
      5     AL      New      211-88-4991       108     0.012116     82.5333
      6     AL      New      222-81-3742        83     0.012116     82.5333
      7     AL      New      238-46-3776       278     0.012116     82.5333
      8     AL      New      370-01-0671       123     0.012116     82.5333
      9     AL      New      407-07-5479      1580     0.012116     82.5333
     10     AL      New      550-90-3188       177     0.012116     82.5333
     11     AL      New      582-40-9610        46     0.012116     82.5333
     12     AL      New      672-59-9114        66     0.012116     82.5333
     13     AL      New      848-60-3119        28     0.012116     82.5333
     14     AL      New      886-83-4909       170     0.012116     82.5333
     15     AL      New      993-31-7677        64     0.012116     82.5333
     16     AL      Old      124-60-0495        80     0.021246     47.0667
     17     AL      Old      128-54-9590        56     0.021246     47.0667
     18     AL      Old      204-05-4017        17     0.021246     47.0667
     19     AL      Old      210-68-8704      4363     0.021246     47.0667
     20     AL      Old      239-75-4343       430     0.021246     47.0667
     21     AL      Old      317-70-6496       452     0.021246     47.0667
     22     AL      Old      365-37-1340        21     0.021246     47.0667
     23     AL      Old      399-78-7900       108     0.021246     47.0667
     24     AL      Old      404-90-6273       824     0.021246     47.0667
     25     AL      Old      421-04-8548      1332     0.021246     47.0667
     26     AL      Old      604-48-0587        16     0.021246     47.0667
     27     AL      Old      774-04-0162       318     0.021246     47.0667
     28     AL      Old      849-66-4156        79     0.021246     47.0667
     29     AL      Old      937-69-9106       182     0.021246     47.0667
     30     AL      Old      985-09-8691        24     0.021246     47.0667
```

Figure 72.6. Customer Sample (First 30 Observations)

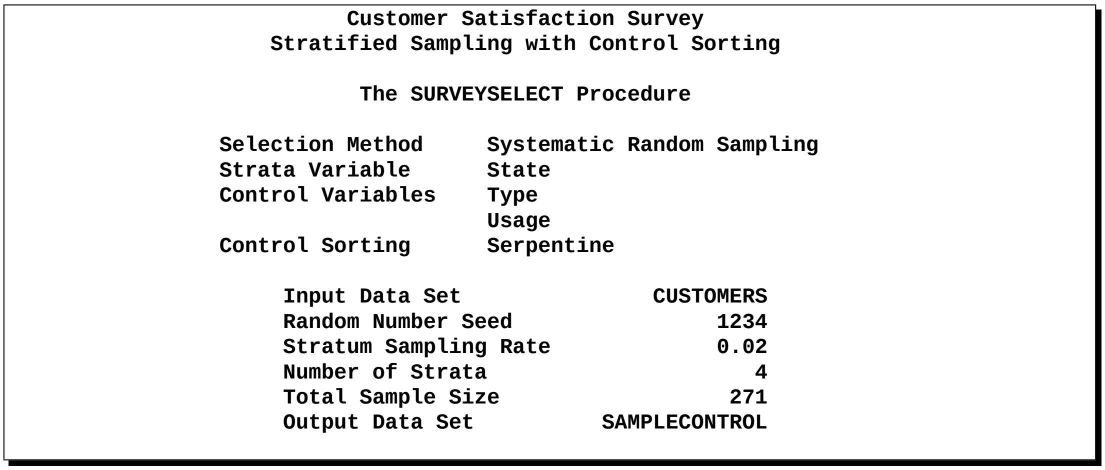
Stratified Sampling with Control Sorting

The next sample design for the customer satisfaction survey uses stratification by State. The sampling frame is also sorted by Type and Usage before sample selection, to provide additional control over the distribution of the sample. Customers are then selected by systematic random sampling within strata. Selection by systematic sampling, together with control sorting, spreads the sample uniformly over the range of type and usage values within each stratum or state. The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set using this design:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling with Control Sorting';
proc surveyselect data=Customers
      method=sys rate=.02
      seed=1234 out=SampleControl;
   strata State;
   control Type Usage;
run;
```

The STRATA statement names the stratification variable State. The CONTROL statement names the control variables Type and Usage. In the PROC SURVEYSELECT statement, the METHOD=SYS option requests systematic random sampling. The RATE=.02 option specifies a sampling rate of 2% for each stratum. The SEED=1234 option specifies the initial seed for random number generation.

Figure 72.7 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A sample of 271 customers is selected, using systematic random sampling within strata determined by State. The sampling frame Customers is sorted by control variables Type and Usage within strata. The type of sorting is serpentine, which is used by default since SORT=NEST is not specified. See the section "Sorting by CONTROL Variables" on page 4445 for a description of serpentine sorting. The sorted data set replaces the input data set. (To store the sorted input data in another data set, leaving the input data set unsorted, use the OUTSORT= option.) The output data set SampleControl contains the sample of customers.



```
                   Customer Satisfaction Survey
             Stratified Sampling with Control Sorting

                    The SURVEYSELECT Procedure

        Selection Method     Systematic Random Sampling
        Strata Variable      State
        Control Variables    Type
                             Usage
        Control Sorting      Serpentine

             Input Data Set               CUSTOMERS
             Random Number Seed                1234
             Stratum Sampling Rate             0.02
             Number of Strata                     4
             Total Sample Size                  271
             Output Data Set          SAMPLECONTROL
```

Figure 72.7. Sample Selection Summary

Syntax

The following statements are available in PROC SURVEYSELECT:

> **PROC SURVEYSELECT** *options* **;**
> **STRATA** *variables* **;**
> **CONTROL** *variables* **;**
> **SIZE** *variable* **;**
> **ID** *variables* **;**

The PROC SURVEYSELECT statement invokes the procedure and optionally identifies input and output data sets. It also specifies the selection method, the sample size, and other sample design parameters. The SURVEYSELECT statement is required.

The SIZE statement identifies the variable that contains the size measures. It is required for any selection method that is probability proportional to size (PPS).

The remaining statements are optional. The STRATA statement identifies a variable or set of variables that stratify the input data set. When you specify a STRATA statement, PROC SURVEYSELECT selects samples independently from the strata formed by the STRATA variables. The CONTROL statement identifies variables for ordering units within strata. It can be used for systematic and sequential sampling methods. The ID statement identifies variables to copy from the input data set to the output data set of selected units.

The rest of this section gives detailed syntax information for the CONTROL, ID, SIZE, and STRATA statements in alphabetical order after the description of the PROC SURVEYSELECT statement.

PROC SURVEYSELECT Statement

> **PROC SURVEYSELECT** *options* **;**

The PROC SURVEYSELECT statement invokes the procedure and optionally identifies input and output data sets. If you do not name a DATA= input data set, the procedure selects the sample from the most recently created SAS data set. If you do not name an OUT= output data set to contain the sample of selected units, the procedure still creates an output data set and names it according to the DATA*n* convention.

The PROC SURVEYSELECT statement also specifies the sample selection method, the sample size, and other sample design parameters. If you do not specify a selection method, PROC SURVEYSELECT uses simple random sampling (METHOD=SRS) if there is no SIZE statement. If you specify a SIZE statement but do not specify a selection method, PROC SURVEYSELECT uses probability proportional to size selection without replacement (METHOD=PPS). You must specify the sample size or sampling rate unless you request a method that selects two units from each stratum (METHOD=PPS_BREWER or METHOD=PPS_MURTHY).

You can use the SAMPSIZE=*n* option to specify the sample size, or you can use the SAMPSIZE=*SAS-data-set* option to name a secondary input data set that contains stratum sample sizes. You can also specify stratum sampling rates, minimum size measures, maximum size measures, and certainty size measures in the secondary input data set. See the descriptions of the SAMPSIZE=, SAMPRATE=, MINSIZE=, MAXSIZE=, and CERTSIZE= options. You can name only one secondary input data set in each invocation of the procedure.

The following table lists the options available with the PROC SURVEYSELECT statement. Descriptions follow in alphabetical order.

Table 72.1. PROC SURVEYSELECT Statement Options

Task	Options
Specify the input data set	DATA=
Specify output data sets	OUT= OUTSORT=
Suppress displayed output	NOPRINT
Specify selection method	METHOD=
Specify sample size	SAMPSIZE= SELECTALL
Specify sampling rate	SAMPRATE= NMIN= NMAX=
Specify number of replicates	REP=
Adjust size measures	MINSIZE= MAXSIZE=
Specify certainty size measures	CERTSIZE=
Specify sorting type	SORT=
Specify random number seed	SEED=
Control OUT= contents	JTPROBS OUTALL OUTHITS OUTSEED OUTSIZE STATS

You can specify the following options in the PROC SURVEYSELECT statement:

CERTSIZE

requests automatic selection of those units with size measures greater than or equal to the stratum certainty size measure. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum certainty size measures in the secondary input data set variable _CERTSIZE_. Use the CERTSIZE option when you have already named the secondary input data set in another option, such as SAMPSIZE=*SAS-data-set*, SAMPRATE=*SAS-data-set*, MAXSIZE=*SAS-data-set*, or MINSIZE=*SAS-data-set*. You can name only one secondary input data set in each invocation of the procedure.

If any unit's size measure is greater than or equal to the certainty size measure for its stratum, then PROC SURVEYSELECT selects this unit with certainty. Each certainty size measure must be a positive number. The CERTSIZE option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

If you want to specify a single certainty size measure in the PROC SURVEYSELECT statement, use the CERTSIZE=*certain* option.

CERTSIZE=*certain*

specifies the certainty size measure. PROC SURVEYSELECT selects with certainty any unit with size measure greater than or equal to the value *certain*, which must be a positive number. You provide size measures in the DATA= input data set variable named in the SIZE statement. This option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

If you request a stratified sample design with a STRATA statement and specify the CERTSIZE= option, PROC SURVEYSELECT uses the certainty size *certain* for all strata. If you do not want to use the same certainty size for all strata, use the CERTSIZE=*SAS-data-set* option to specify a certainty size for each stratum.

CERTSIZE=*SAS-data-set*

names a SAS data set that contains the certainty size measures for the strata. PROC SURVEYSELECT selects with certainty any unit with size measure greater than or equal to the certainty size measure for its stratum. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum certainty size measures in the CERTSIZE= input data set variable _CERTSIZE_. Each certainty size measure must be a positive number. This option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

The CERTSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the CERTSIZE= data set as in the DATA= data set. The CERTSIZE= data set must include a variable named _CERTSIZE_ that contains the certainty size measure for each stratum.

CERTSIZE=P=*p*

specifies the certainty proportion. PROC SURVEYSELECT selects with certainty any unit with size measure greater than or equal to the proportion *p* of the total size for all units in the stratum. The procedure repeats this process with the remaining units until no more certainty units are selected. You provide size measures in the DATA= input data set variable named in the SIZE statement. This option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

The certainty proportion must be a positive number. You can specify *p* as a number between 0 and 1. Or you can specify *p* in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you request a stratified sample design with a STRATA statement and specify the CERTSIZE=P= option, PROC SURVEYSELECT uses the same certainty proportion *p* for all strata.

DATA=*SAS-data-set*

names the SAS data set from which PROC SURVEYSELECT selects the sample. If you omit the DATA= option, the procedure uses the most recently created SAS data set. In sampling terminology, the input data set is the *sampling frame*, or list of units from which the sample is selected.

JTPROBS

includes joint probabilities of selection in the OUT= output data set. This option is available for the following probability proportional to size selection methods: METHOD=PPS, METHOD=PPS_SAMPFORD, and METHOD=PPS_WR. By default, PROC SURVEYSELECT outputs joint selection probabilities for METHOD=PPS_BREWER and METHOD=PPS_MURTHY, which select two units per stratum.

For details on computation of joint selection probabilities for a particular sampling method, see the method description in the section "Sample Selection Methods" on page 4446. For more information on the contents of the output data set, see the section "Output Data Set" on page 4456.

MAXSIZE

requests that sampling unit size measures be adjusted according to the stratum maximum size measures in the secondary input data set. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum maximum size measures in the secondary input data set variable _MAXSIZE_. Use the MAXSIZE option when you have already named the secondary input data set in another option, such as SAMPSIZE=*SAS-data-set*, SAMPRATE=*SAS-data-set*, MINSIZE=*SAS-data-set*, or CERTSIZE=*SAS-data-set*. You can name only one secondary input data set in each invocation of the procedure.

If any size measure exceeds the maximum size measure for its stratum, then PROC SURVEYSELECT adjusts this size measure downward to equal the maximum size measure. Each maximum size measure must be a positive number. The MAXSIZE option is available whenever you specify a SIZE statement for probability proportional to size selection and a STRATA statement for stratification.

If you want to specify a single maximum size value in the PROC SURVEYSELECT statement, use the MAXSIZE=*max* option.

MAXSIZE=*max*

specifies the maximum size measure allowed. If any size measure exceeds the value *max*, then PROC SURVEYSELECT adjusts this size measure to equal *max*, which must be a positive number. You provide size measures in the DATA= input data set variable named in the SIZE statement. This option is available whenever you specify a SIZE statement for selection with probability proportional to size.

If you request a stratified sample design with a STRATA statement and specify the MAXSIZE= option, PROC SURVEYSELECT uses the maximum size *max* for all strata. If you do not want to use the same maximum size for all strata, use the MAXSIZE=*SAS-data-set* option to specify a maximum size for each stratum.

MAXSIZE=*SAS-data-set*

names a SAS data set that contains the maximum size measures allowed for the strata. If any size measure exceeds the maximum size measure for its stratum, then PROC SURVEYSELECT adjusts this size measure downward to equal the maximum size measure. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum maximum size measures in the MAXSIZE= input data set variable _MAXSIZE_. Each maximum size measure must be a positive number. This option is available whenever you specify a SIZE statement for probability proportional to size selection and a STRATA statement for stratified selection.

The MAXSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the MAXSIZE= data set as in the DATA= data set. The MAXSIZE= data set must include a variable named _MAXSIZE_ that contains the maximum size measure for each stratum.

METHOD=*name*
M=*name*

specifies the method for sample selection. If you do not specify the METHOD= option, by default, PROC SURVEYSELECT uses simple random sampling (METHOD=SRS) if there is no SIZE statement. If you specify a SIZE statement, the default selection method is probability proportional to size without replacement (METHOD=PPS). Valid values for *name* are as follows:

PPS — requests selection with probability proportional to size and without replacement. See the section "PPS Sampling without Replacement" on page 4449 for details. If you specify METHOD=PPS, you must name the size measure variable in the SIZE statement.

PPS_BREWER | BREWER — requests selection according to Brewer's method. Brewer's method selects two units from each stratum with probability proportional to size and without replacement. See the section "Brewer's PPS Method" on page 4453 for details. If you specify METHOD=PPS_BREWER, you must name the size measure variable in the SIZE statement. You do not need to specify the sample size with the SAMPSIZE= option, since Brewer's method selects two units from each stratum.

PPS_MURTHY | MURTHY — requests selection according to Murthy's method. Murthy's method selects two units from each stratum with probability proportional to size and without replacement. See the section "Murthy's PPS Method" on page 4454 for details. If you specify METHOD=PPS_MURTHY, you must name the size measure variable in the SIZE statement. You do not need to specify the sample size with the SAMPSIZE= option, since Murthy's method selects two units from each stratum.

PPS_SAMPFORD | SAMPFORD requests selection according to Sampford's method. Sampford's method selects units with probability proportional to size and without replacement. See the section "Sampford's PPS Method" on page 4455 for details. If you specify METHOD=PPS_SAMPFORD, you must name the size measure variable in the SIZE statement.

PPS_SEQ | CHROMY requests sequential selection with probability proportional to size and with minimum replacement. This method is also known as Chromy's method. See the section "PPS Sequential Sampling" on page 4452 for details. If you specify METHOD=PPS_SEQ, you must name the size measure variable in the SIZE statement.

PPS_SYS requests systematic selection with probability proportional to size. See the section "PPS Systematic Sampling" on page 4451 for details on this method. If you specify METHOD=PPS_SYS, you must name the size measure variable in the SIZE statement.

PPS_WR requests selection with probability proportional to size and with replacement. See the section "PPS Sampling with Replacement" on page 4451 for details on this method. If you specify METHOD=PPS_WR, you must name the size measure variable in the SIZE statement.

SEQ requests sequential selection according to Chromy's method. If you specify METHOD=SEQ and do not specify a size measure variable with the SIZE statement, PROC SURVEYSELECT uses sequential zoned selection with equal probability and without replacement. See the section "Sequential Random Sampling" on page 4448 for details on this method. If you specify METHOD=SEQ and also name a size measure variable in the SIZE statement, PROC SURVEYSELECT uses METHOD=PPS_SEQ, which is sequential selection with probability proportional to size and with minimum replacement. See the section "PPS Sequential Sampling" on page 4452 for details on this method.

SRS requests simple random sampling, which is selection with equal probability and without replacement. See the section "Simple Random Sampling" on page 4447 for details. This method is the default if you do not specify the METHOD= option and also do not specify a SIZE statement.

SYS requests systematic random sampling. If you specify METHOD=SYS and do not specify a size measure variable with the SIZE statement, PROC SURVEYSELECT uses systematic selection with equal probability. See the section "Systematic Random Sampling" on page 4448 for details on this method. If you specify METHOD=SYS and also name a size measure variable in the SIZE statement, PROC SURVEYSELECT uses METHOD=PPS_SYS, which is systematic selection with probability proportional to size. See the section "PPS Systematic Sampling" on page 4451 for details.

URS — requests unrestricted random sampling, which is selection with equal probability and with replacement. See the section "Unrestricted Random Sampling" on page 4447 for details.

MINSIZE

requests that sampling unit size measures be adjusted according to the stratum minimum size measures in the secondary input data set. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum minimum size measures in the secondary input data set variable _MINSIZE_. Use the MINSIZE option when you have already named the secondary input data set in another option, such as SAMPSIZE=*SAS-data-set*, SAMPRATE=*SAS-data-set*, MAXSIZE=*SAS-data-set*, or CERTSIZE=*SAS-data-set*. You can name only one secondary input data set in each invocation of the procedure.

If any size measure is less than the minimum size measure for its stratum, then PROC SURVEYSELECT adjusts this size measure upward to equal the minimum size measure. Each minimum size measure must be a positive number. The MINSIZE option is available whenever you specify a SIZE statement for probability proportional to size selection and a STRATA statement for stratification.

If you want to specify a single minimum size value in the PROC SURVEYSELECT statement, use the MINSIZE=*min* option.

MINSIZE=*min*

specifies the minimum size measure allowed. If any size measure is less than the value *min*, then PROC SURVEYSELECT adjusts this size measure upward to equal *min*, which must be a positive number. You provide size measures in the DATA= input data set variable named in the SIZE statement. This option is available whenever you specify a SIZE statement for selection with probability proportional to size.

If you request a stratified sample design with a STRATA statement and specify the MINSIZE= option, PROC SURVEYSELECT uses the minimum size *min* for all strata. If you do not want to use the same minimum size for all strata, use the MINSIZE=*SAS-data-set* option to specify a minimum size for each stratum.

MINSIZE=*SAS-data-set*

names a SAS data set that contains the minimum size measures allowed for the strata. If any size measure is less than the minimum size measure for its stratum, then PROC SURVEYSELECT adjusts this size measure upward to equal the minimum size measure. You provide sampling unit size measures in the DATA= input data set variable named in the SIZE statement. And you provide the stratum minimum size measures in the MINSIZE= input data set variable _MINSIZE_. Each minimum size measure must be a positive number. This option is available whenever you specify a SIZE statement for probability proportional to size selection and a STRATA statement for stratified selection.

The MINSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the MINSIZE= data set as in the DATA= data set. The MINSIZE= data set must include a variable named _MINSIZE_ that contains the minimum size measure for each stratum.

NMAX=*n*

specifies the maximum stratum sample size *n* for the SAMPRATE= option. When you specify the SAMPRATE= option, PROC SURVEYSELECT calculates the desired stratum sample size from the specified sampling rate and the total number of units in the stratum. If this sample size is greater than the value NMAX=*n*, then PROC SURVEYSELECT selects the maximum of *n* units.

The maximum sample size *n* must be a positive integer. The NMAX= option is available only with the SAMPRATE= option, which may be used with equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

NMIN=*n*

specifies the minimum stratum sample size *n* for the SAMPRATE= option. When you specify the SAMPRATE= option, PROC SURVEYSELECT calculates the desired stratum sample size from the specified sampling rate and the total number of units in the stratum. If this sample size is less than the value NMIN=*n*, then PROC SURVEYSELECT selects the minimum of *n* units.

The minimum sample size *n* must be a positive integer. The NMIN= option is available only with the SAMPRATE= option, which may be used with equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

NOPRINT

suppresses the display of all output. You can use the NOPRINT option when you want only to create an output data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, “Using the Output Delivery System.”

OUT=*SAS-data-set*

names the output data set that contains the sample. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

The output data set contains the units selected for the sample, as well as design information and selection statistics, depending on the selection method and output options you specify. See the descriptions for the options JTPROBS, OUTHITS, OUTSEED, OUTSIZE, and STATS. For information on the contents of the output data set, see the section “Output Data Set” on page 4456.

By default, the output data set contains only those units selected for the sample. To include all observations from the input data set in the output data set, use the OUTALL option.

OUTALL

includes all observations from the input data set in the output data set. By default, the output data set includes only those observations selected for the sample. When you specify the OUTALL option, the output data set includes all observations from DATA= and also contains a variable to indicate each observation’s selection status. The variable Selected equals 1 for an observation selected for the sample, and equals

0 for an observation not selected. For information on the contents of the output data set, see the section "Output Data Set" on page 4456.

The OUTALL option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

OUTHITS

includes a separate observation in the output data set for each selection when the same unit is selected more than once. By default, the output data set contains only one observation for each selected unit, even if it is selected more than once, and the variable NumberHits contains the number of hits or selections for that unit. The OUTHITS option is available for selection methods that select with replacement or with minimum replacement (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).

OUTSEED

includes the initial seed for each stratum in the output data set. The variable InitialSeed contains the stratum initial seed. See the section "Sample Selection Methods" on page 4446 for information on initial seeds and random number generation in PROC SURVEYSELECT.

To reproduce the same sample for any stratum in a subsequent execution of PROC SURVEYSELECT, you can specify the same stratum initial seed with the SEED=*SAS-data-set* option, along with the same sample selection parameters.

OUTSIZE

includes additional design and sampling frame parameters in the output data set. If you specify the OUTSIZE option, PROC SURVEYSELECT includes the sample size or sampling rate in the output data set. When you request the OUTSIZE option and also specify the SIZE statement, the procedure outputs the size measure total for the sampling frame. If you do not specify the SIZE statement, the procedure outputs the total number of sampling units in the frame. Also, PROC SURVEYSELECT includes the minimum size measure if you specify the MINSIZE= option, the maximum size measure if you specify the MAXSIZE= option, and the certainty size measure if you specify the CERTSIZE= option.

If you have a stratified design, the output data set includes the stratum-level values of these parameters. Otherwise, the output data set includes the overall population-level values.

For information on the contents of the output data set, see the section "Output Data Set" on page 4456.

OUTSORT=*SAS-data-set*

names an output data set that contains the sorted input data set. This option is available when you specify a CONTROL statement for systematic or sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ). PROC SURVEYSELECT sorts the input data set by the CONTROL variables within strata before selecting the sample.

If you specify CONTROL variables but do not name an output data set with the OUTSORT= option, then the sorted data set replaces the input data set.

REP=*nrep*

specifies the number of sample replicates. If you specify the REP= option, PROC SURVEYSELECT selects *nrep* independent samples, each with the same specified sample size or sampling rate and the same sample design.

You can use replicated sampling to provide a simple method of variance estimation for any form of statistic, as well as to evaluate variable nonsampling errors such as interviewer differences. Refer to Lohr (1999), Kish (1965, 1987), and Kalton (1983) for information on replicated sampling.

SAMPRATE=*r*
RATE=*r*

specifies the sampling rate, which is the proportion of units selected for the sample. The sampling rate *r* must be a positive number. You can specify *r* as a number between 0 and 1. Or you can specify *r* in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the sampling rate *r* as the interval. See the section "Systematic Random Sampling" on page 4448 for details. For other selection methods, PROC SURVEYSELECT converts the sampling rate *r* to the sample size before selection, multiplying the rate by the number of units in the stratum or frame and rounding up to the nearest integer.

If you request a stratified sample design with a STRATA statement and specify the SAMPRATE=*r* option, PROC SURVEYSELECT uses the sampling rate *r* for each stratum. If you do not want to use the same sampling rate for each stratum, use the SAMPRATE=(*values*) option or the SAMPRATE=*SAS-data-set* option to specify a sampling rate for each stratum.

SAMPRATE=(*values***)**
RATE=(*values***)**

specifies sampling rates for the strata. You can separate *values* with blanks or commas. The number of SAMPRATE= values must equal the number of strata in the input data set.

List the stratum sampling rate values in the order in which the strata appear in the input data set. If you use the SAMPRATE=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED options in the STRATA statement.

Each stratum sampling rate value must be a positive number. You can specify each value as a number between 0 and 1. Or you can specify a value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the stratum sampling rate as the interval for the stratum. See the section "Systematic Random Sampling" on page 4448 for details on systematic sampling. For other selection methods, PROC SURVEYSELECT converts the stratum sampling rate to a stratum sample size before selection, multiplying the rate by the number of units in the stratum and rounding up to the nearest integer.

SAMPRATE=*SAS-data-set*
RATE=*SAS-data-set*

names a SAS data set that contains sampling rates for the strata. This input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SAMPSIZE= data set as in the DATA= data set. The SAMPRATE= data set should have a variable _RATE_ that contains the sampling rate for each stratum.

Each sampling rate value must be a positive number. You can specify each value as a number between 0 and 1. Or you can specify a value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the stratum sampling rate as the interval for the stratum. See the section "Systematic Random Sampling" on page 4448 for details. For other selection methods, PROC SURVEYSELECT converts the stratum sampling rate to the stratum sample size before selection, multiplying the rate by the number of units in the stratum and rounding up to the nearest integer.

SAMPSIZE=*n*
N=*n*

specifies the sample size, which is the number of units selected for the sample. The sample size *n* must be a positive integer. For methods that select without replacement, the sample size *n* must not exceed the number of units in the input data set.

If you request a stratified sample design with a STRATA statement and specify the SAMPSIZE=*n* option, PROC SURVEYSELECT selects *n* units from each stratum. For methods that select without replacement, the sample size *n* must not exceed the number of units in any stratum. If you do not want to select the same number of units from each stratum, use the SAMPSIZE=(*values*) option or the SAMPSIZE=*SAS-data-set* option to specify different sample sizes for the strata.

For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units in the stratum. However, you can change this default by specifying the SELECTALL option. With the SELECTALL option, PROC SURVEYSELECT selects all stratum units whenever the stratum sample size exceeds the number of units in the stratum.

SAMPSIZE=(*values*)
N=(*values*)

specifies sample sizes for the strata. You can separate *values* with blanks or commas. The number of SAMPSIZE= values must equal the number of strata in the input data set.

List the stratum sample size values in the order in which the strata appear in the input data set. If you use the SAMPSIZE=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED options in the STRATA statement.

Each stratum sample size value must be a positive integer. For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units in the stratum. However, you can change this default by specifying the SELECTALL option. With the SELECTALL option, PROC SURVEYSELECT selects all stratum units whenever the stratum sample size exceeds the number of units in the stratum.

SAMPSIZE=*SAS-data-set*
N=*SAS-data-set*

names a SAS data set that contains the sample sizes for the strata. This input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SAMPSIZE= data set as in the DATA= data set. The SAMPSIZE= data set should have a variable _NSIZE_ that contains the sample size for each stratum.

Each stratum sample size value must be a positive integer. For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units in the stratum. However, you can change this default by specifying the SELECTALL option. With the SELECTALL option, PROC SURVEYSELECT selects all stratum units whenever the stratum sample size exceeds the number of units in the stratum.

SEED=*number*

specifies the initial seed for random number generation. The value of the SEED= option must be an integer. If you do not specify the SEED= option, or if the SEED= value is negative or zero, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. See the section "Sample Selection Methods" on page 4446 for more information.

Whether or not you specify the SEED= option, PROC SURVEYSELECT displays the value of the initial seed in the "Sample Selection Summary" table. If you need to reproduce the same sample in a subsequent execution of PROC SURVEYSELECT, you can specify this same seed value with the SEED= option, along with the same sample selection parameters, and PROC SURVEYSELECT will reproduce the sample.

If you request a stratified sample design with a STRATA statement, you can use the SEED=*SAS-data-set* option to specify an initial seed for each stratum. Otherwise, PROC SURVEYSELECT generates random numbers continuously across strata from the random number stream initialized by the SEED= value, as described in the section "Sample Selection Methods" on page 4446.

To include the stratum initial seeds in the output data set, use the OUTSEED option.

SEED=*SAS-data-set*

names a SAS data set that contains initial seeds for the strata. This input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SAMPSIZE= data set as in the DATA= data set. The SEED= data set should have a variable _SEED_ that contains the initial seed for each stratum.

Each stratum initial seed value should be an integer. If the initial seed value for the first stratum is not a positive integer, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. If the initial seed value for a subsequent stratum is not a positive integer, PROC SURVEYSELECT continues to use the random number stream already initialized by the seed for the previous stratum. See the section "Sample Selection Methods" on page 4446 for more information.

To include the stratum initial seeds in the output data set, specify the OUTSEED option.

If you specified initial seeds by strata with the SEED=*SAS-data-set* option, you can reproduce the same sample in a subsequent execution of PROC SURVEYSELECT by specifying these same stratum initial seeds, along with the same sample selection parameters. If you need to reproduce the same sample for only a subset of the strata, you can use the same initial seeds for those strata in the subset.

SELECTALL

requests that PROC SURVEYSELECT select all stratum units whenever the stratum sample size exceeds the total number of units in the stratum, for without-replacement selection methods. By default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units in the stratum, for methods that select without replacement.

The SELECTALL option is available for without-replacement selection methods, which include METHOD=SRS, METHOD=SYS, METHOD=SEQ, METHOD=PPS, and METHOD=PPS_SAMPFORD. The SELECTALL option is not available for with-replacement selection methods, with-minimum-replacement methods, or for those PPS methods that select two units per stratum.

SORT=NEST | SERP

specifies the type of sorting by CONTROL variables. The option SORT=NEST requests nested sorting, and SORT=SERP requests hierarchic serpentine sorting. The default is SORT=SERP. See the section "Sorting by CONTROL Variables" on page 4445 for descriptions of serpentine and nested sorting. Where there is only one CONTROL variable, the two types of sorting are equivalent.

This option is available when you specify a CONTROL statement for systematic or sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ). When you specify a CONTROL statement, PROC SURVEYSELECT sorts the input data set by the CONTROL variables within strata before selecting the sample.

With sorting by CONTROL variables, you can also use the OUTSORT= option to name an output data set that contains the sorted input data set. Otherwise, if you do not specify the OUTSORT= option, then the sorted data set replaces the input data set.

STATS

includes selection probabilities and sampling weights in the OUT= output data set for equal probability selection methods when you do not specify a STRATA statement. This option is available for the following equal probability selection methods: METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ. For PPS selection methods and stratified designs, the output data set contains selection probabilities and sampling weights by default. For more information on the contents of the output data set, see the section "Output Data Set" on page 4456.

CONTROL Statement

CONTROL *variables* **;**

The CONTROL statement names variables for sorting the input data set. The CONTROL variables can be character or numeric.

PROC SURVEYSELECT sorts the input data set by the CONTROL variables before selecting the sample. If you also specify a STRATA statement, PROC SURVEYSELECT sorts by CONTROL variables within strata. Control sorting is available for systematic and sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ).

By default, PROC SURVEYSELECT uses hierarchic serpentine sorting by the CONTROL variables. If you specify the SORT=NEST option, the procedure uses nested sorting. See the description for the SORT= option. For more information on serpentine and nested sorting, see the section "Sorting by CONTROL Variables" on page 4445.

You can use the OUTSORT= option to name an output data set that contains the sorted input data set. If you do not specify the OUTSORT= option when you use the CONTROL statement, then the sorted data set replaces the input data set.

ID Statement

ID *variables* **;**

The ID statement names variables from the DATA= input data set to be included in the OUT= data set of selected units. If there is no ID statement, PROC SURVEYSELECT includes all variables from the DATA= data set in the OUT= data set. The ID variables can be character or numeric.

SIZE Statement

SIZE *variable* **;**

The SIZE statement names one and only one size measure variable, which contains the size measures to be used when sampling with probability proportional to size. The SIZE variable must be numeric. When the value of an observation's SIZE variable is missing or nonpositive, that observation has no chance of being selected for the sample.

The SIZE statement is required for all PPS selection methods, which include METHOD=PPS, METHOD=PPS_BREWER, METHOD=PPS_MURTHY, METHOD=PPS_SAMPFORD, METHOD=PPS_SEQ, METHOD=PPS_SYS, and METHOD=PPS_WR. For details on how size measures are used, see the descriptions of PPS methods in the section "Sample Selection Methods" on page 4446.

Note that a unit's size measure, specified in the SIZE statement and used for PPS selection, is not the same as the sample size. The sample size is the number of units selected for the sample, and you can specify this with the SAMPSIZE= option.

STRATA Statement

STRATA *variables* **;**

You can specify a STRATA statement with PROC SURVEYSELECT to partition the input data set into nonoverlapping groups defined by the STRATA variables. PROC SURVEYSELECT then selects independent samples from these strata, according to the selection method and design parameters specified in the PROC SURVEYSELECT statement. For information on the use of stratification in sample design, refer to Lohr (1999), Kalton (1983), Kish (1965, 1987), and Cochran (1977).

The *variables* are one or more variables in the input data set. The STRATA variables function much like BY variables, and PROC SURVEYSELECT expects the input data set to be sorted in order of the STRATA variables.

If you specify a CONTROL statement, or if you specify METHOD=PPS, the input data set must be sorted in ascending order of the STRATA variables. This means you cannot use the STRATA option NOTSORTED or DESCENDING when you specify a CONTROL statement or METHOD=PPS.

If your input data set is not sorted by the STRATA variables in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with the STRATA variables in a BY statement.
- Specify the option NOTSORTED or DESCENDING in the STRATA statement for the SURVEYSELECT procedure (when you do not specify a CONTROL statement or METHOD=PPS). The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the STRATA variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the STRATA variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

Details

Missing Values

If an observation has a missing or nonpositive value for the SIZE variable, PROC SURVEYSELECT excludes that observation from the sample selection. The procedure writes a note to the log giving the number of observations omitted due to missing or nonpositive size measures.

PROC SURVEYSELECT treats missing STRATA variable values like any other STRATA variable value. The missing values form a separate stratum.

If a value of _NSIZE_ is missing in the SAMPSIZE= input data set, then PROC SURVEYSELECT writes an error message to the log and does not select a sample from that stratum. The procedure treats missing values of _NRATE_, _MINSIZE_, _MAXSIZE_, and _CERTSIZE_ similarly.

Sorting by CONTROL Variables

If you specify a CONTROL statement, PROC SURVEYSELECT sorts the input data set by the CONTROL variables before selecting the sample. If you also specify a STRATA statement, the procedure sorts by CONTROL variables within strata. Sorting by CONTROL variables is available for systematic and sequential selection methods, which include METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ. Sorting provides additional control over the distribution of the sample, giving some benefits of proportionate stratification.

By default, the sorted data set replaces the input data set. Or you can use the OUTSORT= option to name an output data set that contains the sorted input data set.

PROC SURVEYSELECT provides two types of sorting: nested sorting and hierarchic serpentine sorting. If you specify the SORT=NEST option, then the procedure sorts by the CONTROL variables according to nested sorting. If you do not specify the SORT=NEST option, the procedure uses serpentine sorting by default. These two types of sorting are equivalent when there is only one CONTROL variable.

If you request nested sorting, PROC SURVEYSELECT sorts observations in the same order as PROC SORT does for an ascending sort by the CONTROL variables. Refer to the chapter on the SORT procedure in the *SAS Procedures Guide*. PROC SURVEYSELECT sorts within strata if you also specify a STRATA statement. The procedure first arranges the input observations in ascending order of the first CONTROL variable. Then within each level of the first control variable, the procedure arranges the observations in ascending order of the second CONTROL variable. This continues for all CONTROL variables specified.

In hierarchic serpentine sorting, PROC SURVEYSELECT sorts by the first CONTROL variable in ascending order. Then within the first level of the first CONTROL variable, the procedure sorts by the second CONTROL variable in ascending order. Within the second level of the first CONTROL variable, the procedure sorts by the second CONTROL variable in descending order. Sorting by the second CONTROL variable continues to alternate between ascending and descending sorting throughout all levels of the first CONTROL variable. If there is a third CONTROL variable, the procedure sorts by that variable within levels formed from the first two CONTROL variables, again alternating between ascending and descending sorting. This continues for all CONTROL variables specified. This sorting algorithm minimizes the change from one observation to the next with respect to the CONTROL variable values, thus making nearby observations more similar. For more information on serpentine sorting, refer to Chromy (1979) and Williams and Chromy (1980).

Sample Selection Methods

PROC SURVEYSELECT provides a variety of methods for selecting probability-based random samples. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population. Refer to Lohr (1999), Kish (1965, 1987), Kalton (1983), and Cochran (1977) for more information on probability sampling.

In equal probability sampling, each unit in the sampling frame, or in a stratum, has the same probability of being selected for the sample. PROC SURVEYSELECT provides the following methods that select units with equal probability: simple random sampling, unrestricted random sampling, systematic random sampling, and sequential random sampling. In simple random sampling, units are selected *without replacement*, which means that a unit cannot be selected more than once. Both systematic and sequential equal probability sampling are also without replacement. In unrestricted random sampling, units are selected *with replacement*, which means that a unit can be selected more than once. In with-replacement sampling, the *number of hits* refers to the number of times a unit is selected.

In probability proportional to size (PPS) sampling, a unit's selection probability is proportional to its size measure. PROC SURVEYSELECT provides the following methods that select units with probability proportional to size (PPS): PPS sampling without replacement, PPS sampling with replacement, PPS systematic sampling, PPS sequential sampling, Brewer's method, Murthy's method, and Sampford's method. PPS sampling is often used in cluster sampling, where you select clusters (or groups of sampling units) of varying size in the first stage of selection. For example, clusters may be schools, hospitals, or geographical areas, and the final sampling units may be students, patients, or citizens. Cluster sampling can provide efficiencies in frame construction and other survey operations. Refer to Lohr (1999), Kalton (1983), Kish (1965), and the other references cited in the following sections for more information.

All the probability sampling methods provided by PROC SURVEYSELECT use random numbers in their selection algorithms, as described in the following sections and in the references cited. PROC SURVEYSELECT uses a uniform random number function to generate streams of pseudo-random numbers from an initial starting point, or *seed*. You can use the SEED= option to specify the initial seed. If you do not specify the SEED= option, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. PROC SURVEYSELECT generates uniform random numbers according to the method of Fishman and Moore (1982), using a prime modulus multiplicative generator with modulus 2^{31} and multiplier 397204094. PROC SURVEYSELECT uses the same uniform random number generator as the RANUNI function. For more information on the RANUNI function, see the *SAS Language Reference: Dictionary*

The following sections give detailed descriptions of the sample selection methods available in PROC SURVEYSELECT. In these sections, n_h denotes the sample size (the number of units in the sample) for stratum h, and N_h denotes the population size (number of units in the population) for stratum h, for $h = 1, 2, \ldots, H$. When the sample design is not stratified, n denotes the sample size, and N denotes the population size. For PPS sampling, M_{hi} represents the size measure for unit i in stratum h, $M_{h\cdot}$ is the total of all size measures for the population of stratum h, and $Z_{hi} = M_{hi} / M_h$ is the relative size of unit i in stratum h.

Simple Random Sampling

The method of simple random sampling (METHOD=SRS) selects units with equal probability and without replacement. Each possible sample of n different units out of N has the same probability of being selected. The selection probability for each individual unit equals n/N. When you request stratified sampling with a STRATA statement, PROC SURVEYSELECT selects samples independently within strata. The selection probability for a unit in stratum h equals n_h / N_h for stratified simple random sampling.

By default, PROC SURVEYSELECT uses Floyd's ordered hash table algorithm for simple random sampling. This algorithm is fast, efficient, and appropriate for large data sets. Refer to Bentley and Floyd (1987) and Bentley and Knuth (1986).

If there is not enough memory available for Floyd's algorithm, PROC SURVEYSELECT switches to the sequential algorithm of Fann, Muller, and Rezucha (1962), which requires less memory but may require more time to select the sample. When SURVEYSELECT uses the alternative sequential algorithm, it writes a note to the log. To request the sequential algorithm, even if enough memory is available for Floyd's algorithm, you can specify METHOD=SRS2 in the PROC SURVEYSELECT statement.

Unrestricted Random Sampling

The method of unrestricted random sampling (METHOD=URS) selects units with equal probability and with replacement. Because units are selected with replacement, a unit can be selected for the sample more than once. The expected number of selections or hits for each unit equals n/N when sampling without stratification. For stratified sampling, the expected number of hits for a unit in stratum h equals n_h / N_h. Note that the expected number of hits exceeds one when the sample size n is greater than the population size N.

For unrestricted random sampling, by default, the output data set contains one observation for each distinct unit selected for the sample, together with a variable NumberHits that gives the number of times the observation was selected. But if you specify the OUTHITS option, then the output data set contains a separate observation for each selection, so that a unit selected three times, e.g., is represented by three observations in the output data set. For information on the contents of the output data set, see the section "Output Data Set" on page 4456.

Systematic Random Sampling

The method of systematic random sampling (METHOD=SYS) selects units at a fixed interval throughout the sampling frame or stratum after a random start. If you specify the sample size (or the stratum sample sizes) with the SAMPSIZE= option, PROC SURVEYSELECT uses a fractional interval to provide exactly the specified sample size. The interval equals N/n, or N_h/n_h for stratified sampling. The selection probability for each unit equals n/N, or n_h/N_h for stratified sampling. If you specify the sampling rate (or the stratum sampling rates) with the SAMPRATE= option, PROC SURVEYSELECT uses the inverse of the rate as the interval for systematic selection. The selection probability for each unit equals the specified rate.

Systematic random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum at equal intervals, thus providing implicit stratification. You can use the CONTROL statement to order the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies systematic selection to the observations in the order in which they appear in the input data set.

Sequential Random Sampling

If you specify the option METHOD=SEQ and do not include a SIZE statement, PROC SURVEYSELECT uses the equal probability version of Chromy's method for sequential random sampling. This method selects units sequentially with equal probability and without replacement. Refer to Chromy (1979) and Williams and Chromy (1980). See the section "PPS Sequential Sampling" on page 4452 for a description of Chromy's PPS selection method.

Sequential random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum, thus providing implicit stratification according to the order of units in the frame or stratum. You can use the CONTROL statement to sort the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. By default, the procedure uses hierarchic serpentine ordering for sorting. If you specify the SORT=NEST option, the procedure uses nested sorting. See the section "Sorting by CONTROL Variables" on page 4445 for descriptions of serpentine and nested sorting. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies sequential selection to the observations in the order in which they appear in the input data set.

Following Chromy's method of sequential selection, PROC SURVEYSELECT randomly chooses a starting unit from the entire stratum (or frame, if the design is not stratified). Using this unit as the first one, the procedure treats the stratum units as a closed loop. This is done so that all pairwise (joint) selection probabilities are positive and an unbiased variance estimator can be obtained. The procedure numbers units sequentially from the random start to the end of the stratum and then continues from the beginning of the stratum until all units are numbered.

Beginning with the randomly chosen starting unit, PROC SURVEYSELECT accumulates the expected number of selections or hits, where the expected number of selections ES_{hi} equals n_h/N_h for all units i in stratum h. The procedure computes

$$I_{hi} \quad = \quad \mathrm{Int}\,(\sum_{j=1}^{i} ES_{hj}) \quad = \quad \mathrm{Int}\,(i\, n_h/N_h)$$

$$F_{hi} \quad = \quad \mathrm{Frac}\,(\sum_{j=1}^{i} ES_{hj}) \quad = \quad \mathrm{Frac}\,(i\, n_h/N_h)$$

where Int denotes the integer part of the number, and Frac denotes the fractional part.

Considering each unit sequentially, Chromy's method determines whether unit i is selected by comparing the total number of selections for the first $i-1$ units,

$$T_{h(i-1)} \quad = \quad \sum_{j=1}^{i-1} S_{hj}$$

with the value of $I_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)}$, Chromy's method determines whether or not unit i is selected as follows. If $F_{hi} = 0$ or $F_{h(i-1)} > F_{hi}$, then unit i is selected with certainty. Otherwise, unit i is selected with probability

$$(F_{hi} - F_{h(i-1)}) \;/\; (1 - F_{h(i-1)})$$

If $T_{h(i-1)} = I_{h(i-1)} + 1$, Chromy's method determines whether or not unit i is selected as follows. If $F_{hi} = 0$ or $F_{hi} > F_{h(i-1)}$, then the unit is not selected. Otherwise, unit i is selected with probability

$$F_{hi} \;/\; F_{h(i-1)}$$

PPS Sampling without Replacement

If you specify the option METHOD=PPS, PROC SURVEYSELECT selects units with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals $n_h Z_{hi}$. The procedure uses the Hanurav-Vijayan algorithm for PPS selection without replacement. Hanurav (1967) introduced this algorithm for the selection of two units per stratum, and Vijayan (1968) generalized it for the selection of more than two units. The algorithm enables computation of joint selection probabilities and provides joint selection probability values that usually ensure nonnegativity and stability of the Sen-Yates-Grundy variance estimator. Refer to Fox (1989), Golmant (1990), and Watts (1991) for details.

Notation in the remainder of this section drops the stratum subscript h for simplicity, but selection is still done independently within strata if you specify a stratified design. For a stratified design, n now denotes the sample size for the current stratum, N denotes the stratum population size, and M_i denotes the size measure for unit i in the stratum. If the design is not stratified, this notation applies to the entire sampling frame.

According to the Hanurav-Vijayan algorithm, PROC SURVEYSELECT first orders units within the stratum in ascending order by size measure, so that $M_1 \leq M_2 \leq \ldots \leq M_N$. Then the procedure selects the PPS sample of n observations as follows:

1. The procedure randomly chooses one of the integers $1, 2, \ldots, n$ with probability $\theta_1, \theta_2, \ldots, \theta_n$, where

   $$\theta_i = n\,(Z_{N-n+i+1} - Z_{N-n+i})\,(T + i\,Z_{N-n+1})\,/\,T$$

   $Z_j = M_j / M, \quad T = \sum_{j=1}^{N-n} Z_j,$ and, by definition, $Z_{N+1} = 1/n$ to ensure that $\sum_{i=1}^{n} \theta_i = 1$.
2. If i is the integer selected in step 1, the procedure includes the last $(n-i)$ units of the stratum in the sample, where the units are ordered by size measure as described previously. The procedure then selects the remaining i units according to steps 3 through 6 below.
3. The procedure defines new normed size measures for the remaining $(N-n+i)$ stratum units that were not selected in steps 1 and 2:

   $$\begin{aligned} Z_j^* &= Z_j\,/\,(T + i\,Z_{N-n+1}) && \text{for } j = 1, \ldots, N-n+1 \\ Z_j^* &= Z_{N-n+1}\,/\,(T + i\,Z_{N-n+1}) && \text{for } j = N-n+2, \ldots, N-n+i \end{aligned}$$

4. The procedure selects the next unit from the first $(N-n+1)$ stratum units with probability proportional to $a_j(1)$, where

   $$\begin{aligned} a_1(1) &= i\,Z_1^* \\ a_j(1) &= i\,Z_j^* \textstyle\prod_{k=1}^{j-1} [1 - (i-1)\,P_k] \quad \text{for } j = 2, \ldots, N-n+1 \end{aligned}$$

   and $P_k = M_k / (M_{k+1} + M_{k+2} + \cdots + M_{N-n+i})$.
5. If stratum unit j_1 is the unit selected in step 4, then the procedure selects the next unit from units $j_1 + 1$ through $N-n+2$ with probability proportional to $a_j(2, j_1)$, where

   $$\begin{aligned} a_{j_1+1}(2, j_1) &= (i-1)\,Z_{j_1+1}^* \\ a_j(2, j_1) &= (i-1)\,Z_j^* \prod_{k=j_1+1}^{j-1} [1 - (i-2)\,P_k] \quad \text{for } j = j_1+2, \ldots, N-n+2 \end{aligned}$$

6. The procedure repeats step 5 until all n sample units are selected.

If you request the JTPROBS option, PROC SURVEYSELECT computes the joint selection probabilities for all pairs of selected units in each stratum. The joint selection probability for units i and j in the stratum equals

$$P_{(ij)} \quad = \quad \sum_{r=1}^{n} \theta_r K_{ij}^{(r)}$$

where

$$\begin{aligned} K_{ij}^{(r)} &= 1 && N-n+r < i \leq N-1 \\ &= r\, Z_{N-n+1} \,/\, (T + r\, Z_{N-n+1}) && N-n < i \leq N-n+r, \quad j > N-n+r \\ &= r\, Z_i \,/\, (T + r\, Z_{N-n+1}) && 1 \leq i \leq N-n, \quad j > N-n+r \\ &= \pi_{ij}^{(r)} && j \leq N-n+r \end{aligned}$$

and

$$\pi_{ij}^{(r)} = \frac{r(r-1)}{2}\; P_i\; Z_j \prod_{k=1}^{i-1} (1 - P_k)$$

where $P_k = M_k / (M_{k+1} + M_{k+2} + \cdots + M_{N-n+r})$.

PPS Sampling with Replacement

If you specify the option METHOD=PPS_WR, PROC SURVEYSELECT selects units with probability proportional to size and with replacement. The procedure makes n_h independent random selections from the stratum of N_h units, selecting with probability $Z_{hi} = M_{hi} / M_{h\cdot}$. Because units are selected with replacement, a unit can be selected for the sample more than once. The expected number of selections or hits for unit i in stratum h equals $n_h Z_{hi}$. If you request the JTPROBS option, PROC SURVEYSELECT computes the joint expected number of hits for all pairs of selected units in each stratum. The joint expected number of hits for units i and j in stratum h equals

$$\begin{aligned} P_{h(ij)} &= n_h (n_h - 1)\; Z_{hi}\; Z_{hj} && \text{for } j \neq i \\ &= n_h (n_h - 1)\; Z_{hi}\; Z_{hi} \,/\, 2 && \text{for } j = i \end{aligned}$$

PPS Systematic Sampling

If you specify the option METHOD=PPS_SYS, PROC SURVEYSELECT selects units by systematic random sampling with probability proportional to size. Systematic sampling selects units at a fixed interval throughout the stratum or sampling frame after a random start. PROC SURVEYSELECT uses a fractional interval to provide exactly the specified sample size. The interval equals $M_{h\cdot} / n_h$ for stratified sampling and M / n for sampling without stratification. Depending on the sample size and the values of the size measures, it may be possible for a unit to be selected more than once. The expected number of selections or hits for unit i in stratum h equals $n_h M_{hi} / M_{h\cdot} = n_h Z_{hi}$. Refer to Cochran (1977, pp. 265–266) and Madow (1949).

Systematic random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum at equal intervals, thus providing implicit stratification. You can use the CONTROL statement to order the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies systematic selection to the observations in the order in which they appear in the input data set.

PPS Sequential Sampling

If you specify the option METHOD=PPS_SEQ, PROC SURVEYSELECT uses Chromy's method of sequential random sampling. Refer to Chromy (1979) and Williams and Chromy (1980). Chromy's method selects units sequentially with probability proportional to size and with minimum replacement. Selection *with minimum replacement* means that the actual number of hits for a unit can equal the integer part of the expected number of hits for that unit, or the next largest integer. This can be compared to selection *without replacement*, where each unit can be selected only once, so the number of hits can equal 0 or one. The other alternative is selection *with replacement*, where there is no restriction on the number of hits for each unit, so the number of hits can equal $0, 1, \cdots, n_h$, where n_h is the stratum sample size.

Sequential random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum, thus providing implicit stratification according to the order of units in the frame or stratum. You can use the CONTROL statement to sort the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. By default, the procedure uses hierarchic serpentine ordering to sort the sampling frame by the CONTROL variables within strata. If you specify the SORT=NEST option, the procedure uses nested sorting. See the section "Sorting by CONTROL Variables" on page 4445 for descriptions of serpentine and nested sorting. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies sequential selection to the observations in the order in which they appear in the input data set.

According to Chromy's method of sequential selection, PROC SURVEYSELECT first chooses a starting unit randomly from the entire stratum, with probability proportional to size. The procedure uses this unit as the first one and treats the stratum observations as a closed loop. This is done so that all pairwise (joint) expected number of hits are positive and an unbiased variance estimator can be obtained. The procedure numbers observations sequentially from the random start to the end of the stratum and then continues from the beginning of the stratum until all units are numbered.

Beginning with the randomly chosen starting unit, Chromy's method partitions the ordered stratum sampling frame into n_h zones of equal size. There is one selection from each zone and a total of n_h selections or hits, although fewer than n_h distinct units may be selected. Beginning with the random start, the procedure accumulates the expected number of hits and computes

$$\begin{aligned} ES_{hi} &= n_h Z_{hi} \\ I_{hi} &= \text{Int}\,(\sum_{j=1}^{i} ES_{hj}) \\ F_{hi} &= \text{Frac}\,(\sum_{j=1}^{i} ES_{hj}) \end{aligned}$$

where ES_{hi} represents the expected number of hits for unit i in stratum h; Int denotes the integer part of the number; and Frac denotes the fractional part.

Considering each unit sequentially, Chromy's method determines the actual number of hits for unit i by comparing the total number of hits for the first $i-1$ units,

$$T_{h(i-1)} = \sum_{j=1}^{i-1} S_{hj}$$

with the value of $I_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)}$, Chromy's method determines the total number of hits for the first i units as follows. If $F_{hi} = 0$ or $F_{h(i-1)} > F_{hi}$, then $T_{hi} = I_{hi}$. Otherwise, $T_{hi} = I_{hi} + 1$ with probability

$$(F_{hi} - F_{h(i-1)}) \;/\; (1 - F_{h(i-1)})$$

And the number of hits for unit i equals $T_{hi} - T_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)} + 1$, Chromy's method determines the total number of hits for the first i units as follows. If $F_{hi} = 0$, then $T_{hi} = I_{hi}$. If $F_{hi} > F_{h(i-1)}$, then $T_{hi} = I_{hi} + 1$. Otherwise, $T_{hi} = I_{hi} + 1$ with probability

$$F_{hi} \;/\; F_{h(i-1)}$$

Brewer's PPS Method

Brewer's method (METHOD=PPS_BREWER) selects two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals $2M_{hi}/M_{h\cdot} = 2Z_{hi}$.

Brewer's algorithm first selects a unit with probability

$$\frac{Z_{hi}\,(1-Z_{hi})}{D_h\,(1-2Z_{hi})}$$

where

$$D_h \quad = \quad \sum_{i=1}^{N_h} \frac{Z_{hi}\,(1-Z_{hi})}{1-2Z_{hi}}$$

Then a second unit is selected from the remaining units with probability

$$\frac{Z_{hj}}{1-Z_{hi}}$$

where unit i is the first unit selected. The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} \quad = \quad \frac{2\,Z_{hi}\,Z_{hj}}{D_h}\left(\frac{1-Z_{hi}-Z_{hj}}{(1-2Z_{hi})\,(1-2Z_{hj})}\right)$$

Brewer's method requires that the relative size Z_{hi} be less than 0.5 for all units. Refer to Cochran (1977, pp. 261–263) and Brewer (1963). Brewer's method yields the same selection probabilities and joint selection probabilities as Durbin's method. Refer to Cochran (1977) and Durbin (1967).

Murthy's PPS Method

Murthy's method (METHOD=PPS_MURTHY) selects two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals

$$P_{hi} \quad = \quad Z_{hi}\,[1+K-(Z_{hi}/(1-Z_{hi})]$$

where $Z_{hi} = M_{hi}/M_{h\cdot}$ and

$$K \quad = \quad \sum_{j=1}^{N}[Z_{hj}/(1-Z_{hj})]$$

Murthy's algorithm first selects a unit with probability Z_{hi} . Then a second unit is selected from the remaining units with probability $Z_{hj}/(1-Z_{hi})$, where unit i is the first unit selected. The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} \quad = \quad Z_{hi}\,Z_{hj}\,\frac{2-Z_{hi}-Z_{hj}}{(1-Z_{hi})\,(1-Z_{hj})}$$

Refer to Cochran (1977, pp. 263–265) and Murthy (1957).

Sampford's PPS Method

Sampford's method (METHOD=PPS_SAMPFORD) is an extension of Brewer's method that selects more than two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals

$$P_{hi} \quad = \quad n_h \, \frac{M_{hi}}{M_{h\cdot}} = n_h \, Z_{hi}$$

Sampford's method first selects a unit from stratum h with probability Z_{hi} . Then subsequent units are selected with probability proportional to

$$\frac{Z_{hi}}{1 - n_h \, Z_{hi}}$$

and with replacement. If the same unit appears more than once in the sample of size n_h, then Sampford's algorithm rejects that sample and selects a new sample. The sample is accepted if it contains n_h distinct units.

The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} \quad = \quad K_h \, \lambda_i \, \lambda_j \, \sum_{t=2}^{n_h} \left[t - n_h \, (P_{hi} + P_{hj}) \, L_{n_h - t}(ij) \right] \, / \, n_h^{t-2}$$

where

$$\lambda_i \quad = \quad \frac{Z_{hi}}{1 - n_h \, Z_{hi}}$$

$$L_m \quad = \quad \sum_{S(m)} \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m}$$

where $S(m)$ denotes all possible samples of size m, for $m = 1, 2, \ldots, N_h$. The sum $L_m(ij)$ is defined similarly to L_m but sums over all possible samples of size m that do not include units i and j, and

$$K_h \quad = \quad \left(\sum_{t=1}^{n_h} t \, L_{n_h - t} \, / \, n_h^t \right)^{-1}$$

Sampford's method requires that the relative size Z_{hi} be less than $1/n_h$ for all units. Refer to Cochran (1977, pp. 262–263) and Sampford (1967).

Output Data Set

PROC SURVEYSELECT creates a SAS data set that contains the sample of selected units. You can specify the name of this output data set with the OUT= option in the PROC SURVEYSELECT statement. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

By default, the output data set contains one observation for each unit selected for the sample. But if you specify the OUTALL option, the output data set includes all observations from the input data set. With OUTALL, the output data set also contains a variable to indicate each observation's selection status. The variable Selected equals 1 for an observation selected for the sample, and equals 0 for an observation not selected. The OUTALL option is available only for equal probability selection methods.

If you specify the OUTHITS option for methods that may select the same unit more than once (that is, methods that select with replacement or with minimum replacement), the output data set contains a separate observation for each selection. If you do not specify the OUTHITS option, the output data set contains only one observation for each selected unit, even if the unit is selected more than once, and the variable NumberHits contains the number of hits or selections for that unit.

The output data set contains design information and selection statistics, depending on the selection method and output options you specify. The output data set can include the following variables:

- Selected, which indicates whether or not the observation is selected for the sample. This variable is included if you specify the OUTALL option. It equals 1 for an observation selected for the sample, and it equals 0 for an observation not selected.
- STRATA variables, which you specify in the STRATA statement
- Replicate, which is the sample replicate number. This variable is included when you request replicated sampling with the REP= option.
- ID variables, which you name in the ID statement
- CONTROL variables, which you specify in the CONTROL statement
- Zone, which is the selection zone. This variable is included for METHOD=PPS_SEQ.
- SIZE variable, which you specify in the SIZE statement
- AdjustedSize, which is the adjusted size measure. This variable is included if you request adjusted sizes with the MINSIZE= option or the MAXSIZE= option.
- Certain, which indicates certainty selection. This variable is included if you specify the CERTSIZE= option. It equals 1 for units included with certainty because their size measures exceed the certainty size measure. Otherwise, it equals 0.

- NumberHits, which is the number of hits or selections. This variable is included for selection methods that are with replacement or with minimum replacement (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).

The output data set includes the following variables if you request a PPS selection method or if you specify the STATS option for other methods:

- ExpectedHits, which is the expected number of hits or selections. This variable is included for selection methods that are with replacement or with minimum replacement, and so may select the same unit more than once (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).
- SelectionProb, which is the probability of selection. This variable is included for selection methods that are without replacement.
- SamplingWeight, which is the sampling weight. This variable equals the inverse of ExpectedHits or SelectionProb.

For METHOD=PPS_BREWER and METHOD=PPS_MURTHY, which select two units from each stratum with probability proportional to size, the output data set contains the following variable:

- JtSelectionProb, which is the joint probability of selection for the two units selected from the stratum

If you request the JTPROBS option to compute joint probabilities of selection for METHOD=PPS or METHOD=PPS_SAMPFORD, then the output data set contains the following variables:

- Unit, which is an identification variable that numbers the selected units sequentially within each stratum
- JtProb_1, JtProb_2, JtProb_3, . . . , where the variable JtProb_1 contains the joint probability of selection for the current unit and unit 1. Similarly, JtProb_2 contains the joint probability of selection for the current unit and unit 2, and so on.

If you request the JTPROBS option for METHOD=PPS_WR, then the output data set contains the following variables:

- Unit, which is an identification variable that numbers the selected units sequentially within each stratum
- JtHits_1, JtHits_2, JtHits_3, . . . , where the variable JtHits_1 contains the joint expected number of hits for the current unit and unit 1. Similarly, JtHits_2 contains the joint expected number of hits for the current unit and unit 2, and so on.

If you request the OUTSIZE option, the output data set contains the following variables. If you specify a STRATA statement, the output data set includes stratum-level values of these variables. Otherwise, the output data set contains population-level values of these variables.

- MinimumSize, which is the minimum size measure specified with the MINSIZE= option. This variable is included if you request the MINSIZE= option.
- MaximumSize, which is the maximum size measure specified with the MAXSIZE= option. This variable is included if you request the MAXSIZE= option.
- CertaintySize, which is the certainty size measure specified with the CERTSIZE= option. This variable is included if you request the CERTSIZE= option.
- Total, which is the total number of sampling units in the stratum. This variable is included if there is no SIZE statement.
- TotalSize, which is the total of size measures in the stratum. This variable is included if there is a SIZE statement.
- TotalAdjSize, which is the total of adjusted size measures in the stratum. This variable is included if there is a SIZE statement and if you request adjusted sizes with the MAXSIZE= option or the MINSIZE= option.
- SamplingRate, which is the sampling rate. This variable is included if you specify the SAMPRATE= option.
- SampleSize, which is the sample size. This variable is included if you specify the SAMPSIZE= option, or if you specify METHOD=PPS_BREWER or METHOD=PPS_MURTHY, which select two units from each stratum.

If you request the OUTSEED option, the output data set contains the following variable:

- InitialSeed, which is the initial seed for the stratum.

Displayed Output

By default, PROC SURVEYSELECT displays two tables that summarize the sample selection. You can suppress display of these tables by using the NOPRINT option.

PROC SURVEYSELECT creates an output data set that contains the units selected for the sample. The procedure does not display this output data set. Use PROC PRINT, PROC REPORT, or any other SAS reporting tool to display the output data set.

PROC SURVEYSELECT displays the following information in the "Sample Selection Method" table:

- Selection Method
- Size Measure variable, if you specify a SIZE statement
- Minimum Size Measure, if you specify the MINSIZE= option
- Maximum Size Measure, if you specify the MAXSIZE= option
- Certainty Size Measure, if you specify the CERTSIZE= option
- Strata Variables, if you specify a STRATA statement
- Control Variables, if you specify a CONTROL statement
- type of Control Sorting, Serpentine or Nested, if you specify a CONTROL statement

PROC SURVEYSELECT displays the following information in the "Sample Selection Summary" table:

- Input Data Set name
- Sorted Data Set name, if you specify the OUTSORT= option
- Random Number Seed
- Sample Size or Stratum Sample Size, if you specify the SAMPSIZE=*n* option
- Sample Size Data Set, if you specify the SAMPSIZE=*SAS-data-set* option
- Sampling Rate or Stratum Sampling Rate, if you specify the SAMPRATE=*r* option
- Sampling Rate Data Set, if you specify the SAMPRATE=*SAS-data-set* option
- Minimum Sample Size or Stratum Minimum Sample Size, if you specify the NMIN= option with the SAMPRATE= option
- Maximum Sample Size or Stratum Maximum Sample Size, if you specify the NMAX= option with the SAMPRATE= option
- Selection Probability, if you specify METHOD=SRS, METHOD=SYS, or METHOD=SEQ and do not specify a STRATA statement
- Expected Number of Hits, if you specify METHOD=URS and do not specify a STRATA statement
- Sampling Weight for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, METHOD=SEQ) if you do not specify a STRATA statement
- Number of Strata, if you specify a STRATA statement
- Number of Replicates, if you specify the REP= option
- Total Sample Size, if you specify a STRATA statement or the REP= option
- Output Data Set name

ODS Table Names

PROC SURVEYSELECT assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 72.2. ODS Tables Produced in PROC SURVEYSELECT

ODS Table Name	Description	Statement	Option
Method	Sample selection method	PROC	default
Summary	Sample selection summary	PROC	default

Examples

Example 72.1. Replicated Sampling

This example uses the Customers data set from the section “Getting Started” on page 4422. The data set Customers contains an Internet service provider’s current subscribers, and the service provider wants to select a sample from this population for a customer satisfaction survey.

This example illustrates replicated sampling, which selects multiple samples from the survey population according to the same design. You can use replicated sampling to provide a simple method of variance estimation, or to evaluate variable nonsampling errors such as interviewer differences. Refer to Lohr (1999), Kish (1965, 1987), and Kalton (1983) for information on replicated sampling.

This design includes four replicates, each with a sample size of 50 customers. The sampling frame is stratified by State and sorted by Type and Usage within strata. Customers are selected by sequential random sampling with equal probability within strata. The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set using this design.

```
title1 ’Customer Satisfaction Survey’;
title2 ’Replicated Sampling’;
proc surveyselect data=Customers
      method=seq n=(8 12 20 10)
      rep=4
      seed=40070 out=SampleRep;
   strata State;
   control Type Usage;
run;
```

The STRATA statement names the stratification variable State. The CONTROL statement names the control variables Type and Usage. In the PROC SURVEYSELECT statement, the METHOD=SEQ option requests sequential random sampling. The REP=4 option specifies four replicates of this sample. The N=(8 12 20 10) option specifies the stratum sample sizes for each replicate. The

N= option lists the stratum sample sizes in the same order as the strata appear in the Customers data set, which has been sorted by State. The sample size of eight customers corresponds to the first stratum, State = ‘AL’. The sample size 12 corresponds to the next stratum, State = ‘FL’, and so on. The SEED=40070 option specifies ’40070’ as the initial seed for random number generation.

Output 72.1.1 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A total of 200 customers is selected in four replicates. PROC SURVEYSELECT selects each replicate using sequential random sampling within strata determined by State. The sampling frame Customers is sorted by control variables Type and Usage within strata, according to hierarchic serpentine sorting. The output data set SampleRep contains the sample.

Output 72.1.1. Sample Selection Summary

```
                          Customer Satisfaction Survey
                              Replicated Sampling

                           The SURVEYSELECT Procedure

                Selection Method     Sequential Random Sampling
                                     With Equal Probability
                Strata Variable      State
                Control Variables    Type
                                     Usage
                Control Sorting      Serpentine

                       Input Data Set           CUSTOMERS
                       Random Number Seed           40070
                       Number of Strata                 4
                       Number of Replicates             4
                       Total Sample Size              200
                       Output Data Set          SAMPLEREP
```

The following PROC PRINT statements display the selected customers for the first stratum, State = ‘AL’, from the output data set SampleRep.

```
title1 ’Customer Satisfaction Survey’;
title2 ’Sample Selected by Replicated Design’;
title3 ’(First Stratum)’;
proc print data=SampleRep;
   where State = ’AL’;
run;
```

Output 72.1.2. Customer Sample (First Stratum)

```
                          Customer Satisfaction Survey
                      Sample Selected by Replicated Design
                                (First Stratum)

                                                           Selection    Sampling
 Obs    State    Replicate     CustomerID    Type    Usage       Prob      Weight

   1     AL          1        882-37-7496    New       572    .004115226      243
   2     AL          1        581-32-5534    New       863    .004115226      243
   3     AL          1        980-29-2898    Old       571    .004115226      243
   4     AL          1        172-56-4743    Old       128    .004115226      243
   5     AL          1        998-55-5227    Old        35    .004115226      243
   6     AL          1        625-44-3396    New        60    .004115226      243
   7     AL          1        627-48-2509    New       114    .004115226      243
   8     AL          1        257-66-6558    New       172    .004115226      243
   9     AL          2        622-83-1680    New        22    .004115226      243
  10     AL          2        343-57-1186    New        53    .004115226      243
  11     AL          2        976-05-3796    New       110    .004115226      243
  12     AL          2        859-74-0652    New       303    .004115226      243
  13     AL          2        476-48-1066    New       839    .004115226      243
  14     AL          2        109-27-8914    Old      2102    .004115226      243
  15     AL          2        743-25-0298    Old       376    .004115226      243
  16     AL          2        722-08-2215    Old       105    .004115226      243
  17     AL          3        668-57-7696    New       200    .004115226      243
  18     AL          3        300-72-0129    New       471    .004115226      243
  19     AL          3        073-60-0765    New       656    .004115226      243
  20     AL          3        526-87-0258    Old       672    .004115226      243
  21     AL          3        726-61-0387    Old       150    .004115226      243
  22     AL          3        632-29-9020    Old        51    .004115226      243
  23     AL          3        417-17-8378    New        56    .004115226      243
  24     AL          3        091-26-2366    New        93    .004115226      243
  25     AL          4        336-04-1288    New       419    .004115226      243
  26     AL          4        827-04-7407    New       650    .004115226      243
  27     AL          4        317-70-6496    Old       452    .004115226      243
  28     AL          4        002-38-4582    Old       206    .004115226      243
  29     AL          4        181-83-3990    Old        33    .004115226      243
  30     AL          4        675-34-7393    New        47    .004115226      243
  31     AL          4        228-07-6671    New        65    .004115226      243
  32     AL          4        298-46-2434    New       161    .004115226      243
```

Output 72.1.2 displays the 32 sample customers of the first stratum (State = ‘AL’) from the output data set SampleRep, which includes the entire sample of 200 customers. The variable SelectionProb contains the selection probability, and SamplingWeight contains the sampling weight. Since customers are selected with equal probability within strata in this design, all customers in the same stratum have the same selection probability. These selection probabilities and sampling weights apply to a single replicate, and the variable Replicate contains the sample replicate number.

Example 72.2. PPS Selection of Two Units Per Stratum

A state health agency plans to conduct a state-wide survey of a variety of different hospital services. The agency plans to select a probability sample of individual discharge records within hospitals using a two-stage sample design. First stage units are hospitals, and second stage units are patient discharges during the study time period. Hospitals are stratified first according to geographic region and then by rural/urban type and size of hospital. Two hospitals are selected from each stratum with probability proportional to size. This example describes hospital selection for this survey using PROC SURVEYSELECT.

The data set HospitalFrame contains all hospitals in the first geographical region of this state.

```
data HospitalFrame;
   input Hospital$ Type$ SizeMeasure @@;
   if (SizeMeasure < 20) then Size='Small ';
      else if (SizeMeasure < 50) then Size='Medium';
      else Size='Large ';
   datalines;
034 Rural  0.870   107 Rural  1.316
079 Rural  2.127   223 Rural  3.960
236 Rural  5.279   165 Rural  5.893
086 Rural  0.501   141 Rural 11.528
042 Urban  3.104   124 Urban  4.033
006 Urban  4.249   261 Urban  4.376
195 Urban  5.024   190 Urban 10.373
038 Urban 17.125   083 Urban 40.382
259 Urban 44.942   129 Urban 46.702
133 Urban 46.992   218 Urban 48.231
026 Urban 61.460   058 Urban 65.931
119 Urban 66.352
;
```

In the SAS data set HospitalFrame, the variable Hospital identifies the hospital. The variable Type equals ‘Urban’ if the hospital is located in an urbanized area, and ‘Rural’ otherwise. The variable SizeMeasure contains the hospital’s size measure, which is constructed from past data on service utilization for the hospital together with the desired sampling rates for each service. This size measure reflects the amount of relevant survey information expected from the hospital. Refer to Drummond et al. (1982) for details on this type of size measure. The variable Size equals ‘Small’, ‘Medium’, or ‘Large’, depending on the value of the hospital’s size measure.

The following PROC PRINT statements display the data set Hospital Frame.

```
title1 'Hospital Utilization Survey';
title2 'Sampling Frame, Region 1';
proc print data=HospitalFrame;
run;
```

Output 72.2.1. Sampling Frame

```
                           Hospital Utilization Survey
                             Sampling Frame, Region 1

                                                        Size
                  Obs     Hospital     Type           Measure      Size

                    1       034        Rural           0.870     Small
                    2       107        Rural           1.316     Small
                    3       079        Rural           2.127     Small
                    4       223        Rural           3.960     Small
                    5       236        Rural           5.279     Small
                    6       165        Rural           5.893     Small
                    7       086        Rural           0.501     Small
                    8       141        Rural          11.528     Small
                    9       042        Urban           3.104     Small
                   10       124        Urban           4.033     Small
                   11       006        Urban           4.249     Small
                   12       261        Urban           4.376     Small
                   13       195        Urban           5.024     Small
                   14       190        Urban          10.373     Small
                   15       038        Urban          17.125     Small
                   16       083        Urban          40.382     Medium
                   17       259        Urban          44.942     Medium
                   18       129        Urban          46.702     Medium
                   19       133        Urban          46.992     Medium
                   20       218        Urban          48.231     Medium
                   21       026        Urban          61.460     Large
                   22       058        Urban          65.931     Large
                   23       119        Urban          66.352     Large
```

The following PROC SURVEYSELECT statements select a probability sample of hospitals from the HospitalFrame data set, using a stratified design with PPS selection of two units from each stratum.

```
title1 'Hospital Utilization Survey';
proc surveyselect data=HospitalFrame
      method=pps_brewer
      seed=48702 out=SampleHospitals;
   size SizeMeasure;
   strata Type Size notsorted;
run;
```

The STRATA statement names the stratification variables Type and Size. The NOTSORTED option specifies that observations with the same STRATA variable values are grouped together but are not necessarily sorted in alphabetical or increasing numerical order. In the HospitalFrame data set, Size = ‘Small’ precedes Size = ‘Medium’.

In the PROC SURVEYSELECT statement, the METHOD=PPS_BREWER option requests sample selection by Brewer’s method, which selects two units per stratum with probability proportional to size. The SEED=48702 option specifies ’48702’ as the initial seed for random number generation. The SIZE statement specifies the size measure variable. It is not necessary to specify the sample size with the N= option, since Brewer’s method always selects two units from each stratum.

Output 72.2.2 displays the output from PROC SURVEYSELECT. A total of 8 hospitals were selected from the 4 strata. The data set SampleHospitals contains the selected hospitals.

Output 72.2.2. Sample Selection Summary

```
                           Hospital Utilization Survey

                            The SURVEYSELECT Procedure

                     Selection Method    Brewer's PPS Method
                     Size Measure        SizeMeasure
                     Strata Variables    Type
                                         Size

                      Input Data Set            HOSPITALFRAME
                      Random Number Seed                48702
                      Stratum Sample Size                   2
                      Number of Strata                      4
                      Total Sample Size                     8
                      Output Data Set         SAMPLEHOSPITALS
```

The following PROC PRINT statements display the sample hospitals.

```
title1 'Hospital Utilization Survey';
title2 'Sample Selected by Stratified PPS Design';
proc print data=SampleHospitals;
run;
```

Output 72.2.3. Sample Hospitals

```
                           Hospital Utilization Survey
                     Sample Selected by Stratified PPS Design

                                                                           Jt
                                            Size     Selection  Sampling  Selection
Obs   Type     Size      Hospital         Measure     Prob      Weight     Prob

 1    Rural    Small       079             2.127     0.13516    7.39868   0.01851
 2    Rural    Small       236             5.279     0.33545    2.98106   0.01851
 3    Urban    Small       006             4.249     0.17600    5.68181   0.01454
 4    Urban    Small       195             5.024     0.20810    4.80533   0.01454
 5    Urban    Medium      133            46.992     0.41357    2.41795   0.11305
 6    Urban    Medium      218            48.231     0.42448    2.35584   0.11305
 7    Urban    Large       026            61.460     0.63445    1.57617   0.31505
 8    Urban    Large       058            65.931     0.68060    1.46929   0.31505
```

The variable SelectionProb contains the selection probability for each hospital in the sample. The variable JtSelectionProb contains the joint probability of selection for the two sample hospitals in the same stratum. The variable SamplingWeight contains the sampling weight component for this first stage of the design. The final-stage weight components, which correspond to patient record selection within hospitals, can be multiplied by the hospital weight components to obtain the overall sampling weights.

Example 72.3. PPS (Dollar-Unit) Sampling

A small company wants to audit employee travel expenses in an effort to improve the expense reporting procedure and possibly reduce expenses. The company does not have resources to examine all expense reports and wants to use statistical sampling to objectively select expense reports for audit.

The data set TravelExpense contains the dollar amount of all employee travel expense transactions during the past month.

```
data TravelExpense;
   input ID$ Amount @@;
   if (Amount < 500) then Level='1_Low ';
       else if (Amount > 1500) then Level='3_High';
       else Level='2_Avg ';
   datalines;
110  237.18   002  567.89   234  118.50
743   74.38   411 1287.23   782  258.10
216  325.36   174  218.38   568 1670.80
302  134.71   285 2020.70   314   47.80
139 1183.45   775  330.54   425  780.10
506  895.80   239  620.10   011  420.18
672  979.66   142  810.25   738  670.85
192  314.58   243   87.50   263 1893.40
496  753.30   332  540.65   486 2580.35
614  230.56   654  185.60   308  688.43
784  505.14   017  205.48   162  650.42
289 1348.34   691   30.50   545 2214.80
517  940.35   382  217.85   024  142.90
478  806.90   107  560.72
;
```

In the SAS data set TravelExpense, the variable ID identifies the travel expense report. The variable Amount contains the dollar amount of the reported expense. The variable Level equals '1_Low', '2_Avg', or '3_High', depending on the value of Amount.

In the sample design for this audit, expense reports are stratified by Level. This ensures that each of these expense levels is included in the sample and also permits a disproportionate allocation of the sample, selecting proportionately more of the expense reports from the higher levels. Within strata, the sample of expense reports is selected with probability proportional to the amount of the expense, thus giving a greater chance of selection to larger expenses. In auditing terms, this is known as monetary-unit sampling. Refer to Wilburn (1984).

PROC SURVEYSELECT requires that the input data set be sorted by the STRATA variables. The following PROC SORT statements sort the TravelExpense data set by the stratification variable Level.

```
proc sort data=TravelExpense;
   by Level;
run;
```

The following PROC PRINT statements display the sampling frame data set TravelExpense, which contains 41 observations.

```
title1 'Travel Expense Audit';
proc print data=TravelExpense;
run;
```

Output 72.3.1. Sampling Frame

```
   Travel Expense Audit

Obs  ID    Amount  Level

  1  110   237.18  1_Low
  2  002   567.89  2_Avg
  3  234   118.50  1_Low
  4  743    74.38  1_Low
  5  411  1287.23  2_Avg
  6  782   258.10  1_Low
  7  216   325.36  1_Low
  8  174   218.38  1_Low
  9  568  1670.80  3_High
 10  302   134.71  1_Low
 11  285  2020.70  3_High
 12  314    47.80  1_Low
 13  139  1183.45  2_Avg
 14  775   330.54  1_Low
 15  425   780.10  2_Avg
 16  506   895.80  2_Avg
 17  239   620.10  2_Avg
 18  011   420.18  1_Low
 19  672   979.66  2_Avg
 20  142   810.25  2_Avg
 21  738   670.85  2_Avg
 22  192   314.58  1_Low
 23  243    87.50  1_Low
 24  263  1893.40  3_High
 25  496   753.30  2_Avg
 26  332   540.65  2_Avg
 27  486  2580.35  3_High
 28  614   230.56  1_Low
 29  654   185.60  1_Low
 30  308   688.43  2_Avg
 31  784   505.14  2_Avg
 32  017   205.48  1_Low
 33  162   650.42  2_Avg
 34  289  1348.34  2_Avg
 35  691    30.50  1_Low
 36  545  2214.80  3_High
 37  517   940.35  2_Avg
 38  382   217.85  1_Low
 39  024   142.90  1_Low
 40  478   806.90  2_Avg
 41  107   560.72  2_Avg
```

The following PROC SURVEYSELECT statements select a probability sample of expense reports from the TravelExpense data set using the stratified design with PPS selection within strata.

```
title1 'Travel Expense Audit';
proc surveyselect data=TravelExpense
      method=pps n=(6 10 4)
      seed=47279 out=AuditSample;
   size Amount;
   strata Level;
run;
```

The STRATA statement names the stratification variable Level. The SIZE statement specifies the size measure variable Amount. In the PROC SURVEYSELECT statement, the METHOD=PPS option requests sample selection with probability proportional to size and without replacement. The N=(6 10 4) option specifies the stratum sample sizes, listing the sample sizes in the same order that the strata appear in the TravelExpense data set. The sample size of 6 corresponds to the first stratum, Level = '1_Low', the sample size of 10 corresponds to the second stratum, Level = '2_Avg', and 4 corresponds to the last stratum, Level = '3_High'. The SEED=47279 option specifies '47279' as the initial seed for random number generation.

Output 72.3.2 displays the output from PROC SURVEYSELECT. A total of 20 expense reports is selected for audit. The data set AuditSample contains the sample of travel expense reports.

Output 72.3.2. Sample Selection Summary

```
                                    Travel Expense Audit

                                 The SURVEYSELECT Procedure

                     Selection Method    PPS, Without Replacement
                     Size Measure        Amount
                     Strata Variable     Level

                         Input Data Set        TRAVELEXPENSE
                         Random Number Seed            47279
                         Number of Strata                  3
                         Total Sample Size                20
                         Output Data Set         AUDITSAMPLE
```

The following PROC PRINT statements display the audit sample, which is shown in Output 72.3.3.

```
title1 'Travel Expense Audit';
title2 'Sample Selected by Stratified PPS Design';
proc print data=AuditSample;
run;
```

Output 72.3.3. Audit Sample

```
                        Travel Expense Audit
                Sample Selected by Stratified PPS Design

                                               Selection     Sampling
         Obs     Level      ID        Amount        Prob        Weight

           1     1_Low      654       185.60     0.31105       3.21489
           2     1_Low      017       205.48     0.34437       2.90385
           3     1_Low      382       217.85     0.36510       2.73896
           4     1_Low      614       230.56     0.38640       2.58797
           5     1_Low      782       258.10     0.43256       2.31183
           6     1_Low      775       330.54     0.55396       1.80518
           7     2_Avg      784       505.14     0.34623       2.88823
           8     2_Avg      332       540.65     0.37057       2.69853
           9     2_Avg      002       567.89     0.38924       2.56909
          10     2_Avg      239       620.10     0.42503       2.35278
          11     2_Avg      738       670.85     0.45981       2.17479
          12     2_Avg      496       753.30     0.51633       1.93676
          13     2_Avg      425       780.10     0.53470       1.87022
          14     2_Avg      478       806.90     0.55307       1.80810
          15     2_Avg      672       979.66     0.67148       1.48925
          16     2_Avg      139      1183.45     0.81116       1.23280
          17     3_High     568      1670.80     0.64385       1.55316
          18     3_High     263      1893.40     0.72963       1.37056
          19     3_High     285      2020.70     0.77869       1.28421
          20     3_High     486      2580.35     0.99435       1.00568
```

References

Bentley, J.L. and Floyd, R. (1987), “A Sample of Brilliance,” *Communications of the Association for Computing Machinery*, 30, 754–757.

Bentley, J.L. and Knuth, D. (1986), “Literate Programming,” *Communications of the Association for Computing Machinery*, 29, 364–369.

Brewer, K.W.R. (1963), “A Model of Systematic Sampling with Unequal Probabilities,” *Australian Journal of Statistics*, 5, 93–105.

Chromy, J.R. (1979), “Sequential Sample Selection Methods,” *Proceedings of the American Statistical Association, Survey Research Methods Section*, 401–406.

Cochran, W.G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons, Inc.

Drummond, D., Lessler, J., Watts, D., and Williams, S. (1982), “A Design for Achieving Prespecified Levels of Representation for Multiple Domains in Health Record Samples,” *Proceedings of the Fourth Conference on Health Survey Research Methods*, DHHS Publication No. (PHS) 84-3346, Washington, D.C.: National Center for Health Services Research, 233–248.

Durbin, J. (1967), “Design of Multi-stage Surveys for the Estimation of Sampling Errors,” *Applied Statistics*, 16, 152–164.

Fan, C.T., Muller, M.E., and Rezucha, I. (1962), "Development of Sampling Plans by Using Sequential (Item by Item) Selection Techniques and Digital Computers," *Journal of the American Statistical Association*, 57, 387–402.

Fishman, G.S. and Moore, L.R. (1982), "A Statistical Evaluation of Multiplicative Congruential Generators with Modulus $(2^{31} - 1)$," *Journal of the American Statistical Association*, 77, 129–136.

Fox, D.R. (1989), "Computer Selection of Size-Biased Samples," *The American Statistician*, 43(3), 168–171.

Golmant, J. (1990), "Correction: Computer Selection of Size-Biased Samples," *The American Statistician*, 44(2), 194.

Hanurav, T.V. (1967), "Optimum Utilization of Auxiliary Information: π_{ps} Sampling of Two Units from a Stratum," *Journal of the Royal Statistical Society, Series B*, 29, 374–391.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA and London: Sage Publications, Inc.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons, Inc.

Kish, L. (1987), *Statistical Design for Research*, New York: John Wiley & Sons, Inc.

Lohr, S.L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Madow, W.G. (1949), "On the Theory of Systematic Sampling, II," *Annals of Mathematical Statistics*, 20, 333–354.

McLeod, A.I. and Bellhouse, D.R. (1983), "A Convenient Algorithm for Drawing a Simple Random Sample," *Applied Statistics*, 32, 182–183.

Murthy, M.N. (1957), "Ordered and Unordered Estimators in Sampling Without Replacement," *Sankhyā*, 18, 379–390.

Murthy, M.N. (1967), *Sampling Theory and Methods*, Calcutta, India: Statistical Publishing Society.

Sampford, M.R. (1967), "On Sampling without Replacement with Unequal Probabilities of Selection," *Biometrika*, 54, 499–513.

Vijayan, K. (1968), "An Exact π_{ps} Sampling Scheme: Generalization of a Method of Hanurav," *Journal of the Royal Statistical Society, Series B*, 30, 556–566.

Watts, D.L. (1991), "Correction: Computer Selection of Size-Biased Samples," *The American Statistician*, 45(2), 172.

Wilburn, A.J. (1984), *Practical Statistical Sampling for Auditors*, New York: Marcel Dekker, Inc.

Williams, R.L. and Chromy, J.R. (1980), "SAS Sample Selection Macros," *Proceedings of the Fifth Annual SAS Users Group International Conference*, 5, 392–396.

Chapter 73
The TPHREG Procedure (Experimental)

Chapter Contents

Chapter 73
The TPHREG Procedure (Experimental)

Overview

The TPHREG procedure, experimental in this release, includes most of the functionality of the PHREG procedure (see Chapter 54, “The PHREG Procedure”) with the additional benefits of the CLASS statement. The CLASS statement enables you to specify categorical variables (also known as factors or CLASS variables) to be used in the analysis. Model effects, including covariates, main effects (CLASS variables), crossed effects (interactions), and nested effects, can be specified in the same way as in the GLM procedure. The CLASS statement supports the less-than-full-rank parameterization as in the GLM procedure as well as various full-rank parameterization methods such as reference coding, effect coding, and orthogonal polynomial coding. For some of the full-rank coding schemes, you can designate a specific value (category or level) of the CLASS variable as the reference level. The CLASS statement also enables you to specify the ordering of the categories of CLASS variables, to reverse the ordering of the categories, and to treat categories with missing values as valid categories.

With the TPHREG procedure, you can control how to move model effects in and out of a model with various model-building strategies such as forward selection, backward elimination, or stepwise selection. When there are no interaction terms, a main effect can enter or leave a model in a single step based on the p-value of the score or Wald statistic, respectively. When there are crossed or nested effects, the selection process also depends on whether you want to preserve model hierarchy. The HIERARCHY= option in the MODEL statement enables you to specify whether model hierarchy is to be preserved, how model hierarchy is applied, and whether a single effect or multiple effects can be moved in a single step.

The TPHREG procedure also enables you to specify CONTRAST statements for testing customized hypotheses concerning the regression parameters. Each CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful in comparing the hazards between the categories of a CLASS variable.

Syntax

The PROC TPHREG statement invokes the TPHREG procedure. All the other statements in the PHREG procedure (with the exception of the experimental ASSESS statement) are available in the TPHREG procedure. The MODEL statement in the TPHREG procedure enables you to specify explanatory effects, not just individual continuous variables, and has additional options specifically for having CLASS variables. In addition, you can specify the CLASS statement and the CONTRAST statement as follows:

CLASS *variable <(options)> <variable <(options)>... >*
< / options >;

CONTRAST *'label' effect values <,... effect values>< /options >*;

The PROC TPHREG statement and the MODEL statement are required. The CLASS statement (if used) must precede the MODEL statement, and CONTRAST statements (if used) must follow the MODEL statement. The CLASS statement specifies the categorical variables to be used as explanatory variables. Variables specified in the CLASS statement have to exist in the input data. You cannot use programming statements to modify their values. The CONTRAST statement tests customized hypotheses concerning the regression parameters and estimates the parameters for the corresponding rows of contrasts.

PROC TPHREG Statement

PROC TPHREG *< options >* **;**

All PROC PHREG statement options can be used in the PROC TPHREG statement. In addition, you can specify the following option:

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

MODEL Statement

MODEL *time < *censor (list) > = effects < /options >* **;**

MODEL *(t1, t2) < *censor(list) > = effects < /options >* **;**

The specifications of the time variables, the censoring indicator, and censored values are same as those in the PHREG procedure. The model *effects*, which follow the equal sign, include continuous or CLASS variables as the main effects. Categorical variables, which can be character or numeric, must be declared in the CLASS statement. Crossed and nested effects can be specified in the same fashion as in the GLM procedure (see the section "Specification of Effects" on page 1784 of Chapter 32, "The GLM Procedure," for more information).

Any MODEL statement options in the PHREG procedure can be used in the TPHREG procedure. To accommodate the broader specification of model effects, the variable-selection options INCLUDE=, START=, and STOP= have been modified.

INCLUDE=*n*

includes the first n effects in the MODEL statement in every model. By default, INCLUDE=0. The INCLUDE= option has no effect when SELECTION=NONE.

START=*n*

begins the FORWARD, BACKWARD, or STEPWISE selection process with the first n effects listed in the MODEL statement. The value of n ranges from 0 to s, where s is the total number of effects in the MODEL statement. The default value of n is s for the BACKWARD method and 0 for the FORWARD and STEPWISE methods. Note that START=n specifies only that the first n effects appear in the first model, while INCLUDE=n requires that the first n effects be included in every model. For the SCORE method, START=n specifies that the smallest models contain n effects, where n ranges from 1 to s; the default value is 1. The START= option has no effect when SELECTION=NONE.

STOP=*n*

specifies the maximum (FORWARD method) or minimum (BACKWARD method) number of effects to be included in the final model. The effect selection process is stopped when n effects are found. The value of n ranges from 0 to s, where s is the total number of effects in the MODEL statement. The default value of n is s for the FORWARD method and 0 for the BACKWARD method. For the SCORE method, STOP=n specifies that the smallest models contain n effects, where n ranges from 1 to s; the default value of n is s. The STOP= option has no effect when SELECTION=NONE or STEPWISE.

Two new options are added to the MODEL statement in the TPHREG procedure.

HIERARCHY=*keyword*
HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied and whether a single effect or multiple effects are allowed to enter or leave the model in one step. You can specify that only CLASS variable effects, or both CLASS and continuous variable effects, be subject to the hierarchy requirement. The HIERARCHY= option is ignored unless you also specify the FORWARD, BACKWARD, or STEPWISE selection method.

Model hierarchy refers to the requirement that, for any term to be in the model, all effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are described as follows:

NONE

Model hierarchy is not maintained. Any single effect can enter or leave the model at any given step of the selection process.

SINGLE

Only one effect can enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that you specify the main effects A and B and the interaction of A*B in the model. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already been entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and continuous variables) are subject to the hierarchy requirement.

SINGLECLASS

This is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

MULTIPLE

More than one effect can enter or leave the model at one time, subject to the model hierarchy requirement. In a forward selection step, a single main effect can enter the model, or an interaction can enter the model together with all the effects that are contained in the interaction. In a backward elimination step, an interaction itself, or the interaction together with all the effects that the interaction contains, can be removed. All effects (CLASS and continuous variable) are subject to the hierarchy requirement.

MULTIPLECLASS

This is the same as HIERARCHY=MULTIPLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=SINGLE, which means that model hierarchy is to be maintained for all effects (that is, both CLASS and continuous variable effects) and that only a single effect can enter or leave the model at each step.

NODUMMYPRINT
NODESIGNPRINT
NODP

suppresses the “Class Level Information” table, which shows how the design matrix columns for the CLASS variables are coded.

CLASS Statement

CLASS *variable <(options)> <variable <(options)>... >*
< / options >;

The CLASS statement names the categorical variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* for the CLASS statement by placing them after a slash (/). Global *options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *options* specified on any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the global *options*.

CPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding dummy variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING
DESC

reverses the sorting order of the categorical variable.

LPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding dummy variables.

MISSING

allows missing value (for example,'.' for a numeric variable and blanks for a character variable) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the categories of cateogrical variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use the CONTRAST statement. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. The following table shows how PROC TPHREG interprets values of the ORDER= option.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the categorical variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=REF. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the "CLASS Variable Parameterization" section on page 4482 for further details.

EFFECT	specifies effect coding
GLM	specifies less-than-full-rank, reference-cell coding; this option can only be used as a global option
ORDINAL	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding
REFERENCE REF	specifies reference cell coding
ORTHEFFECT	orthogonalizes PARAM=EFFECT
ORTHORDINAL	orthogonalizes PARAM=ORDINAL
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL
ORTHREF	orthogonalizes PARAM=REFERENCE

The EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank parameterization. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization.

REF=*'level'* **|** *keyword*

specifies the reference level for PARAM=EFFECT or PARAM=REF. For an individual variable, you can specify a specific *level* of the variable in the REF= option. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference
LAST	designates the last ordered level as reference

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of CLASS variables. This is a global option, not an individual CLASS variable option.

CONTRAST Statement

CONTRAST *'label' row-description* <,... *row-description*>< /*options*> **;**

where a *row-description* is: *effect values* <,...*effect values*>

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST statement in PROC GLM and PROC CATMOD, depending on the coding schemes used with any categorical variables involved.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$. You must be familiar with the details of the model parameterization that PROC TPHREG uses (for more information, see the PARAM= option in the "CLASS Statement" section on page 4477). Optionally, the CONTRAST statement enables you to estimate each row, $\boldsymbol{l}'_i\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$ and test the hypothesis $\boldsymbol{l}'_i\boldsymbol{\beta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The "Class Level Information" table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters.

The rows of $\mathbf{L}$ are specified in order and are separated by commas. Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*. For any of the full-rank parameterizations, if an effect is not specified in the CONTRAST statement, all of its coefficients in the $\mathbf{L}$ matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

When you use effect coding (by specifying PARAM=EFFECT in the CLASS statement), all parameters are directly estimable (involve no other parameters). For example, suppose an effect coded CLASS variable A has four levels. Then there are three parameters ($\alpha_1, \alpha_2, \alpha_3$) representing the first three levels, and the fourth parameter is represented by

$$-\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\beta} = 0$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast ’1 vs. 4’ A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast ’1&2 vs. 3’ A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example,

```
contrast ’1 vs. 2    ’  A  1 -1  0;
contrast ’1&2 vs. 4  ’  A  3  3  2;
contrast ’1&2 vs. 3&4’  A  2  2  0;
contrast ’Main Effect’  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability. If PROC TPHREG finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC TPHREG handles missing level combinations of categorical variables in the same manner as PROC GLM. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the $\mathbf{L}$ matrix in your CONTRAST statement. If the elements of $\mathbf{L}$ are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the $\mathbf{L}$ matrix contains nonzero terms for both A and A*B, since A*B contains A.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement, that is, the rank of $\mathbf{L}$.

You can specify the following options after a slash (/).

ALPHA= *p*

specifies the level of significance p for the $100(1-p)$% confidence interval for each contrast when the ESTIMATE option is specified. The value p must be between 0 and 1. By default, p is equal to the value of the ALPHA= option in the PROC TPHREG statement, or 0.05 if that option is not specified.

E

requests that the $\mathbf{L}$ matrix be displayed.

ESTIMATE=*keyword*

requests that each individual contrast (that is, each row, $l_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$) or exponentiated contrast ($\mathrm{e}^{l_i'\boldsymbol{\beta}}$) be estimated and tested. PROC TPHREG displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the contrast or the exponentiated contrast ($\mathrm{e}^{l_i'\boldsymbol{\beta}}$), or both, by specifying one of the following *keywords*:

PARM	specifies that the contrast itself be estimated
EXP	specifies that the exponentiated contrast be estimated
BOTH	specifies that both the contrast and the exponentiated contrast be estimated

SINGULAR = *number*

tunes the estimability check. This option is ignored when the full-rank parameterization is used. If $\boldsymbol{v}$ is a vector, define ABS($\boldsymbol{v}$) to be the largest absolute value of the elements of $\boldsymbol{v}$. For a row vector $\boldsymbol{l}'$ of the contrast matrix $\mathbf{L}$, define c to be equal to ABS$(\boldsymbol{l})$ if ABS$(\boldsymbol{l})$ is greater than 0; otherwise, c equals 1. If ABS$(\boldsymbol{l}' - \boldsymbol{l}'\boldsymbol{T})$ is greater than $c * number$, then $\boldsymbol{l}$ is declared nonestimable. The $\boldsymbol{T}$ matrix is the Hermite form matrix $\boldsymbol{I}_0^{-}\boldsymbol{I}_0$, where $\boldsymbol{I}_0^{-}$ represents a generalized inverse of the information matrix $\boldsymbol{I}_0$

of the null model. The value for *number* must be between 0 and 1; the default value is 1E−4.

Details

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of −1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects using the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding

	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects using the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects using the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, . . . according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects using the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	**AOPOLY1**	**AOPOLY2**	**AOPOLY5**
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	**AOREF1**	**AOREF2**	**AOREF3**
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Miscellaneous Changes from PROC PHREG

The default method of computing the survivor function estimate is METHOD=CH which is based on the empirical cumulative hazard function estimate rather than the product-limit estimate (METHOD=PL) as in PROC PHREG. This applies to both the OUTPUT and BASELINE statements.

The OUT= data set in the OUTPUT statement contains the entire input data set along with statistics you request using the *keyword=name* options. Observations in the OUT= data set follow the same order as the input data set. The ORDER=SORTED option in the OUTPUT statement of PROC PHREG (see the “OUTPUT Statement” section (page 3233) in Chapter 54, “The PHREG Procedure,”) is no longer available.

The BASELINE statement in PROC PHREG enables you to to predict the cumulative mean function (CMF) and the cumulative hazard function (CUMHAZ) for recurrent events models. However, such features are not yet available in the BASELINE statement of PROC TPHREG.

Displayed Output

If you use the NOPRINT option in the PROC TPHREG statement, the procedure does not display any output. Otherwise, the displayed output of the TPHREG procedure includes the following:

- the “Model Information” table, which contains
  - the two-level name of the input data set
  - the name and label of the failure-time variable
  - if you specify the censoring variable,
    - the name and label of the censoring variable
    - the values that the censoring variable assumes to indicate censored times
  - if you use the OFFSET= option in the MODEL statement, the name and label of the offset variable
  - if you specify the FREQ statement, the name and label of the frequency variable
  - if you specify the WEIGHT statement, the name and label of the weight variable
  - the method of handling ties in the failure time
- the “Class Level Information” table, which shows the levels and the corresponding design variables for each CLASS explanatory variable
- the “Summary of the Number of Event and Censored Values” table, which gives, for each stratum, the breakdown of the number of events and censored values. This table is not produced if the NOSUMMARY option is specified.
- if you specify the SIMPLE option in the PROC TPHREG statement, the “Descriptive Statistics for Continuous Explanatory Variables” table for continuous explanatory variables, and the “Frequency Distribution of CLASS Variables” table. The “Descriptive Statistics for Continuous Explanatory Variables” table contains the mean, standard deviation, maximum and minimum of each continuous variable specified in the MODEL statement. If the WEIGHT statement is specified, the “Frequency Distribution of Class Variables” table also contains the weight distributions of the CLASS variables.
- if you specify the ITPRINT option in the MODEL statement, the “Iteration History” table, which displays the iteration number, step size, log likelihood, and parameter estimates at each iteration The last evaluation of the gradient vector is also displayed.
- the “Model Fit Statistics” table, which gives the values of -2 log likelihood for fitting a model with no explanatory variable and for fitting a model with all the explanatory variables. The AIC and SBC are also given in this table.
- the “Testing Global Null Hypothesis: BETA=0” table, which displays results of the likelihood ratio test, the score test, and the Wald test

- if the model contains an effect involving a CLASS variable, the “Type 3 Tests” table, which gives the Wald chi-square statistic, the degrees of freedom, and the p-value for each effect in the model
- the “Analysis of Maximum Likelihood Estimates” table, which contains the following:
  - the maximum likelihood estimate of the parameter
  - the estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
  - if you specify the COVS option in the PROC statement, the ratio of the robust standard error estimate to the model-based standard error estimate
  - the Wald Chi-Square statistic, computed as the square of the parameter estimate divided by its standard error estimate
  - the degrees of freedom of the Wald chi-square statistic. It has a value of 1 unless the corresponding parameter is redundant or infinite, in which case the value is 0.
  - the *p*-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom
  - the hazards ratio estimate computed by exponentiating the parameter estimate
  - if you specified the RISKLIMITS option in the MODEL statement, the confidence limits for the hazards ratio
- if you specify SELECTION=SCORE in the MODEL statement, the “Regression Models Selected by Score Criterion” table, which gives the number of explanatory variables in each model, the score chi-square statistic, and the names of the variables included in the model
- if you use the FORWARD or STEPWISE selection method and specify the DETAILS option in the MODEL statement, the “Effects to Enter” table, which gives the score chi-square statistic for testing the significance of each candidate effect for entry (after adjusting for the effects already in the model), the degrees of freedom of the score chi-square statistic, and the corresponding *p*-value. This table is produced before an effect is selected for entry.
- if you use the BACKWARD or STEPWISE selection method and specify the DETAILS option in the MODEL statement, the “Effects to Remove” table, which gives the Wald chi-square statistic for testing the significance of each candidate effect for removal, the degrees of freedom of the Wald chi-square, and the corresponding *p*-value. This table is produced before an effect is selected for removal.
- if you use the BACKWARD, FORWARD, or STEPWISE selection method, a table summarizing the model-building process, which gives the step number, the effect entered or removed at each step, the chi-square statistic, and the corresponding *p*-value on which the selection is based
- if you use the COVB option in the MODEL statement, the estimated covariance matrix of the parameter estimates

- if you use the CORRB option in the MODEL statement, the estimated correlation matrix of the parameter estimates
- if you specify a CONTRAST statement, the "Contrast Test Results" table, which gives the result of the Wald test for each CONTRAST specified. If you specify the E option in the CONTRAST statement, then the contrast matrix is displayed. If you specify the ESTIMATE= option in the CONTRAST statement, the "Contrast Rows Estimation and Testing Results" table is produced, which includes the point estimate and confidence interval for each row of the contrast matrix, and the corresponding Wald test as well.
- if you specify a TEST statement,
  - the "Linear Coefficients" table, which gives the coefficients and constants of the linear hypothesis (if the E option is specified)
  - the printing of the intermediate calculations of the Wald test (if the option PRINT is specified)
  - the "Test Results" table, which gives the Wald chi-square statistic, the degrees of freedom, and the *p*-value
  - the "Average Effect" table, which gives the weighted average of the parameter estimates for the variables in the TEST statement, the estimated standard error, the z-score, and the *p*-value (if the AVERAGE option is specified)

ODS Table Names

PROC TPHREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

Table 73.1. ODS Tables Produced in PROC TPHREG

ODS Table Name	Description	Statement	Option
BestSubsets	Best subset selection	MODEL	SELECTION=SCORE
CensoredSummary	Summary of event and censored observations	MODEL	default
ClassLevelFreq	Frequency breakdown of CLASS variables	PROC	SIMPLE (with CLASS vars)
ClassLevelInfo	CLASS variable levels and design variables	MODEL	default (with CLASS vars)
ClassWgt	Weight breakdown of CLASS variables	WEIGHT	SIMPLE (with CLASS vars)
ContrastCoeff	L matrix for contrasts	CONTRAST	E
ContrastEstimate	Individual contrast estimates	CONTRAST	ESTIMATE=
ContrastTest	Wald test for contrasts	CONTRAST	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB

Table 73.1. (continued)

ODS Table Name	Description	Statement	Option
EffectsToEnter	Eligible effects for entry to model	MODEL	SELECTION=F/S
EffectsToRemove	Eligible effects for removal from model	MODEL	SELECTION=B/S
FitStatistics	Model fit statistics	MODEL	default
GlobalScore	Global chi-square test	MODEL	NOFIT
GlobalTests	Tests of the global null hypothesis	MODEL	default
IterHistory	Iteration history	MODEL	ITPRINT
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
ModelBuildingSummary	Summary of model building	MODEL	SELECTION=B/F/S
ModelInfo	Model information	PROC	default
NObs	Number of observations		default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
ResidualChiSq	Residual chi-square	MODEL	SELECTION=F/B
SimpleStatistics	Summary statistics for continuous explanatory variables	PROC	SIMPLE
TestAverage	Average Effect for test	TEST	AVERAGE
TestCoeff	coefficients for linear hypotheses	TEST	E
TestPrint1	**L**[cov(**b**)]**L**’ and **Lb-c**	TEST	PRINT
TestPrint2	Ginv(**L**[cov(**b**)]**L**’) and Ginv(**L**[cov(**b**)]**L**’)(**Lb-c**)	TEST	PRINT
TestStmts	Linear Hypotheses Test Results	TEST	default
Type3	Type 3 tests of effects	MODEL	default (with CLASS vars)

Example

Example 73.1. Analysis of the VA Lung Cancer Data

This example uses the Veteran’s Administration lung cancer data presented in Appendix 1 of Kalbfleisch and Prentice (1980). In this trial, males with advanced inoperable lung cancer were randomized to a standard therapy and a test chemotherapy. The primary end point for the therapy comparison was time to death in days, represented by the variable Time. Negative values of Time are censored values. The data include information on a number of explanatory variables: Therapy (type of therapy: standard or test), Cell (type of tumor cell: adeno, large, small, or squamous), Prior (prior therapy: 0=no, 10=yes), Age (age in years), Duration (months from diagnosis to randomization), and Kps (Karnofsky performance scale). A censoring indicator variable Censor is created from the data, with value 1 indicating a censored time and value 0 an event time.

```
proc format;
   value yesno 0='no' 10='yes';
   run;

data VALung;
   drop check m;
   retain Therapy Cell;
   infile cards column=column;
   length Check $ 1;
   label Time='time to death in days'
         Kps='Karnofsky performance scale'
         Duration='months from diagnosis to randomization'
         Age='age in years'
         Prior='prior therapy'
         Cell='cell type'
         Therapy='type of treatment';
   format Prior yesno.;
   M=Column;
   input Check $ @@;
   if M>Column then M=1;
   if Check='s'|Check='t' then do;
      input @M Therapy $ Cell $;
      delete;
      end;
   else do;
      input @M Time Kps Duration Age Prior @@;
      censor=(Time<0);
      Time=abs(Time);
      end;
   datalines;
standard squamous
 72 60  7 69  0   411 70  5 64 10   228 60  3 38  0   126 60  9 63 10
118 70 11 65 10    10 20  5 49  0    82 40 10 69 10   110 80 29 68  0
314 50 18 43  0  -100 70  6 70  0    42 60  4 81  0     8 40 58 63 10
144 30  4 63  0   -25 80  9 52 10    11 70 11 48 10
standard small
 30 60  3 61  0   384 60  9 42  0     4 40  2 35  0    54 80  4 63 10
 13 60  4 56  0  -123 40  3 55  0   -97 60  5 67  0   153 60 14 63 10
 59 30  2 65  0   117 80  3 46  0    16 30  4 53 10   151 50 12 69  0
 22 60  4 68  0    56 80 12 43 10    21 40  2 55 10    18 20 15 42  0
139 80  2 64  0    20 30  5 65  0    31 75  3 65  0    52 70  2 55  0
287 60 25 66 10    18 30  4 60  0    51 60  1 67  0   122 80 28 53  0
 27 60  8 62  0    54 70  1 67  0     7 50  7 72  0    63 50 11 48  0
392 40  4 68  0    10 40 23 67 10
standard adeno
  8 20 19 61 10    92 70 10 60  0    35 40  6 62  0   117 80  2 38  0
132 80  5 50  0    12 50  4 63 10   162 80  5 64  0     3 30  3 43  0
 95 80  4 34  0
standard large
177 50 16 66 10   162 80  5 62  0   216 50 15 52  0   553 70  2 47  0
278 60 12 63  0    12 40 12 68 10   260 80  5 45  0   200 80 12 41 10
156 70  2 66  0  -182 90  2 62  0   143 90  8 60  0   105 80 11 66  0
103 80  5 38  0   250 70  8 53 10   100 60 13 37 10
test squamous
999 90 12 54 10   112 80  6 60  0   -87 80  3 48  0  -231 50  8 52 10
242 50  1 70  0   991 70  7 50 10   111 70  3 62  0     1 20 21 65 10
587 60  3 58  0   389 90  2 62  0    33 30  6 64  0    25 20 36 63  0
357 70 13 58  0   467 90  2 64  0   201 80 28 52 10     1 50  7 35  0
 30 70 11 63  0    44 60 13 70 10   283 90  2 51  0    15 50 13 40 10
```

```
test small
 25 30  2 69  0  -103 70 22 36 10    21 20  4 71  0    13 30  2 62  0
 87 60  2 60  0     2 40 36 44 10    20 30  9 54 10     7 20 11 66  0
 24 60  8 49  0    99 70  3 72  0     8 80  2 68  0    99 85  4 62  0
 61 70  2 71  0    25 70  2 70  0    95 70  1 61  0    80 50 17 71  0
 51 30 87 59 10    29 40  8 67  0
test adeno
 24 40  2 60  0    18 40  5 69 10   -83 99  3 57  0    31 80  3 39  0
 51 60  5 62  0    90 60 22 50 10    52 60  3 43  0    73 60  3 70  0
  8 50  5 66  0    36 70  8 61  0    48 10  4 81  0     7 40  4 58  0
140 70  3 63  0   186 90  3 60  0    84 80  4 62 10    19 50 10 42  0
 45 40  3 69  0    80 40  4 63  0
test large
 52 60  4 45  0   164 70 15 68 10    19 30  4 39 10    53 60 12 66  0
 15 30  5 63  0    43 60 11 49 10   340 80 10 64 10   133 75  1 65  0
111 60  5 64  0   231 70 18 67 10   378 80  4 65  0    49 30  3 37  0
;
```

PROC TPHREG is invoked to fit the Cox proportional hazards model to these data. Variables Prior, Cell, and Therapy, which are categorical variables, are declared in the CLASS statement. By default, PROC TPHREG parameterizes the CLASS variables using the reference coding with the last category as the reference category. However, you can explicitly specify the reference category of your choice. Here, Prior=no is chosen as the reference category for prior therapy, Cell=large is chosen as the reference category for type of tumor cell, and Therapy=standard is chosen as the reference category for the type of therapy. Both the continuous explanatory variables (Kps, Duration, and Age) and the CLASS explanatory variables (Prior, Cell, and Therapy) are specified in the MODEL statement. Knowing how the Cell variable is parameterized, the hazards ratios of all pairs of cell-type groups can be estimated using the ESTIMATE=EXP option in a CONTRAST statement.

```
proc tphreg data=VALung;
   class Prior(ref=’no’) Cell(ref=’large’) Therapy(ref=’standard’);
   model Time*censor(1) = Kps Duration Age Prior Cell Therapy;
   contrast ’Pairwise’ cell 1  0  0, /* adeno vs large    */
                       cell 0  1  0, /* small vs large    */
                       cell 0  0  1, /* squamous vs large */
                       cell 1 -1  0, /* adeno vs small    */
                       cell 1  0 -1, /* adeno vs squamous */
                       cell 0  1 -1  /* small vs squamous */
                       / estimate=exp;
   run;
```

The output of PROC TPHREG is very similar to that of PROC PHREG, with additional tables for displaying the parameterization of the CLASS variables, the multiparameter tests for the model effects, and the analysis results of the specified contrasts.

Output 73.1.1. Reference Coding of CLASS Variables

```
                         The TPHREG Procedure

                       Class Level Information

              Class       Value        Design Variables

              Prior       no             0
                          yes            1

              Cell        adeno          1      0      0
                          large          0      0      0
                          small          0      1      0
                          squamous       0      0      1

              Therapy     standard       0
                          test           1
```

Coding of the CLASS variables is displayed in Output 73.1.1. There is one dummy variable for Prior and one for Therapy, since both variables are binary. The dummy variable has a value of 0 for the reference category (Prior=no, Therapy=standard). The CLASS variable Cell has four categories and are represented by three dummy variables. Note that the reference category, Cell=large, has a value of 0 for all three dummy variables.

Output 73.1.2. Wald Tests for Individual Model Effects

```
                              Type 3 Tests

                                         Wald
             Effect         DF     Chi-Square    Pr > ChiSq

             Kps             1        35.1124        <.0001
             Duration        1         0.0001        0.9920
             Age             1         0.8443        0.3582
             Prior           1         0.0971        0.7554
             Cell            3        17.9164        0.0005
             Therapy         1         1.9579        0.1617
```

The test results of individual model effects are shown in Output 73.1.2. There is a strong prognostic effect of the Karnofsky performance status on patient survival ($p < 0.0001$), and the survival times in the various cell-type groups differ significantly ($p = 0.0005$). However, there is a lack of evidence that the test chemotherapy differs from the standard therapy ($p = 0.1617$) after accounting for the prognostic effects of other variables.

Output 73.1.3. Inference about the Regression Parameters

```
                    Analysis of Maximum Likelihood Estimates

                                  Parameter      Standard
Parameter            DF      Estimate         Error    Chi-Square    Pr > ChiSq

Kps                   1      -0.03262       0.00551       35.1124        <.0001
Duration              1    -0.0000916       0.00913        0.0001        0.9920
Age                   1      -0.00855       0.00930        0.8443        0.3582
Prior      yes        1       0.07232       0.23213        0.0971        0.7554
Cell       adeno      1       0.78867       0.30267        6.7899        0.0092
Cell       small      1       0.45686       0.26627        2.9438        0.0862
Cell       squamous   1      -0.39963       0.28266        1.9988        0.1574
Therapy    test       1       0.28994       0.20721        1.9579        0.1617

                    Analysis of Maximum Likelihood Estimates

                                 Hazard
    Parameter                     Ratio    Variable Label

    Kps                           0.968    Karnofsky performance scale
    Duration                      1.000    months from diagnosis to randomization
    Age                           0.991    age in years
    Prior      yes                1.075    prior therapy yes
    Cell       adeno              2.200    cell type adeno
    Cell       small              1.579    cell type small
    Cell       squamous           0.671    cell type squamous
    Therapy    test               1.336    type of treatment test
```

In the Cox proportional hazards model, the effects of the covariates are to act multiplicatively on the hazard of the survival time, and therefore it is a little easier to interpret the corresponding hazards ratios than the regression parameters. For a parameter that corresponds to an continous variable, the hazard ratio is the ratio of hazard rates for a increase of one unit of the variable. From Output 73.1.3, the hazard ratio estimate for Kps is 0.968, meaning that an increase of 10 units in Karnofsky performance scale will shrink the hazard rate by $1 - (0.968)^{10}$=28%. For a CLASS variable parameter, the hazard ratio is the ratio of the hazard rates between the given category and the reference category. The hazard rate of Cell=adeno is 220% that of Cell=large, the hazard rate of Cell=small is 158% that of Cell=large, and the hazard rate of Cell=squamous is only 67% that of Cell=large.

Output 73.1.4. Overall Test for All Paired Cell-type Groups

```
                              Contrast Test Results

                                               Wald
                  Contrast      DF     Chi-Square     Pr > ChiSq

                  Pairwise       3        17.9164         0.0005
```

Although there are six pairwise comparisons for the four types of tumor cells in the CONTRAST statement, the overall test has only 3 degrees of freedom (Output

73.1.4). In fact this is the very same testing of no prognostics effect between the cell-type groups as shown in Output 73.1.2.

Output 73.1.5. Hazards Ratios for All Paired Cell-type Groups

```
                  Contrast Rows Estimation and Testing Results

                                          Standard
Contrast   Type        Row   Estimate        Error    Alpha    Confidence Limits

Pairwise   EXP           1     2.2005       0.6660     0.05     1.2159      3.9824
Pairwise   EXP           2     1.5791       0.4205     0.05     0.9370      2.6611
Pairwise   EXP           3     0.6706       0.1895     0.05     0.3853      1.1669
Pairwise   EXP           4     1.3935       0.3840     0.05     0.8119      2.3916
Pairwise   EXP           5     3.2815       0.9870     0.05     1.8200      5.9167
Pairwise   EXP           6     2.3549       0.6480     0.05     1.3732      4.0384

                  Contrast Rows Estimation and Testing Results

                                                  Wald
             Contrast   Type        Row     Chi-Square    Pr > ChiSq

             Pairwise   EXP           1         6.7899        0.0092
             Pairwise   EXP           2         2.9438        0.0862
             Pairwise   EXP           3         1.9988        0.1574
             Pairwise   EXP           4         1.4497        0.2286
             Pairwise   EXP           5        15.6101        <.0001
             Pairwise   EXP           6         9.6866        0.0019
```

Output 73.1.5 is generated by the ESTIMATE=EXP option in the CONTRAST statement. Values of the Estimate column are the estimated hazard ratios: 2.200 for ‘adeno’ vs ‘large’, 1.579 for ‘small’ versus ‘large’, 0.671 for ‘squamous’ versus ‘large’, 1.394 for ‘adeno’ versus ’small’, 3.282 for ‘adeno’ versus ‘squamous’, and 2.355 for ‘small’ versus ‘squamous’. Note that the first three hazard ratio estimates are already given in the parameter estimate table (Output 73.1.3), and therefore there is no need to specify those first three rows in the CONTRAST statement.

References

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons, Inc.

Chapter 74
The TPSPLINE Procedure

Chapter Contents

Chapter 74
The TPSPLINE Procedure

Overview

The TPSPLINE procedure uses the penalized least squares method to fit a nonparametric regression model. It computes thin-plate smoothing splines to approximate smooth multivariate functions observed with noise. The TPSPLINE procedure allows great flexibility in the possible form of the regression surface. In particular, PROC TPSPLINE makes no assumptions of a parametric form for the model. The generalized cross validation (GCV) function may be used to select the amount of smoothing.

The TPSPLINE procedure complements the methods provided by the standard SAS regression procedures such as the GLM, REG and NLIN procedures. These procedures can handle most situations in which you specify the regression model and the model is known up to a fixed number of parameters. However, when you have no prior knowledge about the model, or when you know that the data cannot be represented by a model with a fixed number of parameters, you can use the TPSPLINE procedure to model the data.

The TPSPLINE procedure uses the penalized least squares method to fit the data with a flexible model in which the number of effective parameters can be as large as the number of unique design points. Hence, as the sample size increases, the model space increases as well, enabling the thin-plate smoothing spline to fit more complicated situations.

The main features of the TPSPLINE procedure are as follows:

- provides penalized least squares estimates
- supports the use of multidimensional data
- supports multiple SCORE statements
- fits both semiparametric models and nonparametric models
- provides options for handling large data sets
- supports multiple dependent variables
- enables you to choose a particular model by specifying the model degrees of freedom or smoothing parameter

The Penalized Least Squares Estimate

Penalized least squares estimates provide a way to balance fitting the data closely and avoiding excessive roughness or rapid variation. A penalized least squares estimate is a surface that minimizes the penalized least squares over the class of all surfaces satisfying sufficient regularity conditions.

Define $\mathbf{x}_i$ as a d-dimensional covariate vector, $\mathbf{z}_i$ as a p-dimensional covariate vector, and y_i as the observation associated with $(\mathbf{x}_i, \mathbf{z}_i)$. Assuming that the relation between $\mathbf{z}_i$ and y_i is linear but the relation between $\mathbf{x}_i$ and y_i is unknown, you can fit the data using a semiparametric model as follows:

$$y_i = f(\mathbf{x}_i) + \mathbf{z}_i\beta + \epsilon_i$$

where f is an unknown function that is assumed to be reasonably smooth, $\epsilon_i, i = 1, \cdots, n$ are independent, zero-mean random errors, and β is a p-dimensional unknown parametric vector.

This model consists of two parts. The $\mathbf{z}_i\beta$ is the parametric part of the model, and the $\mathbf{z}_i$ are the regression variables. The $f(\mathbf{x}_i)$ is the nonparametric part of the model, and the $\mathbf{x}_i$ are the smoothing variables.

The ordinary least squares method estimates $f(x_i)$ and β by minimizing the quantity:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\beta)^2$$

However, the functional space of $f(\mathbf{x})$ is so large that you can always find a function f that interpolates the data points. In order to obtain an estimate that fits the data well and has some degree of smoothness, you can use the penalized least squares method.

The penalized least squares function is defined as

$$S_\lambda(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\beta)^2 + \lambda J_2(f)$$

where $J_2(f)$ is the penalty on the roughness of f and is defined, in most cases, as the integral of the square of the second derivative of f.

The first term measures the goodness of fit and the second term measures the smoothness associated with f. The λ term is the smoothing parameter, which governs the tradeoff between smoothness and goodness of fit. When λ is large, it more heavily penalizes rougher fits. Conversely, a small value of λ puts more emphasis on the goodness of fit.

The estimate f_λ is selected from a reproducing kernel Hilbert space, and it can be represented as a linear combination of a sequence of basis functions. Hence, the final estimates of f can be written as

$$f_\lambda(\mathbf{x}_i) = \theta_0 + \sum_{j=1}^{d}\theta_j x_{ij} + \sum_{j=1}^{n}\delta_j B_j(\mathbf{x}_i)$$

where B_j is the basis function, which depends on where the data $\mathbf{x}_j$ is located, and θ_j and δ_j are the coefficients that need to be estimated.

For a fixed λ, the coefficients (θ, δ, β) can be estimated by solving an $n \times n$ system.

The smoothing parameter can be chosen by minimizing the generalized cross validation (GCV) function.

If you write

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y}$$

then $\mathbf{A}(\lambda)$ is referred to as the *hat* or *smoothing* matrix, and the GCV function $V(\lambda)$ is defined as

$$V(\lambda) = \frac{(1/n)||(\mathbf{I} - \mathbf{A}(\lambda))\mathbf{y}||^2}{[(1/n)tr(\mathbf{I} - \mathbf{A}(\lambda))]^2}$$

PROC TPSPLINE with Large Data Sets

The calculation of the penalized least squares estimate is computationally intensive. The amount of memory and CPU time needed for the analysis depend on the number of unique design points, which corresponds to the number of unknown parameters to be estimated.

You can specify the D= *value* option in the MODEL statement to reduce the number of unknown parameters. The option groups design points by the specified range (see the D= option on page 4509).

PROC TPSPLINE selects one design point from the group and treats all observations in the group as replicates of that design point. Calculation of the thin-plate smoothing spline estimates are based on the reprocessed data. The way to choose the design point from a group depends on the order of the data. Therefore, different orders of input data may result in different estimates.

This option, by combining several design points into one, reduces the number of unique design points, thereby approximating the original data. The D= value you specify determines the width of the range used to group the data.

Getting Started

The following example demonstrates how you can use the TPSPLINE procedure to fit a semiparametric model.

Suppose that y is a continuous variable and x1 and x2 are two explanatory variables of interest. To fit a smoothing spline model, you can use a MODEL statement similar to that used in many regression procedures in the SAS System.

```
proc tpspline;
   model y = (x1 x2);
run;
```

The TPSPLINE procedure can fit semiparametric models; the parentheses in the preceding MODEL statement separates the smoothing variables from the regression variables. The following statements illustrates this syntax.

```
   proc tpspline;
      model y = x3 (x1 x2);
   run;
```

This model assumes a linear relation with x3 and an unknown functional relation with x1 and x2.

If you want to fit several responses using the same explanatory variables, you can save computation time by using the multiple responses feature in the MODEL statement. For example, if y1 and y2 are two response variables, the following MODEL statement can be used to fit two models. Separate analyses are then performed for each response variable.

```
   proc tpspline;
      model y1 y2 = (x1 x2);
   run;
```

The following example illustrates the use of PROC TPSPLINE. The data are from Bates, Lindstrom, Wahba, and Yandell (1987).

```
   data Measure;
      input x1 x2 y @@;
      datalines;
      -1.0 -1.0   15.54483570   -1.0 -1.0    15.76312613
       -.5 -1.0   18.67397826    -.5 -1.0    18.49722167
        .0 -1.0   19.66086310     .0 -1.0    19.80231311
        .5 -1.0   18.59838649     .5 -1.0    18.51904737
       1.0 -1.0   15.86842815    1.0 -1.0    16.03913832
      -1.0  -.5   10.92383867   -1.0  -.5    11.14066546
       -.5  -.5   14.81392847    -.5  -.5    14.82830425
        .0  -.5   16.56449698     .0  -.5    16.44307297
        .5  -.5   14.90792284     .5  -.5    15.05653924
       1.0  -.5   10.91956264    1.0  -.5    10.94227538
      -1.0   .0    9.61492010   -1.0   .0     9.64648093
       -.5   .0   14.03133439    -.5   .0    14.03122345
        .0   .0   15.77400253     .0   .0    16.00412514
        .5   .0   13.99627680     .5   .0    14.02826553
       1.0   .0    9.55700164    1.0   .0     9.58467047
      -1.0   .5   11.20625177   -1.0   .5    11.08651907
       -.5   .5   14.83723493    -.5   .5    14.99369172
        .0   .5   16.55494349     .0   .5    16.51294369
        .5   .5   14.98448603     .5   .5    14.71816070
       1.0   .5   11.14575565    1.0   .5    11.17168689
      -1.0  1.0   15.82595514   -1.0  1.0    15.96022497
       -.5  1.0   18.64014953    -.5  1.0    18.56095997
        .0  1.0   19.54375504     .0  1.0    19.80902641
        .5  1.0   18.56884576     .5  1.0    18.61010439
       1.0  1.0   15.86586951    1.0  1.0    15.90136745
      ;
```

The data set Measure contains three variables x1, x2, and y. Suppose that you want to fit a surface by using the variables x1 and x2 to model the response y. The variables x1 and x2 are spaced evenly on a $[-1 \times 1] \times [-1 \times 1]$ square, and the response y is generated by adding a random error to a function $f(x1, x2)$. The raw data are plotted using the G3D procedure. In order to plot those replicates, the data are jittered a little bit.

```
data Measure1;
   set Measure;
run;

proc sort data=Measure1;
   by x2 x1;
run;

data measure1;
   set measure1; by x1;
   if last.x1 then x1=x1+0.00001;
run;

proc g3d data=Measure1;
   scatter x2*x1=y /size=.5
                    zmin=9 zmax=21
                    zticknum=4;
title "Raw Data";
run;
```

Figure 74.1 displays the raw data.

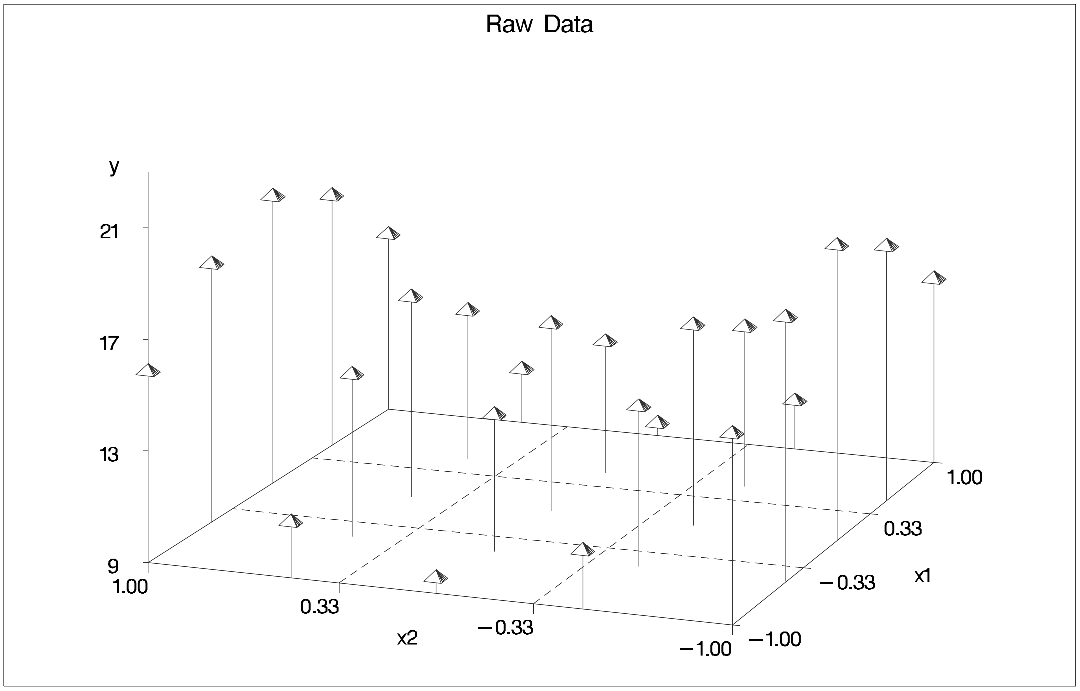


Figure 74.1. Plot of Data Set MEASURE

The following statements invoke the TPSPLINE procedure, using the Measure data set as input. In the MODEL statement, the x1 and x2 variables are listed as smoothing variables. The LOGNLAMBDA= option returns a list of GCV values with $log_{10}(n\lambda)$ ranging from -4 to -2. The OUTPUT statement creates the data set estimate to contain the predicted values and the 95% upper and lower confidence limits.

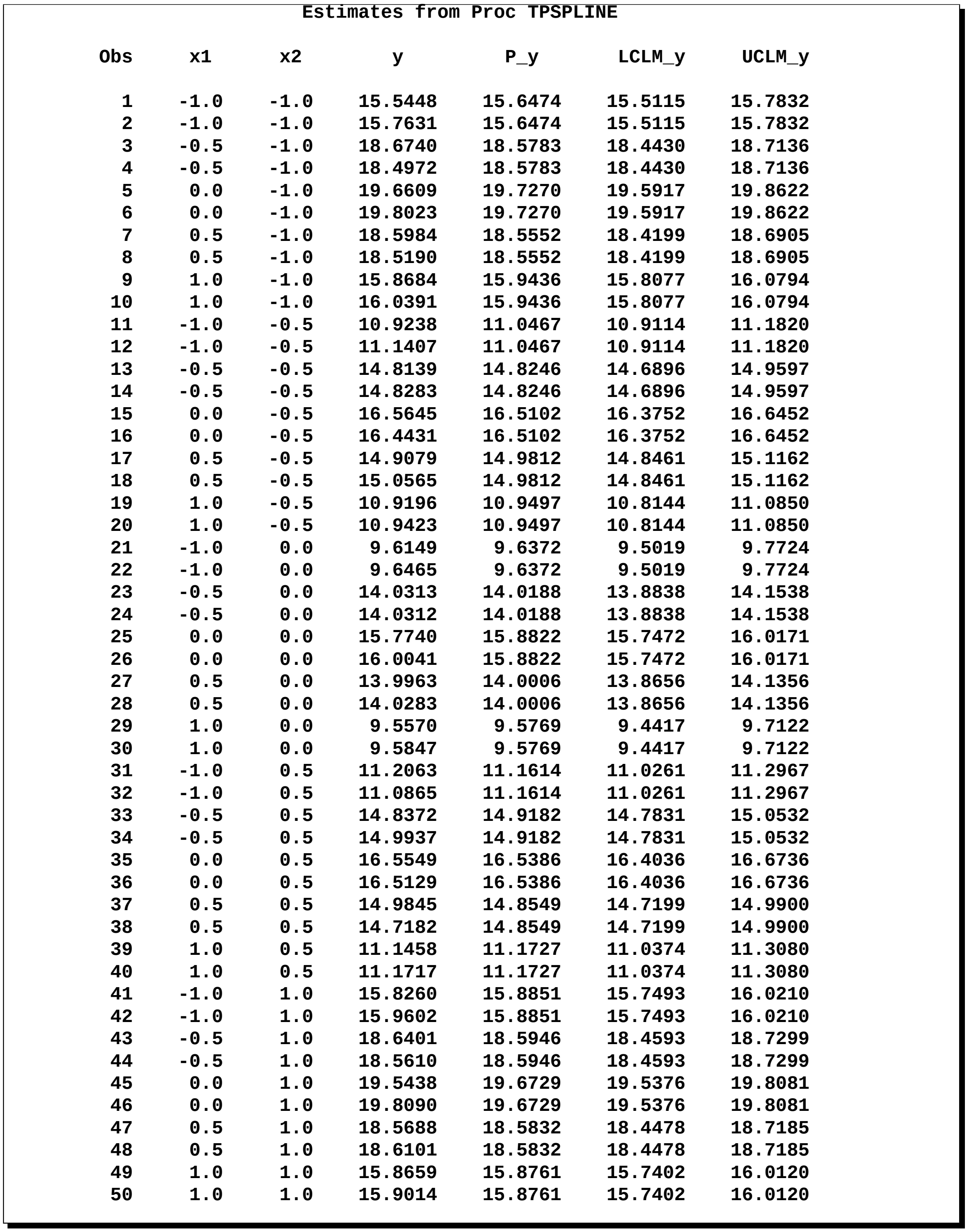
```
proc tpspline data=Measure;
   model y=(x1 x2) /lognlambda=(-4 to -2 by 0.1);
   output out=estimate pred uclm lclm;
run;

proc print data=estimate;
run;
```

The results of this analysis are displayed in the following figures. Figure 74.2 shows that the data set Measure contains 50 observations with 25 unique design points. The GCV values are listed along with the log_{10} of $n\lambda$. The value of $log_{10}(n\lambda)$ that minimizes the GCV function is around -3.5. The final thin-plate smoothing spline estimate is based on LOGNLAMBDA = –3.4762. The residual sum of squares is 0.246110, and the degrees of freedom is 24.593203. The standard deviation, defined as RSS/(Tr(I-A)), is 0.098421. The predictions and 95% confidence limits are displayed in Figure 74.3.

```
                             The TPSPLINE Procedure
                           Dependent Variable:  y

                           Summary of Input Data Set

                    Number of Non-Missing Observations    50
                    Number of Missing Observations         0
                    Unique Smoothing Design Points        25


                            Summary of Final Model

                    Number of Regression Variables        0
                    Number of Smoothing Variables         2
                    Order of Derivative in the Penalty    2
                    Dimension of Polynomial Space         3
```

Figure 74.2. Output from PROC TPSPLINE

```
                    The TPSPLINE Procedure
                  Dependent Variable:  y

                         GCV Function

               log10(n*Lambda)             GCV

                     -4.000000        0.019215
                     -3.900000        0.019183
                     -3.800000        0.019148
                     -3.700000        0.019113
                     -3.600000        0.019082
                     -3.500000        0.019064*
                     -3.400000        0.019074
                     -3.300000        0.019135
                     -3.200000        0.019286
                     -3.100000        0.019584
                     -3.000000        0.020117
                     -2.900000        0.021015
                     -2.800000        0.022462
                     -2.700000        0.024718
                     -2.600000        0.028132
                     -2.500000        0.033165
                     -2.400000        0.040411
                     -2.300000        0.050614
                     -2.200000        0.064699
                     -2.100000        0.083813
                     -2.000000        0.109387

             Note: * indicates minimum GCV value.


                      Summary Statistics
                     of Final Estimation

              log10(n*Lambda)            -3.4762
              Smoothing Penalty        2558.1432
              Residual SS                 0.2461
              Tr(I-A)                    25.4068
              Model DF                   24.5932
              Standard Deviation          0.0984
```



```
                          Estimates from Proc TPSPLINE

       Obs     x1      x2         y          P_y       LCLM_y      UCLM_y

         1    -1.0    -1.0    15.5448    15.6474    15.5115    15.7832
         2    -1.0    -1.0    15.7631    15.6474    15.5115    15.7832
         3    -0.5    -1.0    18.6740    18.5783    18.4430    18.7136
         4    -0.5    -1.0    18.4972    18.5783    18.4430    18.7136
         5     0.0    -1.0    19.6609    19.7270    19.5917    19.8622
         6     0.0    -1.0    19.8023    19.7270    19.5917    19.8622
         7     0.5    -1.0    18.5984    18.5552    18.4199    18.6905
         8     0.5    -1.0    18.5190    18.5552    18.4199    18.6905
         9     1.0    -1.0    15.8684    15.9436    15.8077    16.0794
        10     1.0    -1.0    16.0391    15.9436    15.8077    16.0794
        11    -1.0    -0.5    10.9238    11.0467    10.9114    11.1820
        12    -1.0    -0.5    11.1407    11.0467    10.9114    11.1820
        13    -0.5    -0.5    14.8139    14.8246    14.6896    14.9597
        14    -0.5    -0.5    14.8283    14.8246    14.6896    14.9597
        15     0.0    -0.5    16.5645    16.5102    16.3752    16.6452
        16     0.0    -0.5    16.4431    16.5102    16.3752    16.6452
        17     0.5    -0.5    14.9079    14.9812    14.8461    15.1162
        18     0.5    -0.5    15.0565    14.9812    14.8461    15.1162
        19     1.0    -0.5    10.9196    10.9497    10.8144    11.0850
        20     1.0    -0.5    10.9423    10.9497    10.8144    11.0850
        21    -1.0     0.0     9.6149     9.6372     9.5019     9.7724
        22    -1.0     0.0     9.6465     9.6372     9.5019     9.7724
        23    -0.5     0.0    14.0313    14.0188    13.8838    14.1538
        24    -0.5     0.0    14.0312    14.0188    13.8838    14.1538
        25     0.0     0.0    15.7740    15.8822    15.7472    16.0171
        26     0.0     0.0    16.0041    15.8822    15.7472    16.0171
        27     0.5     0.0    13.9963    14.0006    13.8656    14.1356
        28     0.5     0.0    14.0283    14.0006    13.8656    14.1356
        29     1.0     0.0     9.5570     9.5769     9.4417     9.7122
        30     1.0     0.0     9.5847     9.5769     9.4417     9.7122
        31    -1.0     0.5    11.2063    11.1614    11.0261    11.2967
        32    -1.0     0.5    11.0865    11.1614    11.0261    11.2967
        33    -0.5     0.5    14.8372    14.9182    14.7831    15.0532
        34    -0.5     0.5    14.9937    14.9182    14.7831    15.0532
        35     0.0     0.5    16.5549    16.5386    16.4036    16.6736
        36     0.0     0.5    16.5129    16.5386    16.4036    16.6736
        37     0.5     0.5    14.9845    14.8549    14.7199    14.9900
        38     0.5     0.5    14.7182    14.8549    14.7199    14.9900
        39     1.0     0.5    11.1458    11.1727    11.0374    11.3080
        40     1.0     0.5    11.1717    11.1727    11.0374    11.3080
        41    -1.0     1.0    15.8260    15.8851    15.7493    16.0210
        42    -1.0     1.0    15.9602    15.8851    15.7493    16.0210
        43    -0.5     1.0    18.6401    18.5946    18.4593    18.7299
        44    -0.5     1.0    18.5610    18.5946    18.4593    18.7299
        45     0.0     1.0    19.5438    19.6729    19.5376    19.8081
        46     0.0     1.0    19.8090    19.6729    19.5376    19.8081
        47     0.5     1.0    18.5688    18.5832    18.4478    18.7185
        48     0.5     1.0    18.6101    18.5832    18.4478    18.7185
        49     1.0     1.0    15.8659    15.8761    15.7402    16.0120
        50     1.0     1.0    15.9014    15.8761    15.7402    16.0120
```

Figure 74.3. Data Set ESTIMATE

The fitted surface is plotted with PROC G3D as follows.

```
proc g3d data=estimate;
   plot x2*x1=p_y/grid
                  zmin=9 zmax=21 zticknum=4;
title 'Plot of Fitted Surface';
run;
```

The resulting plot is displayed in Figure 74.4.

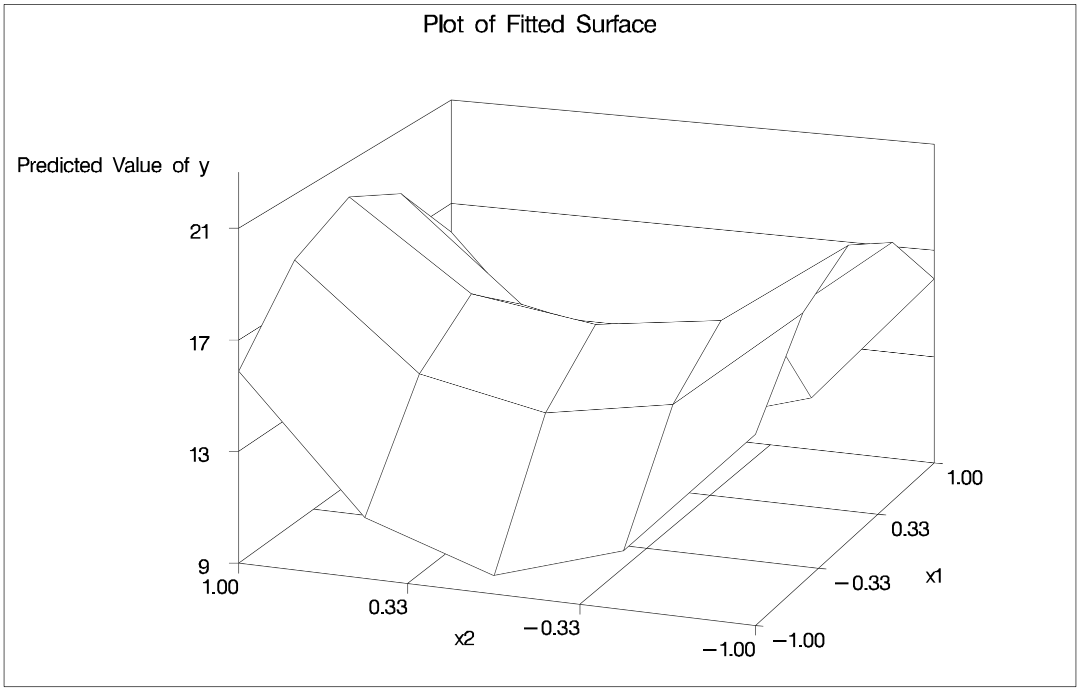


Figure 74.4. Plot of TPSPLINE Fit of Data Set Measure

Because the data in data set Measure are very sparse, the fitted surface is not smooth. To produce a smoother surface, the following statements generate the data set Pred in order to obtain a finer grid. The SCORE statement evaluates the fitted surface at those new design points.

```
data pred;
   do x1=-1 to 1 by 0.1;
      do x2=-1 to 1 by 0.1;
         output;
      end;
   end;
run;

proc tpspline data=measure;
   model y=(x1 x2)/lognlambda=(-4 to -2 by 0.1);
   score data=pred out=predy;
run;

proc g3d data=predy;
   plot x2*x1=p_y/grid
                  zmin=9 zmax=21 zticknum=4;
title 'Plot of Fitted Surface on a Fine Grid';
run;
```

The surface plot based on the finer grid is displayed in Figure 74.5. The plot shows that a parametric model with quadratic terms of x1 and x2 provides a reasonable fit to the data.

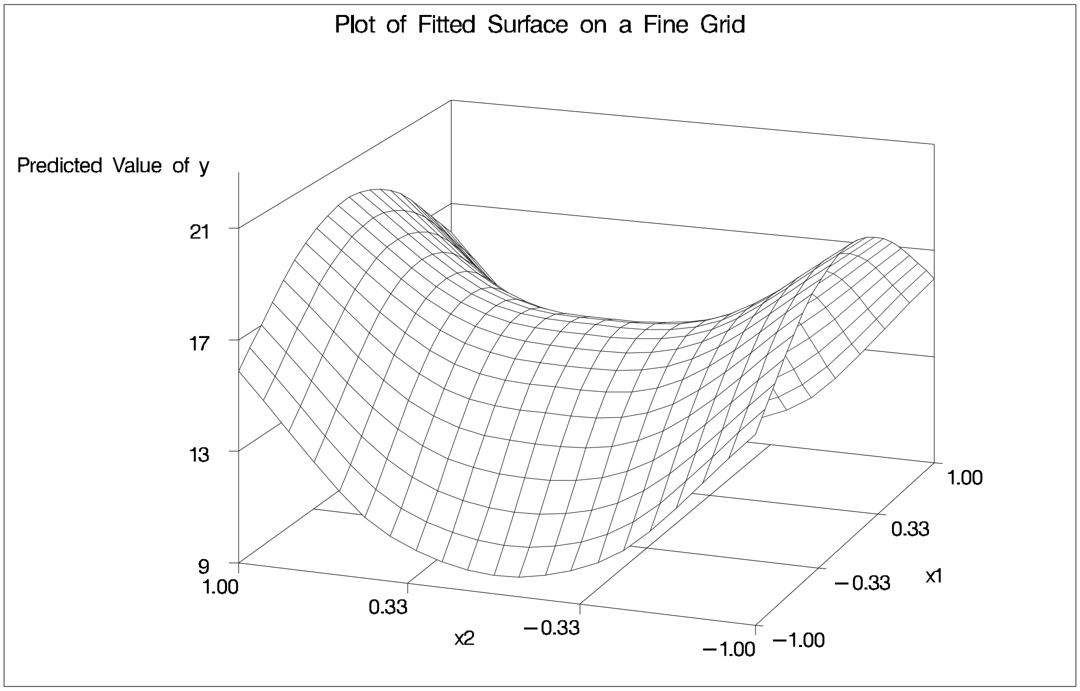


Figure 74.5. Plot of TPSPLINE fit

Syntax

PROC TPSPLINE < *option* > **;**
  MODEL *dependents* = < *variables* > *(variables)* < */options* > **;**
  SCORE *data=SAS-data-set out=SAS-data-set* **;**
  OUTPUT < *out=SAS-data-set* > *keyword* < · · · *keyword* > **;**
  BY *variables* **;**
  FREQ *variable* **;**
  ID *variables* **;**

The syntax in PROC TPSPLINE is similar to that of other regression procedures in the SAS System. The PROC TPSPLINE and MODEL statements are required. The SCORE statement can appear multiple times; all other statements appear only once.

The syntax for PROC TPSPLINE is described in the following sections in alphabetical order after the description of the PROC TPSPLINE statement.

PROC TPSPLINE Statement

PROC TPSPLINE < *option* > **;**

The PROC TPSPLINE statement invokes the procedure. You can specify the following option.

DATA=*SAS-data-set*
specifies the SAS data set to be read by PROC TPSPLINE. The default value is the most recently created data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TPSPLINE to obtain separate analysis on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TPSPLINE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC TPSPLINE treats the data as if each observation appears n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used.

ID Statement

ID *variables* **;**

The variables in the ID statement are copied from the input data set to the OUT= data set. If you omit the ID statement, only the variables used in the MODEL statement and requested statistics are included in the output data set.

MODEL Statement

MODEL *dependents* = $<$ *regression variables* $>$ *(smoothing variables)* $<$ */options* $>$ **;**

The MODEL statement specifies the dependent variables, the independent regression variables, which are listed with no parentheses, and the independent smoothing variables, which are listed inside parentheses.

The regression variables are optional. At least one smoothing variable is required, and it must be listed after the regression variables. No variables can be listed in both the regression variable list and the smoothing variable list.

If you specify more than one dependent variable, PROC TPSPLINE calculates a thin-plate smoothing spline estimate for each dependent variable, using the regression variables and smoothing variables specified on the right-hand side.

If you specify regression variables, PROC TPSPLINE fits a semiparametric model using the regression variables as the linear part of the model.

You can specify the following options in the MODEL statement.

ALPHA=*number*

specifies the significance level α of the confidence limits on the final thin-plate smoothing spline estimate when you request confidence limits to be included in the output data set. Specify *number* as a value between 0 and 1. The default value is 0.05. See the "OUTPUT Statement" section on page 4510 for more information on the OUTPUT statement.

DF=*number*

specifies the degrees of freedom of the thin-plate smoothing spline estimate, defined as

$$df = trace(\mathbf{A}(\lambda))$$

where $\mathbf{A}(\lambda)$ is the *hat* matrix. Specify *number* as a value between zero and the number of unique design points.

DISTANCE=*number*
D=*number*

defines a range such that if two data points $(\mathbf{x_i}, \mathbf{z_i})$ and $(\mathbf{x_j}, \mathbf{z_j})$ satisfy

$$max_k|x_{ik} - x_{jk}| \leq D/2$$

then these data points are treated as replicates, where $\mathbf{x_i}$ are the smoothing variables and $\mathbf{z_i}$ are the regression variables.

You can use the DISTANCE= option to reduce the number of unique design points by treating nearby data as replicates. This can be useful when you have a large data set. The default value is 0.

LAMBDA0=*number*

specifies the smoothing parameter, λ_0, to be used in the thin-plate smoothing spline estimate. By default, PROC TPSPLINE uses the λ parameter that minimizes the GCV function for the final fit. The LAMBDA0= value must be positive.

LAMBDA=*list-of-values*

specifies a set of values for the λ parameter. PROC TPSPLINE returns a GCV value for each λ point that you specify. You can use the LAMBDA= option to study the GCV function curve for a set of values for λ. All values listed in the LAMBDA= option must be positive.

LOGNLAMBDA0=*number*
LOGNL0=*number*

specifies the smoothing parameter λ_0 on the $log10(n\lambda)$ scale. If you specify both the LOGNL0= and LAMBDA0= options, only the value provided by the LOGNL0= option is used. By default, PROC TPSPLINE uses the λ parameter that minimizes the GCV function for the estimate.

LOGNLAMBDA=*list-of-values*
LOGNL=*list-of-values*

specifies a set of values for the λ parameter on the $log10(n\lambda)$ scale. PROC TPSPLINE returns a GCV value for each λ point that you specify. You can use the LOGNLAMBDA= option to study the GCV function curve for a set of λ values. If you specify both the LOGNL= and LAMBDA= options, only the list of values provided by LOGNL= option is used.

In some cases, the LOGNL= option may be prefered over the LAMBDA= option. Because the LAMBDA= value must be positive, a small change in that value can result in a major change in the GCV value. If you instead specify λ on the log_{10} scale, the allowable range is enlarged to include negative values. Thus, the GCV function is less sensitive to changes in LOGNLAMBDA.

M=*number*

specifies the order of the derivative in the penalty term. The M= value must be a positive integer. The default value is the $max(2, INT(d/2) + 1)$, where d is the number of smoothing variables.

SCORE Statement

SCORE *DATA=SAS-data-set OUT=SAS-data-set* **;**

The SCORE statement calculates predicted values for a new data set. If you have multiple data sets to predict, you can specify multiple SCORE statements. You must use a SCORE statement for each data set.

The following keywords must be specified in the SCORE statement.

DATA=*SAS-data-set*

specifies the input SAS data set containing the smoothing variables $\mathbf{x}$ and regression variables $\mathbf{z}$. The predicted response $(\mathbf{y})$ value is computed for each $(\mathbf{x}, \mathbf{z})$ pair. The data set must include all independent variables specified in the MODEL statement.

OUT=*SAS-data-set*

specifies the name of the SAS data set to contain the predictions.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set* $< keyword \cdots$ *keyword* $>$ **;**

The OUTPUT statement creates a new SAS data set containing diagnostic measures calculated after fitting the model.

You can request a variety of diagnostic measures that are calculated for each observation in the data set. The new data set contains the variables specified in the MODEL statement in addition to the requested variables. If no *keyword* is present, the data set contains only the predicted values.

Details on the specifications in the OUTPUT statement are as follows.

OUT=*SAS-data-set*

specifies the name of the new data set to contain the diagnostic measures. This specification is required.

keyword

specifies the statistics to include in the output data set. The names of the new variables that contain the statistics are formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

For example, suppose that you have two dependent variables, say y1 and y2, and you specify the keywords PRED, ADIAG, and UCLM. The output SAS data set will contain the following variables:

- P_y1 and P_y2
- ADIAG_y1 and ADIAG_y2
- UCLM_y1 and UCLM_y2

The keywords and the statistics they represent are as follows:

RESID | R — residual values, calculated as ACTUAL - PREDICTED

PRED — predicted values

STD — standard error of the mean predicted value

UCLM — upper limit of the confidence interval for the expected value of the dependent variables. By default, PROC TPSPLINE computes 95% confidence limits.

LCLM — lower limit of the confidence interval for the expected value of the dependent variables. By default, PROC TPSPLINE computes 95% confidence limits.

ADIAG — diagonal element of the hat matrix associated with the observation

COEF — coefficients arranged in the order of $(\theta_0, \theta_1, \cdots, \theta_d, \delta_1, \cdots \delta_{\textit{nUnique}})$ where *nUnique* is the number of unique data points. This option can only be used when there is only one dependent variable in the model.

Details

Computational Formulas

The theoretical foundations for the thin-plate smoothing spline are described in Duchon (1976, 1977) and Meinguet (1979). Further results and applications are given in Wahba and Wendelberger (1980), Hutchinson and Bischof (1983), and Seaman and Hutchinson (1985).

Suppose that $\mathcal{H}_m$ is a space of functions whose partial derivatives of total order m are in $L_2(E^d)$ where E^d is the domain of $\mathbf{x}$.

Now, consider the data model

$$y_i = f(x_1(i), \ldots, x_d(i)) + \epsilon_i, i = 1, \ldots, n$$

where $f \in \mathcal{H}_m$.

Using the notation from the section "The Penalized Least Squares Estimate" on page 4497, for a fixed λ, estimate f by minimizing the penalized least squares function

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\beta)^2 + \lambda J_m(f)$$

There are several ways to define $J_m(f)$. For the thin-plate smoothing spline, with x of dimension d, define $J_m(f)$ as

$$J_m(f) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \sum \frac{m!}{\alpha_1! \cdots \alpha_d!} \left[\frac{\partial^m f}{\partial {x_1}^{\alpha_1} \cdots \partial {x_d}^{\alpha_d}} \right]^2 dx_1 \cdots dx_d$$

where $\sum_i \alpha_i = m$.

When $d = 2$ and $m = 2$, $J_m(f)$ is as follows:

$$J_2(f) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\left[\frac{\partial^2 f}{\partial {x_1}^2} \right]^2 + 2 \left[\frac{\partial^2 f}{\partial x_1 \partial x_2} \right]^2 + \left[\frac{\partial^2 f}{\partial {x_2}^2} \right]^2) dx_1 dx_2$$

In general, m and d must satisfy the condition that $2m - d > 0$. For the sake of simplicity, the formulas and equations that follow assume $m = 2$. Refer to Wahba (1990) and Bates et al. (1987) for more details.

Duchon (1976) showed that f_λ can be represented as

$$f_\lambda(\mathbf{x_i}) = \theta_\mathbf{0} + \sum_{\mathbf{j=1}}^{\mathbf{d}} \theta_\mathbf{j} \mathbf{x_{ij}} + \sum_{\mathbf{j=1}}^{\mathbf{n}} \delta_\mathbf{j} \mathbf{E_2}(\mathbf{x_i} - \mathbf{x_j})$$

where $E_2(\mathbf{s}) = \frac{\mathbf{1}}{\mathbf{2^3}\pi} ||\mathbf{s}||^\mathbf{2} \, \mathbf{ln}(||\mathbf{s}||)$.

If you define $\mathbf{K} = (\mathbf{K})_\mathbf{ij} = \mathbf{E_2}(\mathbf{x_i} - \mathbf{x_j})$ and $\mathbf{T} = (\mathbf{T})_\mathbf{ij} = (\mathbf{x_{ij}})$, the goal is to find coefficients β, θ, and δ that minimize

$$S_\lambda(\beta, \theta, \delta) = \frac{1}{n} ||\mathbf{y} - \mathbf{T}\theta - \mathbf{K}\delta - \mathbf{Z}\beta||^\mathbf{2} + \lambda \delta^\mathbf{T} \mathbf{K} \delta$$

A unique solution is guaranteed if the matrix $\mathbf{T}$ is of full rank and $\delta^T \mathbf{K} \delta \geq \mathbf{0}$.

If $\alpha = \begin{pmatrix} \theta \\ \beta \end{pmatrix}$ and $\mathbf{X} = (\mathbf{T} : \mathbf{Z})$, the expression for S_λ becomes

$$\frac{1}{n} ||\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\delta||^\mathbf{2} + \lambda \delta^\mathbf{T} \mathbf{K} \delta$$

The coefficients α and δ can be obtained by solving

$$\begin{aligned} (\mathbf{K} + \mathbf{n}\lambda\mathbf{I_n})\delta + \mathbf{X}\alpha &= y \\ \mathbf{X^T}\delta &= \mathbf{0} \end{aligned}$$

To compute α and δ, let the QR decomposition of $\mathbf{X}$ be

$$\mathbf{X} = (\mathbf{Q_1} : \mathbf{Q_2}) \begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

where $(\mathbf{Q_1} : \mathbf{Q_2})$ is an orthogonal matrix and $\mathbf{R}$ is upper triangular, with $\mathbf{X^T Q_2} = \mathbf{0}$ (Dongarra et al. 1979).

Since $\mathbf{X^T}\delta = \mathbf{0}$, δ must be in the column space of $\mathbf{Q_2}$. Therefore, δ can be expressed as $\delta = \mathbf{Q_2}\gamma$ for a vector γ. Substituting $\delta = \mathbf{Q_2}\gamma$ into the preceding equation and multiplying through by $\mathbf{Q_2^T}$ gives

$$\mathbf{Q_2^T}(\mathbf{K} + \mathbf{n}\lambda\mathbf{I_n})\mathbf{Q_2}\gamma = \mathbf{Q_2^T y}$$

or

$$\delta = \mathbf{Q_2}\gamma = \mathbf{Q_2}[\mathbf{Q_2^T}(\mathbf{K} + \mathbf{n}\lambda\mathbf{I_n})\mathbf{Q_2}]^{-1}\mathbf{Q_2^T y}$$

The coefficient α can be obtained by solving

$$\mathbf{R}\alpha = \mathbf{Q_1^T}[\mathbf{y} - (\mathbf{K} + \mathbf{n}\lambda\mathbf{I_n})\delta]$$

The influence matrix $\mathbf{A}(\lambda)$ is defined as

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y}$$

and has the form

$$\mathbf{A}(\lambda) = \mathbf{I} - \mathbf{n}\lambda\mathbf{Q_2}[\mathbf{Q_2^T}(\mathbf{K} + \mathbf{n}\lambda\mathbf{I_n})\mathbf{Q_2}]^{-1}\mathbf{Q_2^T}$$

Similar to the regression case, and if you consider the trace of $\mathbf{A}(\lambda)$ as the degrees of freedom for the information signal and the trace of $(\mathbf{I_n} - \mathbf{A}(\lambda))$ as the degrees of freedom for the noise component, the estimate σ^2 can be represented as

$$\hat{\sigma}^2 = \frac{RSS(\lambda)}{Trace(\mathbf{I_n} - \mathbf{A}(\lambda))}$$

where $RSS(\lambda)$ is the residual sum of squares. Theoretical properties of these estimates have not yet been published. However, good numerical results in simulation studies have been described by several authors. For more information, refer to O'Sullivan and Wong (1987), Nychka (1986a, 1986b, and 1988), and Hall and Titterington (1987).

Confidence Intervals

Viewing the spline model as a Bayesian model, Wahba (1983) proposed Bayesian confidence intervals for smoothing spline estimates as follows:

$$\hat{f}_\lambda(x_i) \pm z_{\alpha/2}\sqrt{\hat{\sigma}^2 a_{ii}(\lambda)}$$

where $a_{ii}(\lambda)$ is the ith diagonal element of the $\mathbf{A}(\lambda)$ matrix and $z_{\alpha/2}$ is the $\alpha/2$ point of the normal distribution. The confidence intervals are interpreted as intervals "across the function" as opposed to point-wise intervals.

Suppose that you fit a spline estimate to experimental data that consists of a true function f and a random error term, ϵ_i. In repeated experiments, it is likely that about $100(1-\alpha)\%$ of the confidence intervals cover the corresponding true values, although some values are covered every time and other values are not covered by the confidence intervals most of the time. This effect is more pronounced when the true surface or surface has small regions of particularly rapid change.

Smoothing Parameter

The quantity λ is called the smoothing parameter, which controls the balance between the goodness of fit and the smoothness of the final estimate.

A large λ heavily penalizes the mth derivative of the function, thus forcing $f^{(m)}$ close to 0. A small λ places less of a penalty on rapid change in $f^{(m)}(x)$, resulting in an estimate that tends to interpolate the data points.

The smoothing parameter greatly affects the analysis, and it should be selected with care. One method is to perform several analyses with different values for λ and compare the resulting final estimates.

A more objective way to select the smoothing parameter λ is to use the "leave-out-one" cross validation function, which is an approximation of the predicted mean squares error. A generalized version of the leave-out-one cross validation function is proposed by Wahba (1990) and is easy to calculate. This Generalized Cross Validation (GCV) function $(V(\lambda))$ is defined as

$$V(\lambda) = \frac{(1/n)||(\mathbf{I}-\mathbf{A}(\lambda))\mathbf{y}||^2}{[(1/n)tr(\mathbf{I}-\mathbf{A}(\lambda))]^2}$$

The justification for using the GCV function to select λ relies on asymptotic theory. Thus, you cannot expect good results for very small sample sizes or when there is not enough information in the data to separate the information signal from the noise component. Simulation studies suggest that for independent and identically distributed Gaussian noise, you can obtain reliable estimates of λ for n greater than 25 or 30. Note that, even for large values of n (say $n \geq 50$), in extreme Monte Carlo simulations there may be a small percentage of unwarranted extreme estimates in which $\hat{\lambda} = 0$ or $\hat{\lambda} = \infty$ (Wahba 1983). Generally, if σ^2 is known to within an order of magnitude, the occasional extreme case can be readily identified. As n gets larger, the effect becomes weaker.

The GCV function is fairly robust against nonhomogeneity of variances and non-Gaussian errors (Villalobos and Wahba 1987). Andrews (1988) has provided favorable theoretical results when variances are unequal. However, this selection method is likely to give unsatisfactory results when the errors are highly correlated.

The GCV value may be suspect when λ is extremely small because computed values may become indistinguishable from zero. In practice, calculations with $\lambda = 0$

or λ near 0 can cause numerical instabilities resulting in an unsatisfactory solution. Simulation studies have shown that a λ with $log_{10}(n\lambda) > -8$ is small enough that the final estimate based on this λ almost interpolates the data points. A GCV value based on a $\lambda \le 10^{-8}$ may not be accurate.

ODS Tables Produced by PROC TPSPLINE

PROC TPSPLINE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 74.1. ODS Tables Produced by PROC TPSPLINE

ODS Table Name	Description	Statement	Option
DataSummary	Data summary	PROC	default
FitSummary	Fit parameters and fit summary	PROC	default
FitStatistics	Model fit statistics	PROC	default
GCVFunction	GCV table	MODEL	LOGNLAMBDA, LAMBDA

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

For example, the following statements create an output data set named FitStats containing the FitStatistics table, an output data set named DataInfo containing the DataSummary table, an output data set named ModelInfo containing the FitSummary and an output data set named GCVFunc containing the GCVFunction.

```
proc tpspline data=Melanoma;
   model Incidences=Year /LOGNLAMBDA=(-4 to 0 by 0.2);
   ods output FitStatistics = FitStats
              DataSummary   = DataInfo
              FitSummary    = ModelInfo
              GCVFunction   = GCVFunc;
run;
```

Examples

Example 74.1. Partial Spline Model Fit

The following example analyzes the data set Measure that was introduced in the "Getting Started" section on page 4499. That analysis determined that the final estimated surface can be represented by a quadratic function for one or both of the independent variables. This example illustrates how you can use PROC TPSPLINE to fit a partial spline model. The data set Measure is fit using the following model:

$$f(x1, x2) = 1 + x_1 + x_1^2 + h(x_2)$$

The model has a parametric component (associated with the x_1 variable) and a nonparametric component (associated with the x_2 variable). The following statements fit a partial spline model.

```
data Measure;
   set Measure;
   x1sq = x1*x1;
run;

data pred;
   do x1=-1 to 1 by 0.1;
      do x2=-1 to 1 by 0.1;
         x1sq = x1*x1;
         output;
      end;
   end;
run;

proc tpspline data= measure;
   model y = x1 x1sq (x2);
   score data = pred
         out  = predy;
run;
```

Output 74.1.1 displays the results from these statements.

Output 74.1.1. Output from PROC TPSPLINE

```
                          The TPSPLINE Procedure
                        Dependent Variable:  y

                        Summary of Input Data Set

                 Number of Non-Missing Observations    50
                 Number of Missing Observations         0
                 Unique Smoothing Design Points         5


                         Summary of Final Model

                 Number of Regression Variables        2
                 Number of Smoothing Variables         1
                 Order of Derivative in the Penalty    2
                 Dimension of Polynomial Space         4


                            Summary Statistics
                           of Final Estimation

                    log10(n*Lambda)            -2.2374
                    Smoothing Penalty         205.3461
                    Residual SS                 8.5821
                    Tr(I-A)                    43.1534
                    Model DF                    6.8466
                    Standard Deviation          0.4460
```

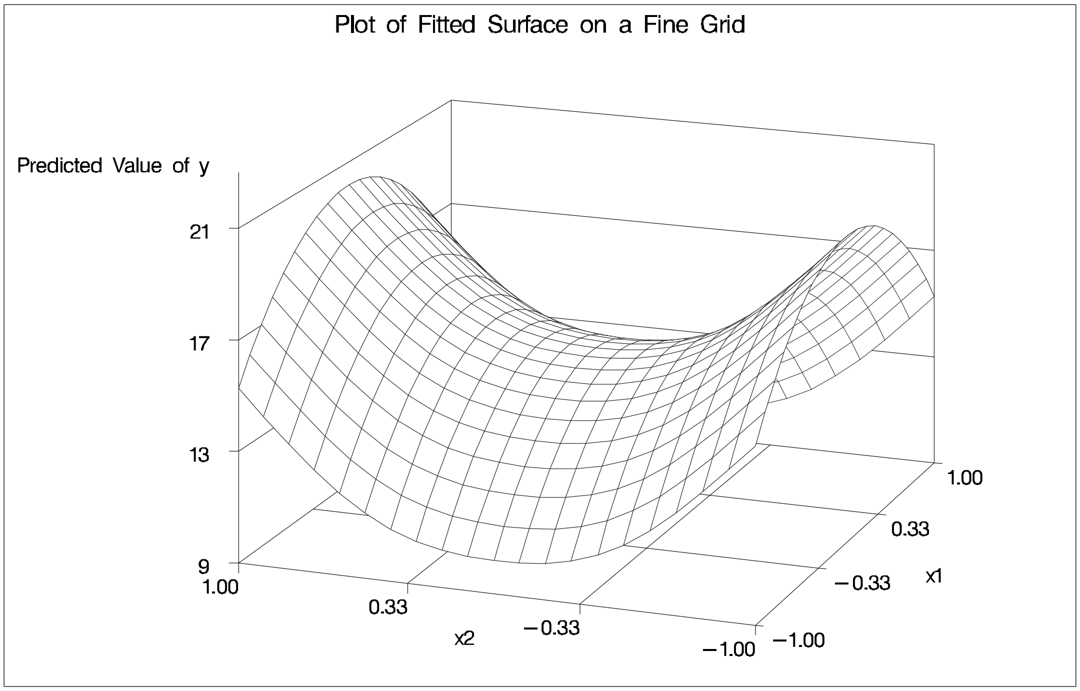
As displayed in Output 74.1.1, there are five unique design points for the smoothing variable x2 and two regression variables in the model (x1,x1sq). The dimension of the null space (polynomial space) is 4. The standard deviation of the estimate is much larger than the one based on the model with both $x1$ and $x2$ as smoothing variables (0.445954 compared to 0.098421). One of the many possible explanations may be that the number of unique design points of the smoothing variable is too small to warrant an accurate estimate for $h(x2)$.

The following statements produce a surface plot for the partial spline model:

```
title 'Plot of Fitted Surface on a Fine Grid';

proc g3d data=predy;
   plot x2*x1=p_y/grid
                  zmin=9
                  zmax=21
                  zticknum=4;
run;
```

The surface displayed in Output 74.1.2 is similar to the one estimated by using the full nonparametric model (displayed in Figure 74.5).

Output 74.1.2. Plot of TPSPLINE Fit from the Partial Spline Model



Example 74.2. Spline Model With Higher-Order Penalty

The following example continues the analysis of the data set Measure to illustrate how you can use PROC TPSPLINE to fit a spline model with a higher-order penalty term. Spline models with high-order penalty terms move low-order polynomial terms into the null space. Hence, there is no penalty for these terms, and they can vary without constraint.

As shown in the previous analyses, the final model for the data set Measure must include quadratic terms for both x1 and x2. This example fits the following model:

$$f(x_1, x_2) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_2 + \theta_4 x_2^2 + \theta_5 x_1 * x_2 + g(x_1, x_2)$$

The model includes quadratic terms for both variables, although it differs from the usual linear model. The nonparametric term $g(x_1, x_2)$ explains the variation of the data unaccounted for by a simple quadratic surface.

To modify the order of the derivative in the penalty term, specify the M= option. The following statements specify the option M=3 in order to include the quadratic terms in the null space:

```
data measure;
   set measure;
   x1sq = x1*x1;
   x2sq = x2*x2;
   x1x2 = x1*x2;
;

proc tpspline data= measure;
   model y = (x1 x2) / m=3;
   score data = pred
          out = predy;
run;
```

The output resulting from these statements is displayed in Output 74.2.1.

Output 74.2.1. Output from PROC TPSPLINE with M=3

```
                    The TPSPLINE Procedure
                  Dependent Variable:  y

                  Summary of Input Data Set

           Number of Non-Missing Observations    50
           Number of Missing Observations         0
           Unique Smoothing Design Points        25


                   Summary of Final Model

           Number of Regression Variables        0
           Number of Smoothing Variables         2
           Order of Derivative in the Penalty    3
           Dimension of Polynomial Space         6


                      Summary Statistics
                     of Final Estimation

              log10(n*Lambda)           -3.7831
              Smoothing Penalty       2092.4495
              Residual SS                0.2731
              Tr(I-A)                   29.1716
              Model DF                  20.8284
              Standard Deviation         0.0968
```

The model contains six terms in the null space. Compare Output 74.2.1 with Figure 74.2: the LOGNLAMBDA value and the smoothing penalty differ significantly. Note that, in general, these terms are not directly comparable for different models. The final estimate based on this model is close to the estimate based on the model using the default, M=2.

In the following statements, the REG procedure fits a quadratic surface model to the data set Measure.

```
proc reg data= measure;
   model y = x1 x1sq x2 x2sq x1x2;
run;
```

The results are displayed in Output 74.2.2.

Output 74.2.2. Quadratic Surface Model: The REG Procedure

```
                                The REG Procedure
                                  Model: MODEL1
                              Dependent Variable: y

                               Analysis of Variance

                                       Sum of           Mean
Source                   DF           Squares         Square    F Value    Pr > F

Model                     5         443.20502       88.64100     436.33    <.0001
Error                    44           8.93874        0.20315
Corrected Total          49         452.14376


             Root MSE              0.45073    R-Square     0.9802
             Dependent Mean       15.08548    Adj R-Sq     0.9780
             Coeff Var             2.98781


                               Parameter Estimates

                             Parameter       Standard
     Variable     DF          Estimate          Error    t Value    Pr > |t|

     Intercept     1          14.90834        0.12519     119.09      <.0001
     x1            1           0.01292        0.09015       0.14      0.8867
     x1sq          1          -4.85194        0.15237     -31.84      <.0001
     x2            1           0.02618        0.09015       0.29      0.7729
     x2sq          1           5.20624        0.15237      34.17      <.0001
     x1x2          1          -0.04814        0.12748      -0.38      0.7076
```

The REG procedure produces slightly different results. To fit a similar model with PROC TPSPLINE, you can use a MODEL statement specifying the degrees of freedom with the DF= option. You can also use a large value for the LOGNLAMBDA0= option to force a parametric model fit.

Because there is one degree of freedom for each of the following terms, Intercept, x1, x2, x1sq, x2sq, and x1x2, the DF=6 option is used.

```
proc tpspline data=measure;
   model  y=(x1 x2) /m=3 df=6 lognlambda=(-4 to 1 by 0.2);
   score data = pred
         out = predy;
run;
```

The results are displayed in Output 74.2.3. PROC TPSPLINE displays the list of GCV values for comparison.

Output 74.2.3. Output from PROC TPSPLINE Using M=3 and DF=6

```
                    The TPSPLINE Procedure
                  Dependent Variable:  y

                  Summary of Input Data Set

           Number of Non-Missing Observations    50
           Number of Missing Observations         0
           Unique Smoothing Design Points        25


                   Summary of Final Model

           Number of Regression Variables        0
           Number of Smoothing Variables         2
           Order of Derivative in the Penalty    3
           Dimension of Polynomial Space         6


                         GCV Function

               log10(n*Lambda)             GCV

                     -4.000000        0.016330
                     -3.800000        0.016051*
                     -3.600000        0.016363
                     -3.400000        0.017770
                     -3.200000        0.021071
                     -3.000000        0.027496
                     -2.800000        0.038707
                     -2.600000        0.056292
                     -2.400000        0.080613
                     -2.200000        0.109714
                     -2.000000        0.139642
                     -1.800000        0.166338
                     -1.600000        0.187437
                     -1.400000        0.202625
                     -1.200000        0.212871
                     -1.000000        0.219512
                     -0.800000        0.223727
                     -0.600000        0.226377
                     -0.400000        0.228041
                     -0.200000        0.229085
                             0        0.229740
                      0.200000        0.230153
                      0.400000        0.230413
                      0.600000        0.230576
                      0.800000        0.230680
                      1.000000        0.230745

             Note: * indicates minimum GCV value.
```

```
                     The TPSPLINE Procedure
                   Dependent Variable:  y

                       Summary Statistics
                      of Final Estimation

               log10(n*Lambda)            2.3830
               Smoothing Penalty          0.0000
               Residual SS                8.9384
               Tr(I-A)                   43.9997
               Model DF                   6.0003
               Standard Deviation         0.4507
```

The final estimate is based on 6.000330 degrees of freedom because there are already 6 degrees of freedom in the null space and the search range for lambda is not large enough (in this case, setting DF=6 is equivalent to setting lambda = ∞).

The standard deviation and RSS (Output 74.2.3) are close to the sum of squares for the error term and the root MSE from the the linear regression model (Output 74.2.2), respectively.

For this model, the optimal LOGNLAMBDA is around -3.8, which produces a standard deviation estimate of 0.096765 (see Output 74.2.1) and a GCV value of 0.016051, while the model specifying DF=6 results in a LOGNLAMBDA larger than 1 and a GCV value larger than 0.23074. The nonparametric model, based on the GCV, should provide better prediction, but the linear regression model can be more easily interpreted.

Example 74.3. Multiple Minima of the GCV Function

The following data represent the deposition of sulfate (SO_4) at 179 sites in 48 contiguous states of the United States in 1990. Each observation records the latitude and longitude of the site as well as the SO_4 deposition at the site measured in gram per square meter (g/m^2).

You can use PROC TPSPLINE to fit a surface that reflects the general trend and that reveals underlying features of the data.

```
data so4;
   input latitude longitude so4 @@;
   datalines;
   32.45833  87.24222 1.403 34.28778  85.96889 2.103
   33.07139 109.86472 0.299 36.07167 112.15500 0.304
   31.95056 112.80000 0.263 33.60500  92.09722 1.950
   34.17944  93.09861 2.168 36.08389  92.58694 1.578
             .
             .
             .
     162 additional observations
             .
             .
             .
   45.82278  91.87444 0.984 41.34028 106.19083 0.335
   42.73389 108.85000 0.236 42.49472 108.82917 0.313
   42.92889 109.78667 0.182 43.22278 109.99111 0.161
   43.87333 104.19222 0.306 44.91722 110.42028 0.210
   45.07611  72.67556 2.646
   ;

data pred;
   do latitude = 25 to 47 by 1;
      do longitude = 68 to 124 by 1;
         output;
      end;
   end;
run;
```

The preceding statements create the SAS data set so4 and the data set pred in order to make predictions on a regular grid. The following statements fit a surface for SO_4 deposition. The ODS OUTPUT statement creates a data set called GCV to contain the GCV values for LOGNLAMBDA in the range from -6 to 1.

```
proc tpspline data=so4;
   ods output GCVFunction=gcv;
   model so4 = (latitude longitude) /lognlambda=(-6 to 1 by 0.1);
   score data=pred out=prediction1;
run;
```

Partial output from these statements is displayed in Output 74.3.1.

Output 74.3.1. Partial Output from PROC TPSPLINE for Data Set SO4

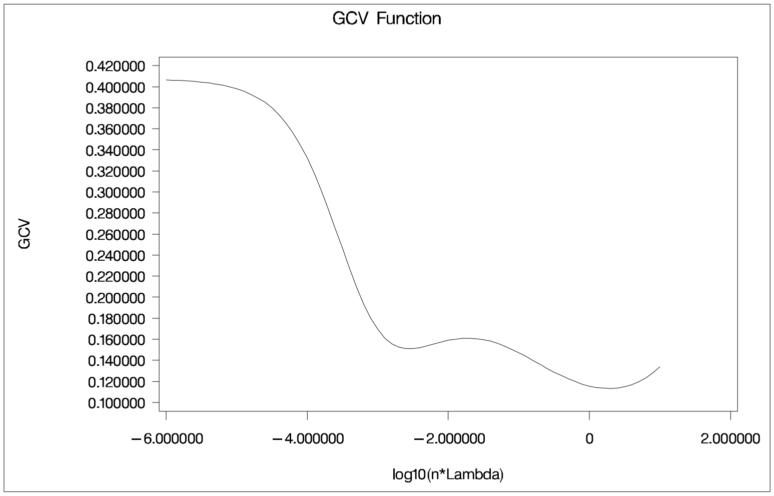
```
                         The TPSPLINE Procedure
                        Dependent Variable:  so4

                         Summary of Input Data Set

                 Number of Non-Missing Observations    179
                 Number of Missing Observations          0
                 Unique Smoothing Design Points        179


                          Summary of Final Model

                  Number of Regression Variables        0
                  Number of Smoothing Variables         2
                  Order of Derivative in the Penalty    2
                  Dimension of Polynomial Space         3


                            Summary Statistics
                           of Final Estimation

                     log10(n*Lambda)            0.2770
                     Smoothing Penalty          2.4588
                     Residual SS               12.4450
                     Tr(I-A)                  140.2750
                     Model DF                  38.7250
                     Standard Deviation         0.2979
```

The following statements produce Output 74.3.2:

```
symbol1 interpol=join value=none;
title "GCV Function";

proc gplot data=gcv;
   plot gcv*lognlambda/frame cframe=ligr
                        vaxis=axis1 haxis=axis2;
run;
```

Output 74.3.2 displays the plot of the GCV function versus nlambda in $log10$ scale. The GCV function has two minima. PROC TPSPLINE locates the minimum at 0.277005. The figure also displays a local minimum located around -2.56. Note that the TPSPLINE procedure may not always find the global minimum, although it did in this case.

Output 74.3.2. GCV Function of SO4 Data Set



The following analysis specifies the option LOGNLAMBDA0=-2.56. The output is displayed in Output 74.3.3.

```
proc tpspline data=so4;
   model so4 = (latitude longitude) /lognlambda0=-2.56;
   score data=pred out=prediction2;
run;
```

Output 74.3.3. Output from PROC TPSPLINE for Data Set SO4 with LOGNLAMBDA=-2.56

```
                    The TPSPLINE Procedure
                 Dependent Variable:  so4

                  Summary of Input Data Set

        Number of Non-Missing Observations    179
        Number of Missing Observations          0
        Unique Smoothing Design Points        179


                   Summary of Final Model

         Number of Regression Variables        0
         Number of Smoothing Variables         2
         Order of Derivative in the Penalty    2
         Dimension of Polynomial Space         3


                      Summary Statistics
                     of Final Estimation

            log10(n*Lambda)             -2.5600
            Smoothing Penalty          177.2144
            Residual SS                  0.0438
            Tr(I-A)                      7.2086
            Model DF                   171.7914
            Standard Deviation           0.0779
```

The smoothing penalty is much larger in Output 74.3.3 than that displayed in Output 74.3.1. The estimate in Output 74.3.1 uses a large lambda value and, therefore, the surface is smoother than the estimate using LOGNLAMBDA=-2.56 (Output 74.3.3).

The estimate based on LOGNLAMBDA=-2.56 has a larger value for the degrees of freedom, and it has a much smaller standard deviation.

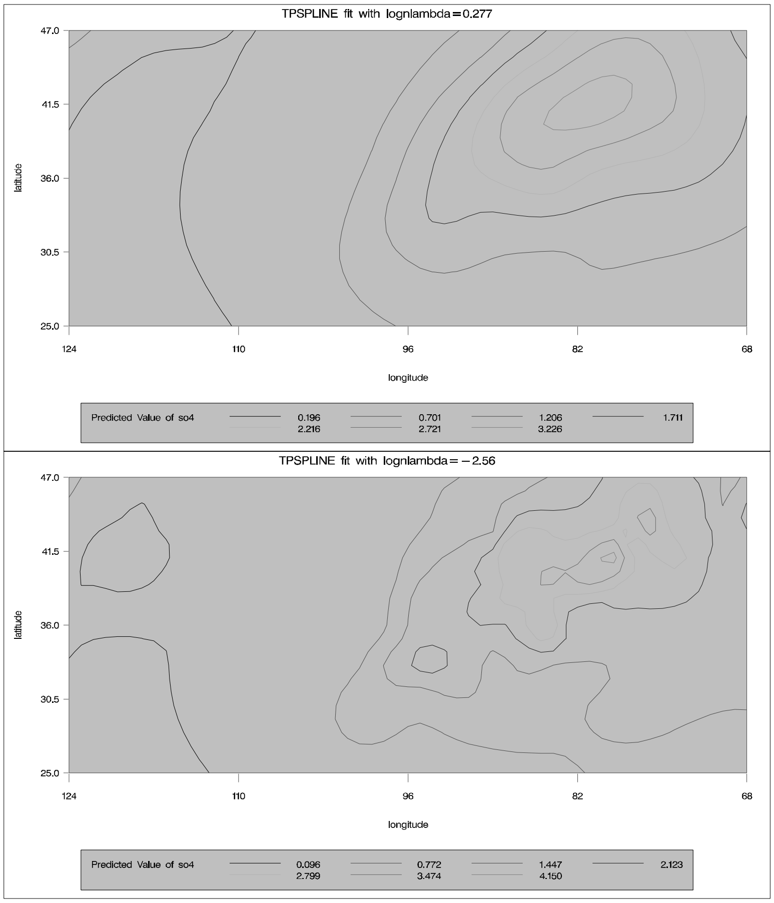
However, a smaller standard deviation in nonparametric regression does not necessarily mean that the estimate is good: a small λ value always produces an estimate closer to the data and, therefore, a smaller standard deviation.

The following statements produce two contour plots of the estimates using the GCONTOUR procedure. In the final step, the plots are placed into a single graphic with the GREPLAY procedure.

```
title "TPSPLINE fit with lognlambda=0.277";
proc gcontour data=prediction1 gout=grafcat;
   plot latitude*longitude = P_so4/
                  name="tpscon1" legend=legend1
                  vaxis=axis1 haxis=axis2 cframe=ligr hreverse;
run;

title "TPSPLINE fit with lognlambda=-2.56";
proc gcontour data=prediction2 gout=grafcat;
   plot latitude*longitude = P_so4/
                  name="tpscon2" legend=legend1
                  vaxis=axis1 haxis=axis2 cframe=ligr hreverse;
run;

title;
proc greplay igout=grafcat tc=sashelp.templt template=v2 nofs;
   treplay 1:tpscon1 2:tpscon2;
quit;
```

Compare the two estimates by examining the contour plots of both estimates (Output 74.3.4).

Output 74.3.4. Contour Plot of TPSPLINE Estimates with Different Lambdas



As the contour plots show, the estimate with LOGNLAMBDA=0.277 may represent the underlying trend, while the estimate with the LOGNLAMBDA=-2.56 is very rough and may be modeling the noise component.

Example 74.4. Large Data Set Application

The following example illustrates how you can use the D= option to decrease the computation time needed by the TPSPLINE procedure. Note that, while the D= option can be helpful in decreasing computation time for large data sets, it may produce unexpected results when used with small data sets.

The following statements generate the data set large:

```
data large;
   do x=-5 to 5 by 0.02;
      y=5*sin(3*x)+1*rannor(57391);
      output;
   end;
run;
```

The data set large contains 501 observations with one independent variable x and one dependent variable y. The following statements invoke PROC TPSPLINE to produce a thin-plate smoothing spline estimate and the associated 99% confidence interval. The output statistics are saved in the data set fit1.

```
proc tpspline data=large;
   model y  =(x) /lambda=(-5 to -1 by 0.2) alpha=0.01;
   output out=fit1 pred lclm uclm;
run;
```

The results from this MODEL statement are displayed in Output 74.4.1.

Output 74.4.1. Output from PROC TPSPLINE without the D= Option

```
                         The TPSPLINE Procedure
                        Dependent Variable:  y

                        Summary of Input Data Set

                Number of Non-Missing Observations    501
                Number of Missing Observations          0
                Unique Smoothing Design Points        501


                         Summary of Final Model

                 Number of Regression Variables        0
                 Number of Smoothing Variables         1
                 Order of Derivative in the Penalty    2
                 Dimension of Polynomial Space         2


                              GCV Function

                     log10(n*Lambda)             GCV

                           -5.000000        1.258653
                           -4.800000        1.228743
                           -4.600000        1.205835
                           -4.400000        1.188371
                           -4.200000        1.174644
                           -4.000000        1.163102
                           -3.800000        1.152627
                           -3.600000        1.142590
                           -3.400000        1.132700
                           -3.200000        1.122789
                           -3.000000        1.112755
                           -2.800000        1.102642
                           -2.600000        1.092769
                           -2.400000        1.083779
                           -2.200000        1.076636
                           -2.000000        1.072763*
                           -1.800000        1.074636
                           -1.600000        1.087152
                           -1.400000        1.120339
                           -1.200000        1.194023
                           -1.000000        1.344213

                   Note: * indicates minimum GCV value.
```

```
                         The TPSPLINE Procedure
                        Dependent Variable:  y

                            Summary Statistics
                           of Final Estimation

                    log10(n*Lambda)           -1.9483
                    Smoothing Penalty       9953.7063
                    Residual SS              475.0984
                    Tr(I-A)                  471.0861
                    Model DF                  29.9139
                    Standard Deviation         1.0042
```

The following statements specify an identical model, but with the additional specification of the D= option. The estimates are obtained by treating nearby points as replicates.

```
proc tpspline data=large;
   model y  =(x) /lambda=(-5 to -1 by 0.2) d=0.05 alpha=0.01;
   output out=fit2 pred lclm uclm;
run;
```

The output is displayed in Output 74.4.2.

Output 74.4.2. Output from PROC TPSPLINE with the D= Option

```
                             The TPSPLINE Procedure
                           Dependent Variable:  y

                           Summary of Input Data Set

                   Number of Non-Missing Observations    501
                   Number of Missing Observations          0
                   Unique Smoothing Design Points        251


                            Summary of Final Model

                    Number of Regression Variables        0
                    Number of Smoothing Variables         1
                    Order of Derivative in the Penalty    2
                    Dimension of Polynomial Space         2


                                  GCV Function

                        log10(n*Lambda)            GCV

                              -5.000000       1.306536
                              -4.800000       1.261692
                              -4.600000       1.226881
                              -4.400000       1.200060
                              -4.200000       1.179284
                              -4.000000       1.162776
                              -3.800000       1.149072
                              -3.600000       1.137120
                              -3.400000       1.126220
                              -3.200000       1.115884
                              -3.000000       1.105766
                              -2.800000       1.095730
                              -2.600000       1.085972
                              -2.400000       1.077066
                              -2.200000       1.069954
                              -2.000000       1.066076*
                              -1.800000       1.067929
                              -1.600000       1.080419
                              -1.400000       1.113564
                              -1.200000       1.187172
                              -1.000000       1.337252

                      Note: * indicates minimum GCV value.
```

```
                          The TPSPLINE Procedure
                        Dependent Variable:  y

                            Summary Statistics
                           of Final Estimation

                    log10(n*Lambda)            -1.9477
                    Smoothing Penalty        9943.5615
                    Residual SS               472.1424
                    Tr(I-A)                   471.0901
                    Model DF                   29.9099
                    Standard Deviation          1.0011
```

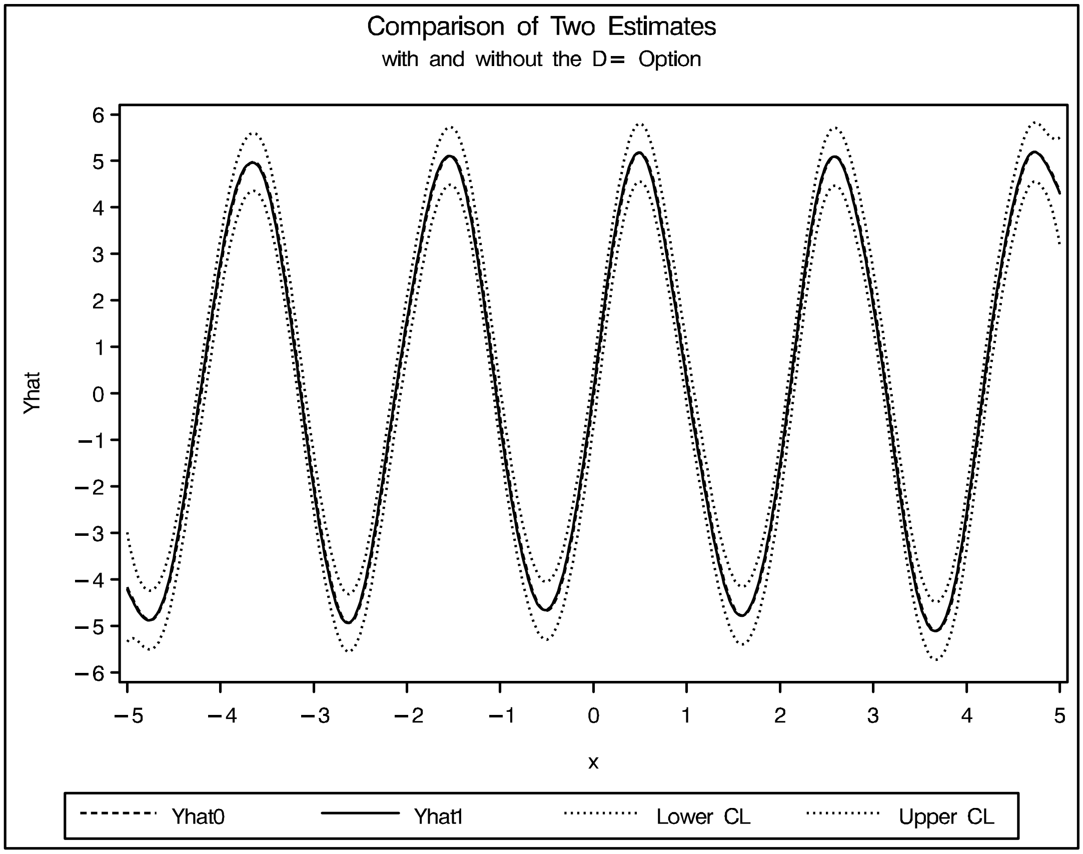
The difference between the two estimates is minimal. However, the CPU time for the second MODEL statement is only about 1/8 of the CPU time used in the first model fit.

The following statements produce a plot for comparison of the two estimates:

```
data fit2;
   set fit2;
   P1_y     = P_y;
   LCLM1_y  = LCLM_y;
   UCLM1_y  = UCLM_y;
   drop P_y LCLM_y UCLM_y;

proc sort data=fit1;
   by x y;
proc sort data=fit2;
   by x y;

data comp;
   merge fit1 fit2;
      by x y;
   label p1_y   ="Yhat1" p_y="Yhat0"
         lclm_y ="Lower CL"
         uclm_y ="Upper CL";

symbol1  i=join v=none ;
symbol2  i=join v=none ;
symbol3  i=join v=none color=cyan;
symbol4  i=join v=none color=cyan;

title ’Comparison of Two Estimates’;
title2 ’with and without the D= Option’;

proc gplot data=comp;
   plot P_y*x=1
        P1_y*x=2
        LCLM_y*x=4
        UCLM_y*x=4/overlay     legend=legend1
                   vaxis=axis1 haxis=axis2
                   frame       cframe=ligr;
run;
```

The estimates from fit1 and fit2 are displayed in Output 74.4.3 with the 99% confidence interval from the fit1 output data set.

Output 74.4.3. Comparison of Two Fits with and without the D= Option



Example 74.5. Computing a Bootstrap Confidence Interval

The following example illustrates how you can construct a bootstrap confidence interval by using the multiple responses feature in PROC TPSPLINE.

Numerous epidemiological observations have indicated that exposure to solar radiation is an important factor in the etiology of melanoma. The following data present age-adjusted melanoma incidences for 37 years from the Connecticut Tumor Registry (Houghton, Flannery, and Viola 1980). The data are analyzed by Ramsay and Silverman (1997).

```
data melanoma;
   input  year incidences @@;
   datalines;
   1936    0.9   1937   0.8  1938   0.8  1939   1.3
   1940    1.4   1941   1.2  1942   1.7  1943   1.8
   1944    1.6   1945   1.5  1946   1.5  1947   2.0
   1948    2.5   1949   2.7  1950   2.9  1951   2.5
   1952    3.1   1953   2.4  1954   2.2  1955   2.9
   1956    2.5   1957   2.6  1958   3.2  1959   3.8
   1960    4.2   1961   3.9  1962   3.7  1963   3.3
   1964    3.7   1965   3.9  1966   4.1  1967   3.8
   1968    4.7   1969   4.4  1970   4.8  1971   4.8
   1972    4.8
   ;
run;
```

The variable incidences records the number of melanoma cases per 100,000 people for the years 1936 to 1972. The following model fits the data and requests a 90% Bayesian confidence interval along with the estimate.

```
proc tpspline data=melanoma;
   model incidences = (year) /alpha = 0.1;
   output out = result pred uclm lclm;
run;
```

The output is displayed in Output 74.5.1

Output 74.5.1. Output from PROC TPSPLINE for the Melanoma Data Set

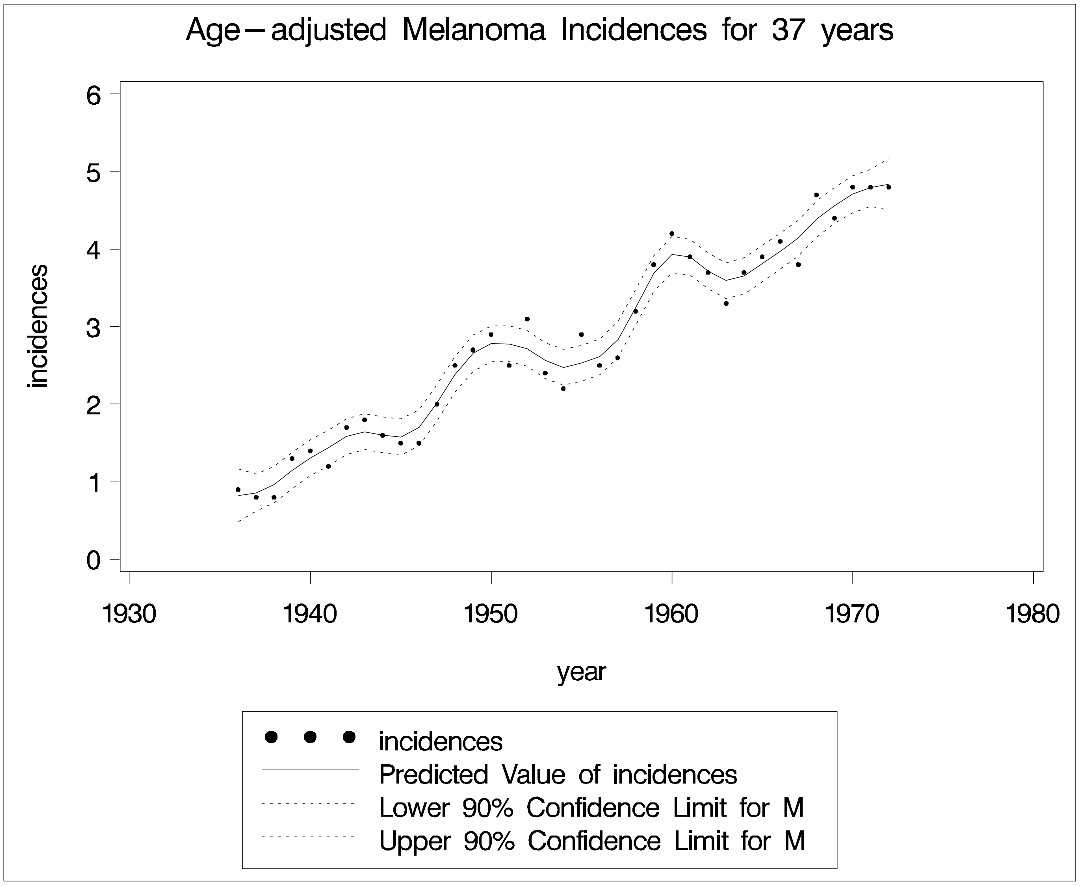
```
                             The TPSPLINE Procedure
                      Dependent Variable:  incidences

                           Summary of Input Data Set

                    Number of Non-Missing Observations    37
                    Number of Missing Observations         0
                    Unique Smoothing Design Points        37


                            Summary of Final Model

                    Number of Regression Variables        0
                    Number of Smoothing Variables         1
                    Order of Derivative in the Penalty    2
                    Dimension of Polynomial Space         2


                               Summary Statistics
                              of Final Estimation

                       log10(n*Lambda)           -0.0607
                       Smoothing Penalty          0.5171
                       Residual SS                1.2243
                       Tr(I-A)                   22.5852
                       Model DF                  14.4148
                       Standard Deviation         0.2328
```

The following statements produce a plot of the estimated curve:

```
symbol1 h=1pct ;
symbol2 i=join v=none;
symbol3 i=join v=none;
symbol4 i=join v=none c=cyan;

legend1 frame cframe=ligr cborder=black
        label=none position=center;
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;

title1 'Age-adjusted Melanoma Incidences for 37 years';

proc gplot data=result;
   plot  incidences*year=1
         p_incidences*year=2
         lclm_incidences*year=3
         uclm_incidences*year=4 /overlay legend=legend1
                                 vaxis=axis1 haxis=axis2
                                 frame cframe=ligr;
run;
```

The estimated curve is displayed with 90% confidence interval bands in Output 74.5.2. The number of melanoma incidences exhibits a periodic pattern and increases over the years. The periodic pattern is related to sunspot activity and the accompanying fluctuations in solar radiation.

Output 74.5.2. TPSPLINE Estimate and 90% Confidence Interval of Melanoma Data



Wang and Wahba (1995) compared several bootstrap confidence intervals to Bayesian confidence intervals for smoothing splines. Both bootstrap and Bayesian confidence intervals are across-the-curve intervals, not point-wise intervals. They concluded that bootstrap confidence intervals work as well as Bayesian intervals concerning average coverage probability. Additionally, bootstrap confidence intervals appear to be better for small sample sizes. Based on their simulation, the "percentile-t interval" bootstrap interval performs better than the other types of bootstrap intervals.

Suppose that $\hat{f}_{\hat{\lambda}}$ and $\hat{\sigma}$ are the estimates of f and σ from the data. Assume that $\hat{f}_{\hat{\lambda}}$ is the "true" f, and generate the bootstrap sample as follows:

$$y_i^* = \hat{f}_{\hat{\lambda}}(x_i) + \epsilon_i^*, \quad i = 1, \cdots, n$$

where $\epsilon^* = (\epsilon_{\mathbf{1}}^*, \cdots, \epsilon_{\mathbf{n}}^*)^{\mathbf{T}} \approx \mathbf{N}(\mathbf{0}, \hat{\sigma}\mathbf{I}_{\mathbf{n}\times\mathbf{n}})$. Denote $f_{\hat{\lambda}}^*(x_i)$ as the random variable of the bootstrap estimate at x_i. Repeat this process K times, so that at each point x_i, you have K bootstrap estimates $\hat{f}_{\hat{\lambda}}(x_i)$ or K realizations of $f_{\hat{\lambda}}^*(x_i)$. For each fixed x_i, consider the following statistic D_i^*, which is similar to a Student's t statistic:

$$D_i^* = (f_{\hat{\lambda}}^*(x_i) - \hat{f}_{\hat{\lambda}}(x_i))/\hat{\sigma}_i{}^*$$

where $\hat{\sigma_i}^*$ is the estimate of $\hat{\sigma}$ based on the ith bootstrap sample.

Suppose $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$ are the lower and upper $\alpha/2$ points of the empirical distribution of D_i^*. The $(1-\alpha)100\%$ bootstrap confidence interval is defined as

$$(\hat{f}_{\hat{\lambda}}(x_i) - \chi_{1-\alpha/2}\hat{\sigma},\ \ \hat{f}_{\hat{\lambda}}(x_i) - \chi_{\alpha/2}\hat{\sigma})$$

Bootstrap confidence intervals are easy to interpret and can be used with any distribution. However, because they require K model fits, their construction is computationally intensive.

The multiple dependent variables feature in PROC TPSPLINE enables you to fit multiple models with the same independent variables. The procedure calculates the matrix decomposition part of the calculations only once regardless of the number of dependent variables in the model. These calculations are responsible for most of the computing time used by the TPSPLINE procedure. This feature is particularly useful when you need to generate a bootstrap confidence interval.

To construct a bootstrap confidence interval, perform the following tasks:

- Fit the data using PROC TPSPLINE and obtain estimates $\hat{f}_{\hat{\lambda}}(x_i)$ and $\hat{\sigma}$.
- Generate K bootstrap samples based on $\hat{f}_{\hat{\lambda}}(x_i)$ and $\hat{\sigma}$.
- Fit the K bootstrap samples with the TPSPLINE procedure to obtain estimates of $\hat{f}^*_{\hat{\lambda}}(x_i)$ and $\hat{\sigma}^*_i$.
- Compute D_i^* and the values $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$.

The following statements illustrate this process:

```
proc tpspline data=melanoma;
   model incidences = (year) /alpha = 0.05;
   output out = result pred uclm lclm;
run;
```

The output from the initial PROC TPSPLINE analysis is displayed in Output 74.5.3. The data set result contains the predicted values and confidence limits from the analysis.

Output 74.5.3. Output from PROC TPSPLINE for the Melanoma Data Set

```
                         The TPSPLINE Procedure
                    Dependent Variable:  incidences

                         Summary of Input Data Set

                  Number of Non-Missing Observations    37
                  Number of Missing Observations         0
                  Unique Smoothing Design Points        37


                          Summary of Final Model

                  Number of Regression Variables        0
                  Number of Smoothing Variables         1
                  Order of Derivative in the Penalty    2
                  Dimension of Polynomial Space         2


                            Summary Statistics
                           of Final Estimation

                     log10(n*Lambda)            -0.0607
                     Smoothing Penalty           0.5171
                     Residual SS                 1.2243
                     Tr(I-A)                    22.5852
                     Model DF                   14.4148
                     Standard Deviation          0.2328
```

The following statements illustrate how you can obtain a bootstrap confidence interval for the Melanoma data. The following statements create the data set bootstrap. The observations are created with information from the preceding PROC TPSPLINE execution; as displayed in Output 74.5.3, $\hat{\sigma} = 0.232823$. The values of $\hat{f}_{\hat{\lambda}}(x_i)$ are stored in the data set result in the variable P_incidence.

```
   data bootstrap;
      set result;
      array y{1070} y1-y1070;
      do i=1 to 1070;
         y{i} = p_incidences + 0.232823*rannor(123456789);
      end;
      keep y1-y1070 p_incidences year;
   run;

  ods listing close;

   proc tpspline data=bootstrap;
      ods output FitStatistics=FitResult;
      id p_incidences;
      model y1-y1070 = (year);
      output out=result2;
   run;
   ods listing;
```

The DATA step generates 1,070 bootstrap samples based on the previous estimate from PROC TPSPLINE. For this data set, some of the bootstrap samples result in λs (selected by the GCV function) that cause problematic behavior. Thus, an additional 70 bootstrap samples are generated.

The ODS listing destination is closed before PROC TPSPLINE is invoked. The model fits all the y1–y1070 variables as dependent variables, and the models are fit for all bootstrap samples simultaneously. The output data set result2 contains the variables year, y1–y1070, p_y1–p_y1070, and P_incidences.

The ODS OUTPUT statement writes the FitStatistics table to the data set FitResult. The data set FitResult contains the two variables. They are Parameter and Value. The FitResult data set is used in subsequent calculations for D_i^*.

In the data set FitResult, there are 63 estimates with a standard deviation of zero, suggesting that the estimates provide perfect fits of the data and are caused by $\hat{\lambda}$s that are approximately equal to zero. For small sample sizes, there is a positive probability that the λ chosen by the GCV function will be zero (refer to Wang and Wahba 1995).

In the following steps, these cases are removed from the bootstrap samples as “bad” samples: they represent failure of the GCV function.

The following SAS statements manipulate the data set FitResult, retaining the standard deviations for all bootstrap samples and merging FitResult with the data set result2, which contains the estimates for bootstrap samples. In the final data set boot, the D_i^* statistics are calculated.

```
data FitResult; set FitResult;
   if Parameter="Standard Deviation";
   keep Value;
run;

proc transpose data=FitResult out=sd prefix=sd;

data result2;
   if _N_ = 1 then set sd;
   set result2;

data boot;
   set result2;
   array y{1070}  p_y1-p_y1070;
   array sd{1070} sd1-sd1070;
   do i=1 to 1070;
      if sd{i} > 0 then do;
         d = (y{i} - P_incidences)/sd{i};
         obs = _N_;
         output;
      end;
   end;
   keep d obs P_incidences year;
run;
```

The following SAS statements retain the first 1000 bootstrap samples and calculate the values $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$ with $\alpha = 0.1$.

```
proc sort data=boot;
   by obs;
run;

data boot;
   set boot;
      by obs;
   retain n;

   if first.obs then n=1;
      else n=n+1;
   if n > 1000 then delete;
run;


proc sort data=boot;
   by obs d;
run;

data chi1 chi2 ;
   set boot;
   if (_N_ = (obs-1)*1000+50)  then output chi1;
   if (_N_ = (obs-1)*1000+950) then output chi2;
run;

proc sort data=result;
   by year;
run;

proc sort data=chi1;
   by year;
run;

proc sort data=chi2;
   by year;
run;

data result;
   merge result
      chi1(rename=(d=chi05))
      chi2(rename=(d=chi95));
   keep year incidences P_incidences lower upper
        LCLM_incidences UCLM_incidences;

   lower = -chi95*0.232823 + P_incidences;
   upper = -chi05*0.232823 + P_incidences;

   label  lower="Lower 90% CL (Bootstrap)"
          upper="Upper 90% CL (Bootstrap)"
          lclm_incidences="Lower 90% CL (Bayesian)"
          uclm_incidences="Upper 90% CL (Bayesian)";
run;
```

The data set result contains the variables year, incidences, the TPSPLINE estimate

P_incidences, and the 90% Bayesian and 90% bootstrap confidence intervals.

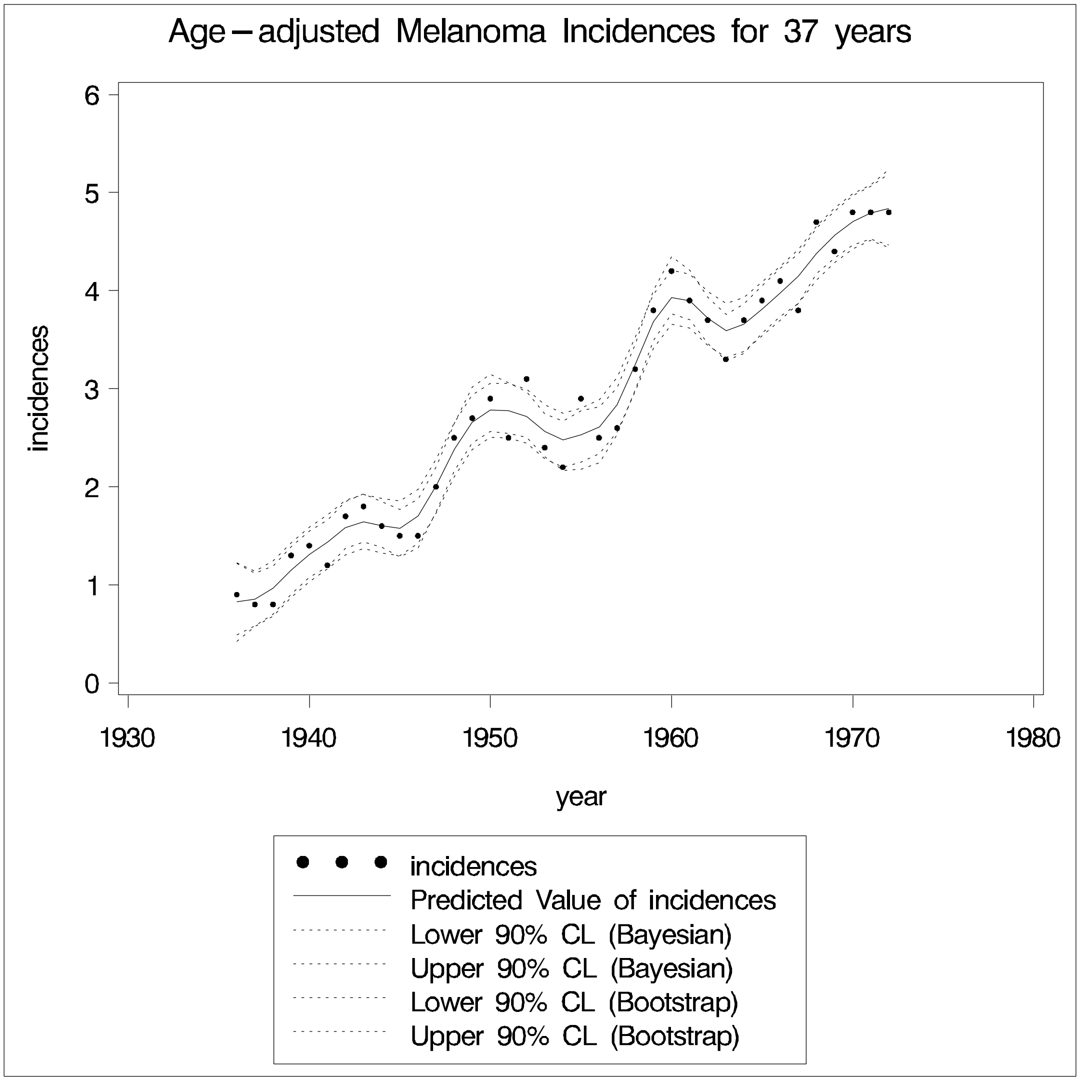
The following statements produce Output 74.5.4:

```
symbol1  v=dot  h=1pct ;
symbol2  i=join v=none l=1;
symbol3  i=join v=none l=33;
symbol4  i=join v=none l=33;
symbol5  i=join v=none l=43 c=green;
symbol6  i=join v=none l=43 c=green;

title1 'Age-adjusted Melanoma Incidences for 37 years';
proc gplot data=result;
   plot       incidences * year=1
            p_incidences * year=2
         lclm_incidences * year=3
         uclm_incidences * year=3
                   lower * year=4
                   upper * year=4
         /overlay      legend=legend1
          vaxis=axis1  haxis=axis2
          frame        cframe=ligr;
run;
```

Output 74.5.4 displays the plot of the variable incidences, the predicted values, and the Bayesian and bootstrap confidence intervals.

The plot shows that the bootstrap confidence interval is similar to the Bayesian confidence interval. However, the Bayesian confidence interval is symmetric around the estimates, while the bootstrap confidence interval is not.

Output 74.5.4. Comparison of Bayesian and Bootstrap Confidence Interval for Melanoma Data



References

Andrews, D (1988), *Asymptotic Optimality of GC_L, Cross-Validation, and GCV in Regression with Heteroscedastic Errors*, Cowles Foundation, Yale University, New Haven, CT, manuscript.

Bates, D.; Lindstrom, M.; Wahba, G. and Yandell, B. (1987), "GCVPACK-Routines for Generalized Cross Validation," *Comm. Statist. B-Simulation Comput.*, 16, 263–297.

Duchon, J. (1976), "Fonctions-Spline et Esperances Conditionnelles de Champs Gaussiens," *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14, 19–27.

Duchon, J. (1977), "Splines Minimizing Rotation-Invariant Semi-Norms in Sovolev Spaces," in *Constructive Theory of Functions of Several Variables*, eds. W.

Schempp and K. Zeller, 85–100.

Dongarra, J.; Bunch, J.; Moler, C. and Steward, G. (1979), *Linpack Users' Guide*, Philadelphia: Society for Industrial and Applied Mathematics.

Hall, P. and Titterington, D. (1987), "Common Structure of Techniques for Choosing Smoothing Parameters in Regression Problems," *J. Roy. Statist. Soc. Ser. B*, 49, 184–198.

Houghton, A. N.; Flannery, J. and Viola, M. V. (1980), "Malignant Melanoma in Connecticut and Denmark," *International Journal of Cancer*, 25, 95–104.

Hutchinson, M. and Bischof, R. (1983), "A New Method for Estimating the Spatial Distribution of Mean Seasonal and Annual Rainfall Applied to the Hunter Valley," *New South Wales, Aust. Met. Mag.*, 31, 179–184.

Meinguet, J. (1979), "Multivariate Interpolation at Arbitrary Points Made Simple," *J. Appl. Math. Phys. (ZAMP)*, 5, 439–468.

Nychka, D (1986a), "The Average Posterior Variance of a Smoothing Spline and a Consistent Estimate of the Mean Square Error," Tech. Report 168, The Institute of Statistics, North Carolina State University, Raleigh, NC.

Nychka, D (1986b), "A Frequency Interpretation of Bayesian "Confidence" Interval for Smoothing Splines," Tech. Report 169, The Institute of Statistics, North Carolina State University, Raleigh, NC.

Nychka, D (1988), "Confidence Intervals for Smoothing Splines," *J. Amer. Statist. Assoc.*, 83, 1134–1143.

O'Sullivan, F. and Wong, T. (1987), "Determining a Function Diffusion Coefficient in the Heat Equation," Tech. Report 98, Department of Statistics, University of California, Berkeley, CA.

Ramsay, J. and Silverman, B (1997), *Functional Data Analysis*, New York: Springer-Verlag.

Seaman, R. and Hutchinson, M. (1985), "Comparative Real Data Tests of Some Objective Analysis Methods by Withholding," *Aust. Met. Mag.*, 33, 37–46.

Villalobos, M. and Wahba, G. (1987), "Inequality Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities," *J. Amer. Statist. Assoc.*, 82 239–248.

Wahba, G. (1983), "Bayesian "Confidence Intervals" for the Cross Validated Smoothing Spline," *J. Roy. Statist. Soc. Ser. B*, 45, 133–150.

Wahba, G., (1990), *Spline Models for Observational Data*, Philadelphia: Society for Industrial and Applied Mathematics.

Wahba, G. and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation," *Monthly Weather Rev.*, 108, 1122–1145.

Wang, Y. and Wahba, G. (1995), "Bootstrap Confidence Intervals for Smoothing Splines and their Comparison to Bayesian Confidence Intervals," *J. Statistical Computation and Simulation*, 51, 263–279.

Chapter 75
The TRANSREG Procedure

Chapter Contents

Chapter 75
The TRANSREG Procedure

Overview

The TRANSREG (transformation regression) procedure fits linear models, optionally with spline and other nonlinear transformations, and it can be used to code experimental designs prior to their use in other analyses.

The TRANSREG procedure fits many types of linear models, including

- ordinary regression and ANOVA
- metric and nonmetric conjoint analysis (Green and Wind 1975; de Leeuw, Young, and Takane 1976)
- metric and nonmetric vector and ideal point preference mapping (Carroll 1972)
- simple, multiple, and multivariate regression with variable transformations (Young, de Leeuw, and Takane 1976; Winsberg and Ramsay 1980; Breiman and Friedman 1985)
- redundancy analysis (Stewart and Love 1968) with variable transformations (Israels 1984)
- canonical correlation analysis with variable transformations (van der Burg and de Leeuw 1983)
- response surface regression (Meyers 1976; Khuri and Cornell 1987) with variable transformations
- linear models with Box-Cox (1964) transformations of the dependent variables

The data set can contain variables measured on nominal, ordinal, interval, and ratio scales (Siegel 1956). Any mix of these variable types is allowed for the dependent and independent variables. The TRANSREG procedure can transform

- nominal variables by scoring the categories to minimize squared error (Fisher 1938), or they can be expanded into dummy variables
- ordinal variables by monotonically scoring the ordered categories so that order is weakly preserved (adjacent categories can be merged) and squared error is minimized. Ties can be optimally untied or left tied (Kruskal 1964). Ordinal variables can also be transformed to ranks.
- interval and ratio scale of measurement variables linearly or nonlinearly with spline (de Boor 1978; van Rijckevorsel 1982) or monotone spline (Winsberg and Ramsay 1980) transformations. In addition, smooth, logarithmic, exponential, power, logit, and inverse trigonometric sine transformations are available.

Transformations produced by the PROC TRANSREG multiple regression algorithm, requesting spline transformations, are often similar to transformations produced by the ACE smooth regression method of Breiman and Friedman (1985). However, ACE does not explicitly optimize a loss function (de Leeuw 1986), while PROC TRANSREG always explicitly optimizes a squared-error loss function.

PROC TRANSREG extends the ordinary general linear model by providing optimal variable transformations that are iteratively derived using the method of alternating least squares (Young 1981). PROC TRANSREG iterates until convergence, alternating

- finding least-squares estimates of the parameters of the model given the current scoring of the data (that is, the current vectors)
- finding least-squares estimates of the scoring parameters given the current set of model parameters

For more background on alternating least-squares optimal scaling methods and transformation regression methods, refer to Young, de Leeuw, and Takane (1976), Winsberg and Ramsay (1980), Young (1981), Gifi (1990), Schiffman, Reynolds, and Young (1981), van der Burg and de Leeuw (1983), Israels (1984), Breiman and Friedman (1985), and Hastie and Tibshirani (1986). (These are just a few of the many relevant sources.)

Getting Started

This section provides several examples that illustrate features of the TRANSREG procedure.

Main-Effects ANOVA

This example shows how to use the TRANSREG procedure to code and fit a main-effects ANOVA model. The input data set contains the dependent variables Y, factors X1 and X2, and 11 observations. The following statements perform a main-effects ANOVA:

```
title 'Introductory Main-Effects ANOVA Example';

data A;
   input Y X1 $ X2 $;
   datalines;
8 a a
7 a a
4 a b
3 a b
5 b a
4 b a
2 b b
1 b b
8 c a
```

```
7 c a
5 c b
2 c b
;

*---Fit a Main-Effects ANOVA model with 1, 0, -1 coding. ---;
proc transreg ss2;
   model identity(Y) = class(X1 X2 / effects);
   output coefficients replace;
run;

*---Print TRANSREG output data set---;
proc print label;
   format Intercept -- X2a 5.2;
run;
```

```
                  Introductory Main-Effects ANOVA Example

                           The TRANSREG Procedure

                       Dependent Variable Identity(Y)


                          Class Level Information

                         Class    Levels    Values

                         X1            3    a b c

                         X2            2    a b


                 Number of Observations Read             12
                 Number of Observations Used             12


      TRANSREG Univariate Algorithm Iteration History for Identity(Y)

 Iteration    Average    Maximum                Criterion
    Number     Change     Change    R-Square       Change    Note
 -------------------------------------------------------------------------
         1    0.00000    0.00000     0.88144                 Converged

 Algorithm converged.


          The TRANSREG Procedure Hypothesis Tests for Identity(Y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
    Source                DF     Squares       Square    F Value    Pr > F

    Model                  3    57.00000    19.00000      19.83    0.0005
    Error                  8     7.66667     0.95833
    Corrected Total       11    64.66667


              Root MSE            0.97895    R-Square    0.8814
              Dependent Mean      4.66667    Adj R-Sq    0.8370
              Coeff Var          20.97739


     Univariate Regression Table Based on the Usual Degrees of Freedom

                               Type II
                                Sum of     Mean
Variable      DF  Coefficient  Squares   Square  F Value  Pr > F  Label

Intercept      1    4.6666667  261.333  261.333   272.70  <.0001  Intercept
Class.X1a      1    0.8333333    4.167    4.167     4.35  0.0705  X1 a
Class.X1b      1   -1.6666667   16.667   16.667    17.39  0.0031  X1 b
Class.X2a      1    1.8333333   40.333   40.333    42.09  0.0002  X2 a
```

Figure 75.1. ANOVA Example Output from PROC TRANSREG

The iteration history in Figure 75.1 shows that the final R-Square of 0.88144 is reached on the first iteration.

This is followed by ANOVA, fit statistics, and regression tables. PROC TRANSREG uses an effects (also called deviations from means or 0, 1, -1) coding in this example. For more information on using PROC TRANSREG for ANOVA and other codings, see the "ANOVA Codings" section on page 4662.

The TRANSREG procedure produces the data set displayed in Figure 75.2.

```
                    Introductory Main-Effects ANOVA Example

   Obs    _TYPE_     _NAME_    Y    Intercept    X1 a     X1 b     X2 a    X1    X2

     1    SCORE      ROW1      8      1.00       1.00     0.00     1.00    a     a
     2    SCORE      ROW2      7      1.00       1.00     0.00     1.00    a     a
     3    SCORE      ROW3      4      1.00       1.00     0.00    -1.00    a     b
     4    SCORE      ROW4      3      1.00       1.00     0.00    -1.00    a     b
     5    SCORE      ROW5      5      1.00       0.00     1.00     1.00    b     a
     6    SCORE      ROW6      4      1.00       0.00     1.00     1.00    b     a
     7    SCORE      ROW7      2      1.00       0.00     1.00    -1.00    b     b
     8    SCORE      ROW8      1      1.00       0.00     1.00    -1.00    b     b
     9    SCORE      ROW9      8      1.00      -1.00    -1.00     1.00    c     a
    10    SCORE      ROW10     7      1.00      -1.00    -1.00     1.00    c     a
    11    SCORE      ROW11     5      1.00      -1.00    -1.00    -1.00    c     b
    12    SCORE      ROW12     2      1.00      -1.00    -1.00    -1.00    c     b
    13    M COEFFI   Y         .      4.67       0.83    -1.67     1.83
    14    MEAN       Y         .       .         5.50     3.00     6.50
```

Figure 75.2. Output Data Set from PROC TRANSREG

The output data set has three kinds of observations, identified by values of _TYPE_.

- When _TYPE_='SCORE', the observation contains information on the dependent and independent variables as follows:
  - Y is the original dependent variable.
  - X1 and X2 are the independent classification variables, and the Intercept through X2 a columns contain the main effects design matrix that PROC TRANSREG creates. The variable names are Intercept, X1a, X1b, and X2a. Their labels are shown in the listing.
- When _TYPE_='M COEFFI', the observation contains coefficients of the final linear model.
- When _TYPE_='MEAN', the observation contains the marginal means.

The observations with _TYPE_='SCORE' form the score partition of the data set, and the observations with _TYPE_='M COEFFI' and _TYPE_='MEAN' form the coefficient partition of the data set.

Detecting Nonlinear Relationships

The TRANSREG procedure can detect nonlinear relationships among variables. For example, suppose 400 observations are generated from the following function

$$t = \frac{x}{4} + \sin(x)$$

and data are created as follows

$$y = t + \epsilon$$

where ϵ is random normal error.

The following statements find a cubic spline transformation of X with four knots. For information on using splines and knots, see the "Smoothing Splines" section on page 4596, the "Solving Standard Least-Squares Problems" section on page 4628, Example 75.1, and Example 75.4.

The following statements produce Figure 75.3 through Figure 75.4:

```
title 'Curve Fitting Example';

*---Create An Artificial Nonlinear Scatter Plot---;
data Curve;
   Pi=constant('pi');
   Pi4=4*Pi;
   Increment=Pi4/400;
   do X=Increment to Pi4 by Increment;
      T=X/4 + sin(X);
      Y=T + normal(7);
      output;
      end;
run;

*---Request a Spline Transformation of X---;
proc transreg data=Curve dummy;
   model identity(Y)=spline(X / nknots=4);
   output predicted;
   id T;
run;

*---Plot the Results---;
goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot;
   title;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   plot T*X=2              / &opts name='tregin1';
   plot Y*X=1              / &opts name='tregin2';
   plot Y*X=1 T*X=2 PY*X=3 / &opts name='tregin3' overlay ;
   symbol1 color=blue   v=star i=none;

   symbol2 color=yellow v=none i=join line=1;
   symbol3 color=red    v=none i=join line=2;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:tregin1 2:tregin3 3:tregin2;
run; quit;
```

PROC TRANSREG increases the squared multiple correlation from the original value of 0.19945 to 0.47062. The plot of T by X shows the original function, the plot of Y by X shows the error-perturbed data, and the third plot shows the data, the true function as a solid curve, and the regression function as the dashed curve. The regression function closely approximates the true function.

```
                          Curve Fitting Example

                          The TRANSREG Procedure

       TRANSREG MORALS Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.74855    1.29047     0.19945
        1    0.00000    0.00000     0.47062      0.27117    Converged

Algorithm converged.
```

Figure 75.3. Curve Fitting Example Output

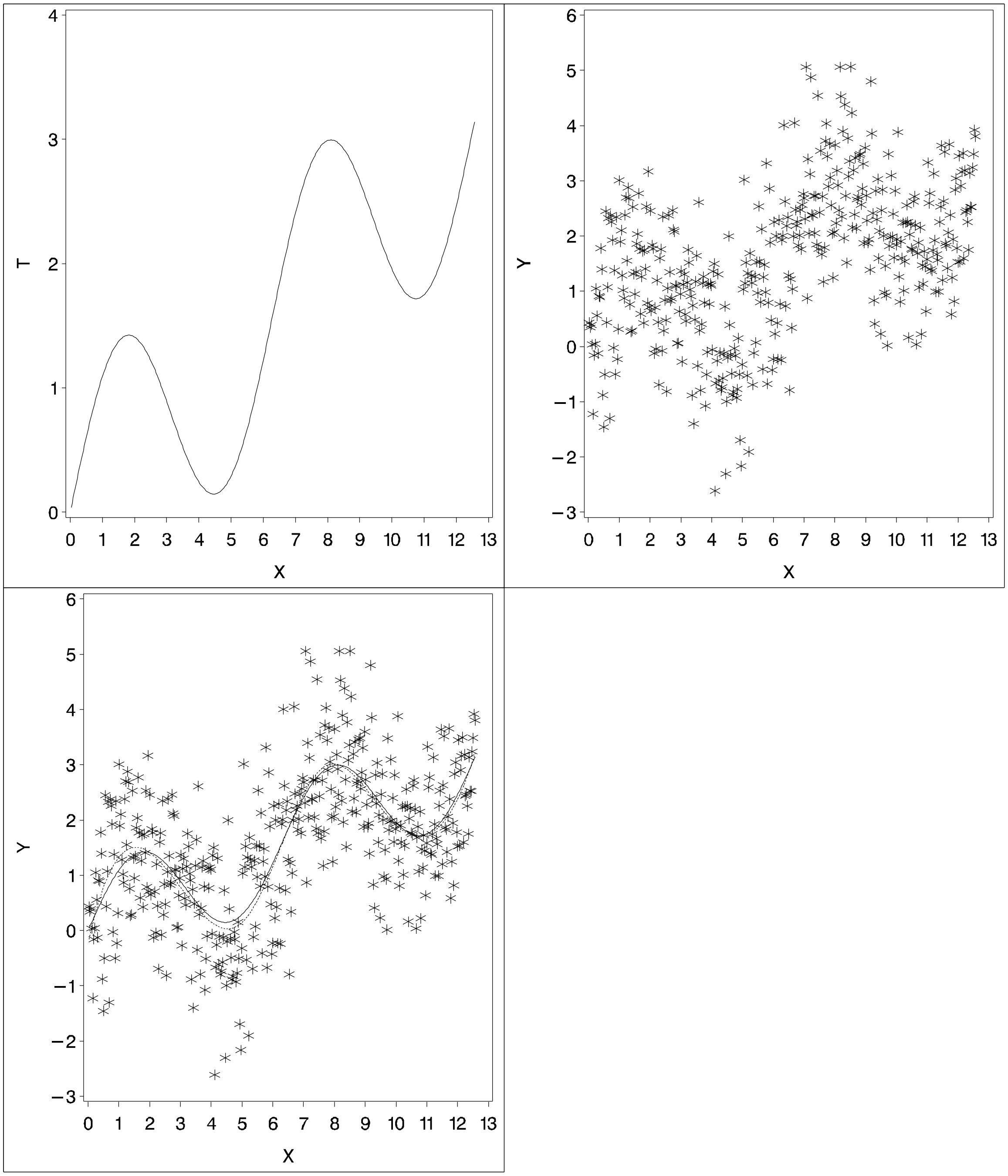


Figure 75.4. Plots for the Curve Fitting Example

Syntax

The following statements are available in PROC TRANSREG.

PROC TRANSREG < **DATA=***SAS-data-set* >
        < **OUTTEST=***SAS-data-set* >< *a-options* >< *o-options* > **;**
  MODEL < *transform(dependents < / t-options >)*
        < transform(dependents < / t-options >)...> = >
        transform(independents < / t-options >)
        < transform(independents < / t-options >)...>< / a-options > **;**
  OUTPUT < **OUT=***SAS-data-set* >< *o-options* > **;**
  ID *variables* **;**
  FREQ *variable* **;**
  WEIGHT *variable* **;**
  BY *variables* **;**

To use the TRANSREG procedure, you need the PROC TRANSREG and MODEL statements. To produce an OUT= output data set, the OUTPUT statement is required. PROC TRANSREG enables you to specify the same options in more than one statement. All of the MODEL statement *a-options* (algorithm options) and all of the OUTPUT statement *o-options* (output options) can also be specified in the PROC TRANSREG statement. You can abbreviate all *a-options*, *o-options*, and *t-options* (transformation options) to their first three letters. This is a special feature of the TRANSREG procedure and is not generally true of other SAS/STAT procedures. See Table 75.1 on page 4554.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC TRANSREG statement. The remaining statements are described in alphabetical order.

PROC TRANSREG Statement

PROC TRANSREG < **DATA=***SAS-data-set* >
        < **OUTTEST=***SAS-data-set* >< *a-options*
      >< *o-options* > **;**

The PROC TRANSREG statement starts the TRANSREG procedure. Optionally, this statement identifies an input and an OUTTEST= data set, specifies the algorithm and other computational details, requests displayed output, and controls the contents of the OUT= data set (which is created with the OUTPUT statement). The DATA= and OUTTEST= options can appear only in the PROC TRANSREG statement.

The following table summarizes options available in the PROC TRANSREG statement. All *a-options* and *o-options* are described in the sections on either the MODEL or OUTPUT statement, in which these options can also be specified.

Table 75.1. Options Available in the TRANSREG Procedure

Task	Option	Statement
Identify input data set		
specifies input SAS data set	DATA=	PROC
Output data set with test statistics		
specifies output test statistics data set	OUTTEST=	PROC
Input data set		
specifies input observation type	TYPE=	MODEL
restarts iterations	REITERATE	MODEL
Specify method and control iterations		
specifies minimum criterion change	CCONVERGE=	MODEL
specifies minimum data change	CONVERGE=	MODEL
specifies canonical dummy-variable initialization	DUMMY	MODEL
specifies maximum number of iterations	MAXITER=	MODEL
specifies iterative algorithm	METHOD=	MODEL
specifies number of canonical variables	NCAN=	MODEL
specifies singularity criterion	SINGULAR=	MODEL
Control missing data handling		
METHOD=MORALS fists each model individually	INDIVIDUAL	MODEL
includes monotone special missing values	MONOTONE=	MODEL
excludes observations with missing values	NOMISS	MODEL
unties special missing values	UNTIE=	MODEL
Control intercept and CLASS variables		
CLASS dummy variable name prefix	CPREFIX=	MODEL
CLASS dummy variable label prefix	LPREFIX=	MODEL
no intercept or centering	NOINT	MODEL
order of class variable levels	ORDER=	MODEL
controls output of reference levels	REFERENCE=	MODEL
CLASS dummy variable label separators	SEPARATORS=	MODEL
Control displayed output		
confidence limits alpha	ALPHA=	MODEL
displays parameter estimate confidence limits	CL	MODEL
displays model specification details	DETAIL	MODEL
displays iteration histories	HISTORY	MODEL
suppresses displayed output	NOPRINT	MODEL
suppresses the iteration histories	SHORT	MODEL
displays regression results	SS2	MODEL
displays ANOVA table	TEST	MODEL
displays conjoint part-worth utilities	UTILITIES	MODEL
Control standardization		
fits additive model	ADDITIVE	MODEL
do not zero constant variables	NOZEROCONSTANT	MODEL
specifies transformation standardization	TSTANDARD=	MODEL
Predicted values, residuals, scores		
outputs canonical scores	CANONICAL	OUTPUT
outputs individual confidence limits	CLI	OUTPUT

Table 75.1. (continued)

Task	Option	Statement
outputs mean confidence limits	CLM	OUTPUT
specifies design matrix coding	DESIGN=	OUTPUT
outputs leverage	LEVERAGE	OUTPUT
does not restore missing values	NORESTOREMISSING	OUTPUT
suppresses output of scores	NOSCORES	OUTPUT
outputs predicted values	PREDICTED	OUTPUT
outputs redundancy variables	REDUNDANCY=	OUTPUT
outputs residuals	RESIDUALS	OUTPUT
Output data set replacement		
replaces dependent variables	DREPLACE	OUTPUT
replaces independent variables	IREPLACE	OUTPUT
replaces all variables	REPLACE	OUTPUT
Output data set coefficients		
outputs coefficients	COEFFICIENTS	OUTPUT
outputs ideal point coordinates	COORDINATES	OUTPUT
outputs marginal means	MEANS	OUTPUT
outputs redundancy analysis coefficients	MREDUNDANCY	OUTPUT
Output data set variable name prefixes		
dependent variable approximations	ADPREFIX=	OUTPUT
independent variable approximations	AIPREFIX=	OUTPUT
canonical dependent variables	CDPREFIX=	OUTPUT
conservative individual lower CL	CILPREFIX=	OUTPUT
canonical independent variables	CIPREFIX=	OUTPUT
conservative-individual-upper CL	CIUPREFIX=	OUTPUT
conservative-mean-lower CL	CMLPREFIX=	OUTPUT
conservative-mean-upper CL	CMUPREFIX=	OUTPUT
METHOD=MORALS untransformed dependent	DEPENDENT=	OUTPUT
liberal-individual-lower CL	LILPREFIX=	OUTPUT
liberal-individual-upper CL	LIUPREFIX=	OUTPUT
liberal-mean-lower CL	LMLPREFIX=	OUTPUT
liberal-mean-upper CL	LMUPREFIX=	OUTPUT
residuals	RDPREFIX=	OUTPUT
predicted values	PPREFIX=	OUTPUT
redundancy variables	RPREFIX=	OUTPUT
transformed dependents	TDPREFIX=	OUTPUT
transformed independents	TIPREFIX=	OUTPUT
Output data set macros		
creates macro variables	MACRO	OUTPUT
Output data set details		
dependent and independent approximations	APPROXIMATIONS	OUTPUT
canonical correlation coefficients	CCC	OUTPUT
canonical elliptical point coordinate	CEC	OUTPUT
canonical point coordinates	CPC	OUTPUT
canonical quadratic point coordinates	CQC	OUTPUT

Table 75.1. (continued)

Task	**Option**	**Statement**
approximations to transformed dependents	DAPPROXIMATIONS	OUTPUT
approximations to transformed independents	IAPPROXIMATIONS	OUTPUT
elliptical point coordinates	MEC	OUTPUT
point coordinates	MPC	OUTPUT
quadratic point coordinates	MQC	OUTPUT
multiple regression coefficients	MRC	OUTPUT

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. If you do not specify the DATA= option, PROC TRANSREG uses the most recently created SAS data set. The data set must be an ordinary SAS data set; it cannot be a special TYPE= data set.

OUTTEST=*SAS-data-set*

specifies an output data set to contain hypothesis tests results. When you specify the OUTTEST= option, the data set contains ANOVA results. When you specify the SS2 *a-option*, regression tables are also output. When you specify the UTILITIES *o-option*, conjoint analysis part-worth utilities are also output. For more information on the OUTTEST= data set, see the "OUTTEST= Output Data Set" section on page 4626.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TRANSREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TRANSREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC TRANSREG then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ statement variable is greater than or equal to 1.

ID Statement

ID *variables* **;**

The ID statement includes additional character or numeric variables in the OUT= data set. The variables must be contained in the input data set.

MODEL Statement

MODEL < *transform(dependents* < */ t-options* >*)*
< *transform(dependents* < */ t-options* >*)...*> = >
transform(independents < */ t-options* >*)*
< *transform(independents* < */ t-options* >*)...*>< */ a-options* > **;**

The MODEL statement specifies the dependent and independent variables (*dependents* and *independents*, respectively) and specifies the transformation (*transform*) to apply to each variable. Only one MODEL statement can appear in the TRANSREG procedure. The *t-options* are transformation options, and the *a-options* are the algorithm options. The *t-options* provide details for the transformation; these depend on the *transform* chosen. The *t-options* are listed after a slash in the parentheses that enclose the variable list (either *dependents* or *independents*). The *a-options* control the algorithm used, details of iteration, details of how the intercept and dummy variables are generated, and displayed output details. The *a-options* are listed after the entire model specification (the *dependents*, *independents*, transformations, and *t-options*) and after a slash. You can also specify the algorithm options in the PROC TRANSREG statement. When you specify the DESIGN *o-option*, *dependents* and an equal sign are not required. The operators "*", "|", and "@" from the GLM procedure are available for interactions with the CLASS expansion and the IDENTITY transformation.

```
   Class(a * b ...
         c | d ...
         e | f ... @ n)
Identity(a * b ...
         c | d ...
         e | f ... @ n)
```

In addition, transformations and spline expansions can be crossed with classification variables:

```
transform(var) * class(group)
transform(var) | class(group)
```

See the “Types of Effects” section on page 1784 in Chapter 32, “The GLM Procedure,” for a description of the @, *, and | operators and see the “Model Statement Usage” section on page 4592 for information on how to use these operators in PROC TRANSREG. Note that nesting is not allowed in PROC TRANSREG.

The next three sections discuss the transformations available (*transforms*) (see the “Families of Transformations” section on page 4558), the transformation options (*t-options*) (see the “Transformation Options (t-options)” section on page 4564), and the algorithm options (*a-options*) (see the “Algorithm Options (a-options)” section on page 4573).

Families of Transformations

In the MODEL statement, *transform* specifies a transformation in one of four families.

Variable expansions
preprocess the specified variables, replacing them with more variables.

Nonoptimal transformations
preprocess the specified variables, replacing each one with a single new nonoptimal, nonlinear transformation.

Optimal transformations
replace the specified variables with new, iteratively derived optimal transformation variables that fit the specified model better than the original variable (except for contrived cases where the transformation fits the model exactly as well as the original variable).

Other transformations
are the IDENTITY and SSPLINE transformations. These do not fit into the preceding categories.

The following table summarizes the transformations in each family.

Family	Members of Family
Variable expansions	
B-spline basis	BSPLINE
set of dummy variables	CLASS
elliptical response surface	EPOINT
circular response surface	POINT
piecewise polynomial basis	PSPLINE
quadratic response surface	QPOINT
Nonoptimal transformations	
inverse trigonometric sine	ARSIN
Box-Cox	BOXCOX
exponential	EXP
logarithm	LOG
logit	LOGIT
raises variables to specified power	POWER
transforms to ranks	RANK
noniterative smoothing spline	SMOOTH
Optimal transformations	
linear	LINEAR
monotonic, ties preserved	MONOTONE
monotonic B-spline	MSPLINE
optimal scoring	OPSCORE
B-spline	SPLINE
monotonic, ties not preserved	UNTIE
Other transformations	
identity, no transformation	IDENTITY
iterative smoothing spline	SSPLINE

You can use any transformation with either dependent or independent variables (except the SMOOTH transformation, which can be used only with independent variables, and BOXCOX, which can be used only with dependent variables). However, the variable expansions are usually more appropriate for independent variables.

The *transform* is followed by a variable (or list of variables) enclosed in parentheses. Optionally, depending on the *transform*, the parentheses can also contain *t-options*, which follow the variables and a slash. For example,

```
   model log(y)=class(x);
```

finds a LOG transformation of Y and performs a CLASS expansion of X.

```
   model identity(y) = spline(x1 x2 / nknots=3);
```

The preceding statement finds SPLINE transformations of X1 and X2. The NKNOTS= *t-option* used with the SPLINE transformation specifies three knots. The IDENTITY(Y) transformation specifies that Y is not to be transformed.

The rest of this section provides syntax details for members of the four families of transformations. The *t-options* are discussed in the "Transformation Options (t-options)" section on page 4564.

Variable Expansions

The TRANSREG procedure performs variable expansions before iteration begins. Variable expansions expand the original variables into a typically larger set of new variables. The original variables are those that are listed in parentheses after *transform*, and they are sometimes referred to by the name of the *transform*. For example, in CLASS(X1 X2), X1 and X2 are sometimes referred to as CLASS expansion variables or simply CLASS variables, and the expanded variables are referred to as dummy variables. Similarly, in POINT(Dim1 Dim2), Dim1 and Dim2 are sometimes referred to as POINT variables.

The resulting variables are not transformed by the iterative algorithms after the initial preprocessing. Observations with missing values for these types of variables are excluded from the analysis.

The POINT, EPOINT, and QPOINT variable expansions are used in preference mapping analyses (also called PREFMAP, external unfolding, ideal point regression) (Carroll 1972) and for response surface regressions. These three expansions create circular, elliptical, and quadratic response or preference surfaces (see the "Point Models" section on page 4605 and Example 75.5). The CLASS variable expansion is used for main effects ANOVA.

The following list provides syntax and details for the variable expansion *transforms*.

BSPLINE
BSP

expands each variable to a B-spline basis. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with the BSPLINE expansion. When DEGREE=n (3 by default) with k knots (0 by default), $n + k + 1$ variables are created. In addition, the original variable appears in the OUT= data set before the ID variables. For example, BSPLINE(X) expands X into X_0 X_1 X_2 X_3 and outputs X as well. The X_: variables contain the B-spline (which are the same basis vectors that the SPLINE and MSPLINE transformations use internally). The columns of the BSPLINE expansion sum to a column of ones, so an implicit intercept model is fit when the BSPLINE expansion is specified. If you specify the BSPLINE expansion for more than one variable, the model is less than full rank. See the section "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 4614. Variables following BSPLINE must be numeric, and they are typically continuous.

CLASS
CLA

expands the variables to a set of dummy variables. For example, CLASS(X1 X2) is used for a simple main-effects model, CLASS(X1 | X2) fits a main-effects and interactions model, and CLASS(X1|X2|X3|X4@2 X1*X2*X3) creates all main effects, all two-way interactions, and one three-way interaction. See the "Model Statement Usage" section on page 4592 for information on how to use the operators @, *, and | in PROC TRANSREG. To determine class membership, PROC TRANSREG uses the values of the formatted variables. Variables following CLASS can be either character or numeric; numeric variables should be discrete.

EPOINT
EPO

expands the variables for an elliptical response surface regression or for an elliptical ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP ideal elliptical point model coordinates to the OUT= data set. Each axis of the ellipse (or ellipsoid) is oriented in the same direction as one of the variables. The EPOINT expansion creates a new variable for each original variable. The value of each new variable is the square of each observed value for the corresponding parenthesized variable. The regression analysis then uses both sets of variables (original and squared). Variables following EPOINT must be numeric, and they are typically continuous.

POINT
POI

expands the variables for a circular response surface regression or for a circular ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP ideal point model coordinates to the OUT= data set. The POINT expansion creates a new variable having a value for each observation that is the sums of squares of all the POINT variables. This new variable is added to the set of variables and is used in the regression analysis. For more on ideal point regression, refer to Carroll (1972). Variables following POINT must be numeric, and they are typically continuous.

PSPLINE
PSP

expands each variable to a piecewise polynomial basis. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with PSPLINE. When DEGREE=n (3 by default) with k knots (0 by default), $n + k$ variables are created. In addition, the original variable appears in the OUT= data set before the ID variables. For example, PSPLINE(X / NKNOTS=1) expands X into X_1 X_2 X_3 X_4 and outputs X as well. Unlike BSPLINE, an intercept is not implicit in the columns of PSPLINE. Refer to Smith (1979) for a good introduction to piecewise polynomial splines. Also see the section "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 4614. Variables following PSPLINE must be numeric, and they are typically continuous.

QPOINT
QPO

expands the variables for a quadratic response surface regression or for a quadratic ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP quadratic ideal point model coordinates to the OUT= data set. For m QPOINT variables, $m(m+1)/2$ new variables are created containing the squares and crossproducts of the original variables. The regression analysis uses both sets (original and crossed). Variables following QPOINT must be numeric, and they are typically continuous.

Nonoptimal Transformations

Like variable expansions, nonoptimal transformations are computed before the iterative algorithm begins. Nonoptimal transformations create a single new transformed variable that replaces the original variable. The new variable is not transformed by the subsequent iterative algorithms (except for a possible linear transformation with missing value estimation).

The following list provides syntax and details for nonoptimal variable transformations.

ARSIN
ARS

finds an inverse trigonometric sine transformation. Variables following ARSIN must be numeric, in the interval $(-1.0 \le \mathsf{X} \le 1.0)$, and they are typically continuous.

BOXCOX
BOX

finds a Box-Cox transformation of the specified variables (see the "Box-Cox Transformations" section on page 4595 and Example 75.6). The BOXCOX transformation can be used only with dependent variables. The ALPHA=, CLL=, CONVENIENT, GEOMETRICMEAN, LAMBDA=, and PARAMETER= *t-options* can be used with the BOXCOX transformation. Variables following BOXCOX must be numeric, and they are typically continuous.

EXP

exponentiates variables (the variable X is transformed to a^X). To specify the value of a, use the PARAMETER= *t-option*. By default, a is the mathematical constant $e = 2.718\ldots$. Variables following EXP must be numeric, and they are typically continuous.

LOG

transforms variables to logarithms (the variable X is transformed to $\log_a(X)$). To specify the base of the logarithm, use the PARAMETER= *t-option*. The default is a natural logarithm with base $e = 2.718\ldots$. Variables following LOG must be numeric and positive, and they are typically continuous.

LOGIT

finds a logit transformation on the variables. The logit of X is $\log(X/(1-X))$. Unlike other transformations, LOGIT does not have a three-letter abbreviation. Variables following LOGIT must be numeric, in the interval $(0.0 < \mathsf{X} < 1.0)$, and they are typically continuous.

POWER
POW

raises variables to a specified power (the variable X is transformed to X^a). You must specify the power parameter a by specifying the PARAMETER= *t-option* following the variables:

```
power(variable / parameter=number)
```

You can use POWER for squaring variables (PARAMETER=2), reciprocal transformations (PARAMETER=−1), square roots (PARAMETER=0.5), and so on. Variables following POWER must be numeric, and they are typically continuous.

RANK
RAN

transforms variables to ranks. Ranks are averaged within ties. The smallest input value is assigned the smallest rank. Variables following RANK must be numeric.

SMOOTH
SMO

is a noniterative smoothing spline transformation. You can specify the smoothing parameter with either the SM= or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables following SMOOTH must be numeric, and they are typically continuous. The SMOOTH transformation can be used only with independent variables. For more information, see the "Smoothing Splines" section on page 4596.

Optimal Transformations

Optimal transformations are iteratively derived. Missing values for these types of variables can be optimally estimated (see the "Missing Values" section on page 4599).

The following list provides syntax and details for optimal transformations.

LINEAR
LIN

finds an optimal linear transformation of each variable. For variables with no missing values, the transformed variable is the same as the original variable. For variables with missing values, the transformed nonmissing values have a different scale and origin than the original values. Variables following LINEAR must be numeric.

MONOTONE
MON

finds a monotonic transformation of each variable, with the restriction that ties are preserved. The Kruskal (1964) secondary least-squares monotonic transformation is used. This transformation weakly preserves order and category membership (ties). Variables following MONOTONE must be numeric, and they are typically discrete.

MSPLINE
MSP

finds a monotonically increasing B-spline transformation with monotonic coefficients (de Boor 1978; de Leeuw 1986) of each variable. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with MSPLINE. By default, PROC TRANSREG uses a quadratic spline. Variables following MSPLINE must be numeric, and they are typically continuous.

OPSCORE
OPS

finds an optimal scoring of each variable. The OPSCORE transformation assigns scores to each class (level) of the variable. Fisher's (1938) optimal scoring method is used. Variables following OPSCORE can be either character or numeric; numeric variables should be discrete.

SPLINE
SPL

finds a B-spline transformation (de Boor 1978) of each variable. By default, PROC TRANSREG uses a cubic polynomial transformation. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with SPLINE. Variables following SPLINE must be numeric, and they are typically continuous.

UNTIE
UNT

finds a monotonic transformation of each variable without the restriction that ties are preserved. The TRANSREG procedure uses the Kruskal (1964) primary least-squares monotonic transformation method. This transformation weakly preserves order but not category membership (it may untie some previously tied values). Variables following UNTIE must be numeric, and they are typically discrete.

Other Transformations

IDENTITY
IDE

specifies variables that are not changed by the iterations. Typically, the IDENTITY transformation is used with a simple variable list, such as IDENTITY(X1-X5). However, you can also specify interaction terms. For example, IDENTITY(X1 | X2) creates X1, X2, and the product X1*X2; and IDENTITY(X1 | X2 | X3) creates X1, X2, X1*X2, X3, X1*X3, X2*X3, and X1*X2*X3. See the "Model Statement Usage" section on page 4592 for information on how to use the operators @, *, and | in PROC TRANSREG.

The IDENTITY transformation is used for variables when no transformation and no missing data estimation are desired. However, the REFLECT *t-option*, the ADDITIVE *a-option*, and the TSTANDARD=Z, and TSTANDARD=CENTER options can linearly transform all variables, including IDENTITY variables, after the iterations. Observations with missing values in IDENTITY variables are excluded from the analysis, and no optimal scores are computed for missing values in IDENTITY variables. Variables following IDENTITY must be numeric.

SSPLINE
SSP

finds an iterative smoothing spline transformation of each variable. The SSPLINE transformation does not generally minimize squared error. You can specify the smoothing parameter with either the SM= *t-option* or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables following SSPLINE must be numeric, and they are typically continuous.

Transformation Options (t-options)

If you use a nonoptimal, optimal, or other transformation, you can use *t-options*, which specify additional details of the transformation. The *t-options* are specified within the parentheses that enclose variables and are listed after a slash. You can use *t-options* with both dependent and independent variables. For example,

```
proc transreg;
   model identity(y)=spline(x / nknots=3);
   output;
run;
```

The preceding statements find an optimal variable transformation (SPLINE) of the independent variable, and they use a *t-option* to specify the number of knots (NKNOTS=). The following is a more complex example:

```
proc transreg;
   model mspline(y / nknots=3)=class(x1 x2 / effects);
   output;
run;
```

These statements find a monotone spline transformation (MSPLINE with three knots) of the dependent variable and perform a CLASS expansion with effects coding of the independents.

The following sections discuss the *t-options* available for nonoptimal, optimal, and other transformations.

The following table summarizes the *t-options*.

Table 75.2. t-options Available in the MODEL Statement

Task	**Option**
Nonoptimal transformation t-options	
uses original mean and variance	ORIGINAL
Parameter t-options	
specifies miscellaneous parameters	PARAMETER=
specifies smoothing parameter	SM=
Spline t-options	
specifies the degree of the spline	DEGREE=
spaces the knots evenly	EVENLY
exterior knots	EXKNOTS=
specifies the interior knots or break points	KNOTS=
creates n knots	NKNOTS=
CLASS Variable t-options	
CLASS dummy variable name prefix	CPREFIX=
requests a deviations-from-means coding	DEVIATIONS
requests a deviations-from-means coding	EFFECTS
CLASS dummy variable label prefix	LPREFIX=
order of class variable levels	ORDER=
CLASS dummy variable label separators	SEPARATORS=
controls reference levels	ZERO=
BOXCOX t-options	
confidence interval alpha	ALPHA=
convenient lambda list	CLL=

Table 75.2. (continued)

Task	**Option**
use a convenient lambda	CONVENIENT
scale the transformation using geometric mean	GEOMETRICMEAN
power parameter list	LAMBDA=
Other t-options	
operations occur after the expansion	AFTER
centers before the analysis begins	CENTER
renames variables	NAME=
reflects the variable around the mean	REFLECT
specifies transformation standardization	TSTANDARD=
standardizes before the analysis begins	Z

Nonoptimal Transformation t-options

ORIGINAL
ORI

matches the variable's final mean and variance to the mean and variance of the original variable. By default, the mean and variance are based on the transformed values. The ORIGINAL *t-option* is available for all of the nonoptimal transformations.

Parameter t-options

PARAMETER=*number*
PAR=*number*

specifies the transformation parameter. The PARAMETER= *t-option* is available for the BOXCOX, EXP, LOG, POWER, SMOOTH, and SSPLINE transformations. For BOXCOX, the parameter is the value to add to each value of the BOXCOX variable before a Box-Cox transformation. For EXP, the parameter is the value to be exponentiated; for LOG, the parameter is the base value; and for POWER, the parameter is the power. For SMOOTH and SSPLINE, the parameter is the raw smoothing parameter. (You can specify a SAS/GRAPH-style smoothing parameter with the SM= *t-option*.) The default for the PARAMETER= *t-option* for the BOXCOX transformation is 0 and for the LOG and EXP transformations is $e = 2.718\ldots$. The default parameter for SMOOTH and SSPLINE is computed from SM=0. For the POWER transformation, you must specify the PARAMETER= *t-option*; there is no default.

SM=*n*

specifies a SAS/GRAPH-style smoothing parameter in the range 0 to 100. You can specify the SM= *t-option* only with the SMOOTH and SSPLINE transformations. The smoothness of the function increases as the value of the smoothing parameter increases. By default, SM=0.

Spline t-options

The following *t-options* are available with the SPLINE and MSPLINE optimal transformations and the PSPLINE and BSPLINE expansions.

DEGREE=*n*
DEG=*n*

specifies the degree of the spline transformation. The degree must be a nonnegative integer. The defaults are DEGREE=3 for SPLINE, PSPLINE, and BSPLINE variables and DEGREE=2 for MSPLINE variables.

The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

EVENLY
EVE

is used with the NKNOTS= *t-option* to space the knots evenly. The differences between adjacent knots are constant.

If you specify NKNOTS=k, k knots are created at

$$\text{minimum} + i((\text{maximum} - \text{minimum})/(k+1))$$

for $i = 1, \ldots, k$. For example, if you specify

```
spline(X / knots=2 evenly)
```

and the variable X has a minimum of 4 and a maximum of 10, then the two interior knots are 6 and 8. Without the EVENLY *t-option*, the NKNOTS= *t-option* places knots at percentiles, so the knots are not evenly spaced.

EXKNOTS=*number-list* | *n* TO *m* BY *p*
EXK=*number-list* | *n* TO *m* BY *p*

specifies exterior knots for SPLINE and MSPLINE transformations and BSPLINE expansions. Usually, this option is not needed; PROC TRANSREG automatically picks suitable exterior knots. The only time you need to use this option is when you want to ensure that the exact same basis is used for different splines, for example when applying coefficients from one spline transformation to a variable in a different data set (see “Scoring Spline Variables” at the end of Example 75.1).

Specify one or two values. If the minimum EXKNOTS= value is less than the minimum data value, it is used as the exterior knot. If the maximum EXKNOTS= value is greater than the maximum data value, it is used as the exterior knot. Otherwise these values are ignored. When EXKNOTS= is specified with the CENTER or Z *t-options*, the knots apply to the original variable, not to the centered or standardized variable.

The B-spline transformations and expansions use a knot list consisting of exterior knots (values just smaller than the minimum), the specified (interior) knots, and exterior knots (values just larger than the minimum). You can use the DETAILS option to see all of these knots. Using different external knots gives different but equivalent B-spline bases. You can specify exterior knots on either the KNOTS= or EXKNOTS= *t-options*, however for the BSPLINE expansion, the KNOTS= *t-option* creates extra all-zero basis columns, whereas the EXKNOTS= *t-option* will give you the correct basis.

KNOTS=*number-list* **|** *n* **TO** *m* **BY** *p*
KNO=*number-list* **|** *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. By default, there are no knots. The first time you specify a value in the knot list, it indicates a discontinuity in the nth (from DEGREE=n) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$th derivative of the transformation function at the value of the knot. Knots can be repeated any number of times for decreasing smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= *t-option* with the NKNOTS= *t-option*. You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 4613).

NKNOTS=*n*
NKN=*n*

creates n knots, the first at the $100/(n+1)$ percentile, the second at the $200/(n+1)$ percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the twenty-fifth percentile, the median, and the seventy-fifth percentile. By default, NKNOTS=0. The NKNOTS= *t-option* must be ≥ 0.

You cannot use the NKNOTS= *t-option* with the KNOTS= *t-option*.

You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 4613).

CLASS Variable t-options

CPREFIX=*n* **|** *number-list*
CPR=*n* **|** *number-list*

specifies the number of first characters of a CLASS expansion variable's name to use in constructing names for dummy variables. When CPREFIX= is specified as an *a-option* (see the description of the CPREFIX= *a-option* on page 4575) or an *o-option*, it specifies the default for all CLASS variables. When you specify CPREFIX= as a *t-option*, it overrides the default only for selected variables. A different CPREFIX= value can be specified for each CLASS variable by specifying the CPREFIX=number-list *t-option*, like the ZERO=formatted-value-list *t-option*.

DEVIATIONS
DEV
EFFECTS
EFF

requests a deviations-from-means coding of CLASS variables. The coded design matrix has values of 0, 1, and -1 for reference levels. This coding is referred to as "deviations-from-means," "effects," "center-point," or "full-rank" coding.

LPREFIX=*n* **|** *number-list*
LPR=*n* **|** *number-list*

specifies the number of first characters of a CLASS expansion variable's label (or name if no label is specified) to use in constructing labels for dummy variables. When LPREFIX= is specified as an *a-option* (see the description of the LPREFIX= *a-option* on page 4576) or an *o-option*, it specifies the default for all CLASS variables. When you specify LPREFIX= as a *t-option*, it overrides the default only for selected variables. A different LPREFIX= value can be specified for each CLASS variable by specifying the LPREFIX=number-list *t-option*, like the ZERO=formatted-value-list *t-option*.

ORDER=DATA | FREQ | FORMATTED | INTERNAL
ORD=DAT | FRE | FOR | INT

specifies the order in which the CLASS variable levels are to be reported. The default is ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER= is specified as an *a-option* (see the description of the ORDER= *a-option* on page 4578) or as an *o-option*, it specifies the default ordering for all CLASS variables. When you specify ORDER= as a *t-option*, it overrides the default ordering only for selected variables. You can specify a different ORDER= value for each CLASS specification.

SEPARATORS=’*string-1* **’**<**’***string-2* **’** >
SEP=’*string-1* **’**<**’***string-2* **’** >

specifies separators for creating CLASS expansion variable labels. By default, SEPARATORS=’ ’ ’ * ’ (“blank” and “blank asterisk blank”). When SEPARATORS= is specified as an *a-option* (see the description of the SEPARATORS= *a-option* on page 4579) or an *o-option*, it specifies the default separators for all CLASS variables. When you specify SEPARATORS= as a *t-option*, it overrides the default only for selected variables. You can specify a different SEPARATORS= value for each CLASS specification.

ZERO=FIRST | LAST | NONE | SUM
ZER=FIR | LAS | NON | SUM
ZERO=’*formatted-value* **’** <**’***formatted-value* **’ ...**>

is used with CLASS variables. The default is ZERO=LAST.

The specification CLASS(variable / ZERO=FIRST) sets to missing the dummy variable for the first of the sorted categories, implying a zero coefficient for that category.

The specification CLASS(variable / ZERO=LAST) sets to missing the dummy variable for the last of the sorted categories, implying a zero coefficient for that category.

The specification CLASS(variable / ZERO=’*formatted-value*’) sets to missing the dummy variable for the category with a formatted value that matches ’*formatted-value*’, implying a zero coefficient for that category. With ZERO=formatted-value-list, the first formatted value applies to the first variable in the specification, the second formatted value applies to the next variable that was not previously mentioned and so on. For example, CLASS(A A*B B B*C C / ZERO=’x’ ’y’ ’z’) specifies that the reference level for A is ’x’, for B is ’y’, and for C is ’z’. With ZERO=’*formatted-value*’, the procedure first looks for exact matches between the formatted values and the specified value. If none are found, leading blanks are stripped from both and the values are compared again. If zero or two or more matches are found, warnings are issued.

The specifications ZERO=FIRST, ZERO=LAST, and ZERO=’*formatted-value*’ are used for reference cell models. The Intercept parameter estimate is the marginal mean for the reference cell, and the other marginal means are obtained by adding the intercept to the dummy variable coefficients.

The specification CLASS(variable / ZERO=NONE) sets to missing none of the dummy variables. The columns of the expansion sum to a column of ones, so an implicit intercept model is fit. If you specify ZERO=NONE for more than one variable, the model is less than full rank. In the model MODEL IDENTITY(Y) = CLASS(X / ZERO=NONE), the coefficients are cell means.

The specification CLASS(variable / ZERO=SUM) sets to missing none of the dummy variables, and the coefficients for the dummy variables created from the variable sum to 0. This creates a less-than-full-rank model, but the coefficients are uniquely determined due to the sum-to-zero constraint.

In the presence of iterative transformations, hypothesis tests for ZERO=NONE and ZERO=SUM levels are not exact; they are liberal because a model with an explicit intercept is fit inside the iterations. There is no provision for adjusting the transformations while setting to 0 a parameter that is redundant given the explicit intercept and the other parameters.

Box-Cox t-options

The following *t-options* are available only with the BOXCOX transformation of the dependent variable (see the “Box-Cox Transformations” section on page 4595 and Example 75.6).

ALPHA=*p*
ALP=*p*

specifies the Box-Cox alpha for the confidence interval for the power parameter. By default, ALPHA=0.05.

CLL=*number-list*

specifies the Box-Cox convenient lambda list. When the confidence interval for the power parameter includes one of the values in this list, PROC TRANSREG reports it and can optionally use the convenient power parameter instead of the more optimal power parameter. The default is CLL=1.0 0.0 0.5 -1.0 -0.5 2.0 -2.0 3.0 -3.0. By default, a linear transformation is preferred over log, square root, inverse, inverse square root, quadratic, inverse quadratic, cubic, and inverse cubic. If you specify the CONVENIENT *t-option*, then PROC TRANSREG uses the first convenient power parameter in the list that is in the confidence interval. For example, if the optimal power parameter is 0.25 and 0.0 is in the confidence interval but not 1.0, then the convenient power parameter is 0.0.

CONVENIENT
CON

specifies that a power parameter from the CLL= *t-option* list is to be used for the final transformation instead of the LAMBDA= *t-option* value if a CLL= value is in the confidence interval. See the CLL= *t-option* for more information on its usage.

GEOMETRICMEAN
GEO

divides the Box-Cox transformation by $\dot{y}^{\lambda-1}$ where $\dot{y}$ is the geometric mean of the variable to be transformed. This form of the Box-Cox transformation essentially converts the transformation back to original units and hence allows direct comparison of the residual sums of squares for models with different power parameters.

LAMBDA=*number-list*
LAM=*number-list*

specifies a list of Box-Cox power parameters. The default is LAMBDA=-3 TO 3 BY 0.25. PROC TRANSREG tries each power parameter in the list and picks the best one. However, when the CONVENIENT *t-option* value is in the confidence interval. See the CLL= *t-option* for more information on its usage.

Other t-options

AFTER
AFT

requests that certain operations occur after the expansion. This *t-option* affects the NKNOTS= *t-option* when the SPLINE or MSPLINE transformation is crossed with a CLASS specification. For example, if the original spline variable (1 2 3 4 5 6 7 8 9) is expanded into the three variables (1 2 3 0 0 0 0 0 0), (0 0 0 4 5 6 0 0 0), and (0 0 0 0 0 0 7 8 9), then, by default, NKNOTS=1 would use the overall median of 5 as the knot for all three variables. When you specify the AFTER *t-option*, the knots for the three variables are 2, 5, and 8. Note that the structural zeros are ignored when the internal knot list is created, but they are not ignored for the external knots.

You can also specify the AFTER *t-option* with the RANK and SMOOTH transformations. The following specifications compute ranks and smooth within groups, after crossing, ignoring the structural zeros.

```
class(x / zero=none) | rank(z / after)
class(x / zero=none) | smooth(z / after)
```

CENTER
CEN

centers the variables before the analysis begins (in contrast to the TSTANDARD=CENTER option which centers after the analysis ends). The CENTER *t-option* can be used instead of running PROC STANDARD before PROC TRANSREG (see the "Centering" section on page 4675). When the KNOTS= *t-option* is specified with CENTER, the knots apply to the original variable, not to the centered variable. PROC TRANSREG will center the knots.

NAME=(*variable-list***)**
NAM=(*variable-list***)**

renames variables as they are used in the MODEL statement. This *t-option* allows a variable to be used more than once.

For example, if X is a character variable, then the following step stores both the original character variable X and a numeric variable XC that contains category numbers in the OUT= data set.

```
proc transreg data=a;
   model identity(y) = opscore(x / name=(xc));
   output;
   id x;
run;
```

With the CLASS and IDENTITY transformations, which allow interaction effects, the first name applies to the first variable in the specification, the second name applies to the next variable that was not previously mentioned, and so on. For example, IDENTITY(A A*B B B*C C / NAME=(G H I)) specifies that the new name for A is G, for B is H, and for C is I. The same assignment is used for the (not useful) specification IDENTITY(A A B B C C / NAME=(G H I)). For all *transforms* other than CLASS and IDENTITY (all those in which interactions are not supported), repeated variables are not handled specially. For example, SPLINE(A A B B C C / NAME=(A G B H C I)) creates six variables, a copy of A named A, another copy of A named G, a copy of B named B, another copy of B named H, a copy of C named C, and another copy of C named I.

REFLECT
REF

reflects the transformation

$$y = -(y - \bar{y}) + \bar{y}$$

after the iterations are completed and before the final standardization and results calculations. This *t-option* is particularly useful with the dependent variable in a conjoint analysis. When the dependent variable consists of ranks with the most preferred combination assigned 1.0, the REFLECT *t-option* reflects the transformation so that positive utilities mean high preference. (See Example 75.2.)

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables for the hypothesis tests and in the OUT= data set (see the “Centering” section on page 4675). By default, TSTANDARD=ORIGINAL. When TSTANDARD= is specified as an *a-option* (see the description of the TSTANDARD= *a-option* on page 4580) or an *o-option*, it determines the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables. You can specify a different TSTANDARD= value for each transformation. For example, to perform a redundancy analysis with standardized dependent variables, specify

```
model identity(y1-y4 / tstandard=z) = identity(x1-x10);
```

Z

centers and standardizes the variables to variance one before the analysis begins (in contrast to the TSTANDARD=Z option, which standardizes after the analysis ends). The Z *t-option* can be used instead of running PROC STANDARD before PROC

TRANSREG (see the "Centering" section on page 4675). When the KNOTS= *t-option* is specified with Z, the knots apply to the original variable, not to the centered variable. PROC TRANSREG will standardize the knots.

Algorithm Options (a-options)

This section discusses the options that can appear in the PROC TRANSREG or MODEL statements as *a-options*. They are listed after the entire model specification and after a slash.

For example,

```
proc transreg;
   model spline(y / nknots=3)=log(x1 x2 / parameter=2)
         / nomiss maxiter=50;
   output;
run;
```

In the preceding statements, NOMISS and MAXITER= are *a-options*. (SPLINE and LOG are *transforms*, and NKNOTS= and PARAMETER= are *t-options*.) The statements find a spline transformation with 3 knots on Y and a base 2 logarithmic transformation on X1 and X2. The NOMISS *a-option* excludes all observations with missing values, and the MAXITER= *a-option* specifies the maximum number of iterations.

Table 75.3. Options Available in the PROC TRANSREG or MODEL Statements

Task	Option
Input data set	
specifies input observation type	TYPE=
restarts iterations	REITERATE
Specify method and control iterations	
specifies minimum criterion change	CCONVERGE=
specifies minimum data change	CONVERGE=
specifies canonical dummy-variable initialization	DUMMY
specifies maximum number of iterations	MAXITER=
specifies iterative algorithm	METHOD=
specifies number of canonical variables	NCAN=
specifies singularity criterion	SINGULAR=
Control missing data handling	
METHOD=MORALS fists each model individually	INDIVIDUAL
includes monotone special missing values	MONOTONE=
excludes observations with missing values	NOMISS
unties special missing values	UNTIE=
Control intercept and CLASS variables	
CLASS dummy variable name prefix	CPREFIX=
CLASS dummy variable label prefix	LPREFIX=
no intercept or centering	NOINT
order of class variable levels	ORDER=
controls output of reference levels	REFERENCE=

Table 75.3. (continued)

Task	**Option**
CLASS dummy variable label separators	SEPARATORS=
Control displayed output	
confidence limits alpha	ALPHA=
displays parameter estimate confidence limits	CL
displays model specification details	DETAIL
displays iteration histories	HISTORY
suppresses displayed output	NOPRINT
suppresses the iteration histories	SHORT
displays regression results	SS2
displays ANOVA table	TEST
displays conjoint part-worth utilities	UTILITIES
Control standardization	
fits additive model	ADDITIVE
do not zero constant variables	NOZEROCONSTANT
specifies transformation standardization	TSTANDARD=

The following list provides details on these *a-options*.

ADDITIVE
ADD

creates an additive model by multiplying the values of each independent variable (after the TSTANDARD= standardization) by that variable's corresponding multiple regression coefficient. This process scales the independent variables so that the predicted-values variable for the final dependent variable is simply the sum of the final independent variables. An additive model is a univariate multiple regression model. As a result, the ADDITIVE *a-option* is not valid if METHOD=CANALS, or if METHOD=REDUNDANCY or METHOD=UNIVARIATE with more than one dependent variable.

ALPHA=*number*
ALP=*number*

specifies the level of significance for all of the confidence limits. By default, ALPHA=0.05.

CCONVERGE=*n*
CCO=*n*

specifies the minimum change in the criterion being optimized (squared multiple correlation for METHOD=MORALS and METHOD=UNIVARIATE, average squared multiple correlation for METHOD=REDUNDANCY, average squared canonical correlation for METHOD=CANALS) that is required to continue iterating. By default, CCONVERGE=0.0.

CL

requests confidence limits on the parameter estimates in the displayed output.

CONVERGE=*n*
CON=*n*

specifies the minimum average absolute change in standardized variable scores that is required to continue iterating. By default, CONVERGE=0.00001. Average change is computed over only those variables that can be transformed by the iterations; that is, all LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE variables and nonoptimal transformation variables with missing values.

CPREFIX=*n*
CPR=*n*

specifies the number of first characters of a CLASS expansion variable's name to use in constructing names for dummy variables. Dummy variable names are constructed from the first n characters of the CLASS expansion variable's name and the first $32 - n$ characters of the formatted CLASS expansion variable's value. For example, if the variable ClassVariable has values 1, 2, and 3, then, by default, the dummy variables are named ClassVariable1, ClassVariable2, and ClassVariable3. However, with CPREFIX=5, the dummy variables are named Class1, Class2, and Class3. When CPREFIX=0, dummy variable names are created entirely from the CLASS expansion variable's formatted values. Valid values range from -1 to 31, where -1 indicates the default calculation and 0 to 31 are the number of prefix characters to use. The default, -1, sets n to 32 - min(32, max(2, *fl*)), where *fl* is the format length. When CPREFIX= is specified as an *a-option* or an *o-option*, it specifies the default for all CLASS variables. When you specify CPREFIX= as a *t-option*, it overrides the default only for selected variables.

DETAIL
DET

reports on details of the model specification. For example, it reports the knots and coefficients for splines, reference levels for CLASS variables, Box-Cox results, and so on.

DUMMY
DUM

provides a canonical dummy variable initialization. When there are no monotonicity constraints and there is only one canonical variable in each set, PROC TRANSREG (with the DUMMY *a-option*) can usually find the optimal solution in only one iteration. The initialization iteration is number 0, which is slower and uses more memory than other iterations. However, when there are no monotonicity constraints, when there is only one canonical variable in each set, and when there is enough available memory, specifying the DUMMY *a-option* can greatly decrease the amount of time required to find the optimal transformations. Furthermore, by solving for the transformations directly instead of iteratively, PROC TRANSREG avoids certain nonoptimal solutions.

HISTORY
HIS

displays the iteration histories even when the NOPRINT *a-option* is specified.

INDIVIDUAL
IND

fits each model for each dependent variable individually. This means, for example, that when INDIVIDUAL is specified, missing values in one dependent variable will not cause that observation to be deleted for the other models with the other dependent variables. In contrast, by default, missing values in any variable in any model can cause the observation to be deleted for all models. The INDIVIDUAL option can only be specified with METHOD=MORALS.

This option also affects the order of the output. By default, the number of observations table is printed once at the beginning of the output. With INDIVIDUAL, a number of observations table appears for each model.

LPREFIX=*n*
LPR=*n*

specifies the number of first characters of a CLASS expansion variable's label (or name if no label is specified) to use in constructing labels for dummy variables. Dummy variable labels are constructed from the first n characters of the CLASS expansion variable's name and the first $127 - n$ characters of the formatted CLASS expansion variable's value. Valid values range from -1 to 127. Values of 0 to 127 specify the number of name or label characters to use. The default is -1, which specifies that PROC TRANSREG should pick a value depending on the length of the prefix and the formatted class value. When LPREFIX= is specified as an *a-option* or an *o-option*, it determines the default for all CLASS variables. When you specify LPREFIX= as a *t-option*, it overrides the default only for selected variables.

MAXITER=*n*
MAX=*n*

specifies the maximum number of iterations (see the "Controlling the Number of Iterations" section on page 4601). By default, MAXITER=30. A specification of MAXITER=0 is allowed to save time when no transformations are requested.

METHOD=CANALS | MORALS | REDUNDANCY | UNIVARIATE
MET=CAN | MOR | RED | UNI

specifies the iterative algorithm. By default, METHOD=UNIVARIATE, unless you specify options that cannot be handled by the UNIVARIATE algorithm. Specifically, the default is METHOD=MORALS for the following situations:

- if you specify LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, or SSPLINE transformations for the independent variables
- if you specify the ADDITIVE *a-option* with more than one dependent variable
- if you specify the IAPPROXIMATIONS *o-option*

CANALS specifies canonical correlation with alternating least squares. This jointly transforms all dependent and independent variables to maximize the average of the first n squared canonical correlations, where n is the value of the NCAN= *a-option*.

MORALS specifies multiple optimal regression with alternating least squares. This transforms each dependent variable, along with the set of independent variables, to maximize the squared multiple correlation.

REDUNDANCY jointly transforms all dependent and independent variables to maximize the average of the squared multiple correlations (see the "Redundancy Analysis" section on page 4606).

UNIVARIATE transforms each dependent variable to maximize the squared multiple correlation, while the independent variables are not transformed.

MONOTONE=*two-letters*
MON=*two-letters*

specifies the first and last special missing value in the list of those special missing values to be estimated using within-variable order and category constraints. By default, there are no order constraints on missing value estimates. The *two-letters* value must consist of two letters in alphabetical order. For example, MONOTONE=DF means that the estimate of .D must be less than or equal to the estimate of .E, which must be less than or equal to the estimate of .F; no order constraints are placed on estimates of ._, .A through .C, and .G through .Z. For details, see the "Missing Values" section on page 4599.

NCAN=*n*
NCA=*n*

specifies the number of canonical variables to use in the METHOD=CANALS algorithm. By default, NCAN=1. The value of the NCAN= *a-option* must be ≥ 1.

When canonical coefficients and coordinates are included in the OUT= data set, the NCAN= *a-option* also controls the number of rows of the canonical coefficient matrices in the data set. If you specify an NCAN= value larger than the minimum of the number of dependent variables and the number of independent variables, PROC TRANSREG displays a warning and sets the NCAN= *a-option* to the maximum allowable value.

NOINT
NOI

omits the intercept from the OUT= data set and suppresses centering of data. The NOINT *a-option* is not allowed with iterative transformations since there is no provision for optimal scaling without an intercept. The NOINT *a-option* is allowed only when there is no implicit intercept and when all of the data in a BY group absolutely will not change during the iterations.

NOMISS
NOM

excludes all observations with missing values from the analysis, but does not exclude them from the OUT= data set. If you omit the NOMISS *a-option*, PROC TRANSREG simultaneously computes the optimal transformations of the nonmissing values and estimates the missing values that minimize squared error. For details, see the "Missing Values" section on page 4599.

Casewise deletion of observations with missing values occurs when the NOMISS *a-option* is specified, when there are missing values in expansions, when there are missing values in METHOD=UNIVARIATE independent variables, when there are weights less than or equal to 0, or when there are frequencies less than 1. Excluded observations are output with a blank value for the _TYPE_ variable, and they have a weight of 0. They do not contribute to the analysis but are scored and transformed as *supplementary* or passive observations.

See the "Passive Observations" section on page 4605 for more information on excluded observations.

NOPRINT
NOP

suppresses the display of all output unless you specify the HISTORY *a-option*. The NOPRINT *a-option* without the HISTORY *a-option* temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

NOZEROCONSTANT
NOZERO
NOZ

specifies that constant variables are expected and should not be zeroed. By default, constant variables are zeroed. This option is useful when PROC TRANSREG is used to code experimental designs for discrete choice models (see the "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" section on page 4660). When these designs are very large, it may be more efficient to use the DESIGN=n option. It may be that attributes are constant within a block of n observations, so you need to specify the NOZEROCONSTANT option to get the correct results. You can specify this option in the PROC TRANSREG, MODEL, and OUTPUT statements.

ORDER=DATA | FREQ | FORMATTED | INTERNAL
ORD=DAT | FRE | FOR | INT

specifies the order in which the CLASS variable levels are to be reported. The default is ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER= is specified as an *a-option* or an *o-option*, it determines the default ordering for all CLASS variables. When you specify ORDER= as a *t-option*, it overrides the default ordering only for selected variables.

DATA	sorts by order of appearance in the input data set.
FORMATTED	sorts by formatted value.

FREQ sorts by descending frequency count; levels with the most observations appear first.

INTERNAL sorts by unformatted value.

REFERENCE=NONE | MISSING | ZERO
REF=NON | MIS | ZER

specifies how reference levels of CLASS variables are to be treated. The options are REFERENCE=NONE, the default, in which reference levels are suppressed; REFERENCE=MISSING, in which reference levels are displayed and output with missing values; and REFERENCE=ZERO, in which reference levels are displayed and output with zeros. The REFERENCE= option can be specified in the PROC TRANSREG, MODEL, or OUTPUT statement, and it can be independently specified for the OUT= data set and the displayed output. When you specify it in only one statement, it sets the option for both the displayed output and the OUT= data set.

REITERATE
REI

enables the TRANSREG procedure to use previous transformations as starting points. The REITERATE *a-option* affects only variables that are iteratively transformed (specified as LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE). For iterative transformations, the REITERATE *a-option* requests a search in the input data set for a variable that consists of the value of the TDPREFIX= or TIPREFIX= *o-option* followed by the original variable name. If such a variable is found, it is used to provide the initial values for the first iteration. The final transformation is a member of the transformation family defined by the original variable, not the transformation family defined by the initialization variable. See the section “Using the REITERATE Algorithm Option” on page 4602.

SEPARATORS=’*string-1* **’**<**’***string-2* **’** >
SEP=’*string-1* **’**<**’***string-2* **’** >

specifies separators for creating CLASS expansion variable labels. By default, SEPARATORS=’ ’ ’ * ’ (“blank” and “blank asterisk blank”). The first value is used to separate variable names and values in interactions. The second value is used to separate interaction components. For example, the label for the dummy variable for the A=1 and B=2 cell is, by default, ’A 1 * B 2’. If SEPARATORS=’=’ ’x’ is specified, then the label is ’A=1xB=2’. When SEPARATORS= is specified as an *a-option* or an *o-option*, it determines the default separators for all CLASS variables. When you specify SEPARATORS= as a *t-option*, it overrides the default only for selected variables.

SHORT
SHO

suppresses the iteration histories.

SINGULAR=*n*
SIN=*n*

specifies the largest value within rounding error of zero. By default, SINGULAR=1E−12. The TRANSREG procedure uses the value of the SINGULAR= *a-option* for checking $1 - \mathrm{R}^2$ when constructing full-rank matrices of predictor variables, checking denominators before dividing, and so on. PROC TRANSREG computes the regression coefficients by sweeping with rational pivoting.

SS2

produces a regression table based on Type II sums of squares. Tests of the contribution of each transformation to the overall model are displayed and output to the OUTTEST= data set when you specify the OUTTEST= option. When you specify the SS2 *a-option*, the TEST *a-option* is implied. See the section "Hypothesis Tests" on page 4615. You can suppress the variable labels in the regression tables by specifying the NOLABEL option in the OPTIONS statement.

TEST
TES

generates an ANOVA table. PROC TRANSREG tests the null hypothesis that the vector of scoring coefficients for all of the transformations is zero. See the section "Hypothesis Tests" on page 4615.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables for the hypothesis tests and in the OUT= data set. By default, TSTANDARD=ORIGINAL. When TSTANDARD= is specified as an *a-option* or an *o-option*, it determines the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables.

CENTER centers the output variables to mean zero, but the variances are the same as the variances of the input variables.

NOMISS sets the means and variances of the transformed variables in the OUT= data set, computed over all output values that correspond to nonmissing values in the input data set, to the means and variances computed from the nonmissing observations of the original variables. The TSTANDARD=NOMISS specification is useful with missing data. When a variable is linearly transformed, the final variable contains the original nonmissing values and the missing value estimates. In other words, the nonmissing values are unchanged. If your data have no missing values, TSTANDARD=NOMISS and TSTANDARD=ORIGINAL produce the same results.

ORIGINAL sets the means and variances of the transformed variables to the means and variances of the original variables. This is the default.

Z standardizes the variables to mean zero, variance one.

The final standardization is affected by other options. If you also specify the ADDITIVE *a-option*, the TSTANDARD= option specifies an intermediate step in computing the final means and variances. The final independent variables, along with their means and standard deviations, are scaled by the regression coefficients, creating an additive model with all coefficients equal to one.

For nonoptimal variable transformations, the means and variances of the original variables are actually the means and variances of the nonlinearly transformed variables, unless you specify the ORIGINAL nonoptimal *t-option* in the MODEL statement. For example, if a variable X with no missing values is specified as LOG, then, by default, the final transformation of X is simply LOG(X), not LOG(X) standardized to the mean of X and variance of X.

TYPE='*text* **'|***name*
TYP='*text* **'|***name*

specifies the valid value for the _TYPE_ variable in the input data set. If PROC TRANSREG finds an input _TYPE_ variable, it uses only observations with a _TYPE_ value that matches the TYPE= value. This enables a PROC TRANSREG OUT= data set containing coefficients to be used as input to PROC TRANSREG without requiring a WHERE statement to exclude the coefficients. If a _TYPE_ variable is not in the data set, all observations are used. The default is TYPE='SCORE', so if you do not specify the TYPE= *a-option*, only observations with _TYPE_='SCORE' are used. Do not confuse this option with the data set TYPE= option. The DATA= data set must be an ordinary SAS data set.

PROC TRANSREG displays a note when it reads observations with blank values of _TYPE_, but it does not automatically exclude those observations. Data sets created by the TRANSREG and PRINQUAL procedures have blank _TYPE_ values for those observations that were excluded from the analysis due to nonpositive weights, nonpositive frequencies, or missing data. When these observations are read again, they are excluded for the same reason that they were excluded from their original analysis, not because their _TYPE_ value is blank.

UNTIE=*two-letters*
UNT=*two-letters*

specifies the first and last special missing value in the list of those special missing values that are to be estimated with within-variable order constraints but no category constraints. The *two-letters* value must consist of two letters in alphabetical order. By default, there are category constraints but no order constraints on special missing value estimates. For details, see the "Missing Values" section on page 4599 and the "Optimal Scaling" section on page 4609.

UTILITIES
UTI

produces a table of the part-worth utilities from a conjoint analysis. Utilities, their standard errors, and the relative importance of each factor are displayed and output to the OUTTEST= data set when you specify the OUTTEST= option. When you specify the UTILITIES *a-option*, the TEST *a-option* is implied. Refer to SAS Technical Report R-109, *Conjoint Analysis Examples*, for more information on conjoint analysis.

OUTPUT Statement

OUTPUT OUT=*SAS-data-set* < *o-options* > **;**

The OUTPUT statement creates a new SAS data set that contains coefficients, marginal means, and information on original and transformed variables. The information on original and transformed variables composes the score partition of the data set; observations have _TYPE_='SCORE'. The coefficients and marginal means compose the coefficient partition of the data set; observations have _TYPE_='M COEFFI' or _TYPE_='MEAN'. Other values of _TYPE_ are possible; for details, see "_TYPE_ and _NAME_ Variables" later in this chapter. For details on data set structure, see the "Output Data Set" section on page 4617.

To specify the data set, use the OUT= specification.

OUT=*SAS-data-set*

specifies the output data set for the data, transformed data, predicted values, residuals, scores, coefficients, and so on. When you use an OUTPUT statement but do not use the OUT= specification, PROC TRANSREG creates a data set and uses the DATA*n* convention. If you want to create a permanent SAS data set, you must specify a two-level name (refer to "SAS Files" in *SAS Language Reference: Concepts* and "Introduction to DATA Step Processing" in the *SAS Procedures Guide* for details).

To control the contents of the data set and variable names, use one or more of the *o-options*. You can also specify these options in the PROC TRANSREG statement.

Output Options (o-options)

The following table provides a summary of options in the OUTPUT statement. These options include the OUT= option and all of the *o-options*.

Table 75.4. Options Available in the OUTPUT Statement

Task	Option
Identify output data set	
output data set	OUT=
Predicted values, residuals, scores	
outputs canonical scores	CANONICAL
outputs individual confidence limits	CLI
outputs mean confidence limits	CLM
specifies design matrix coding	DESIGN=
outputs leverage	LEVERAGE
does not restore missings	NORESTOREMISSING
suppresses output of scores	NOSCORES
outputs predicted values	PREDICTED
outputs redundancy variables	REDUNDANCY=
outputs residuals	RESIDUALS
Output data set replacement	
replaces dependent variables	DREPLACE
replaces independent variables	IREPLACE

Table 75.4. (continued)

Task	Option
replaces all variables	REPLACE
Output data set coefficients	
outputs coefficients	COEFFICIENTS
outputs ideal point coordinates	COORDINATES
outputs marginal means	MEANS
outputs redundancy analysis coefficients	MREDUNDANCY
Output data set variable name prefixes	
dependent variable approximations	ADPREFIX=
independent variable approximations	AIPREFIX=
canonical dependent variables	CDPREFIX=
conservative individual lower CL	CILPREFIX=
canonical independent variables	CIPREFIX=
conservative-individual-upper CL	CIUPREFIX=
conservative-mean-lower CL	CMLPREFIX=
conservative-mean-upper CL	CMUPREFIX=
METHOD=MORALS untransformed dependent	DEPENDENT=
liberal-individual-lower CL	LILPREFIX=
liberal-individual-upper CL	LIUPREFIX=
liberal-mean-lower CL	LMLPREFIX=
liberal-mean-upper CL	LMUPREFIX=
residuals	RDPREFIX=
predicted values	PPREFIX=
redundancy variables	RPREFIX=
transformed dependents	TDPREFIX=
transformed independents	TIPREFIX=
Output data set macros	
creates macro variables	MACRO
Control CLASS variables	
controls output of reference levels	REFERENCE=
Output data set details	
dependent and independent approximations	APPROXIMATIONS
canonical correlation coefficients	CCC
canonical elliptical point coordinate	CEC
canonical point coordinates	CPC
canonical quadratic point coordinates	CQC
approximations to transformed dependents	DAPPROXIMATIONS
approximations to transformed independents	IAPPROXIMATIONS
elliptical point coordinates	MEC
point coordinates	MPC
quadratic point coordinates	MQC
multiple regression coefficients	MRC

For the coefficients partition, the COEFFICIENTS, COORDINATES, and MEANS *o-options* provide the coefficients that are appropriate for your model. For more explicit control of the coefficient partition, use the options that control details and prefixes.

The following list provides details on these options.

ADPREFIX=*name*
ADP=*name*

specifies a prefix for naming the dependent variable predicted values. The default is ADPREFIX=P when you specify the PREDICTED *o-option*; otherwise, it is ADPREFIX=A. Specifying the ADPREFIX= *o-option* also implies the PREDICTED *o-option*, and the ADPREFIX= *o-option* is the same as the PPREFIX= *o-option*.

AIPREFIX=*name*
AIP=*name*

specifies a prefix for naming the independent variable approximations. The default is AIPREFIX=A. Specifying the AIPREFIX= *o-option* also implies the IAPPROXIMATIONS *o-option*.

APPROXIMATIONS
APPROX
APP

is equivalent to specifying both the DAPPROXIMATIONS and the IAPPROXIMATIONS *o-options*. If METHOD=UNIVARIATE, then the APPROXIMATIONS *o-option* implies only the DAPPROXIMATIONS *o-option*.

CANONICAL
CAN

outputs canonical variables to the OUT= data set. When METHOD=CANALS, the CANONICAL *o-option* is implied. The CDPREFIX= *o-option* specifies a prefix for naming the dependent canonical variables (default Cand), and the CIPREFIX= *o-option* specifies a prefix for naming the independent canonical variables (default Cani).

CCC

outputs canonical correlation coefficients to the OUT= data set.

CDPREFIX=*name*
CDP=*name*

provides a prefix for naming the canonical dependent variables. The default is CDPREFIX=Cand. Specifying the CDPREFIX= *o-option* also implies the CANONICAL *o-option*.

CEC

outputs canonical elliptical point model coordinates to the OUT= data set.

CILPREFIX=*name*
CIL=*name*

specifies a prefix for naming the conservative-individual-lower confidence limits. The default prefix is CIL. Specifying the CILPREFIX= *o-option* also implies the CLI *o-option*.

CIPREFIX=*name*
CIP=*name*

provides a prefix for naming the canonical independent variables. The default is CIPREFIX=Cani. Specifying the CIPREFIX= *o-option* also implies the CANONICAL *o-option*.

CIUPREFIX=*name*
CIU=*name*

specifies a prefix for naming the conservative-individual-upper confidence limits. The default prefix is CIU. Specifying the CIUPREFIX= *o-option* also implies the CLI *o-option*.

CLI

outputs individual confidence limits to the OUT= data set. The names of the confidence limits variables are constructed from the original dependent variable names and the prefixes specified in the following *o-options*: LILPREFIX= (default LIL for liberal individual lower), CILPREFIX= (default CIL for conservative individual lower), LIUPREFIX= (default LIU for liberal individual upper), and CIUPREFIX= (default CIU for conservative individual upper). When there are no monotonicity constraints, the liberal and conservative limits are the same.

CLM

outputs mean confidence limits to the OUT= data set. The names of the confidence limits variables are constructed from the original dependent variable names and the prefixes specified in the following *o-options*: LMLPREFIX= (default LML for liberal mean lower), CMLPREFIX= (default CML for conservative mean lower), LMUPREFIX= (default LMU for liberal mean upper), and CMUPREFIX= (default CMU for conservative mean upper). When there are no monotonicity constraints, the liberal and conservative limits are the same.

CMLPREFIX=*name*
CML=*name*

specifies a prefix for naming the conservative-mean-lower confidence limits. The default prefix is CML. Specifying the CMLPREFIX= *o-option* also implies the CLM *o-option*.

CMUPREFIX=*name*
CMU=*name*

specifies a prefix for naming the conservative-mean-upper confidence limits. The default prefix is CMU. Specifying the CMUPREFIX= *o-option* also implies the CLM *o-option*.

COEFFICIENTS
COE

outputs either multiple regression coefficients or raw canonical coefficients to the OUT= data set. If you specify METHOD=CANALS (in the MODEL or PROC TRANSREG statement), then the COEFFICIENTS *o-option* outputs the first n canonical variables, where n is the value of the NCAN= *a-option* (specified in the MODEL or PROC TRANSREG statement). Otherwise, the COEFFICIENTS *o-option* includes multiple regression coefficients in the OUT= data set. In addition, when you specify the CLASS expansion for any independent variable, the COEFFICIENTS *o-option* also outputs marginal means.

COORDINATES
COO

outputs either ideal point or vector model coordinates for preference mapping to the OUT= data set. When METHOD=CANALS, these coordinates are computed from canonical coefficients; otherwise, the coordinates are computed from multiple regression coefficients. For details, see the "Point Models" section on page 4605.

CPC

outputs canonical point model coordinates to the OUT= data set.

CQC

outputs canonical quadratic point model coordinates to the OUT= data set.

DAPPROXIMATIONS
DAP

outputs the approximations of the transformed dependent variables to the OUT= data set. These are the target values for the optimal transformations. With METHOD=UNIVARIATE and METHOD=MORALS, the dependent variable approximations are the ordinary predicted values from the linear model. The names of the approximation variables are constructed from the ADPREFIX= *o-option* (default A) and the original dependent variable names. For ordinary predicted values, use the PREDICTED *o-option* instead of the DAPPROXIMATIONS *o-option*, since the PREDICTED *o-option* uses a more relevant prefix ("P" instead of "A") and a more relevant variable label suffix ("Predicted Values" instead of "Approximations").

DESIGN<**=***n*>
DES<**=***n*>

specifies that your primary goal is design matrix coding, not analysis. Specifying the DESIGN *o-option* makes the procedure run faster. The DESIGN *o-option* sets the default method to UNIVARIATE and the default MAXITER= value to zero. It suppresses computing the regression coefficients, unless they are needed for some other option. Furthermore, when the DESIGN *o-option* is specified, the MODEL statement is not required to have an equal sign. When no MODEL statement equal sign is specified, all variables are considered independent variables, all options that require dependent variables are ignored, and the IREPLACE *o-option* is implied.

You can use DESIGN=n for coding very large data sets, where n is the number of observations to code at one time. For example, to code a data set with a large number of observations, you can specify DESIGN=100 or DESIGN=1000 to process the data set in blocks of 100 or 1000 observations. If you specify the DESIGN *o-option* rather than DESIGN=n, PROC TRANSREG tries to process all observations at once, which will not work with very large data sets. Specify the NOZEROCONSTANT *a-option* with DESIGN=n to ensure that constant variables within blocks are not zeroed. See the section "Using the DESIGN Output Option" on page 4654 and the section "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 4660.

DEPENDENT=*name*
DEP=*name*

specifies the untransformed dependent variable for OUT= data sets with METHOD=MORALS when there is more than one dependent variable. The default is DEPENDENT=_DEPEND_.

DREPLACE
DRE

replaces the original dependent variables with the transformed dependent variables in the OUT= data set. The names of the transformed variables in the OUT= data set correspond to the names of the original dependent variables in the input data set. By default, both the original dependent variables and transformed dependent variables (with names constructed from the TDPREFIX= (default T) *o-option* and the original dependent variable names) are included in the OUT= data set.

IAPPROXIMATIONS
IAP

outputs the approximations of the transformed independent variables to the OUT= data set. These are the target values for the optimal transformations. The names of the approximation variables are constructed from the AIPREFIX= *o-option* (default A) and the original independent variable names. Specifying the AIPREFIX= *o-option* also implies the IAPPROXIMATIONS *o-option*. The IAPPROXIMATIONS *o-option* is not valid when METHOD=UNIVARIATE.

IREPLACE
IRE

replaces the original independent variables with the transformed independent variables in the OUT= data set. The names of the transformed variables in the OUT= data set correspond to the names of the original independent variables in the input data set. By default, both the original independent variables and transformed independent variables (with names constructed from the TIPREFIX= *o-option* (default T) and the original independent variable names) are included in the OUT= data set.

LEVERAGE<=*name*>
LEV<=*name*>

creates a variable with the specified name in the OUT= data set that contains leverages. Specifying the LEVERAGE *o-option* is equivalent to specifying LEVERAGE=Leverage.

LILPREFIX=*name*
LIL=*name*

specifies a prefix for naming the liberal-individual-lower confidence limits. The default prefix is LIL. Specifying the LILPREFIX= *o-option* also implies the CLI *o-option*.

LIUPREFIX=*name*
LIU=*name*

specifies a prefix for naming the liberal-individual-upper confidence limits. The default prefix is LIU. Specifying the LIUPREFIX= *o-option* also implies the CLI *o-option*.

LMLPREFIX=*name*
LML=*name*

specifies a prefix for naming the liberal-mean-lower confidence limits. The default prefix is LML. Specifying the LMLPREFIX= *o-option* also implies the CLM *o-option*.

LMUPREFIX=*name*
LMU=*name*

specifies a prefix for naming the liberal-mean-upper confidence limits. The default prefix is LMU. Specifying the LMUPREFIX= *o-option* also implies the CLM *o-option*.

MACRO(*keyword=name...***)**
MAC(*keyword=name...***)**

creates macro variables. Most of the options available within the MACRO *o-option* are rarely needed. By default, the TRANSREG procedure creates a macro variable named _TRGIND with a complete list of independent variables created by the procedure. When the TRANSREG procedure is being used for design matrix creation prior to running a procedure without a CLASS statement, this macro provides a convenient way to use the results from PROC TRANSREG. For example, a PROC LOGISTIC step that uses a design matrix coded by PROC TRANSREG could use the following MODEL statement:

```
model y=&_trgind;
```

The TRANSREG procedure, also by default, creates a macro variable named _TRGINDN, which contains the number of variables in the _TRGIND list. This macro variable could be used in an ARRAY statement as follows:

```
array indvars[&_trgindn] &_trgind;
```

See the section "Using the DESIGN Output Option" on page 4654 and the section "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 4660 for examples of using the default macro variables.

The available *keywords* are as follows.

DN=*name* specifies the name of a macro variable that contains the number of dependent variables. By default, a macro variable named _TRGDEPN is created. This is the number of variables in the DL= list and the number of macro variables created by the DV= and DE= specifications.

IN=*name* specifies the name of a macro variable that contains the number of independent variables. By default, a macro variable named _TRGINDN is created. This is the number of variables in the IL= list and the number of macro variables created by the IV= and IE= specifications.

DL=*name* specifies the name of a macro variable that contains the list of the dependent variables. By default, a macro variable named _TRGDEP is created. These are the variable names of the final transformed variables in the OUT= data set. For example, if there are three dependent variables, Y1–Y3, then _TRGDEP contains, by default, TY1 TY2 TY3 (or Y1 Y2 Y3 if you specify the REPLACE *o-option*).

IL=*name* specifies the name of a macro variable that contains the list of the independent variables. By default, a macro variable named _TRGIND is created. These are the variable names of the final transformed variables in the OUT= data set. For example, if there are three independent variables, X1–X3, then _TRGIND contains, by default, TX1 TX2 TX3 (or X1 X2 X3 if you specify the REPLACE *o-option*).

DV=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one dependent variable name. For example, if there are three dependent variables, Y1–Y3, and you specify MACRO(DV=DEP), then three macro variables, DEP1, DEP2, and DEP3, are created, containing TY1, TY2, and TY3, respectively (or Y1, Y2, and Y3 if you specify the REPLACE *o-option*). By default, no list is created.

IV=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one independent variable name. For example, if there are three independent variables, X1–X3, and you specify MACRO(IV=IND), then three macro variables, IND1, IND2, and IND3, are created, containing TX1, TX2, and TX3, respectively (or X1, X2, and X3 if you specify the REPLACE *o-option*). By default, no list is created.

DE=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one dependent variable effect. This list shows the origin of each model term. Each effect consists of two or more parts, and each part consists of a value in 32 columns followed by a blank. For example, if you specify MACRO(DE=D), then a macro variable D1 is created for IDENTITY(Y). The D1 macro variable is shown next, wrapped onto two lines.

```
4                                TY
IDENTITY                         Y
```

The first part is the number of parts (4), the second part is the transformed variable name, the third part is the transformation, and the last part is the input variable name. By default, no list is created.

IE=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one independent variable effect. This list shows the origin of each model term. Each effect consists of two or more parts, and each part consists of a value in 32 columns followed by a blank. For example, if you specify MACRO(IE=I), then three macro variables, I1, I2, and I3, are created for CLASS(X1 | X2) when both X1 and X2 have values of 1 and 2. These macro variables are shown next, but with extra white space removed.

```
5     Tx11     CLASS    x1   1
5     Tx21     CLASS    x2   1
8     Tx11x21  CLASS    x1   1      CLASS    x2   1
```

For CLASS variables, the formatted level appears after the variable name. The first two effects are the main effects, and the last is the interaction term. By default, no list is created.

MEANS
MEA

outputs marginal means for CLASS variable expansions to the OUT= data set.

MEC

outputs multiple regression elliptical point model coordinates to the OUT= data set.

MPC

outputs multiple regression point model coordinates to the OUT= data set.

MQC

outputs multiple regression quadratic point model coordinates to the OUT= data set.

MRC

outputs multiple regression coefficients to the OUT= data set.

MREDUNDANCY
MRE

outputs multiple redundancy analysis coefficients to the OUT= data set.

NORESTOREMISSING
NORESTORE
NOR

specifies that missing values should not be restored when the OUT= data set is created. By default, the coded CLASS variable contains a row of missing values for observations in which the CLASS variable is missing. When you specify the NORESTOREMISSING *o-option*, these observations contain a row of zeros instead. This is useful when the TRANSREG procedure is used to code experimental designs for discrete choice models and there is a constant alternative indicated by a missing value.

NOSCORES
NOS

excludes original variables, transformed variables, predicted values, residuals, and scores from the OUT= data set. You can use the NOSCORES *o-option* with various other options to create an OUT= data set that contains only a coefficient partition (for example, a data set consisting entirely of coefficients and coordinates).

PREDICTED
PRE
P

outputs predicted values, which for METHOD=UNIVARIATE and METHOD=MORALS are the ordinary predicted values from the linear model, to the OUT= data set. The names of the predicted values' variables are constructed from the PPREFIX= *o-option* (default P) and the original dependent variable names. Specifying the PPREFIX= *o-option* also implies the PREDICTED *o-option*.

PPREFIX=*name*
PDPREFIX=*name*
PDP=*name*

specifies a prefix for naming the dependent variable predicted values. The default is PPREFIX=P when you specify the PREDICTED *o-option*; otherwise, it is PPREFIX=A. Specifying the PPREFIX= *o-option* also implies the PREDICTED *o-option*, and the PPREFIX= *o-option* is the same as the ADPREFIX= *o-option*.

RDPREFIX=*name*
RDP=*name*

specifies a prefix for naming the residual (dependent) variables to the OUT= data set. The default is RDPREFIX=R. Specifying the RDPREFIX= *o-option* also implies the RESIDUALS *o-option*.

REDUNDANCY<=STANDARDIZE | UNSTANDARDIZE>
RED<=STA | UNS>

outputs redundancy variables to the OUT= data set, either standardized or unstandardized. Specifying the REDUNDANCY *o-option* is the same as specifying REDUNDANCY=STANDARDIZE. The results of the REDUNDANCY *o-option* depends on the TSTANDARD= option. You must specify TSTANDARD=Z to get results based on standardized data. The TSTANDARD= option controls how the data that go into the redundancy analysis are scaled, and REDUNDANCY=STANDARDIZE|UNSTANDARDIZE controls how the redundancy variables are scaled. The REDUNDANCY *o-option* is implied by METHOD=REDUNDANCY. The RPREFIX= *o-option* specifies a prefix (default Red) for naming the redundancy variables.

REFERENCE=NONE | MISSING | ZERO
REF=NON | MIS | ZER

specifies how reference levels of CLASS variables are to be treated. The options are REFERENCE=NONE, the default, in which reference levels are suppressed; REFERENCE=MISSING, in which reference levels are displayed and output with missing values; and REFERENCE=ZERO, in which reference levels are displayed and output with zeros. The REFERENCE= option can be specified in the PROC TRANSREG, MODEL, or OUTPUT statement, and it can be independently specified for the OUT= data set and the displayed output. When you specify it in only one statement, it sets the option for both the displayed output and the OUT= data set.

REPLACE
REP

is equivalent to specifying both the DREPLACE and the IREPLACE *o-options*.

RESIDUALS
RES
R

outputs the differences between the transformed dependent variables and their predicted values. The names of the residual variables are constructed from the RDPREFIX= *o-option* (default R) and the original dependent variable names.

RPREFIX=*name*
RPR=*name*

provides a prefix for naming the redundancy variables. The default is RPREFIX=Red. Specifying the RPREFIX= *o-option* also implies the REDUNDANCY *o-option*.

TDPREFIX=*name*
TDP=*name*

specifies a prefix for naming the transformed dependent variables. By default, TDPREFIX=T. The TDPREFIX= *o-option* is ignored when you specify the DREPLACE *o-option*.

TIPREFIX=*name*
TIP=*name*

specifies a prefix for naming the transformed independent variables. By default, TIPREFIX=T. The TIPREFIX= *o-option* is ignored when you specify the IREPLACE *o-option*.

WEIGHT Statement

WEIGHT *variable* **;**

When you use a WEIGHT statement, a weighted residual sum of squares is minimized. The WEIGHT statement has no effect on degrees of freedom or number of observations, but the weights affect most other calculations. The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than 0.

Details

Model Statement Usage

MODEL < transform(dependents < / t-options >)
< transform(dependents < / t-options >)...> = >
transform(independents < / t-options >)
< transform(independents < / t-options >)...>< / a-options > **;**

Here are some examples of model statements:

- linear regression

```
model identity(y) = identity(x);
```

- a linear model with a nonlinear regression function

```
model identity(y) = spline(x / nknots=5);
```

- multiple regression

```
model identity(y) = identity(x1-x5);
```

- multiple regression with nonlinear transformations

```
model spline(y / nknots=3) = spline(x1-x5 / nknots=3);
```

- multiple regression with nonlinear but monotone transformations

```
model mspline(y / nknots=3) = mspline(x1-x5 / nknots=3);
```

- multivariate multiple regression

```
model identity(y1-y4) = identity(x1-x5);
```

- canonical correlation

```
model identity(y1-y4) = identity(x1-x5) / method=canals;
```

- redundancy analysis

```
model identity(y1-y4) = identity(x1-x5) / method=redundancy;
```

- preference mapping, vector model (Carroll 1972)

```
model identity(Attrib1-Attrib3) = identity(Dim1-Dim2);
```

- preference mapping, ideal point model (Carroll 1972)

```
model identity(Attrib1-Attrib3) = point(Dim1-Dim2);
```

- preference mapping, ideal point model, elliptical (Carroll 1972)

```
model identity(Attrib1-Attrib3) = epoint(Dim1-Dim2);
```

- preference mapping, ideal point model, quadratic (Carroll 1972)

```
model identity(Attrib1-Attrib3) = qpoint(Dim1-Dim2);
```

- metric conjoint analysis

```
model identity(Subj1-Subj50) = class(a b c d e f / zero=sum);
```

- nonmetric conjoint analysis

```
model monotone(Subj1-Subj50) = class(a b c d e f / zero=sum);
```

- main effects, two-way interaction

  ```
  model identity(y) = class(a|b);
  ```

- less-than-full-rank model—main effects and two-way interaction are constrained to sum to zero

  ```
  model identity(y) = class(a|b / zero=sum);
  ```

- main effects and all two-way interactions

  ```
  model identity(y) = class(a|b|c@2);
  ```

- main effects and all two- and three-way interactions

  ```
  model identity(y) = class(a|b|c);
  ```

- main effects and just B*C two-way interaction

  ```
  model identity(y) = class(a b c b*c);
  ```

- seven main effects, three two-way interactions

  ```
  model identity(y) = class(a b c d e f g a*b a*c a*d);
  ```

- deviations-from-means (effects or $(1, 0, -1)$) coding, with an A reference level of ’1’ and a B reference level of ’2’

  ```
  model identity(y) = class(a|b / deviations zero=’1’ ’2’);
  ```

- cell-means coding (implicit intercept)

  ```
  model identity(y) = class(a*b / zero=none);
  ```

- reference cell model

  ```
  model identity(y) = class(a|b / zero=’1’ ’1’);
  ```

- reference line with change in line parameters

  ```
  model identity(y) = class(a) | identity(x);
  ```

- reference curve with change in curve parameters

  ```
  model identity(y) = class(a) | spline(x);
  ```

- separate curves and intercepts

  ```
  model identity(y) = class(a / zero=none) | spline(x);
  ```

- quantitative effects with interaction

  ```
  model identity(y) = identity(x1 | x2);
  ```

- separate quantitative effects with interaction within each cell

  ```
  model identity(y) = class(a * b / zero=none) | identity(x1 | x2);
  ```

Box-Cox Transformations

The Box-Cox (1964) transformation has the form

$$\begin{array}{ll} (y^{\lambda} - 1)/\lambda & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{array}$$

This family of transformations of the positive dependent variable y is controlled by the parameter λ. Transformations linearly related to square root, inverse, quadratic, cubic, and so on are all special cases. The limit as λ approaches 0 is the log transformation. More generally, Box-Cox transformations of the following form can be fit:

$$\begin{array}{ll} ((y+c)^{\lambda} - 1)/(\lambda g) & \lambda \neq 0 \\ \log(y+c)/g & \lambda = 0 \end{array}$$

By default, $c = 0$. The parameter c can be used to rescale y so that it is strictly positive. By default, $g = 1$. Alternatively, g can be $\dot{y}^{\lambda-1}$ where $\dot{y}$ is the geometric mean of y.

The BOXCOX transformation in PROC TRANSREG can be used to perform a Box-Cox transformation of the dependent variable. You can specify a list of power parameters using the LAMBDA= transformation option. By default, LAMBDA=-3 TO 3 BY 0.25. The procedure chooses the optimal power parameter using a maximum likelihood criterion (Draper and Smith 1981, pp. 225-226). You can specify the PARAMETER=c transformation option when you want to shift the values of y, usually to avoid negatives. To divide by $\dot{y}^{\lambda-1}$, specify the GEOMETRICMEAN transformation option.

Here are some examples of usage of the LAMBDA= option:

```
model BoxCox(y / lambda=0) = identity(x1-x5);
model BoxCox(y / lambda=-2 to 2 by 0.1) = identity(x1-x5);
model BoxCox(y) = identity(x1-x5);
```

In the first example

```
model BoxCox(y / lambda=0) = identity(x1-x5);
```

LAMBDA=0 specifies a Box-Cox transformation with a power parameter of 0. Since a single value of 0 was specified for LAMBDA=, there is no difference between the following models:

```
model BoxCox(y / lambda=0) = identity(x1-x5);
model log(y) = identity(x1-x5);
```

In the second example

```
model BoxCox(y / lambda=-2 to 2 by 0.1) = identity(x1-x5);
```

there is a list of power parameters specified. This tells PROC TRANSREG to find a Box-Cox transformation before the usual iterations begin. PROC TRANSREG tries each power parameter in the list and picks the best transformation. A maximum likelihood approach (Draper and Smith 1981, pp. 225-226) is used. Note that this is quite different from TRANSREG's usual approach of iteratively finding optimal transformations. It is analogous to SMOOTH, RANK, and the other nonoptimal transformations that are performed before the iterations begin.

In the third example

```
model BoxCox(y) = identity(x1-x5);
```

the default list of -3 TO 3 BY 0.25 is used.

The procedure prints the optimal power parameter, a confidence interval on the power parameter (using the ALPHA= transformation option), a “convenient” power parameter (selected from the CLL= option list), and the log likelihood for each power parameter tried (see Example 75.6).

Smoothing Splines

You can use PROC TRANSREG to output to a SAS data set the same smoothing splines that the GPLOT procedure creates. The SMOOTH transformation is a noniterative transformation for smoothing splines. The smoothing parameter can be specified with either the SM= or the PARAMETER= *o-option*. The independent variable transformation (Tx in this case) contains the results. The GPLOT request y*x=2 with I=SM50 creates the same curve as Tx*x.

```
title 'Smoothing Splines';

data x;
   do x = 1 to 100 by 2;
      do rep = 1 to 3;
         y = log(x) + sin(x / 10) + normal(7);
         output;
         end;
      end;
run;

proc transreg;
   model identity(y) = smooth(x / sm=50);
   output;
run;

%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
proc gplot;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   plot y*x=1 y*x=2 tx*x=3 / &opts overlay;
   symbol1 color=blue   v=star i=none;
   symbol2 color=yellow v=none i=sm50;
   symbol3 color=cyan   v=dot  i=none;
run; quit;
```

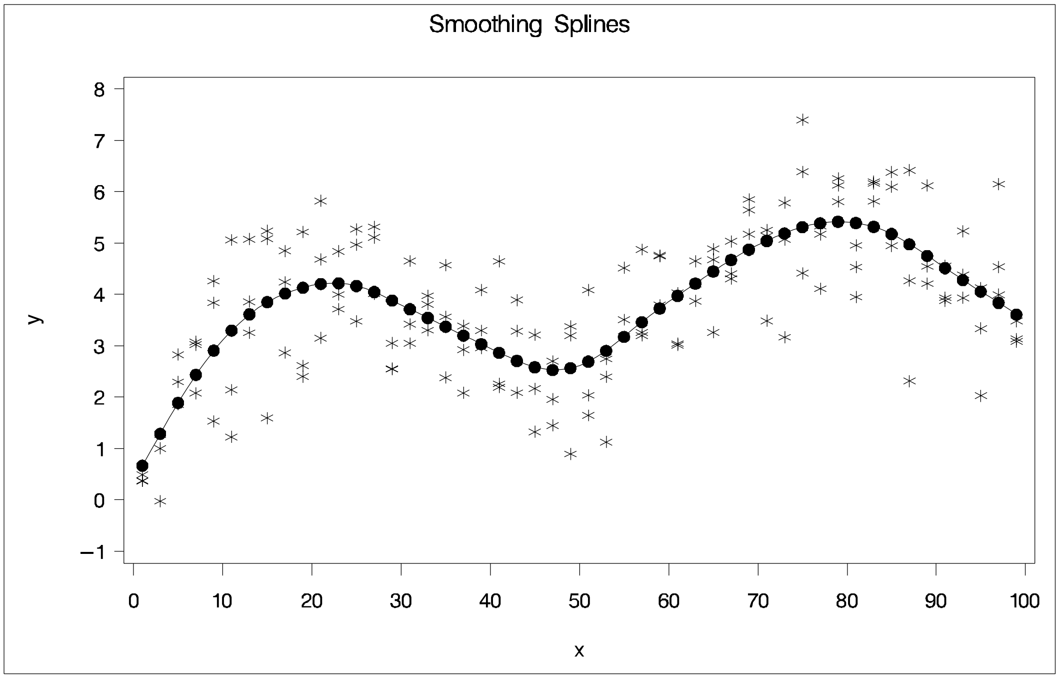


Figure 75.5. Smoothing Spline Example 1

When you cross a SMOOTH variable with a CLASS variable, specify ZERO=NONE with the CLASS expansion and the AFTER *t-option* with the SMOOTH transformation so that separate functions are found within each group.

```
title2 'Two Groups';

data x;
   do x = 1 to 100;
      group = 1;
      do rep = 1 to 3;
         y = log(x) + sin(x / 10) + normal(7);
         output;
         end;
      group = 2;
      do rep = 1 to 3;
         y = -log(x) + cos(x / 10) + normal(7);
         output;
         end;
      end;
run;
```

```
proc transreg;
   model identity(y) = class(group / zero=none) |
                       smooth(x / after sm=50);
   output out=curves;
run;

data curves2;
   set curves;
   if group1 = 0 then tgroup1x = .;
   if group2 = 0 then tgroup2x = .;
run;

%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
proc gplot;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   plot y*x=1 tgroup1x*x=2 tgroup2x*x=2 / &opts overlay;
   symbol1 color=blue   v=star i=none;
   symbol2 color=yellow v=none i=join;
run; quit;
```

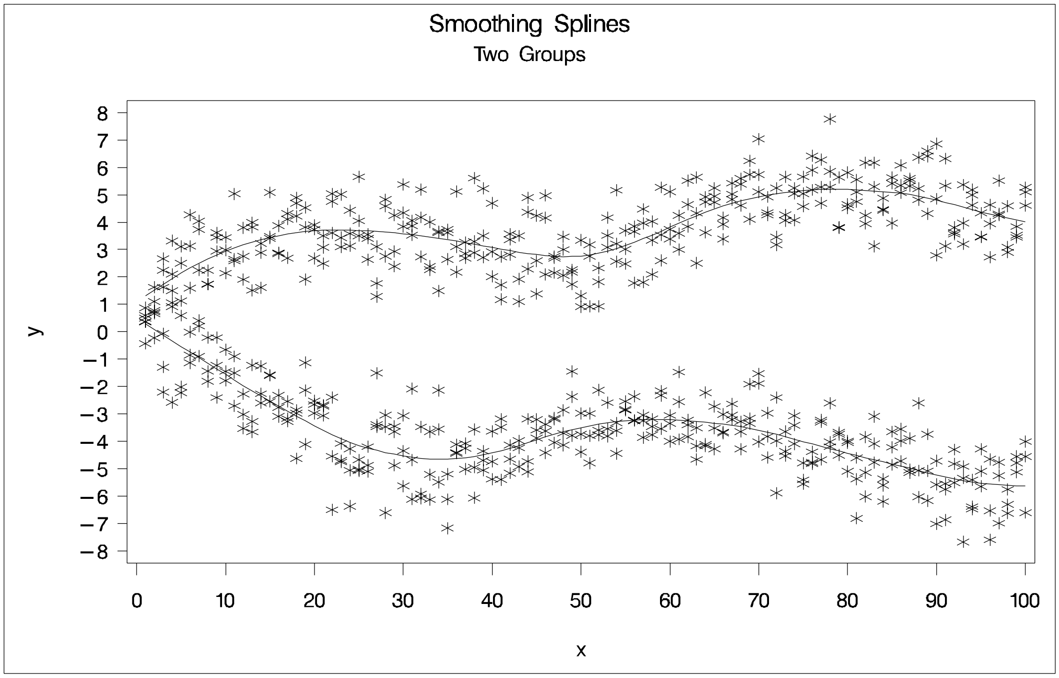


Figure 75.6. Smoothing Spline Example 2

The SMOOTH transformation is valid only with independent variables; typically, it is used in models with a single independent and a single dependent variable. When there are multiple independent variables designated as SMOOTH, the TRANSREG procedure tries to smooth the ith independent variable using the ith dependent variable as a target. When there are more independent variables than dependent variables, the last dependent variable is reused as often as is necessary. For example, for the model

```
   model identity(y1-y3) = smooth(x1-x5);
```

smoothing is based on the pairs (y1, x1), (y2, x2), (y3, x3), (y3, x4), and (y3, x5).

The SMOOTH transformation is a noniterative transformation; smoothing occurs once per variable before the iterations begin. In contrast, SSPLINE provides an iterative smoothing spline transformation. It does not generally minimize squared error; hence, divergence is possible with SSPLINE.

Missing Values

PROC TRANSREG can estimate missing values, with or without category or monotonicity constraints, so that the regression model fit is optimized. Several approaches to missing data handling are provided. All observations with missing values in IDENTITY, CLASS, POINT, EPOINT, QPOINT, SMOOTH, PSPLINE, and BSPLINE variables are excluded from the analysis. When METHOD=UNIVARIATE (specified in the PROC TRANSREG or MODEL statement), observations with missing values in any of the independent variables are excluded from the analysis. When you specify the NOMISS *a-option*, observations with missing values in the other analysis variables are excluded. Otherwise, missing data are estimated, using variable means as initial estimates.

You can specify the LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, SSPLINE, LOG, LOGIT, POWER, ARSIN, BOXCOX, RANK, and EXP transformations in any combination with nonmissing values, ordinary missing values, and special missing values, as long as the nonmissing values in each variable have positive variance. No category or order restrictions are placed on the estimates of ordinary missing values. You can force missing value estimates within a variable to be identical by using special missing values (refer to “DATA Step Processing” in *SAS Language Reference: Concepts*). You can specify up to 27 categories of missing values, in which within-category estimates must be the same, by coding the missing values using ._ and .A through .Z.

You can also specify an ordering of some missing value estimates. You can use the MONOTONE= *a-option* in the PROC TRANSREG or MODEL statement to indicate a range of special missing values (a subset of the list from .A to .Z) with estimates that must be weakly ordered within each variable in which they appear. For example, if MONOTONE=AI, the nine classes, .A, .B,. . ., .I, are monotonically scored and optimally scaled just as MONOTONE transformation values are scored. In this case, category but not order restrictions are placed on the missing values ._ and .J through .Z. You can also use the UNTIE= *a-option* (in the PROC TRANSREG or MODEL statement) to indicate a range of special missing values with estimates that must be weakly ordered within each variable in which they appear but can be untied.

The missing value estimation facilities allow for partitioned or mixed-type variables. For example, a variable can be considered part nominal and part ordinal. Nominal classes of otherwise ordinal variables are coded with special missing values. This feature can be useful with survey research. The class “unfamiliar with the product” in the variable “Rate your preference for ’Brand X’ on a 1 to 9 scale, or if you are unfamiliar with the product, check ’unfamiliar with the product’” is an example. You can code “unfamiliar with the product” as a special missing value, such as .A. The 1s to 9s can be monotonically transformed, while no monotonic restrictions are placed on the quantification of the “unfamiliar with the product” class.

A variable specified for a LINEAR transformation, with special missing values and ordered categorical missing values, can be part interval, part ordinal, and part nominal. A variable specified for a MONOTONE transformation can have two independent ordinal parts. A variable specified for an UNTIE transformation can have an

ordered categorical part and an ordered part without category restrictions. Many other mixes are possible.

Missing Values, UNTIE, and Hypothesis Tests

The TRANSREG procedure has the ability to estimate missing data and monotonically transform variables while untying tied values. Estimates of ordinary missing values (.) may all be different. Analyses with UNTIE transformations, the UNTIE= *a-option*, and ordinary missing data estimation are all prone to degeneracy problems. Consider the following example. A perfect fit is found by collapsing all observations except the one with two missing values into a single value in Y and X1.

```
data x;
   input y x1 x2 @@;
   datalines;
1 3 7    8 3 9    1 8 6    . . 9    3 3 9
8 5 1    6 7 3    2 7 2    1 8 2    . 9 1
;

proc transreg dummy;
   model linear(y) = linear(x1 x2);
   output;
run;

proc print;
run;
```

```
Obs  _TYPE_   _NAME_   y      Ty     Intercept   x1   x2   TIntercept     Tx1     Tx2

  1  SCORE    ROW1     1    2.7680       1        3    7        1        5.1233    7
  2  SCORE    ROW2     8    2.7680       1        3    9        1        5.1233    9
  3  SCORE    ROW3     1    2.7680       1        8    6        1        5.1233    6
  4  SCORE    ROW4     .   12.5878       1        .    9        1       12.7791    9
  5  SCORE    ROW5     3    2.7680       1        3    9        1        5.1233    9
  6  SCORE    ROW6     8    2.7680       1        5    1        1        5.1233    1
  7  SCORE    ROW7     6    2.7680       1        7    3        1        5.1233    3
  8  SCORE    ROW8     2    2.7680       1        7    2        1        5.1233    2
  9  SCORE    ROW9     1    2.7680       1        8    2        1        5.1233    2
 10  SCORE    ROW10    .    2.7680       1        9    1        1        5.1233    1
```

Figure 75.7. Missing Values Example

Generally, the use of ordinary missing data estimation, the UNTIE transformation, and the UNTIE= *a-option* should be avoided, particularly with hypothesis tests. With these options, parameters are estimated based on only a single observation, and they can exert tremendous influence over the results. Each of these parameters has one model degree of freedom associated with it, so small or zero error degrees of freedom can also be a problem.

Controlling the Number of Iterations

Several *a-options* in the PROC TRANSREG or MODEL statement control the number of iterations performed. Iteration terminates when any one of the following conditions is satisfied:

- The number of iterations equals the value of the MAXITER= *a-option*.
- The average absolute change in variable scores from one iteration to the next is less than the value of the CONVERGE= *a-option*.
- The criterion change is less than the value of the CCONVERGE= *a-option*.

You can specify negative values for either convergence option if you wish to define convergence only in terms of the other option. The criterion change can become negative when the data have converged so that it is numerically impossible, within machine precision, to increase the criterion. Usually, a negative criterion change is the result of very small amounts of rounding error since the algorithms are (usually) convergent. However, there are other cases where a negative criterion change is a sign of divergence, which is not necessarily an error. When you specify an SSPLINE transformation or the REITERATE or DUMMY *a-option*, divergence may be perfectly normal.

When there are no monotonicity constraints and there is only one canonical variable in each set, PROC TRANSREG (with the DUMMY *a-option*) can usually find the optimal solution in only one iteration. (There are no monotonicity constraints when the MONOTONE, MSPLINE, or UNTIE transformations and the UNTIE= and MONOTONE= *a-options* are not specified. There is only one canonical variable in each set when METHOD=MORALS or METHOD=UNIVARIATE, or when METHOD=REDUNDANCY with only one dependent variable, or when METHOD=CANALS and NCAN=1.)

The initialization iteration is number 0. When there are no monotonicity constraints and there is only one canonical variable in each set, the next iteration shows no change and iteration stops. At least two iterations (0 and 1) are performed with the DUMMY *a-option* even if nothing changes in iteration 0. The MONOTONE, MSPLINE, and UNTIE variables are not transformed by the dummy variable initialization. Note that divergence with the DUMMY *a-option*, particularly in the second iteration, is not an error. The initialization iteration is slower and uses more memory than other iterations. However, for many models, specifying the DUMMY *a-option* can greatly decrease the amount of time required to find the optimal transformations. Furthermore, by solving for the transformations directly instead of iteratively, PROC TRANSREG avoids certain nonoptimal solutions.

You can increase the number of iterations to ensure convergence by increasing the value of the MAXITER= *a-option* and decreasing the value of the CONVERGE= *a-option*. Since the average absolute change in standardized variable scores seldom decreases below 1E−11, you should not specify a value for the CONVERGE= *a-option* less than 1E−8 or 1E−10. Most of the data changes occur during the first few iterations, but the data can still change after 50 or even 100 iterations. You can try different combinations of values for the CONVERGE= and MAXITER= *a-options* to ensure convergence without extreme overiteration. If the data do not converge with the default specifications, try CONVERGE=1E−8 and MAXITER=50, or CONVERGE=1E−10 and MAXITER=200. Note that you can specify the REITERATE *a-option* to start iterating where the previous analysis stopped.

Using the REITERATE Algorithm Option

You can use the REITERATE *a-option* to perform additional iterations when PROC TRANSREG stops before the data have adequately converged. For example, suppose that you execute the following code:

```
proc transreg data=a;
   model mspline(y) = mspline(x1-x5);
   output out=b coefficients;
run;
```

If the transformations do not converge in the default 30 iterations, you can perform more iterations without repeating the first 30 iterations.

```
proc transreg data=b reiterate;
   model mspline(y) = mspline(x1-x5);
   output out=b coefficients;
run;
```

Note that a WHERE statement is not necessary to exclude the coefficient observations. They are automatically excluded because their _TYPE_ value is not SCORE.

You can also use the REITERATE *a-option* to specify starting values other than the original values for the transformations. Providing alternate starting points may avoid local optima. Here are two examples.

```
proc transreg data=a;
   model rank(y) = rank(x1-x5);
   output out=b;
run;

proc transreg data=b reiterate;
   /* Use ranks as the starting point. */
   model mspline(y) = mspline(x1-x5);
   output out=c coefficients;
run;
```

```
data b;
   set a;
   array tx[6] ty tx1-tx5;
   do j = 1 to 6;
      tx[j] = normal(7);
   end;
run;

proc transreg data=b reiterate;
   /* Use a random starting point. */
   model mspline(y) = mspline(x1-x5);
   output out=c coefficients;
run;
```

Note that divergence with the REITERATE *a-option*, particularly in the second iteration, is not an error since the initial transformation is not required to be a valid member of the transformation family. When you specify the REITERATE *a-option*, the iteration does not terminate when the criterion change is negative during the first 10 iterations.

Avoiding Constant Transformations

There are times when the optimal scaling produces a constant transformed variable. This can happen with the MONOTONE, UNTIE, and MSPLINE transformations when the target is negatively correlated with the original input variable. It can happen with all transformations when the target is uncorrelated with the original input variable. When this happens, the procedure modifies the target to avoid a constant transformation. This strategy avoids certain nonoptimal solutions.

If the transformation is monotonic and a constant transformed variable results, the procedure multiplies the target by -1 and tries the optimal scaling again. If the transformation is not monotonic or if the multiplication by -1 did not help, the procedure tries using a random target. If the transformation is still constant, the previous nonconstant transformation is retained. When a constant transformation is avoided by any strategy, a message is displayed: “A constant transformation was avoided for *name*.”

With extreme collinearity, small amounts of rounding error might interact with the instability of the coefficients to produce target vectors that are not positively correlated with the original scaling. If a regression coefficient for a variable is zero, the formula for the target for that variable contains a zero divide. In a multiple regression model, after many iterations, one independent variable can be scaled the same way as the current scaling of the dependent variable, so the other independent variables have coefficients of zero. When the constant transformation warning appears, you should interpret your results with extreme caution, and recheck your model.

Constant Variables

Constant and almost constant variables are zeroed and ignored. As long as the dependent variable is not constant, PROC TRANSREG produces an iteration history table for all models, not just models in which the variables can change. When constant variables are expected and should not be zeroed, specify the NOZEROCONSTANT option.

Character OPSCORE Variables

Character OPSCORE variables are replaced by a numeric variable containing category numbers before the iterations, and the character values are discarded. Only the first eight characters are considered when determining category membership. If you want the original character variable in the output data set, give it a different name in the OPSCORE specification (OPSCORE(x / name=(x2)) and name the original variable on the ID statement (ID x;).

Convergence and Degeneracies

When you specify the SSPLINE transformation, divergence is normal. The rest of this section assumes that you did not specify SSPLINE. For all the methods available in PROC TRANSREG, the algorithms are convergent, both in terms of the criterion being optimized and the parameters being estimated. The value of the criterion being maximized (squared multiple correlation, average squared multiple correlation, or average squared canonical correlation) can, theoretically, never decrease from one iteration to the next. The values of the parameters being solved for (the scores and weights of the transformed variables) become stable after sufficient iteration.

In practice, the criterion being maximized can decrease with overiteration. When the statistic has very nearly reached its maximum, further iterations might report a decrease in the criterion in the last few decimal places. This is a normal result of very small amounts of rounding error. By default, iteration terminates when this occurs because, by default, CCONVERGE=0.0. Specifying CCONVERGE=-1, an impossible change, turns off this check for convergence.

Even though the algorithms are convergent, they might not converge to a global optimum. Also, under extreme circumstances, the solution might degenerate. Because two points always form a straight line, the algorithms sometimes try to reach this degenerate optimum. This sometimes occurs when one observation is an ordinal outlier (when one observation has the extreme rank on all variables). The algorithm can reach an optimal solution that ties all other categories producing two points. Similar results can occur when there are many missing values. More generally, whenever there are very few constraints on the scoring of one or more points, degeneracies can be a problem. In a well-behaved analysis, the maximum data change, average data change, and criterion change all decrease at a rapid rate with each iteration. When the rate of change increases for several iterations, the solution might be degenerating.

Implicit and Explicit Intercepts

Depending on several options, the model intercept is nonzero, zero, or implicit, or there is no intercept. Ordinarily, the model contains an explicit nonzero intercept, and the Intercept variable in the OUT= data set contains ones. When TSTANDARD=CENTER or TSTANDARD=Z is specified, the model contains an explicit, zero intercept and the Intercept variable contains zeros. When METHOD=CANALS, the model is fit with centered variables and the Intercept variable is set to missing.

If you specify CLASS with ZERO=NONE or BSPLINE for one or more independent variables, and TSTANDARD=NOMISS or TSTANDARD=ORIGINAL (the default), an implicit intercept model is fit. The intercept is implicit in a set of the independent variables since there exists a set of independent variables the sum of which is a column of ones. All statistics are mean corrected. The implicit intercept is not an option; it is implied by the model.

With METHOD=CANALS, the Intercept variable contains the *canonical intercept* for canonical coefficients observations: $\hat{\beta}_0 = \overline{\mathbf{y}}'\hat{\boldsymbol{\alpha}} - \overline{\mathbf{x}}'\hat{\boldsymbol{\beta}}$ where $\mathbf{Y}\hat{\boldsymbol{\alpha}} \approx \mathbf{X}\hat{\boldsymbol{\beta}}$.

Passive Observations

Observations may be excluded from the analysis for several reasons; these include zero weight; zero frequency; missing values in variables designated IDENTITY, CLASS, POINT, EPOINT, QPOINT, SMOOTH, PSPLINE, or BSPLINE; and missing values with the NOMISS *a-option* specified. These observations are passive in that they do not contribute to determining transformations, R^2, sums of squares, degrees of freedom, and so on. However, some information can be computed for them. For example, if no independent variable values are missing, predicted values and redundancy variable values can both be computed. Residuals can be computed for observations with a nonmissing dependent and nonmissing predicted value. Canonical variables for dependent variables can be computed when no dependent variables are missing; canonical variables for independent variables can be computed when no independent variables are missing, and so on. Passive observations in the OUT= data set have a blank value for _TYPE_.

Point Models

The expanded set of independent variables generated from the POINT, EPOINT, and QPOINT expansions can be used to perform ideal point regressions (Carroll 1972) and compute ideal point coordinates for plotting in a biplot (Gabriel 1981). The three types of ideal point coordinates can all be described as transformed coefficients. Assume that m independent variables are specified in one of the three point expansions. Let $\mathbf{b}'$ be a $1 \times m$ row vector of coefficients for these variables and one of the dependent variables. Let $\mathbf{R}$ be a matrix created from the coefficients of the extra variables. When coordinates are requested with the MPC, MEC, or MQC *o-options*, $\mathbf{b}'$ and $\mathbf{R}$ are created from multiple regression coefficients. When coordinates are requested with the CPC, CEC, or CQC *o-options*, $\mathbf{b}'$ and $\mathbf{R}$ are created from canonical coefficients.

If you specify the POINT expansion in the MODEL statement, $\mathbf{R}$ is an $m \times m$ identity matrix times the coefficient for the sums of squares (_ISSQ_) variable. If you specify the EPOINT expansion, $\mathbf{R}$ is an $m \times m$ diagonal matrix of coefficients from the squared variables. If you specify the QPOINT expansion, $\mathbf{R}$ is an $m \times m$ symmetric matrix of coefficients from the squared variables on the diagonal and crossproduct variables off the diagonal. The MPC, MEC, MQC, CPC, CEC, and CQC ideal point coordinates are defined as $-0.5\mathbf{b}'\mathbf{R}^{-1}$. When $\mathbf{R}$ is singular, the ideal point coordinates are infinitely far away and are set to missing, so you should try a simpler version of the model. The version that is simpler than the POINT model is the vector model where no extra variables are created. In the vector model, designate all independent variables as IDENTITY. Then draw vectors from the origin to the COEFFICIENTS points.

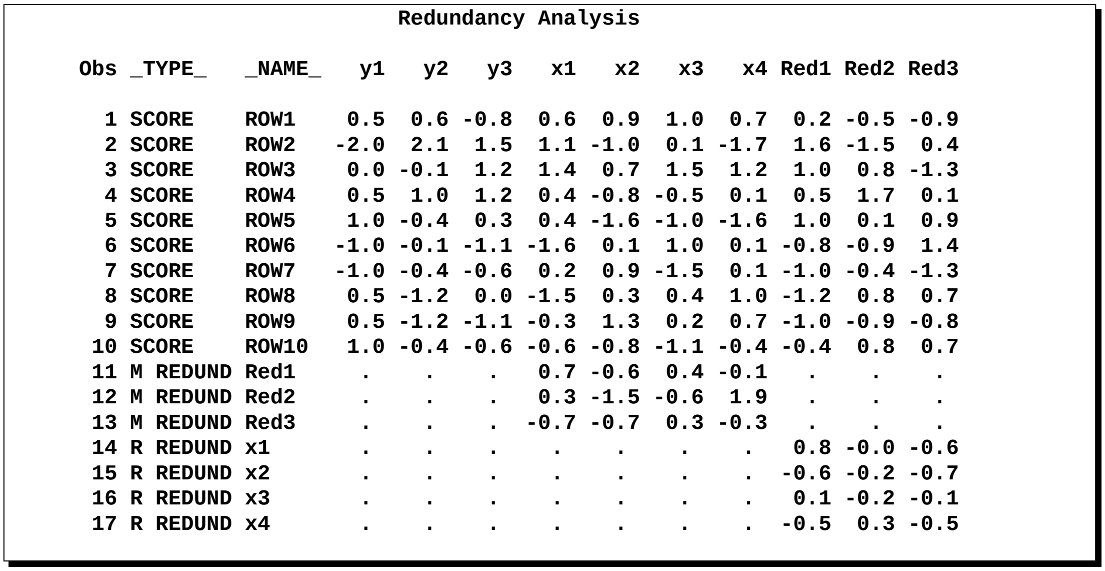
Typically, when you request ideal point coordinates, the MODEL statement should consist of a single transformation for the dependent variables (usually IDENTITY, MONOTONE, or MSPLINE) and a single expansion for the independent variables (one of POINT, EPOINT, or QPOINT).

Redundancy Analysis

Redundancy analysis (Stewart and Love 1968) is a principal component analysis of multivariate regression predicted values. These first steps show the redundancy analysis results produced by PROC TRANSREG. The specification TSTANDARD=Z standardizes all variables to mean zero and variance one. METHOD=REDUNDANCY specifies redundancy analysis and outputs the redundancy variables to the OUT= data set. The MREDUNDANCY *o-option* outputs two sets of redundancy analysis coefficients to the OUT= data set.

```
title 'Redundancy Analysis';

data x;
   input y1-y3 x1-x4;
   datalines;
 6  8  8 15 18 26 27
 1 12 16 18  9 20  8
 5  6 15 20 17 29 31
 6  9 15 14 10 16 22
 7  5 12 14  6 13  9
 3  6  7  2 14 26 22
 3  5  9 13 18 10 22
 6  3 11  3 15 22 29
 6  3  7 10 20 21 27
 7  5  9  8 10 12 18
;

proc transreg data=x tstandard=z method=redundancy;
   model identity(y1-y3) = identity(x1-x4);
   output out=red mredundancy replace;
run;

proc print data=red(drop=Intercept);
```

```
      format _numeric_ 4.1;
   run;
```



```
                                Redundancy Analysis

     Obs _TYPE_   _NAME_   y1   y2   y3   x1   x2   x3   x4 Red1 Red2 Red3

       1 SCORE    ROW1    0.5  0.6 -0.8  0.6  0.9  1.0  0.7  0.2 -0.5 -0.9
       2 SCORE    ROW2   -2.0  2.1  1.5  1.1 -1.0  0.1 -1.7  1.6 -1.5  0.4
       3 SCORE    ROW3    0.0 -0.1  1.2  1.4  0.7  1.5  1.2  1.0  0.8 -1.3
       4 SCORE    ROW4    0.5  1.0  1.2  0.4 -0.8 -0.5  0.1  0.5  1.7  0.1
       5 SCORE    ROW5    1.0 -0.4  0.3  0.4 -1.6 -1.0 -1.6  1.0  0.1  0.9
       6 SCORE    ROW6   -1.0 -0.1 -1.1 -1.6  0.1  1.0  0.1 -0.8 -0.9  1.4
       7 SCORE    ROW7   -1.0 -0.4 -0.6  0.2  0.9 -1.5  0.1 -1.0 -0.4 -1.3
       8 SCORE    ROW8    0.5 -1.2  0.0 -1.5  0.3  0.4  1.0 -1.2  0.8  0.7
       9 SCORE    ROW9    0.5 -1.2 -1.1 -0.3  1.3  0.2  0.7 -1.0 -0.9 -0.8
      10 SCORE    ROW10   1.0 -0.4 -0.6 -0.6 -0.8 -1.1 -0.4 -0.4  0.8  0.7
      11 M REDUND Red1     .    .    .   0.7 -0.6  0.4 -0.1   .    .    .
      12 M REDUND Red2     .    .    .   0.3 -1.5 -0.6  1.9   .    .    .
      13 M REDUND Red3     .    .    .  -0.7 -0.7  0.3 -0.3   .    .    .
      14 R REDUND x1       .    .    .    .    .    .    .   0.8 -0.0 -0.6
      15 R REDUND x2       .    .    .    .    .    .    .  -0.6 -0.2 -0.7
      16 R REDUND x3       .    .    .    .    .    .    .   0.1 -0.2 -0.1
      17 R REDUND x4       .    .    .    .    .    .    .  -0.5  0.3 -0.5
```

Figure 75.8. Redundancy Analysis Example

The _TYPE_='SCORE' observations of the Red1–Red3 variables contain the redundancy variables. The nonmissing "M REDUND" values are coefficients for predicting the redundancy variables from the independent variables. The nonmissing "R REDUND" values are coefficients for predicting the independent variables from the redundancy variables.

These following steps show how to generate the same results manually. The data set is standardized, predicted values are computed, and principal components of the predicted values are computed. The following statements produce the redundancy variables, shown in Figure 75.9:

```
   proc standard data=x out=std m=0 s=1;
      title2 'Manually Generate Redundancy Variables';
   run;

   proc reg noprint data=std;
      model y1-y3 = x1-x4;
      output out=p p=ay1-ay3;
   run; quit;

   proc princomp data=p cov noprint std out=p;
      var ay1-ay3;
   run;

   proc print data=p(keep=Prin:);
      format _numeric_ 4.1;
   run;
```

```
                     Redundancy Analysis
               Manually Generate Redundancy Variables

                   Obs    Prin1    Prin2    Prin3

                     1     0.2     -0.5     -0.9
                     2     1.6     -1.5      0.4
                     3     1.0      0.8     -1.3
                     4     0.5      1.7      0.1
                     5     1.0      0.1      0.9
                     6    -0.8     -0.9      1.4
                     7    -1.0     -0.4     -1.3
                     8    -1.2      0.8      0.7
                     9    -1.0     -0.9     -0.8
                    10    -0.4      0.8      0.7
```

Figure 75.9. Redundancy Analysis Example

The following statements produce the coefficients for predicting the redundancy variables from the independent variables, shown in Figure 75.10:

```
proc reg data=p outest=redcoef noprint;
   title2 'Manually Create Redundancy Coefficients';
   model Prin1-Prin3 = x1-x4;
run; quit;

proc print data=redcoef(keep=x1-x4);
   format _numeric_ 4.1;
run;
```

```
                     Redundancy Analysis
              Manually Create Redundancy Coefficients

                Obs      x1       x2       x3       x4

                  1      0.7     -0.6      0.4     -0.1
                  2      0.3     -1.5     -0.6      1.9
                  3     -0.7     -0.7      0.3     -0.3
```

Figure 75.10. Redundancy Analysis Example

The following statements produce the coefficients for predicting the independent variables from the redundancy variables, shown in Figure 75.11:

```
proc reg data=p outest=redcoef2 noprint;
   title2 'Manually Create Other Coefficients';
   model x1-x4 = prin1-prin3;
run; quit;

proc print data=redcoef2(keep=Prin1-Prin3);
   format _numeric_ 4.1;
run;
```

```
                      Redundancy Analysis
                Manually Create Other Coefficients

                   Obs    Prin1    Prin2    Prin3

                    1      0.8     -0.0     -0.6
                    2     -0.6     -0.2     -0.7
                    3      0.1     -0.2     -0.1
                    4     -0.5      0.3     -0.5
```

Figure 75.11. Redundancy Analysis Example

Optimal Scaling

An alternating least-squares optimal scaling algorithm can be divided into two major stages. The first stage estimates the parameters of the linear model. These parameters are used to create the predicted values or target for each variable that can be transformed. Each target minimizes squared error (as explained in the discussion of the algorithms in *SAS Technical Report R-108*. The definition of the target depends on many factors, such as whether a variable is independent or dependent, which algorithm is used (for example, regression, redundancy, CANALS, principal components), and so on. The definition of the target is independent of the transformation family you specify for the variable. However, the target values for a variable typically do not fit the prescribed transformation family for the variable. They might not have the right category structure; they might not have the right order; they might not be a linear combination of the columns of a B-spline basis; and so on.

The second major stage is optimal scaling. Optimal scaling can be defined as a possibly constrained, least-squares regression problem. When you specify an optimal transformation, or when missing data are estimated for any variable, the full representation of the variable is not simply a vector; it is a matrix with more than one column. The optimal scaling phase finds the vector that is a linear combination of the columns of this matrix, that is closest to the target (in terms of minimum squared error), among those that do not violate any of the constraints imposed by the transformation family. Optimal scaling methods are independent of the data analysis method that generated the target. In all cases, optimal scaling can be accomplished by creating a design matrix based on the original scaling of the variable and the transformation family specified for that variable. The optimally scaled variable is a linear combination of the columns of the design matrix. The coefficients of the linear combination are found using (possibly constrained) least squares. Many optimal scaling problems are solved without actually constructing design and projection matrices. The following two sections describe the algorithms used by PROC TRANSREG for optimal scaling. The first section discusses optimal scaling for OPSCORE, MONOTONE, UNTIE, and LINEAR transformations, including how missing values are handled. The second section addresses SPLINE and MSPLINE transformations.

OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations

Two vectors of information are needed to produce the optimally scaled variable: the initial variable scaling vector $\mathbf{x}$ and the target vector $\mathbf{y}$. For convenience, both vectors are first sorted on the values of the initial scaling vector. If you request an UNTIE transformation, the target vector is sorted within ties in the initial scaling vector. The normal SAS System collating sequence for missing and nonmissing values is used. Sorting simply allows constraints to be specified in terms of relations among adjoining coefficients. The sorting process partitions $\mathbf{x}$ and $\mathbf{y}$ into missing and nonmissing parts $(\mathbf{x}_m'\mathbf{x}_n')'$, and $(\mathbf{y}_m'\mathbf{y}_n')'$.

Next, PROC TRANSREG determines category membership. Every ordinary missing value (.) forms a separate category. (Three ordinary missing values form three categories.) Every special missing value within the range specified in the UNTIE= *a-option* forms a separate category. (If UNTIE= BC and there are three .B and two .C missing values, five categories are formed from them.) For all other special missing values, a separate category is formed for each different value. (If there are four .A missing values, one category is formed from them.)

Each distinct nonmissing value forms a separate category for OPSCORE and MONOTONE transformations (1 1 1 2 2 3 form three categories). Each nonmissing datum forms a separate category for all other transformations (1 1 1 2 2 3 form six categories). Once category membership is determined, category means are computed. Here is an example:

x:	(.	.	.A	.A	.B	1	1	1	2	2	3	3	3	4)′
y:	(5	6	2	4	2	1	2	3	4	6	4	5	6	7)′
OPSCORE and MONOTONE means:	(5	6	3		2	2			5		5			7)′
other means:	(5	6	3		2	1	2	3	4	6	4	5	6	7)′

The category means are the coefficients of a category indicator design matrix. The category means are the Fisher (1938) optimal scores. For MONOTONE and UNTIE transformations, order constraints are imposed on the category means for the nonmissing partition by merging categories that are out of order. The algorithm checks upward until an order violation is found, then averages downward until the order violation is averaged away. (The average of $\bar{x}_1$ computed from n_1 observations and $\bar{x}_2$ computed from n_2 observations is $(n_1\bar{x}_1 + n_2\bar{x}_2)/(n_1 + n_2)$.) The MONOTONE algorithm (Kruskal 1964, secondary approach to ties) for this example with means for the nonmissing values $(2\ 5\ 5\ 7)'$ would do the following checks: $2 < 5$:OK, $5 = 5$:OK, $5 < 7$:OK. The means are in the proper order, so no work is needed.

The UNTIE transformation (Kruskal 1964, primary approach to ties) uses the same algorithm on the means of the nonmissing values $(1\ 2\ 3\ 4\ 6\ 4\ 5\ 6\ 7)'$ but with different results for this example: $1 < 2$:OK, $2 < 3$:OK, $3 < 4$:OK, $4 < 6$:OK, $6 > 4$:average

6 and 4 and replace 6 and 4 by the average. The new means of the nonmissing values are $(1\ 2\ 3\ 4\ 5\ 5\ 5\ 6\ 7)'$. The check resumes: $4 < 5$:OK, $5 = 5$:OK, $5 = 5$:OK, $5 < 6$:OK, $6 < 7$:OK. If some of the special missing values are ordered, the upward checking, downward averaging method is applied to them also, independently of the other missing and nonmissing partitions. Once the means conform to any required category or order constraints, an optimally scaled vector is produced from the means. The following example results from a MONOTONE transformation.

```
     x:    (. . .A .A .B 1 1 1 2 2 3 3 3 4)'
     y:    (5 6  2  4  2 1 2 3 4 6 4 5 6 7)'
result:    (5 6  3  3  2 2 2 2 5 5 5 5 5 7)'
```

The upward checking, downward averaging algorithm is equivalent to creating a category indicator design matrix, solving for least-squares coefficients with order constraints, then computing the linear combination of design matrix columns.

For the optimal transformation LINEAR and for nonoptimal transformations, missing values are handled as just described. The nonmissing target values are regressed onto the matrix defined by the nonmissing initial scaling values and an intercept. In this example, the target vector $y_n = (1\ 2\ 3\ 4\ 6\ 4\ 5\ 6\ 7)'$ is regressed onto the design matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}'$$

Although only a linear transformation is performed, the effect of a linear regression optimal scaling is not eliminated by the later standardization step (unless the variable has no missing values). In the presence of missing values, the linear regression is necessary to minimize squared error.

SPLINE and MSPLINE Transformations

The missing portions of variables subjected to SPLINE or MSPLINE transformations are handled the same way as for OPSCORE, MONOTONE, UNTIE, and LINEAR transformations (see the previous section). The nonmissing partition is handled by first creating a B-spline basis of the specified degree with the specified knots for the nonmissing partition of the initial scaling vector and then regressing the target onto the basis. The optimally scaled vector is a linear combination of the B-spline basis vectors using least-squares regression coefficients. An algorithm for generating the B-spline basis is given in de Boor (1978, pp. 134–135). B-splines are both a computationally accurate and efficient way of constructing a basis for piecewise polynomials; however, they are not the most natural method of describing splines.

Consider an initial scaling vector $x = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)'$ and a degree three spline with interior knots at 3.5 and 6.5. The B-spline basis for the transformation is the left matrix in Table 75.5, and the natural piecewise polynomial spline basis is the right matrix. The two matrices span the same column space. The natural basis has an intercept, a linear term, a quadratic term, a cubic term, and two more terms since there are two interior knots. These terms are generated (for knot k and $\mathbf{x}$ element x) by the formula $(x - k)^3 \times I_{(x>k)}$. The indicator variable $I_{(x>k)}$ evaluates to 1.0 if x is greater than k and to 0.0 otherwise. If knot k had been repeated, there would be a $(x - k)^2 \times I_{(x>k)}$ term also. Notice that the fifth column makes no contribution to the curve before 3.5, makes zero contribution at 3.5 (the transformation is continuous), and makes an increasing contribution beyond 3.5. The same pattern of results holds for the last term with knot 6.5. The coefficient of the fifth column represents the change in the cubic portion of the curve after 3.5. The coefficient of the sixth column represents the change in the cubic portion of the curve after 6.5.

Table 75.5. Spline Bases

$$\begin{array}{cc}
\text{B-Spline Basis} & \begin{array}{c}\text{Piecewise}\\ \text{Polynomial Splines}\end{array} \\
\begin{bmatrix}
1.000 & 0.000 & 0.000 & 0.000 & 0 & 0 \\
0.216 & 0.608 & 0.167 & 0.009 & 0 & 0 \\
0.008 & 0.458 & 0.461 & 0.073 & 0 & 0 \\
0 & 0.172 & 0.585 & 0.241 & 0.001 & 0 \\
0 & 0.037 & 0.463 & 0.463 & 0.037 & 0 \\
0 & 0.001 & 0.241 & 0.585 & 0.172 & 0 \\
0 & 0 & 0.073 & 0.461 & 0.458 & 0.008 \\
0 & 0 & 0.009 & 0.167 & 0.608 & 0.216 \\
0 & 0 & 0.000 & 0.000 & 0.000 & 1.000
\end{bmatrix}
&
\begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 \\
1 & 2 & 4 & 8 & 0 & 0 \\
1 & 3 & 9 & 27 & 0 & 0 \\
1 & 4 & 16 & 64 & 0.125 & 0 \\
1 & 5 & 25 & 125 & 3.375 & 0 \\
1 & 6 & 36 & 216 & 15.625 & 0 \\
1 & 7 & 49 & 343 & 42.875 & 0.125 \\
1 & 8 & 64 & 512 & 91.125 & 3.375 \\
1 & 9 & 81 & 729 & 166.375 & 15.625
\end{bmatrix}
\end{array}$$

The numbers in the B-spline basis do not have a simple interpretation like the numbers in the natural piecewise polynomial basis. The B-spline basis has a diagonally banded structure. The band shifts one column to the right after every knot. The number of entries in each row that may potentially be nonzero is one greater than the degree. The elements within a row always sum to one. The B-spline basis is accurate because of the smallness of the numbers and the lack of extreme collinearity inherent in the natural polynomials. B-splines are efficient because PROC TRANSREG can take advantage of the sparseness of the B-spline basis when it accumulates crossproducts. The number of required multiplications and additions to accumulate the crossproduct matrix does not increase with the number of knots but does increase with the degree of the spline, so it is much more computationally efficient to increase the number of knots than to increase the degree of the polynomial.

MSPLINE transformations are handled like SPLINE transformations except that constraints are placed on the coefficients to ensure monotonicity. When the coefficients of the B-spline basis are monotonically increasing, the transformation is monotonically increasing. When the polynomial degree is two or less, monotone coefficient splines, integrated splines (Winsberg and Ramsay 1980), and the general class of all monotone splines are equivalent.

Specifying the Number of Knots

Keep the number of knots small (usually less than ten, although you can specify more). A degree three spline with nine knots, one at each decile, can closely follow a large variety of curves. Each spline transformation of degree p with q knots fits a model with $p + q$ parameters. The total number of parameters should be much less than the number of observations. Usually in regression analyses, it is recommended that there be at least five or ten observations for each parameter in order to get stable results. For example, when spline transformations of degree three with nine knots are requested for six variables, the number of observations in the data set should be at least five or ten times 72 (since $6 \times (3 + 9)$ is the total number of parameters). The overall model can also have a parameter for the intercept and one or more parameters for each nonspline variable in the model.

Increasing the number of knots gives the spline more freedom to bend and follow the data. Increasing the degree also gives the spline more freedom, but to a lesser extent. Specifying a large number of knots is much better than increasing the degree beyond three.

When you specify NKNOTS=q for a variable with n observations, then each of the $q + 1$ segments of the spline contains $n/(q + 1)$ observations on the average. When you specify KNOTS=number-list, make sure that there is a reasonable number of observations in each interval.

The following statements find a cubic polynomial transformation of X and no transformation of Y:

```
proc transreg;
   model identity(Y)=spline(X);
   output;
run;
```

The following statements find a cubic spline transformation curve for X that consists of the weighted sum of a single constant, a single straight line, a quadratic curve for the portion of the variable less than 3.0, a different quadratic curve for the portion greater than 3.0 (since the 3.0 knot is repeated), and a different cubic curve for each of the intervals: (minimum to 1.5), (1.5 to 2.4), (2.4 to 3.0), (3.0 to 4.0), and (4.0 to maximum). The transformation is continuous everywhere, its first derivative is continuous everywhere, its second derivative is continuous everywhere except at 3.0, and its third derivative is continuous everywhere except at 1.5, 2.4, 3.0, and 4.0.

```
proc transreg;
   model identity(Y)=spline(X / knots=1.5 2.4 3.0 3.0 4.0);
   output;
run;
```

The following statements find a quadratic spline transformation that consists of a polynomial $\text{X\_t} = b_0 + b_1\text{X} + b_2\text{X}^2$ for the range ($\text{X} < 3.0$) and a completely different polynomial $\text{X\_t} = b_3 + b_4\text{X} + b_5\text{X}^2$ for the range ($\text{X} > 3.0$). The two curves are not required to be continuous at 3.0.

```
proc transreg;
   model identity(y)=spline(x / knots=3 3 3 degree=2);
   output;
run;
```

The following statements categorize Y into 10 intervals and find a step-function transformation. One aspect of this transformation family is unlike all other optimal transformation families. The initial scaling of the data does not fit the restrictions imposed by the transformation family. This is because the initial variable can be continuous, but a discrete step function transformation is sought. Zero degree spline variables are categorized before the first iteration.

```
proc transreg;
   model identity(Y)=spline(X / degree=0 nknots=9);
   output;
run;
```

The following statements find a continuous, piecewise linear transformation of X:

```
proc transreg;
   model identity(Y)=spline(X / degree=1 nknots=8);
   output;
run;
```

SPLINE, BSPLINE, and PSPLINE Comparisons

SPLINE is a transformation. It takes a variable as input and produces a transformed variable as output. Internally, with SPLINE, a B-spline basis is used to find the transformation, which is a linear combination of the columns of the B-spline basis. However, with SPLINE, the basis is not made available in any output.

BSPLINE is an expansion. It takes a variable as input and produces more than one variable as output. The output variables comprise the B-spline basis that is used internally by SPLINE.

PSPLINE is an expansion. It takes a variable as input and produces more than one variable as output. The difference between PSPLINE and BSPLINE is that PSPLINE produces a piecewise polynomial, whereas BSPLINE produces a B-spline. A matrix consisting of a piecewise polynomial basis and an intercept spans the same space as the B-spline matrix, but the basis vectors are quite different. The numbers in the piecewise polynomials can get quite large; the numbers in the B-spline basis range between 0 and 1. There are many more zeros in the B-spline basis.

Interchanging SPLINE, BSPLINE, and PSPLINE should have no effect on the fit of the overall model except for the fact that PSPLINE is much more prone to numerical problems. Similarly, interchanging a CLASS expansion and an OPSCORE transformation should have no effect on the fit of the overall model.

Hypothesis Tests

The TRANSREG procedure has a set of options for testing hypotheses in models with a single dependent variable. The TEST *a-option* produces an ANOVA table. It tests the null hypothesis that the vector of coefficients for all of the transformations is zero. The SS2 *a-option* produces a regression table with Type II tests of the contribution of each transformation to the overall model. In some cases, exact tests are provided; in other cases, the tests are approximate, liberal, or conservative.

For two reasons it is typically not appropriate to test hypotheses by using the output from PROC TRANSREG as input to other procedures such as the REG procedure. First, PROC REG has no way of determining how many degrees of freedom were used for each transformation. Second, the Type II sums of squares for the tests of the individual regression coefficients are not correct for the transformation regression model since PROC REG, as it evaluates the effect of each variable, cannot change the transformations of the other variables. PROC TRANSREG uses the correct degrees of freedom and sums of squares.

In an ordinary univariate linear model, there is one parameter for each independent variable, including the intercept. In the transformation regression model, many of the “variables” are used internally in the bases for the transformations. Each basis column has one parameter or *scoring* coefficient, and each linearly independent column has one model degree of freedom associated with it. Coefficients applied to transformed variables, *model coefficients*, do not enter into the degrees of freedom calculations. They are by-products of the standardizations and can be absorbed into the transformations by specifying the ADDITIVE *a-option*. The word *parameter* is reserved for model and scoring coefficients that have a degree of freedom associated with them.

For expansions, there is one model parameter for each variable created by the expansion (except for all missing CLASS columns and expansions that have an implicit intercept). Each IDENTITY variable has one model parameter. If there are m POINT variables, they expand to $m + 1$ variables and, hence, have $m + 1$ model parameters. For m EPOINT variables, there are $2m$ model parameters. For m QPOINT variables, there are $m(m + 3)/2$ model parameters. If a variable with m categories is designated CLASS, there are $m - 1$ parameters. For BSPLINE and PSPLINE variables of DEGREE=n with NKNOTS=k, there are $n + k$ parameters. Note that one of the $n + k + 1$ BSPLINE columns and one of the m CLASS(variable / ZERO=NONE) columns are not counted due to the implicit intercept.

There are scoring parameters for missing values in nonexcluded observations. Each ordinary missing value (.) has one scoring parameter. Each different special missing value (._ and .A through .Z) within each variable has one scoring parameter. Missing values specified in the UNTIE= and MONOTONE= options follow the rules for UNTIE and MONOTONE transformations, which are described later in this chapter.

For all nonoptimal transformations (LOG, LOGIT, ARSIN, POWER, EXP, RANK, BOXCOX), there is one parameter per variable in addition to any missing value scoring parameters.

For SPLINE, OPSCORE, and LINEAR transformations, the number of scoring parameters is the number of basis columns that are used internally to find the transformations minus 1 for the intercept. The number of scoring parameters for SPLINE variables is the same as the number of model parameters for BSPLINE and PSPLINE variables. If DEGREE=n and NKNOTS=k, there are $n + k$ scoring parameters. The number of scoring parameters for OPSCORE, SMOOTH, and SSPLINE variables is the same as the number of model parameters for CLASS variables. If there are m categories, there are $m - 1$ scoring parameters. There is one parameter for each LINEAR variable. For SPLINE, OPSCORE, LINEAR, MONOTONE, UNTIE, and MSPLINE transformations, missing value scoring parameters are computed as described previously with the nonoptimal transformations.

The number of scoring parameters for MONOTONE, UNTIE, and MSPLINE transformations is less precise than for SPLINE, OPSCORE, and LINEAR transformations. One way of handling a MONOTONE transformation is to treat it as if it were the same as an OPSCORE transformation. If there are m categories, there are $m - 1$ potential scoring parameters. However, there are typically fewer than $m - 1$ unique parameter estimates since some of those $m - 1$ scoring parameter estimates may be tied during the optimal scaling to impose the order constraints. Imposing ties on the scoring parameter estimates is equivalent to fitting a model with fewer parameters. So there are two available scoring parameter counts: $m - 1$ and a smaller number that is determined during the analysis. Using $m - 1$ as the model degrees of freedom for MONOTONE variables (treating OPSCORE and MONOTONE transformations the same way) is *conservative*, since the MONOTONE scoring parameter estimates are more restricted than the OPSCORE scoring parameter estimates. Using the smaller count (the number of scoring parameter estimates that are different minus 1 for the intercept) in the model degrees of freedom is *liberal*, since the data and the model together are being used to determine the number of parameters. PROC TRANSREG reports tests using both liberal and conservative degrees of freedom to provide lower and upper bounds on the “true” *p*-values.

For the UNTIE transformation, the conservative scoring parameter count is the number of distinct observations, whereas the liberal scoring parameter count is the number of scoring parameter estimates that are different minus 1 for the intercept. Hence, when you specify UNTIE, conservative tests have zero error degrees of freedom unless there are replicated observations.

For MSPLINE variables of DEGREE=n and NKNOTS=k, the conservative scoring parameter count is $n+k$, whereas the liberal parameter count is the number of scoring parameter estimates that are different, minus 1 for the intercept. A liberal degrees of freedom of 1 does not necessarily imply a linear transformation. It just implies that n plus k minus the number of ties imposed equals 1. An example of a one degree-of-freedom nonlinear transformation is a two-piece linear transformation in which the slope of one piece is 0.

The number of scoring parameters is determined during each iteration. After the last iteration, enough information is available for the TEST *a-option* to produce an ANOVA table that reports the overall fit of the model. If you specify the SS2 *a-option*, further iterations are necessary to test the contribution of each transformation to the overall model.

The liberal tests do not compensate for over-parameterization. For example, requesting a spline transformation with k knots when a linear transformation will suffice results in “liberal” tests that are actually conservative because too many degrees of freedom are being used for the transformations. Use as few knots as possible to avoid this problem.

In ordinary multiple regression, an *F* test of the null hypothesis that the coefficient for variable x_j is zero can be constructed by comparing two linear models. One model is the full model with all parameters, and the other is a reduced model that has all parameters except the parameter for variable x_j. The difference between the model sum of squares for the full model and the model sum of squares for the reduced model is the Type II sum of squares for the test of the null hypothesis that the coefficient for variable x_j is 0. The numerator of the *F* test has one degree of freedom. The mean square error for the full model is the denominator of the *F* test of variable x_j. Note that the estimates of the coefficients for the two models are not usually the same. When variable x_j is removed, the coefficients for the other variables change to compensate for the removal of x_j. In a transformation regression model, the transformations of the other variables must be allowed to change and the numerator degrees of freedom are not always ones. It is not correct to simply let the model coefficients for the transformed variables change and apply the new model coefficients to the old transformations computed with the old scoring parameter estimates. In a transformation regression model, further iteration is needed to test each transformation because all the scoring parameter estimates for other variables must be allowed to change to test the effect of variable x_j. This can be quite time consuming for a large model if the DUMMY *a-option* cannot be used to solve directly for the transformations.

Output Data Set

The OUT= output data set can contain a great deal of information; however, in most cases, the output data set contains a small portion of the entire range of available information and is organized for direct input into the %PLOTIT macro or graphical or analysis procedures. For information on the %PLOTIT macro, see Appendix B, “Using the %PLOTIT Macro.”

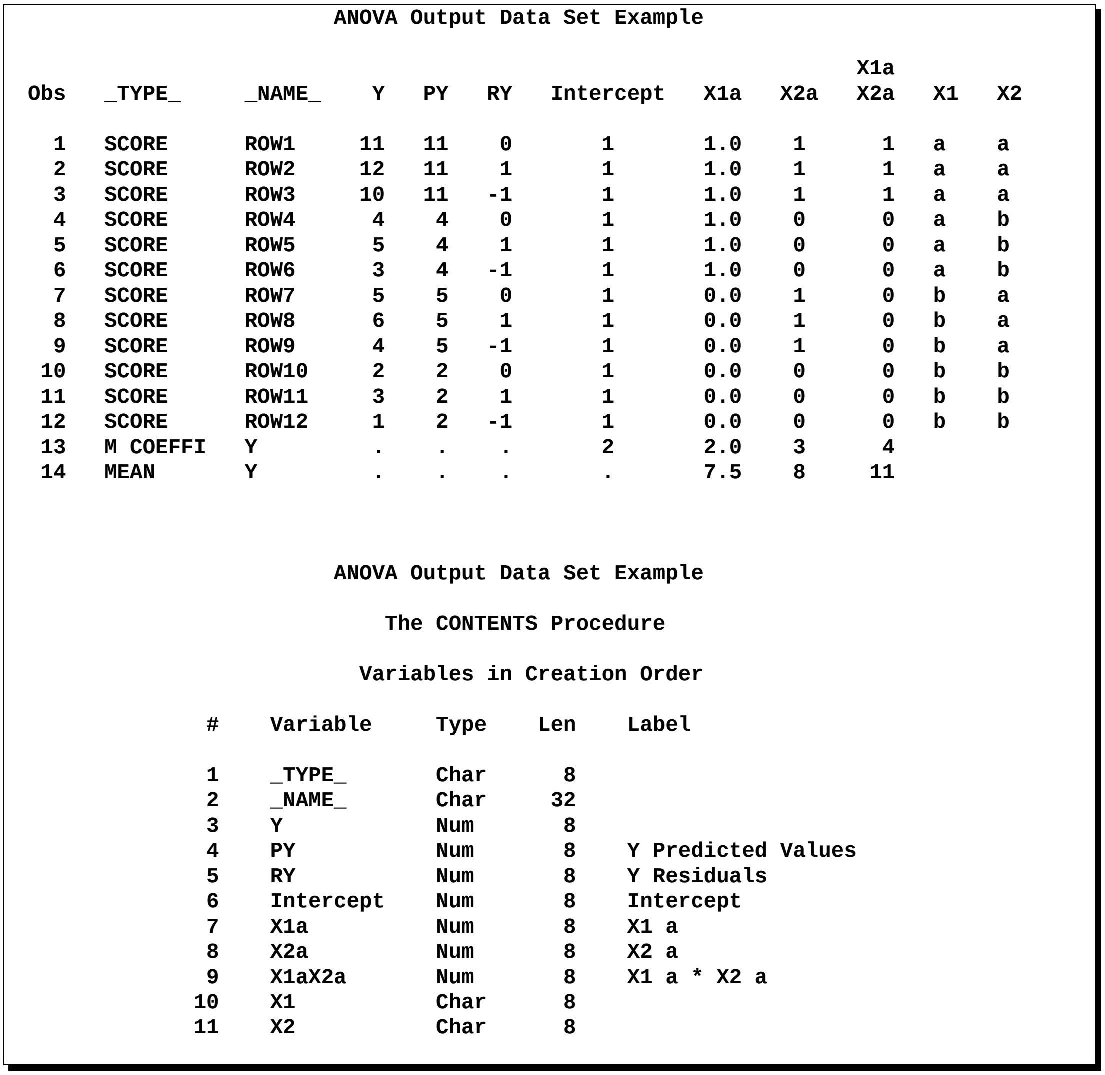
Output Data Set Examples

The next section provides a complete list of the contents of the OUT= data set. However, before presenting complete details, this section provides three brief examples, illustrating some typical output data sets.

The first example shows the output data set from a two-way ANOVA model. The following statements produce Figure 75.12:

```
title 'ANOVA Output Data Set Example';

data ReferenceCell;
   input Y X1 $ X2 $;
   datalines;
```

```
11  a  a
12  a  a
10  a  a
 4  a  b
 5  a  b
 3  a  b
 5  b  a
 6  b  a
 4  b  a
 2  b  b
 3  b  b
 1  b  b
;

*---Fit Reference Cell Two-Way ANOVA Model---;
proc transreg data=ReferenceCell;
   model identity(Y) = class(X1 | X2);
   output coefficients replace predicted residuals;
run;

*---Print the Results---;
proc print;
run;

proc contents position;
   ods select position;
run;
```



```
                        ANOVA Output Data Set Example

                                                                 X1a
Obs   _TYPE_     _NAME_    Y   PY   RY   Intercept   X1a   X2a   X2a   X1   X2

  1   SCORE      ROW1     11   11    0       1       1.0    1      1   a    a
  2   SCORE      ROW2     12   11    1       1       1.0    1      1   a    a
  3   SCORE      ROW3     10   11   -1       1       1.0    1      1   a    a
  4   SCORE      ROW4      4    4    0       1       1.0    0      0   a    b
  5   SCORE      ROW5      5    4    1       1       1.0    0      0   a    b
  6   SCORE      ROW6      3    4   -1       1       1.0    0      0   a    b
  7   SCORE      ROW7      5    5    0       1       0.0    1      0   b    a
  8   SCORE      ROW8      6    5    1       1       0.0    1      0   b    a
  9   SCORE      ROW9      4    5   -1       1       0.0    1      0   b    a
 10   SCORE      ROW10     2    2    0       1       0.0    0      0   b    b
 11   SCORE      ROW11     3    2    1       1       0.0    0      0   b    b
 12   SCORE      ROW12     1    2   -1       1       0.0    0      0   b    b
 13   M COEFFI   Y         .    .    .       2       2.0    3      4
 14   MEAN       Y         .    .    .       .       7.5    8     11



                        ANOVA Output Data Set Example

                            The CONTENTS Procedure

                          Variables in Creation Order

              #    Variable     Type    Len    Label

              1    _TYPE_       Char      8
              2    _NAME_       Char     32
              3    Y            Num       8
              4    PY           Num       8    Y Predicted Values
              5    RY           Num       8    Y Residuals
              6    Intercept    Num       8    Intercept
              7    X1a          Num       8    X1 a
              8    X2a          Num       8    X2 a
              9    X1aX2a       Num       8    X1 a * X2 a
             10    X1           Char      8
             11    X2           Char      8
```

Figure 75.12. ANOVA Example Output Data Set Contents

The _TYPE_ variable indicates observation type: score, multiple regression coefficient (parameter estimates), and marginal means. The _NAME_ variable contains the default observation labels, "ROW1", "ROW2", and so on, and contains the dependent variable name (Y) for the remaining observations. If you specify an ID statement, _NAME_ contains the values of the first ID variable for score observations. The Y variable is the dependent variable, PY contains the predicted values, RY contains the residuals, and the variables Intercept through X1aX2a contain the design matrix. The X1 and X2 variables are the original CLASS variables.

The next example shows the contents of the output data set from fitting a curve through a scatter plot.

```
title 'Output Data Set for Curve Fitting Example';

data A;
```

```
      do X = 1 to 100;
         Y = log(x) + sin(x / 10) + normal(7);
         output;
         end;
   run;

   proc transreg;
      model identity(Y) = spline(X / nknots=9);
      output predicted out=B;
   run;

   proc contents position;
      ods select position;
   run;
```

These statements produce Figure 75.13.

```
                  Output Data Set for Curve Fitting Example

                            The CONTENTS Procedure

                          Variables in Creation Order

            #    Variable      Type    Len    Label

            1    _TYPE_        Char      8
            2    _NAME_        Char     32
            3    Y             Num       8
            4    TY            Num       8    Y Transformation
            5    PY            Num       8    Y Predicted Values
            6    Intercept     Num       8    Intercept
            7    X             Num       8
            8    TIntercept    Num       8    Intercept Transformation
            9    TX            Num       8    X Transformation
```

Figure 75.13. Predicted Values Example Output Data Set Contents

The OUT= data set contains _TYPE_ and _NAME_ variables. Since no coefficients or coordinates are requested, all observations are _TYPE_='SCORE'. The Y variable is the original dependent variable, TY is the transformed dependent variable, PY contains the predicted values, X is the original independent variable, and TX is the transformed independent variable. The data set also contains an Intercept and transformed intercept TIntercept variable. (In this case, the transformed intercept is the same as the intercept. However, if you specify the TSTANDARD= and ADDITIVE options, these are not always the same.)

The next example shows the results from specifying METHOD=MORALS when there is more than one dependent variable.

```
title ’METHOD=MORALS Output Data Set Example’;

data x;
   input Y1 Y2 X1 $ X2 $;
   datalines;
11 1 a a
10 4 b a
 5 2 a b
 5 9 b b
 4 3 c c
 3 6 b a
 1 8 a b
;

*---Fit Reference Cell Two-Way ANOVA Model---;
proc transreg data=x noprint dummy;
   model spline(Y1 Y2) = opscore(X1 X2 / name=(N1 N2));
   output coefficients predicted residuals;
   id x1 x2;
run;

*---Print the Results---;
proc print;
run;

proc contents position;
   ods select position;
run;
```

These statements produce Figure 75.14.

```
                      METHOD=MORALS Output Data Set Example

Obs   _DEPVAR_    _TYPE_    _NAME_  _DEPEND_  T_DEPEND_  P_DEPEND_  R_DEPEND_

  1  Spline(Y1)  SCORE       a        11       13.1600    11.1554    2.00464
  2  Spline(Y1)  SCORE       b        10        6.1931     6.8835   -0.69041
  3  Spline(Y1)  SCORE       a         5        2.4467     4.7140   -2.26724
  4  Spline(Y1)  SCORE       b         5        2.4467     0.4421    2.00464
  5  Spline(Y1)  SCORE       c         4        4.2076     4.2076    0.00000
  6  Spline(Y1)  SCORE       b         3        5.5693     6.8835   -1.31422
  7  Spline(Y1)  SCORE       a         1        4.9766     4.7140    0.26261
  8  Spline(Y1)  M COEFFI    Y1        .         .          .          .
  9  Spline(Y2)  SCORE       a         1       -0.5303    -0.5199   -0.01043
 10  Spline(Y2)  SCORE       b         4        5.5487     4.5689    0.97988
 11  Spline(Y2)  SCORE       a         2        3.8940     4.5575   -0.66347
 12  Spline(Y2)  SCORE       b         9        9.6358     9.6462   -0.01043
 13  Spline(Y2)  SCORE       c         3        5.6210     5.6210    0.00000
 14  Spline(Y2)  SCORE       b         6        3.5994     4.5689   -0.96945
 15  Spline(Y2)  SCORE       a         8        5.2314     4.5575    0.67390
 16  Spline(Y2)  M COEFFI    Y2        .         .          .          .

Obs  Intercept    N1    N2    TIntercept       TN1         TN2      X1    X2

  1      1         0     0      1.0000      0.06711    -0.09384    a     a
  2      1         1     0      1.0000      1.51978    -0.09384    b     a
  3      1         0     1      1.0000      0.06711     1.32038    a     b
  4      1         1     1      1.0000      1.51978     1.32038    b     b
  5      1         2     2      1.0000      0.23932     1.32038    c     c
  6      1         1     0      1.0000      1.51978    -0.09384    b     a
  7      1         0     1      1.0000      0.06711     1.32038    a     b
  8      .         .     .     10.9253     -2.94071    -4.55475    Y1    Y1
  9      1         0     0      1.0000      0.03739    -0.09384    a     a
 10      1         1     0      1.0000      1.51395    -0.09384    b     a
 11      1         0     1      1.0000      0.03739     1.32038    a     b
 12      1         1     1      1.0000      1.51395     1.32038    b     b
 13      1         2     2      1.0000      0.34598     1.32038    c     c
 14      1         1     0      1.0000      1.51395    -0.09384    b     a
 15      1         0     1      1.0000      0.03739     1.32038    a     b
 16      .         .     .     -0.3119      3.44636     3.59024    Y2    Y2
```

Figure 75.14. METHOD=MORALS Rolled Output Data Set

```
                          The CONTENTS Procedure

                        Variables in Creation Order

 #    Variable      Type    Len    Label

 1    _DEPVAR_      Char     42    Dependent Variable Transformation(Name)
 2    _TYPE_        Char      8
 3    _NAME_        Char     32
 4    _DEPEND_      Num       8    Dependent Variable
 5    T_DEPEND_     Num       8    Dependent Variable Transformation
 6    P_DEPEND_     Num       8    Dependent Variable Predicted Values
 7    R_DEPEND_     Num       8    Dependent Variable Residuals
 8    Intercept     Num       8    Intercept
 9    N1            Num       8
10    N2            Num       8
11    TIntercept    Num       8    Intercept Transformation
12    TN1           Num       8    N1 Transformation
13    TN2           Num       8    N2 Transformation
14    X1            Char      8
15    X2            Char      8
```

Figure 75.14. (continued)

If you specify METHOD=MORALS with multiple dependent variables, PROC TRANSREG performs separate univariate analyses and stacks the results in the OUT= data set. For this example, the results of the first analysis are in the partition designated by _DEPVAR_='Spline(Y1)' and the results of the first analysis are in the partition designated by _DEPVAR_='Spline(Y2)', which are the transformation and dependent variable names. Each partition has _TYPE_='SCORE' observations for the variables and a _TYPE_='M COEFFI' observation for the coefficients. In this example, an ID variable is specified, so the _NAME_ variable contains the formatted values of the first ID variable. Since both dependent variables have to go into the same column, the dependent variable is given a new name, _DEPEND_. The dependent variable transformation is named T_DEPEND_, the predicted values variable is named P_DEPEND_, and the residuals variable is named R_DEPEND_.

The independent variables are character OPSCORE variables. By default, PROC TRANSREG replaces character OPSCORE variables with category numbers and discards the original character variables. To avoid this, the input variables are renamed from X1 and X2 to N1 and N2 and the original X1 and X2 are added to the data set as ID variables. The N1 and N2 variables contain the initial values for the OPSCORE transformations, and the TN1 and TN2 variables contain optimal scores. The data set also contains an Intercept and transformed intercept TIntercept variable. The regression coefficients are in the transformation columns, which also contain the variables to which they apply.

Output Data Set Contents

This section presents the various matrices that can result from PROC TRANSREG processing and that appear in the OUT= data set. The exact contents of an OUT= data set depends on many options.

Table 75.6. PROC TRANSREG OUT= Data Set Contents

TYPE	Contents	Options, Default Prefix
SCORE	dependent variables	DREPLACE not specified
SCORE	independent variables	IREPLACE not specified
SCORE	transformed dependent variables	default, TDPREFIX=T
SCORE	transformed independent variables	default, TIPREFIX=T
SCORE	predicted values	PREDICTED, PPREFIX=P
SCORE	residuals	RESIDUALS, RDPREFIX=R
SCORE	leverage	LEVERAGE, LEVERAGE=Leverage
SCORE	lower individual confidence limits	CLI, LILPREFIX=LIL, CILPREFIX=CIL
SCORE	upper individual confidence limits	CLI, LIUPREFIX=LIU, CIUPREFIX=CIU
SCORE	lower mean confidence limits	CLM, LMLPREFIX=LML, CMLPREFIX=CML
SCORE	upper mean confidence limits	CLM, LMUPREFIX=LMU, CMUPREFIX=CMU
SCORE	dependent canonical variables	CANONICAL, CDPREFIX=Cand
SCORE	independent canonical variables	CANONICAL, CIPREFIX=Cani
SCORE	redundancy variables	REDUNDANCY, RPREFIX=Red
SCORE	ID, CLASS, BSPLINE variables	ID, CLASS, BSPLINE,
SCORE	independent variables approximations	IAPPROXIMATIONS, IAPREFIX=A
M COEFFI	multiple regression coefficients	COEFFICIENTS, MRC
C COEFFI	canonical coefficients	COEFFICIENTS, CCC
MEAN	marginal means	COEFFICIENTS, MEANS
M REDUND	multiple redundancy coefficients	MREDUNDANCY
R REDUND	multiple redundancy coefficients	MREDUNDANCY
M POINT	point coordinates	COORDINATES or MPC, POINT
M EPOINT	elliptical point coordinates	COORDINATES or MEC, EPOINT
M QPOINT	quadratic point coordinates	COORDINATES or MQC, QPOINT
C POINT	canonical point coordinates	COORDINATES or CPC, POINT
C EPOINT	canonical elliptical point coordinates	COORDINATES or CEC, EPOINT
C QPOINT	canonical quadratic point coordinates	COORDINATES or CQC, QPOINT

The independent and dependent variables are created from the original input data. Several potential differences exist between these variables and the actual input data. An intercept variable can be added, new variables can be added for POINT, EPOINT, QPOINT, CLASS, IDENTITY, PSPLINE, and BSPLINE variables, and category numbers are substituted for character OPSCORE variables. These matrices are not always what is input to the first iteration. After the expanded data set is stored for inclusion in the output data set, several things happen to the data before they are input to the first iteration: column means are substituted for missing values; zero degree SPLINE and MSPLINE variables are transformed so that the iterative algorithms get step function data as input, which conform to the zero degree transformation family restrictions; and the nonoptimal transformations are performed.

Details for the UNIVARIATE Method

When you specify METHOD=UNIVARIATE (in the MODEL or PROC TRANSREG statement), PROC TRANSREG can perform several analyses, one for each dependent variable. While each dependent variable can be transformed, their independent variables are not transformed. The OUT= data set optionally contains all of the _TYPE_='SCORE' observations, optionally followed by coefficients or coordinates.

Details for the MORALS Method

When you specify METHOD=MORALS (in the MODEL or PROC TRANSREG statement), successive analyses are performed, one for each dependent variable. Each analysis transforms one dependent variable and the entire set of the independent variables. All information for the first dependent variable (scores then, optionally, coefficients) appear first. Then all information for the second dependent variable (scores then, optionally, coefficients) appear next. This arrangement is repeated for all dependent variables.

Details for the CANALS and REDUNDANCY Methods

For METHOD=CANALS and METHOD=REDUNDANCY (specified in either the MODEL or PROC TRANSREG statement), one analysis is performed that simultaneously transforms all dependent and independent variables. The OUT= data set optionally contains all of the _TYPE_='SCORE' observations, optionally followed by coefficients or coordinates.

Variable Names

As shown in the preceding examples, some variables in the output data set directly correspond to input variables and some are created. All original optimal and nonoptimal transformation variable names are unchanged.

The names of the POINT, QPOINT, and EPOINT expansion variables are also left unchanged, but new variables are created. When independent POINT variables are present, the sum-of-squares variable _ISSQ_ is added to the output data set. For each EPOINT and QPOINT variable, a new squared variable is created by appending "_2". For example, Dim1 and Dim2 are expanded into Dim1, Dim2, Dim1_2, and Dim2_2. In addition, for each pair of QPOINT variables, a new crossproduct variable is created by combining the two names, for example, Dim1Dim2.

The names of the CLASS variables are constructed from original variable names and levels. Lengths are controlled by the CPREFIX= *a-option*. For example, when X1 and X2 both have values of 'a' and 'b', CLASS(X1 | X2 / ZERO=NONE) creates X1 main effect variable names X1a X1b, X2 main effect variable names X2a X2b, and interaction variable names X1aX2a X1aX2b X1bX2a X1bX2b.

PROC TRANSREG then uses these variable names when creating the transformed, predicted, and residual variable names by affixing the relevant prefix and possibly dropping extra characters.

METHOD=MORALS Variable Names

When you specify METHOD=MORALS and only one dependent variable is present, the output data set is structured exactly as if METHOD=REDUNDANCY (see the section "Details for the CANALS and REDUNDANCY Methods" on page 4625). When more than one dependent variable is present, the dependent variables are output in the variable _DEPEND_, transformed dependent variables are output in the variable T_DEPEND_, predicted values are output in the variable P_DEPEND_, and residuals are output in the variable R_DEPEND_. You can partition the data set into BY groups, one per dependent variable, by referring to the character variable _DEPVAR_, which contains the original dependent variable names and transformations.

Duplicate Variable Names

When the same name is generated from multiple variables in the OUT= data set, new names are created by appending '2', '3', or '4', and so on, until a unique name is created. For 32-character names, the last character is replaced with a numeric suffix until a unique name is created. For example, if there are two output variables that otherwise would be named X, then X and X2 are created instead. If there are two output variables that otherwise would be named ThisIsAThirtyTwoCharacterVarName, then ThisIsAThirtyTwoCharacterVarName and ThisIsAThirtyTwoCharacterVarNam2 are created instead.

OUTTEST= Output Data Set

The OUTTEST= data set contains hypothesis test results. The OUTTEST= data set always contains ANOVA results. When you specify the SS2 *a-option*, regression tables are also output. When you specify the UTILITIES *a-option*, conjoint analysis part-worth utilities are also output. The OUTTEST= data set has the following variables:

DEPVAR is a 42-character variable that contains the dependent variable transformation and name.

TYPE is an 8-character variable that contains the table type. The first character is "U" for univariate or "M" for multivariate. The second character is blank. The third character is "A" for ANOVA, "2" for Type II sum of squares, or "U" for UTILITIES. The fourth character is blank. The fifth character is "L" for liberal tests, "C" for conservative tests, or "U" for the usual tests.

Title is an 80-character variable that contains the table title.

Variable is a 42-character variable that contains the independent variable transformations and names for regression tables and blanks for ANOVA tables.

Coefficient contains the multiple regression coefficients for regression tables and underscore special missing values for ANOVA tables.

Statistic is a 24-character variable that contains the names for statistics in other variables, such as Value.

Value contains multivariate test statistics and all other information that does not fit in one of the other columns including R-Square, Dependent Mean, Adj R-Sq, and Coeff Var. Whenever Value is not

an underscore special missing value, Statistic describes the contents of Value.

NumDF contains numerator degrees of freedom for *F* tests.

DenDF contains denominator degrees of freedom for *F* tests.

SSq contains sums of squares.

MeanSquare contains mean squares.

F contains *F* statistics.

NumericP contains the *p*-value for the *F* statistic, stored in a numeric variable.

P is a 9-character variable that contains the formatted *p*-value for the *F* statistic, including the appropriate ∼, <=, >=, or blank symbols.

LowerLimit contains lower confidence limits on the parameter estimates.

UpperLimit contains upper confidence limits on the parameter estimates.

StdError contains standard errors. For SS2 and UTILITIES tables, standard errors are output for each coefficient with one degree of freedom.

Importance contains the relative importance of each factor for UTILITIES tables.

Label is a 256-character variable that contains variable labels.

There are several possible tables in the OUTTEST= data set corresponding to combinations of univariate and multivariate tests; ANOVA and regression results; and liberal, conservative, and the usual tests. Each table is composed of only a subset of the variables. Numeric variables contain underscore special missing values when they are not a column in a table. Ordinary missing values (.) appear in variables that are part of a table when a nonmissing value cannot be produced. For example, the *F* is missing for a test with zero degrees of freedom.

Computational Resources

This section provides information on the computational resources required to use PROC TRANSREG.

Let

n = number of observations

q = number of expanded independent variables

r = number of expanded dependent variables

k = maximum spline degree

p = maximum number of knots

- More than $56(q+r)$ plus the maximum of the data matrix size, the optimal scaling work space, and the covariance matrix size bytes of array space are required. The data matrix size is $8n(q+r)$ bytes. The optimal scaling work space requires less than $8(6n+(p+k+2)(p+k+11))$ bytes. The covariance matrix size is $4(q+r)(q+r+1)$ bytes.
- PROC TRANSREG tries to store the original and transformed data in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time. The amount of memory for the preceding data formulas is an underestimate of the amount of memory needed to handle most problems. These formulas give the absolute minimum amount of memory required. If a utility data set is used, and if memory can be used with perfect efficiency, then roughly the amount of memory stated previously is needed. In reality, most problems require at least two or three times the minimum.
- PROC TRANSREG sorts the data once. The sort time is roughly proportional to $(q+r)n^{3/2}$.
- One regression analysis per iteration is required to compute model parameters (or two canonical correlation analyses per iteration for METHOD=CANALS). The time required for accumulating the crossproducts matrix is roughly proportional to $n(q+r)^2$. The time required to compute the regression coefficients is roughly proportional to q^3.
- Each optimal scaling is a multiple regression problem, although some transformations are handled with faster special-case algorithms. The number of regressors for the optimal scaling problems depends on the original values of the variable and the type of transformation. For each monotone spline transformation, an unknown number of multiple regressions is required to find a set of coefficients that satisfies the constraints. The B-spline basis is generated twice for each SPLINE and MSPLINE transformation for each iteration. The time required to generate the B-spline basis is roughly proportional to nk^2.

Solving Standard Least-Squares Problems

This section illustrates how to solve some ordinary least-squares problems and generalizations of those problems by formulating them as transformation regression problems. One problem involves finding linear and nonlinear regression functions in a scatter plot. The next problem involves simultaneously fitting two lines or curves through a scatter plot. The last problem involves finding the overall fit of a multi-way main-effects and interactions analysis-of-variance model.

Nonlinear Regression Functions

This example uses PROC TRANSREG in simple regression to find the optimal regression line, a nonlinear but monotone regression function, and a nonlinear nonmonotone regression function. A regression line can be found by specifying

```
proc transreg;
   model identity(y) = identity(x);
   output predicted;
run;
```

A monotone regression function (in this case, a monotonically decreasing regression function, since the correlation coefficient is negative) can be found by requesting an MSPLINE transformation of the independent variable, as follows.

```
proc transreg;
   model identity(y) = mspline(x / nknots=9);
   output predicted;
run;
```

The monotonicity restriction can be relaxed by requesting a SPLINE transformation of the independent variable, as shown next.

```
proc transreg;
   model identity(y) = spline(x / nknots=9);
   output predicted;
run;
```

In this example, it is not useful to plot the transformation TX, since TX is just an intermediate result used in finding a regression function through the original X and Y scatter plot.

The following statements provide a specific example of using the TRANSREG procedure for fitting nonlinear regression functions. These statements produce Figure 75.15 through Figure 75.18.

```
title 'Linear and Nonlinear Regression Functions';
*---Generate an Artificial Nonlinear Scatter Plot---;
*---SAS/IML Software is Required for this Example---;
proc iml;
   N   = 500;
   X   = (1:N)`;
   X   = X/(N/200);
   Y   = -((X/50)-1.5)##2 + sin(X/8) + sqrt(X)/5 + 2*log(X) + cos(X);
   X   = X - X[:,];
   X   = -X / sqrt(X[##,]/(n-1));
   Y   = Y - Y[:,];
   Y   = Y / sqrt(Y[##,]/(n-1));
   all = Y || X;
   create outset from all;
   append        from all;
   quit;

data A;
   set outset(rename=(col1=Y col2=X));
   if Y<-2 then Y=-2 + ranuni(7654321)/2;
   X1=X; X2=X; X3=X; X4=X;
run;

*---Predicted Values for the Linear Regression Line---;
proc transreg data=A;
```

```
   title2 ’A Linear Regression Line’;
   model identity(Y)=identity(X);
   output out=A pprefix=L;
   id X1-X4;
run;

*---Predicted Values for the Monotone Regression Function---;
proc transreg data=A;
   title2 ’A Monotone Regression Function’;
   model identity(Y)=mspline(X / nknots=9);
   output out=A pprefix=M;
   id X1-X4 LY;
run;

*---Predicted Values for the Nonmonotone Regression Function---;
proc transreg data=A;
   title2 ’A Nonmonotone Regression Function’;
   model identity(Y)=spline(X / nknots=9);
   output out=A predicted;
   id X1-X4 LY MY;
run;

*---Plot the Results---;
goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot data=A;
   title;
   axis1 minor=none label=(angle=90 rotate=0)
         order=(-2 to 2 by 2);
   axis2 minor=none order=(-2 to 2 by 2);
   plot Y*X1=1         /         &opts name=’tregnl1’;
   plot Y*X2=1 LY*X2=2 / overlay &opts name=’tregnl2’;
   plot Y*X3=1 MY*X3=2 / overlay &opts name=’tregnl3’;
   plot Y*X4=1 PY*X4=2 / overlay &opts name=’tregnl4’;
   symbol1 color=blue   v=star i=none;
   symbol2 color=yellow v=none i=join;
   label X1 = ’Nonlinear Scatter Plot’
         X2 = ’Linear Regression, r**2 = 0.14580’
         X3 = ’Monotone Function, r**2 = 0.60576’
         X4 = ’Nonlinear Function, r**2 = 0.89634’;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:tregnl1 2:tregnl3 3:tregnl2 4:tregnl4;
run; quit;
```

```
                    Linear and Nonlinear Regression Functions
                             A Linear Regression Line

                              The TRANSREG Procedure

          TRANSREG Univariate Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.00000    0.00000     0.14580                 Converged

   Algorithm converged.
```

Figure 75.15. A Linear Regression Line

```
                    Linear and Nonlinear Regression Functions
                          A Monotone Regression Function

                              The TRANSREG Procedure

            TRANSREG MORALS Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.62131    1.34209     0.14580
           2    0.00000    0.00000     0.60576      0.45995    Converged

   Algorithm converged.
```

Figure 75.16. A Monotone Regression Function

```
                    Linear and Nonlinear Regression Functions
                        A Nonmonotone Regression Function

                              The TRANSREG Procedure

            TRANSREG MORALS Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.83948    2.78984     0.14580
           2    0.00000    0.00000     0.89634      0.75054    Converged

   Algorithm converged.
```

Figure 75.17. A Nonmonotone Regression Function

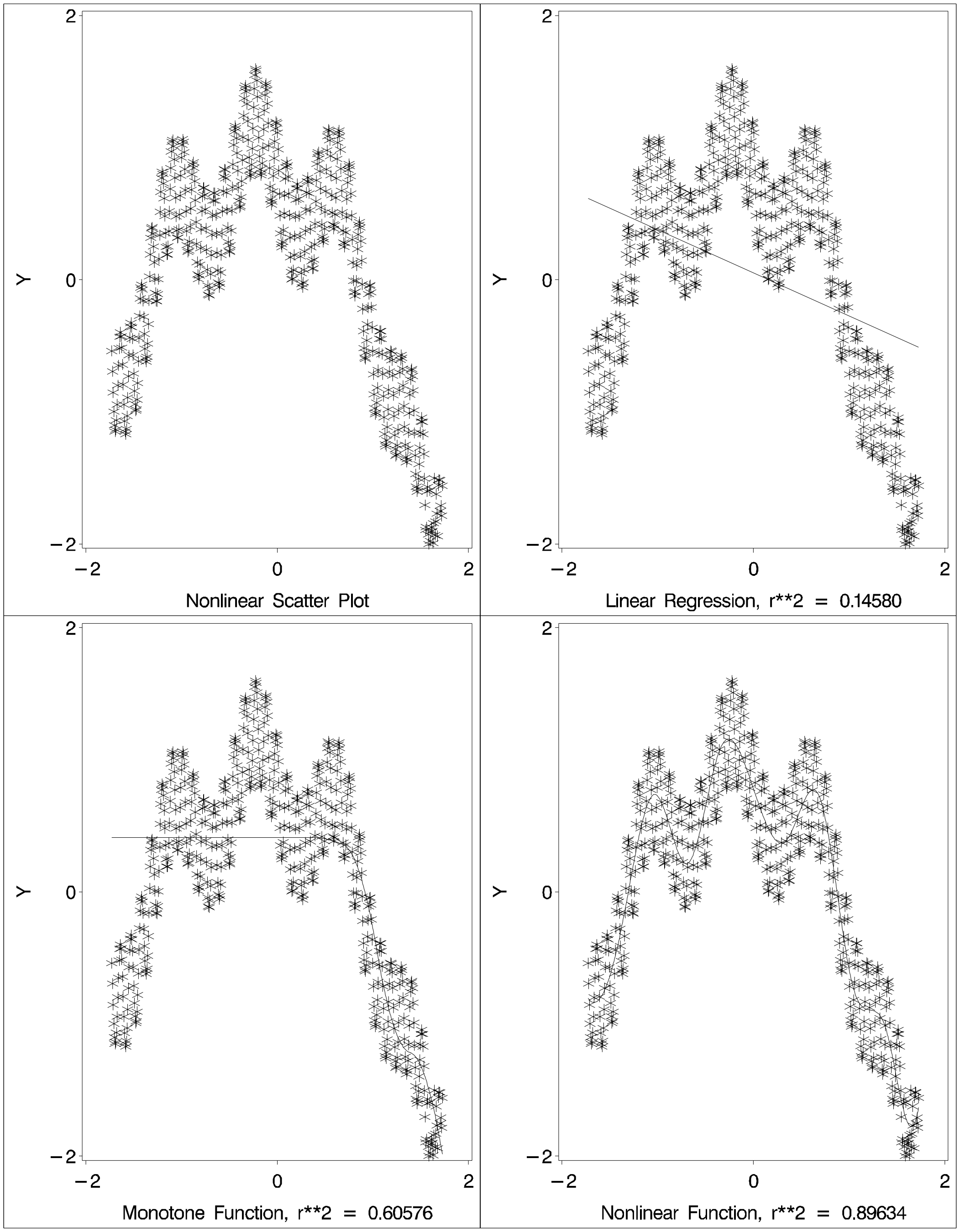


Figure 75.18. Linear, Monotone, and Nonmonotone Regression Functions

The squared correlation is only 0.15 for the linear regression, showing that a simple linear regression model is not appropriate for these data. By relaxing the constraints placed on the regression line, the proportion of variance accounted for increases from 0.15 (linear) to 0.61 (monotone) to 0.90 (nonmonotone). Relaxing the linearity constraint allows the regression function to bend and more closely follow the right portion of the scatter plot. Relaxing the monotonicity constraint allows the regression function to follow the periodic portion of the left side of the plot more closely. The nonlinear MSPLINE transformation is a quadratic spline with knots at the deciles. The nonlinear nonmonotonic SPLINE transformation is a cubic spline with knots at the deciles.

Different knots and different degrees would produce slightly different results. The two nonlinear regression functions could be closely approximated by simpler piecewise linear regression functions. The monotone function could be approximated by a two-piece line with a single knot at the elbow. The nonmonotone function could be approximated by a six-piece function with knots at the five elbows.

With this type of problem (one dependent variable with no missing values that is not transformed and one independent variable that is nonlinearly transformed), PROC TRANSREG always iterates exactly twice (although only one iteration is necessary). The first iteration reports the R^2 for the linear regression line and finds the optimal transformation of X. Since the data change in the first iteration, a second iteration is performed, which reports the R^2 for the final nonlinear regression function, and zero data change. The predicted values, which are a linear function of the optimal transformation of X, contain the y-coordinates for the nonlinear regression function. The variance of the predicted values divided by the variance of Y is the R^2 for the fit of the nonlinear regression function. When X is monotonically transformed, the transformation of X is always monotonically increasing, but the predicted values increase if the correlation is positive and decrease for negative correlations.

Simultaneously Fitting Two Regression Functions

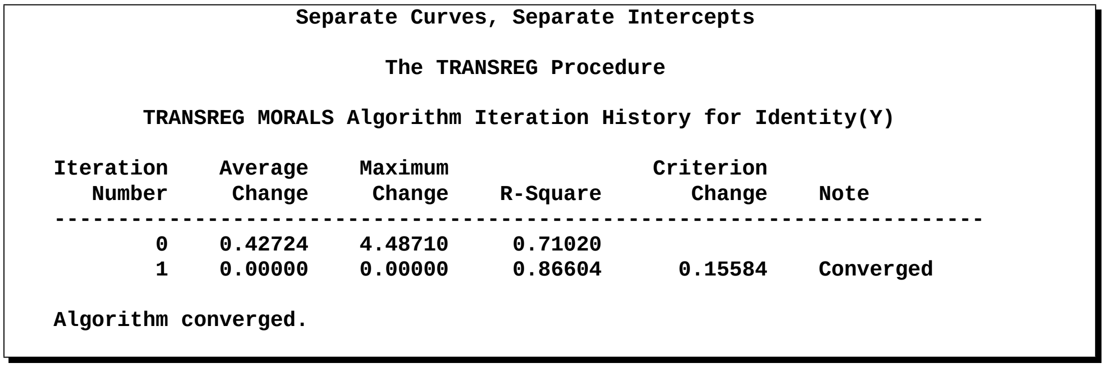
One application of ordinary multiple regression is fitting two or more regression lines through a single scatter plot. With PROC TRANSREG, this application can easily be generalized to fit separate or parallel curves. To illustrate, consider a data set with two groups. The data set has a continuous independent variable X, a continuous dependent variable Y, and a group membership variable G that has the value 1 for one group and 2 for the other group. The following code shows how PROC TRANSREG can be used to fit two lines, curves, and monotone curves simultaneously through a scatter plot. You can use this code with an appropriate number-list for the KNOTS= *t-option*.

```
proc transreg data=A dummy;
   title 'Parallel Lines, Separate Intercepts';
   model identity(Y)=class(G) identity(X);
   output predicted;
run;

proc transreg data=A;
   title 'Parallel Monotone Curves, Separate Intercepts';
   model identity(Y)=class(G) mspline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;

proc transreg data=A dummy;
   title 'Parallel Curves, Separate Intercepts';
   model identity(Y)=class(G) spline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;


proc transreg data=A;
   title 'Separate Slopes, Same Intercept';
   model identity(Y)=class(G / zero=none) * identity(X);
   output predicted;
run;

proc transreg data=A;
   title 'Separate Monotone Curves, Same Intercept';
   model identity(Y) = class(G / zero=none) *
                       mspline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;

proc transreg data=A dummy;
   title 'Separate Curves, Same Intercept';
   model identity(Y) = class(G / zero=none) *
                       spline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;

proc transreg data=A;
   title 'Separate Slopes, Separate Intercepts';
   model identity(Y) = class(G / zero=none) | identity(X);
   output predicted;
run;

proc transreg data=A;
   title 'Separate Monotone Curves, Separate Intercepts';
   model identity(Y) = class(G / zero=none) |
                       mspline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;

proc transreg data=A dummy;
   title 'Separate Curves, Separate Intercepts';
   model identity(Y) = class(G / zero=none) |
                       spline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;
```

Since the variables X1 and X2 both have a large partition of zeros, the KNOTS= *t-option* is specified instead of the NKNOTS= *t-option*. The following example generates an artificial data set with two curves. In the interest of space, only the preceding separate curves, separate intercepts example is run.

```
title 'Separate Curves, Separate Intercepts';

data A;
   do X = -2 to 3 by 0.025;
      G = 1;
      Y = 8*(X*X + 2*cos(X*6)) + 15*normal(7654321);
      output;
      G = 2;
      Y = 4*(-X*X + 4*sin(X*4)) - 40 + 15*normal(7654321);
      output;
      end;
run;

proc transreg data=A dummy;
   model identity(Y) = class(G / zero=none) |
                       spline(X / knots=-1.5 to 2.5 by 0.5);
   output predicted;
run;

proc gplot;
   axis1 minor=none;
   axis2 minor=none label=(angle=90 rotate=0);
   symbol1 color=blue   v=star i=none;
   symbol2 color=yellow v=dot  i=none;
   plot Y*X=1 PY*X=2 /overlay frame cframe=ligr haxis=axis1
                      vaxis=axis2 href=0 vref=0;
run; quit;
```

The previous statements produce Figure 75.19 through Figure 75.20.



```
                    Separate Curves, Separate Intercepts

                           The TRANSREG Procedure

       TRANSREG MORALS Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.42724    4.48710     0.71020
        1    0.00000    0.00000     0.86604      0.15584    Converged

Algorithm converged.
```

Figure 75.19. Fitting Models: Separate Curves, Separate Intercepts

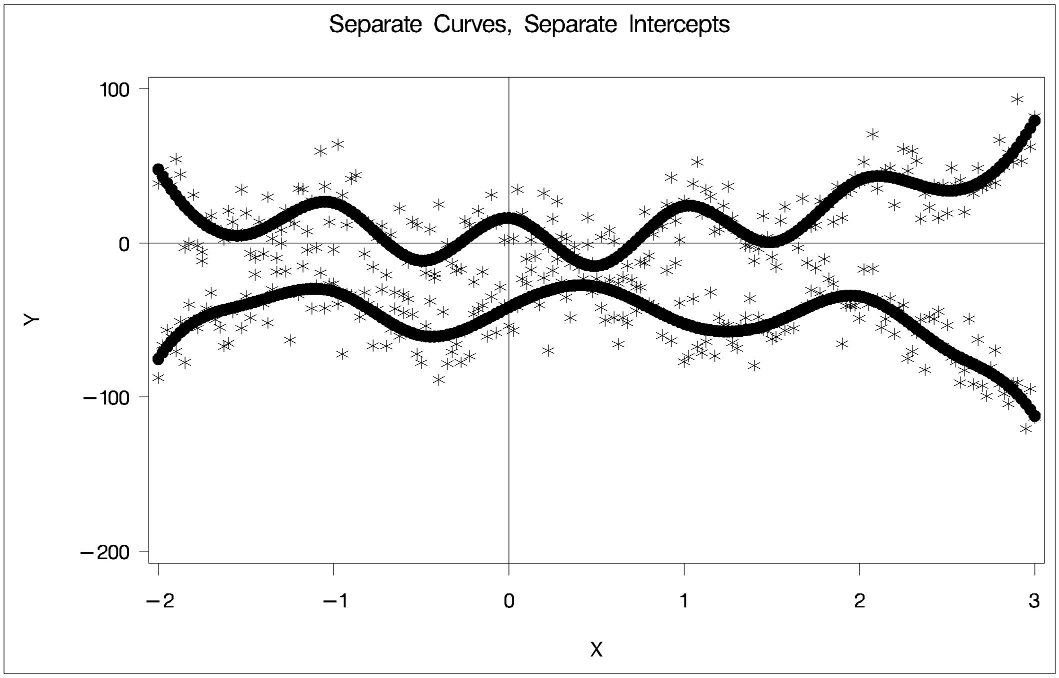


Figure 75.20. Plot for the Separate Curves, Separate Intercepts Example

Unbalanced ANOVA without Dummy Variables

This example illustrates that an analysis of variance model can be formulated as a simple regression model with optimal scoring. The purpose of the example is to explain one aspect of how PROC TRANSREG works, not to propose an alternative way of performing an analysis of variance.

Finding the overall fit of a large, unbalanced analysis of variance model can be handled as an optimal scoring problem without creating large, sparse design matrices. For example, consider an unbalanced full main-effects and interactions ANOVA model with six factors. Assume that a SAS data set is created with factor level indicator variables C1 through C6 and dependent variable Y. If each factor level consists of nonblank single characters, you can create a cell indicator in a DATA step with the statement

```
x=compress(c1||c2||c3||c4||c5||c6);
```

The following statements optimally score X (using the OPSCORE transformation) and do not transform Y. The final R^2 reported is the R^2 for the full analysis of variance model.

```
proc transreg;
   model identity(y)=opscore(x);
   output;
run;
```

The R^2 displayed by the preceding statements is the same as the R^2 that would be reported by both of the following PROC GLM runs.

```
proc glm;
   class x;
   model y=x;
run;

proc glm;
   class c1-c6;
   model y=c1|c2|c3|c4|c5|c6;
run;
```

PROC TRANSREG optimally scores the classes of X, within the space of a single variable with values linearly related to the cell means, so the full ANOVA problem is reduced to a simple regression problem with an optimal independent variable. PROC TRANSREG requires only one iteration to find the optimal scoring of X but, by default, performs a second iteration, which reports no data changes.

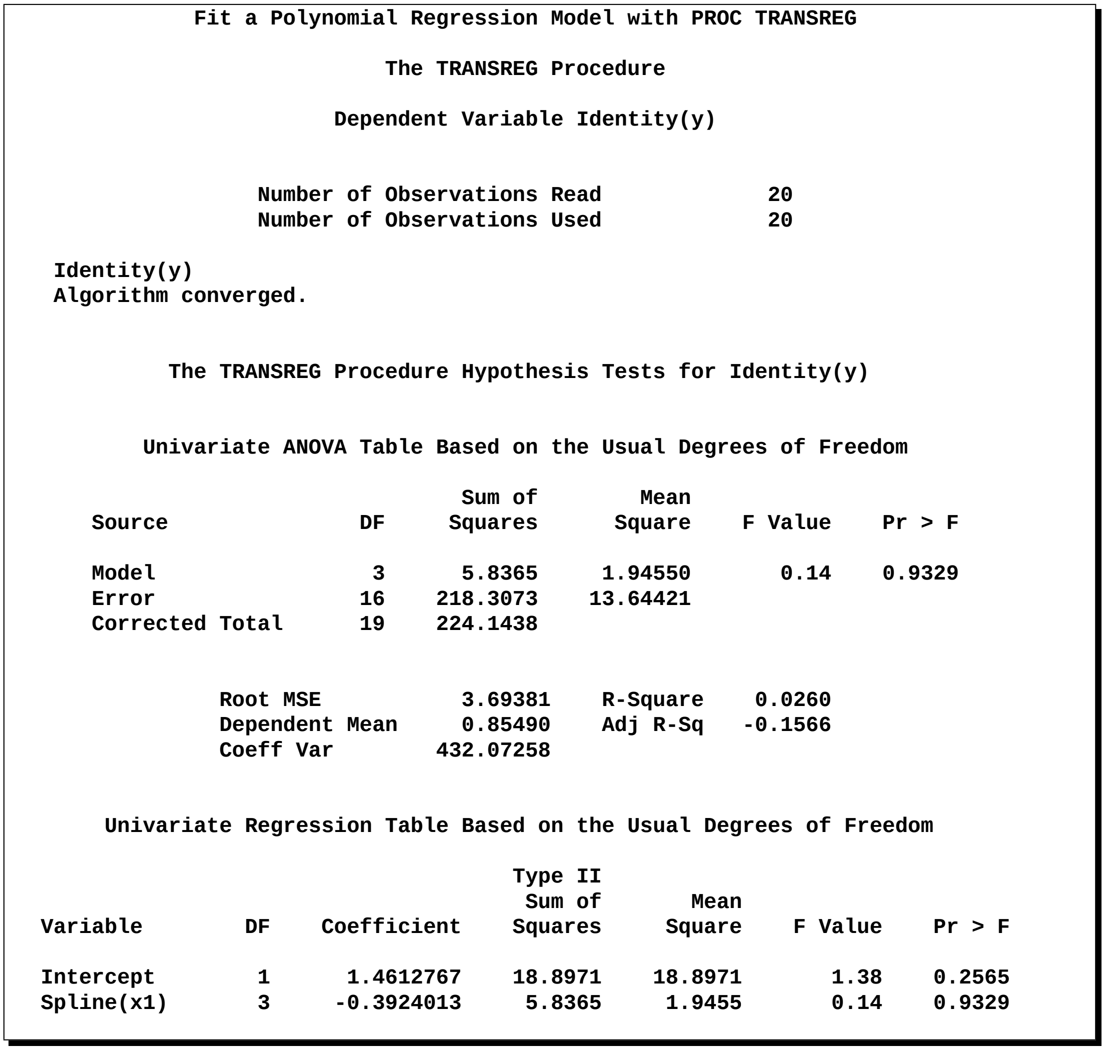
Hypothesis Tests for Simple Univariate Models

If the dependent variable has one parameter (IDENTITY, LINEAR with no missing values, and so on) and if there are no monotonicity constraints, PROC TRANSREG fits univariate models, which can also be fit with a DATA step and PROC REG. This is illustrated with an artificial data set.

```
data htex;
   do i = 0.5 to 10 by 0.5;
      x1 = log(i);
      x2 = sqrt(i) + sin(i);
      x3 = 0.05 * i * i + cos(i);
      y  = x1 - x2 + x3 + 3 * normal(7);
      x1 = x1 + normal(7);
      x2 = x2 + normal(7);
      x3 = x3 + normal(7);
      output;
   end;
run;
```

Both PROC TRANSREG and PROC REG are run to fit the same polynomial regression model. The ANOVA and regression tables from PROC TRANSREG are displayed in Figure 75.21. The ANOVA and regression tables from PROC REG are displayed in Figure 75.22. The SHORT *a-option* is specified to suppress the iteration history.

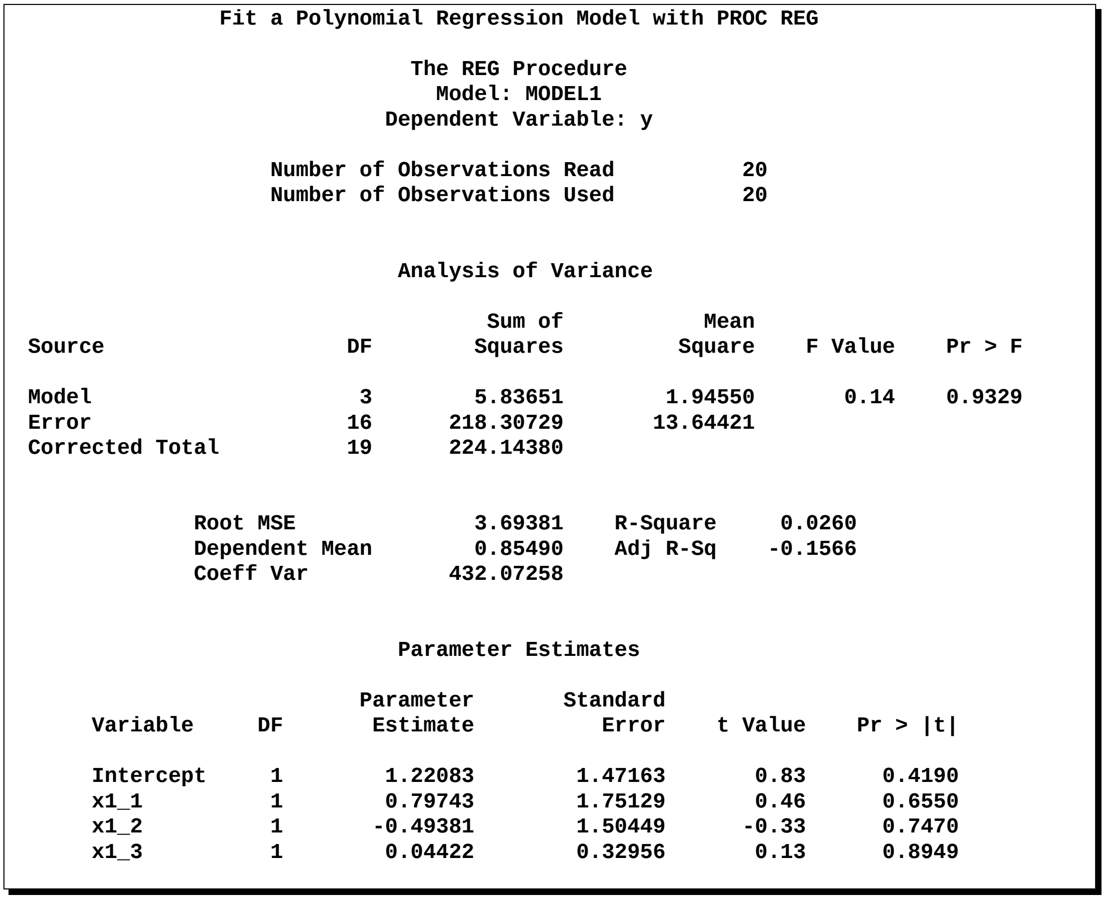
```
proc transreg data=htex ss2 short;
   title 'Fit a Polynomial Regression Model with PROC TRANSREG';
   model identity(y) = spline(x1);
run;
```



```
              Fit a Polynomial Regression Model with PROC TRANSREG

                             The TRANSREG Procedure

                         Dependent Variable Identity(y)


                   Number of Observations Read             20
                   Number of Observations Used             20

 Identity(y)
 Algorithm converged.


             The TRANSREG Procedure Hypothesis Tests for Identity(y)


           Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                    Sum of        Mean
    Source               DF       Squares      Square    F Value    Pr > F

    Model                 3        5.8365     1.94550       0.14    0.9329
    Error                16      218.3073    13.64421
    Corrected Total      19      224.1438


              Root MSE            3.69381    R-Square    0.0260
              Dependent Mean      0.85490    Adj R-Sq   -0.1566
              Coeff Var         432.07258


        Univariate Regression Table Based on the Usual Degrees of Freedom

                                            Type II
                                             Sum of        Mean
Variable         DF     Coefficient         Squares      Square     F Value     Pr > F

Intercept         1       1.4612767         18.8971     18.8971        1.38     0.2565
Spline(x1)        3      -0.3924013          5.8365      1.9455        0.14     0.9329
```

Figure 75.21. ANOVA and Regression Output from PROC TRANSREG

```
data htex2;
   set htex;
   x1_1 = x1;
   x1_2 = x1 * x1;
   x1_3 = x1 * x1 * x1;
run;

proc reg;
   title 'Fit a Polynomial Regression Model with PROC REG';
   model y = x1_1 - x1_3;
run;
```



```
                 Fit a Polynomial Regression Model with PROC REG

                              The REG Procedure
                                Model: MODEL1
                            Dependent Variable: y

                 Number of Observations Read          20
                 Number of Observations Used          20


                             Analysis of Variance

                                    Sum of           Mean
Source                   DF        Squares         Square    F Value    Pr > F

Model                     3        5.83651        1.94550       0.14    0.9329
Error                    16      218.30729       13.64421
Corrected Total          19      224.14380


             Root MSE              3.69381    R-Square     0.0260
             Dependent Mean        0.85490    Adj R-Sq    -0.1566
             Coeff Var           432.07258


                             Parameter Estimates

                         Parameter       Standard
     Variable     DF      Estimate          Error    t Value    Pr > |t|

     Intercept     1       1.22083        1.47163       0.83      0.4190
     x1_1          1       0.79743        1.75129       0.46      0.6550
     x1_2          1      -0.49381        1.50449      -0.33      0.7470
     x1_3          1       0.04422        0.32956       0.13      0.8949
```

Figure 75.22. ANOVA and Regression Output from PROC REG

The PROC TRANSREG regression table differs in several important ways from the parameter estimate table produced by PROC REG. The REG procedure displays standard errors and *t*s. PROC TRANSREG displays Type II sums of squares, mean squares, and *F*s. The difference is because the numerator degrees of freedom are not always 1, so *t*-tests are not uniformly appropriate. When the degrees of freedom for variable x_j is 1, the following relationships hold between the standard errors (s_{β_j}) and the Type II sums of squares (SS_j):

$$s_{\beta_j} = (\hat{\beta}_j^2/F_j)^{1/2}$$

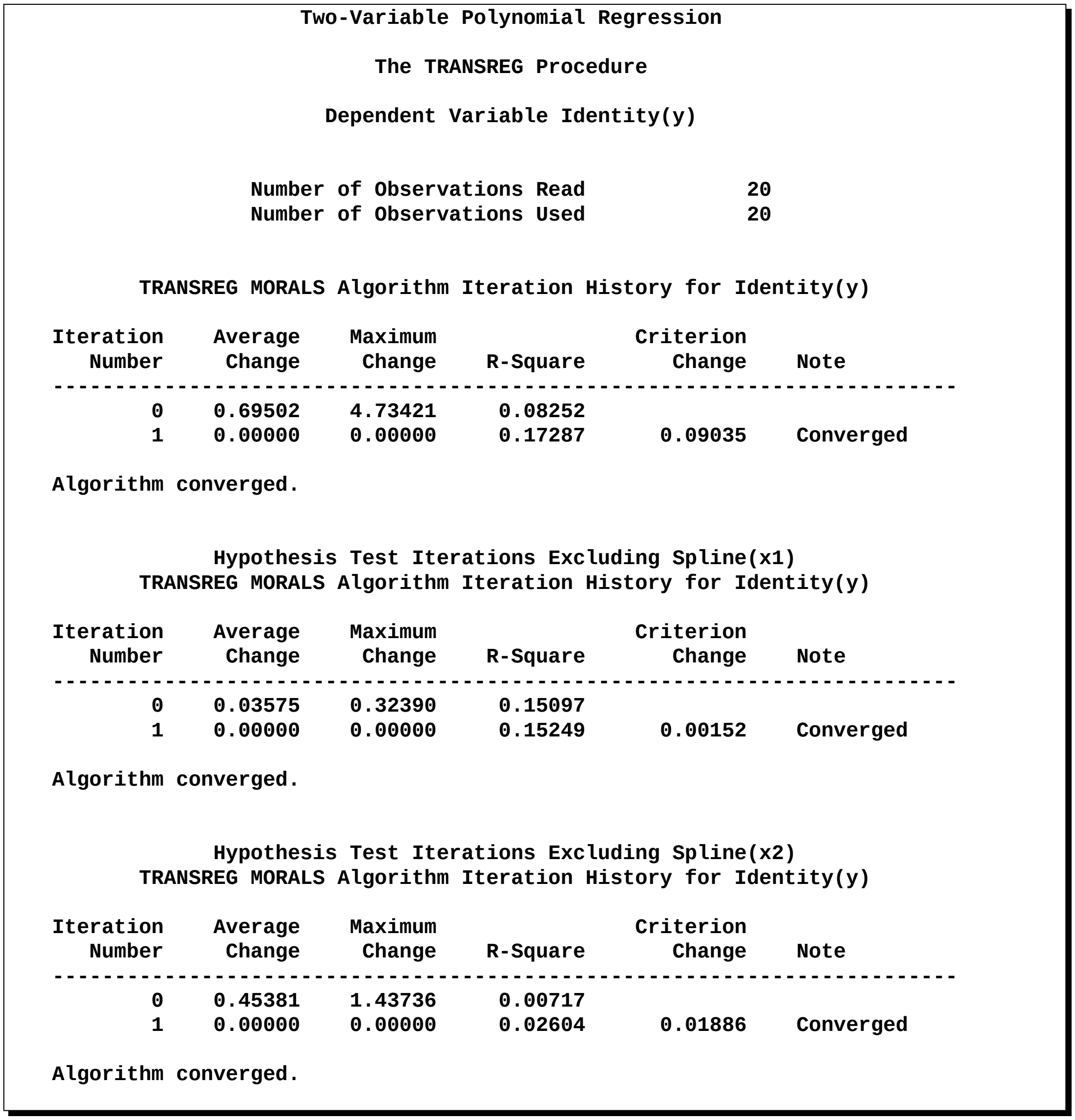
and

$$SS_j = \hat{\beta}_j^2 \times MSE/s_{\beta_j}^2$$

PROC TRANSREG does not provide tests of the individual terms that go into the transformation. (However it could if BSPLINE or PSPLINE had been specified instead of SPLINE.) The test of SPLINE(X1) is the same as the test of the overall model. The intercepts are different due to the different numbers of variables and their standardizations.

In the next example, both X1 and X2 are transformed in the first PROC TRANSREG step, and PROC TRANSREG is used instead of a DATA step to create the polynomials for PROC REG. Both PROC TRANSREG and PROC REG fit the same polynomial regression model. The output from PROC TRANSREG and PROC REG is in Figure 75.23.

```
proc transreg data=htex ss2 dummy;
   title 'Two-Variable Polynomial Regression';
   model identity(y) = spline(x1 x2);
run;

proc transreg noprint data=htex maxiter=0;
   /* Use PROC TRANSREG to prepare input to PROC REG */
   model identity(y) = pspline(x1 x2);
   output out=htex2;
run;

proc reg;
   model y = x1_1-x1_3 x2_1-x2_3;
   test x1_1, x1_2, x1_3;
   test x2_1, x2_2, x2_3;
run;
```

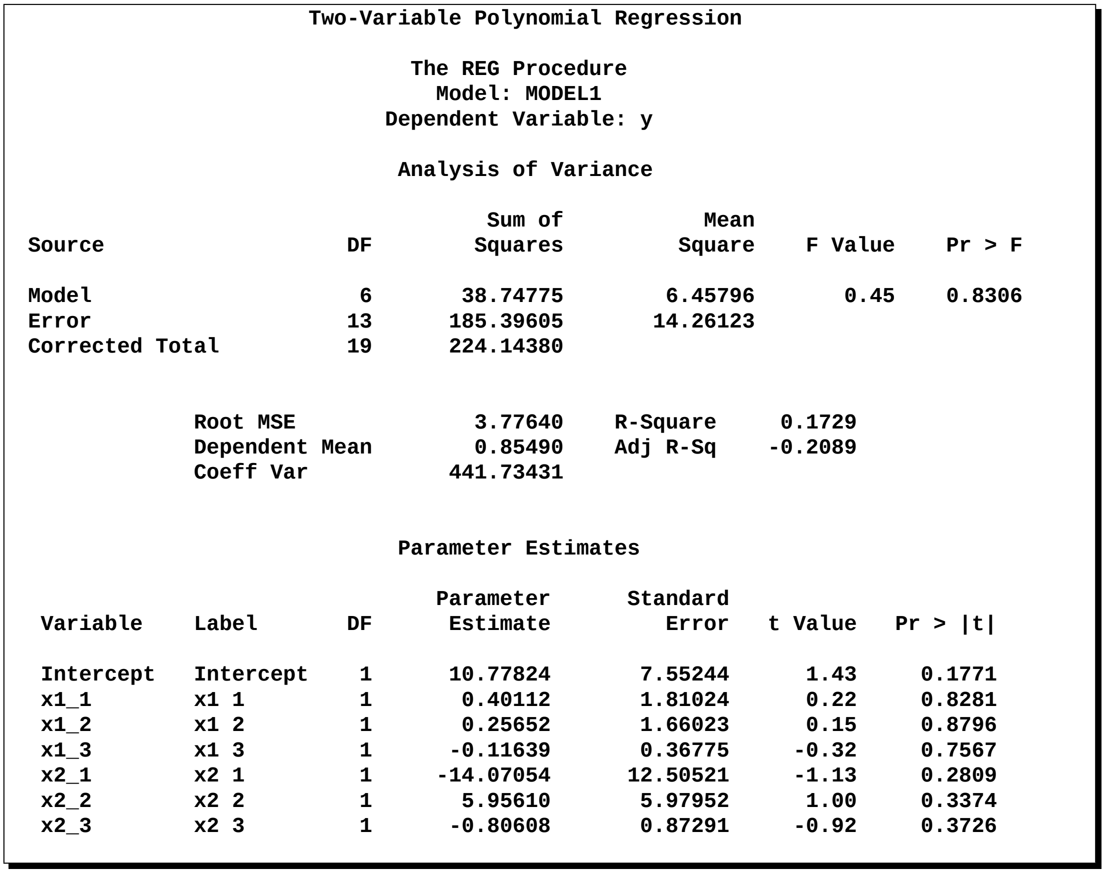


```
                    Two-Variable Polynomial Regression

                          The TRANSREG Procedure

                      Dependent Variable Identity(y)


                Number of Observations Read             20
                Number of Observations Used             20


       TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.69502    4.73421     0.08252
        1    0.00000    0.00000     0.17287      0.09035    Converged

Algorithm converged.


             Hypothesis Test Iterations Excluding Spline(x1)
       TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.03575    0.32390     0.15097
        1    0.00000    0.00000     0.15249      0.00152    Converged

Algorithm converged.


             Hypothesis Test Iterations Excluding Spline(x2)
       TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.45381    1.43736     0.00717
        1    0.00000    0.00000     0.02604      0.01886    Converged

Algorithm converged.
```

Figure 75.23. Two-Variable Polynomial Regression Output from PROC TRANSREG

```
                       Two-Variable Polynomial Regression

                             The TRANSREG Procedure

                The TRANSREG Procedure Hypothesis Tests for Identity(y)


              Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                       Sum of         Mean
       Source                DF       Squares       Square    F Value    Pr > F

       Model                  6       38.7478      6.45796       0.45    0.8306
       Error                 13      185.3960     14.26123
       Corrected Total       19      224.1438


                 Root MSE            3.77640    R-Square    0.1729
                 Dependent Mean      0.85490    Adj R-Sq   -0.2089
                 Coeff Var         441.73431


           Univariate Regression Table Based on the Usual Degrees of Freedom

                                            Type II
                                             Sum of        Mean
   Variable        DF     Coefficient       Squares      Square     F Value    Pr > F

   Intercept        1       3.5437125       35.2282     35.2282        2.47    0.1400
   Spline(x1)       3       0.3644562        4.5682      1.5227        0.11    0.9546
   Spline(x2)       3      -1.3551738       32.9112     10.9704        0.77    0.5315
```

Figure 75.23. (continued)



```
                         Two-Variable Polynomial Regression

                                 The REG Procedure
                                   Model: MODEL1
                               Dependent Variable: y

                               Analysis of Variance

                                        Sum of           Mean
Source                     DF          Squares         Square    F Value    Pr > F

Model                       6         38.74775        6.45796       0.45    0.8306
Error                      13        185.39605       14.26123
Corrected Total            19        224.14380


             Root MSE              3.77640    R-Square     0.1729
             Dependent Mean        0.85490    Adj R-Sq    -0.2089
             Coeff Var           441.73431


                               Parameter Estimates

                                     Parameter      Standard
 Variable    Label        DF          Estimate         Error    t Value    Pr > |t|

 Intercept   Intercept     1          10.77824       7.55244       1.43      0.1771
 x1_1        x1 1          1           0.40112       1.81024       0.22      0.8281
 x1_2        x1 2          1           0.25652       1.66023       0.15      0.8796
 x1_3        x1 3          1          -0.11639       0.36775      -0.32      0.7567
 x2_1        x2 1          1         -14.07054      12.50521      -1.13      0.2809
 x2_2        x2 2          1           5.95610       5.97952       1.00      0.3374
 x2_3        x2 3          1          -0.80608       0.87291      -0.92      0.3726
```

Figure 75.23. (continued)

```
                        Two-Variable Polynomial Regression

                                The REG Procedure
                                  Model: MODEL1

                     Test 1 Results for Dependent Variable y

                                               Mean
            Source                DF          Square    F Value    Pr > F

            Numerator              3         1.52272       0.11    0.9546
            Denominator           13        14.26123


                        Two-Variable Polynomial Regression

                                The REG Procedure
                                  Model: MODEL1

                     Test 2 Results for Dependent Variable y

                                               Mean
            Source                DF          Square    F Value    Pr > F

            Numerator              3        10.97042       0.77    0.5315
            Denominator           13        14.26123
```

Figure 75.23. (continued)

There are three iteration histories: one for the overall model and two for the two independent variables. The first PROC TRANSREG iteration history shows the R^2 of 0.17287 for the fit of the overall model. The second is for

```
   model identity(y) = spline(x2);
```

which excludes SPLINE(X1). The third is for

```
   model identity(y) = spline(x1);
```

which excludes SPLINE(X2). The difference between the first and second R^2 times the total sum of squares is the model sum of squares for SPLINE(X1)

$$(0.17287 - 0.15249) \times 224.143800 = 4.568165$$

The difference between the first and third R^2 times the total sum of squares is the model sum of squares for SPLINE(X2)

$$(0.17287 - 0.02604) \times 224.143800 = 32.911247$$

The TEST statement in PROC REG tests the null hypothesis that the vector of parameters for X1_1 X1_2 X1_3 is zero. This is the same test as the SPLINE(X1) test used by PROC TRANSREG. Similarly, the PROC REG test that the vector of parameters for X2_1 X2_2 X2_3 is zero is the same as the PROC TRANSREG SPLINE(X2) test. So for models with no monotonicity constraints and no dependent variable transformations, PROC TRANSREG provides little more than a different packaging of standard least-squares methodology.

Hypothesis Tests with Monotonicity Constraints

Now consider a model with monotonicity constraints. This model has no counterpart in PROC REG.

```
proc transreg data=htex ss2 short;
   title 'Monotone Splines';
   model identity(y) = mspline(x1-x3 / nknots=3);
run;
```

The SHORT *a-option* is specified to suppress the iteration histories. Two ANOVA tables are displayed—one using liberal degrees of freedom and one using conservative degrees of freedom. All sums of squares and the R^2s are the same for both tables. What differs are the degrees of freedom and statistics that are computed using degrees of freedom. The liberal test has 8 model degrees of freedom and 11 error degrees of freedom, whereas the conservative test has 15 model degrees of freedom and only 4 error degrees of freedom. The "true" *p*-value is between 0.8462 and 0.9997, so clearly you would fail to reject the null hypothesis. Unfortunately, results are not always this clear. See Figure 75.24.

```
                                Monotone Splines

                             The TRANSREG Procedure

                          Dependent Variable Identity(y)


                    Number of Observations Read             20
                    Number of Observations Used             20

Identity(y)
Algorithm converged.


            The TRANSREG Procedure Hypothesis Tests for Identity(y)


          Univariate ANOVA Table Based on Liberal Degrees of Freedom

                                  Sum of        Mean
 Source               DF         Squares      Square    F Value    Liberal p

 Model                 8         58.0534     7.25667       0.48    >= 0.8462
 Error                11        166.0904    15.09913
 Corrected Total      19        224.1438


             Root MSE           3.88576    R-Square    0.2590
             Dependent Mean     0.85490    Adj R-Sq   -0.2799
             Coeff Var        454.52581


        Univariate ANOVA Table Based on Conservative Degrees of Freedom

                                 Sum of        Mean                Conservative
Source               DF         Squares      Square    F Value               p

Model                15         58.0534     3.87022       0.09       <= 0.9997
Error                 4        166.0904    41.52261
Corrected Total      19        224.1438


             Root MSE           6.44380    R-Square    0.2590
             Dependent Mean     0.85490    Adj R-Sq   -2.5197
             Coeff Var        753.74578
```

Figure 75.24. Monotone Spline Transformations

```
                               Monotone Splines

                            The TRANSREG Procedure

         Univariate Regression Table Based on Liberal Degrees of Freedom

                                            Type II
                                             Sum of        Mean
Variable         DF     Coefficient         Squares      Square     F Value     Liberal p

Intercept         1       4.8687676         54.7372     54.7372        3.63    >= 0.0834
Mspline(x1)       2      -0.6886834         12.1943      6.0972        0.40    >= 0.6773
Mspline(x2)       3      -1.8237319         46.3155     15.4385        1.02    >= 0.4199
Mspline(x3)       3       0.8646155         24.6840      8.2280        0.54    >= 0.6616


       Univariate Regression Table Based on Conservative Degrees of Freedom

                                           Type II
                                            Sum of        Mean                Conservative
  Variable         DF    Coefficient       Squares      Square    F Value               p

  Intercept         1      4.8687676       54.7372     54.7372       1.32        <= 0.3149
  Mspline(x1)       5     -0.6886834       12.1943      2.4389       0.06        <= 0.9959
  Mspline(x2)       5     -1.8237319       46.3155      9.2631       0.22        <= 0.9344
  Mspline(x3)       5      0.8646155       24.6840      4.9368       0.12        <= 0.9809
```

Figure 75.24. (continued)

Hypothesis Tests with Dependent Variable Transformations

PROC TRANSREG can also provide approximate tests of hypotheses when the dependent variable is transformed, but the output is more complicated. When a dependent variable has more than one degree of freedom, the problem becomes multivariate. Hypothesis tests are performed in the context of a multivariate linear model with the number of dependent variables equal to the number of scoring parameters for the dependent variable transformation. The transformation regression model with a dependent variable transformation differs from the usual multivariate linear model in two important ways. First, the usual assumption of multivariate normality is always violated. This fact is simply ignored. This is one reason that all hypothesis tests in the presence of a dependent variable transformation should be considered approximate at best. Multivariate normality is assumed even though it is known that the assumption is violated.

The second difference concerns the usual multivariate test statistics: Pillai's Trace, Wilks' Lambda, Hotelling-Lawley Trace, and Roy's Greatest Root. The first three statistics are defined in terms of all the squared canonical correlations. Here, there is only one linear combination (the transformation) and, hence, only one squared canonical correlation of interest, which is equal to the R^2. It may seem that Roy's Greatest Root, which uses only the largest squared canonical correlation, is the only statistic of interest. Unfortunately, Roy's Greatest Root is very liberal and provides only a lower bound on the *p*-value. Approximate upper bounds are provided by adjusting the other three statistics for the one linear combination case. The Wilks' Lambda, Pillai's Trace, and Hotelling-Lawley Trace statistics are a conservative adjustment of the usual statistics.

These statistics are normally defined in terms of the squared canonical correlations, which are the eigenvalues of the matrix $\mathbf{H}(\mathbf{H} + \mathbf{E})^{-1}$, where $\mathbf{H}$ is the hypothesis sum-of-squares matrix and $\mathbf{E}$ is the error sum-of-squares matrix. Here the R^2 is used for the first eigenvalue, and all other eigenvalues are set to 0 since only one linear combination is used. Degrees of freedom are computed assuming that all linear combinations contribute to the Lambda and Trace statistics, so the *F* tests for those statistics are conservative. The *p*-values for the liberal and conservative statistics provide approximate lower and upper bounds on *p*. In practice, the adjusted Pillai's Trace is very conservative—perhaps too conservative to be useful. Wilks' Lambda is less conservative, and the Hotelling-Lawley Trace seems to be the least conservative. The conservative statistics and the liberal Roy's Greatest Root provide a bound on the true *p*-value. Unfortunately, they sometimes report a bound of 0.0001 and 1.0000.

Here is an example with a dependent variable transformation.

```
proc transreg data=htex ss2 dummy short;
   title 'Transform Dependent and Independent Variables';
   model spline(y) = spline(x1-x3);
run;
```

The univariate results match Roy's Greatest Root results. Clearly, the proper action is to fail to reject the null hypothesis. However, as stated previously, results are not always this clear. See Figure 75.25.

```
                 Transform Dependent and Independent Variables

                             The TRANSREG Procedure

                          Dependent Variable Spline(y)


                   Number of Observations Read             20
                   Number of Observations Used             20

   Spline(y)
   Algorithm converged.


             The TRANSREG Procedure Hypothesis Tests for Spline(y)


         Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
    Source               DF     Squares      Square    F Value    Liberal p

    Model                 9    110.8822    12.32025       1.09    >= 0.4452
    Error                10    113.2616    11.32616
    Corrected Total      19    224.1438

    The above statistics are not adjusted for the fact that the dependent
    variable was transformed and so are generally liberal.


                Root MSE           3.36544    R-Square    0.4947
                Dependent Mean     0.85490    Adj R-Sq    0.0399
                Coeff Var        393.66234


    Adjusted Multivariate ANOVA Table Based on the Usual Degrees of Freedom

          Dependent Variable Scoring Parameters=3   S=3   M=2.5   N=3

 Statistic                    Value    F Value    Num DF    Den DF            p

 Wilks' Lambda             0.505308       0.23        27    24.006    <= 0.9998
 Pillai's Trace            0.494692       0.22        27        30    <= 0.9999
 Hotelling-Lawley Trace    0.978992       0.26        27    11.589    <= 0.9980
 Roy's Greatest Root       0.978992       1.09         9        10    >= 0.4452

The Wilks' Lambda, Pillai's Trace, and Hotelling-Lawley Trace statistics are a
conservative adjustment of the normal statistics.  Roy's Greatest Root is
liberal.  These statistics are normally defined in terms of the squared
canonical correlations which are the eigenvalues of the matrix H*inv(H+E).
Here the R-Square is used for the first eigenvalue and all other eigenvalues
are set to zero since only one linear combination is used.  Degrees of freedom
are computed assuming all linear combinations contribute to the Lambda and
Trace statistics, so the F tests for those statistics are conservative.  The p
values for the liberal and conservative statistics provide approximate lower
and upper bounds on p.  A liberal test statistic with conservative degrees of
freedom and a conservative test statistic with liberal degrees of freedom yield
at best an approximate p value, which is indicated by a "~" before the p value.
```

Figure 75.25. Transform Dependent and Independent Variables

```
                  Transform Dependent and Independent Variables

                              The TRANSREG Procedure

        Univariate Regression Table Based on the Usual Degrees of Freedom

                                          Type II
                                           Sum of       Mean
 Variable       DF    Coefficient    Squares      Square    F Value    Liberal p

 Intercept       1      6.9089087    117.452     117.452      10.37    >= 0.0092
 Spline(x1)      3     -1.0832321     32.493      10.831       0.96    >= 0.4504
 Spline(x2)      3     -2.1539191     45.251      15.084       1.33    >= 0.3184
 Spline(x3)      3      0.4779207     10.139       3.380       0.30    >= 0.8259

The above statistics are not adjusted for the fact that the dependent variable
was transformed and so are generally liberal.


  Adjusted Multivariate Regression Table Based on the Usual Degrees of Freedom

Variable   Coefficient Statistic           Value F Value Num DF Den DF         p

Intercept    6.9089087 Wilks' Lambda     0.49092    2.77      3      8    0.1112
                       Pillai's Trace    0.50908    2.77      3      8    0.1112
                       Hotelling-Lawley 1.036993    2.77      3      8    0.1112
                       Trace
                       Roy's Greatest   1.036993    2.77      3      8    0.1112
                       Root

Spline(x1)  -1.0832321 Wilks' Lambda    0.777072    0.24      9 19.621 <= 0.9840
                       Pillai's Trace   0.222928    0.27      9     30 <= 0.9787
                       Hotelling-Lawley 0.286883    0.24      9 9.8113 <= 0.9784
                       Trace
                       Roy's Greatest   0.286883    0.96      3     10 >= 0.4504
                       Root

Spline(x2)  -2.1539191 Wilks' Lambda    0.714529    0.32      9 19.621 <= 0.9572
                       Pillai's Trace   0.285471    0.35      9     30 <= 0.9494
                       Hotelling-Lawley 0.399524    0.33      9 9.8113 <= 0.9424
                       Trace
                       Roy's Greatest   0.399524    1.33      3     10 >= 0.3184
                       Root

Spline(x3)   0.4779207 Wilks' Lambda    0.917838    0.08      9 19.621 <= 0.9998
                       Pillai's Trace   0.082162    0.09      9     30 <= 0.9996
                       Hotelling-Lawley 0.089517    0.07      9 9.8113 <= 0.9997
                       Trace
                       Roy's Greatest   0.089517    0.30      3     10 >= 0.8259
                       Root

These statistics are adjusted in the same way as the multivariate statistics
above.
```

Figure 75.25. (continued)

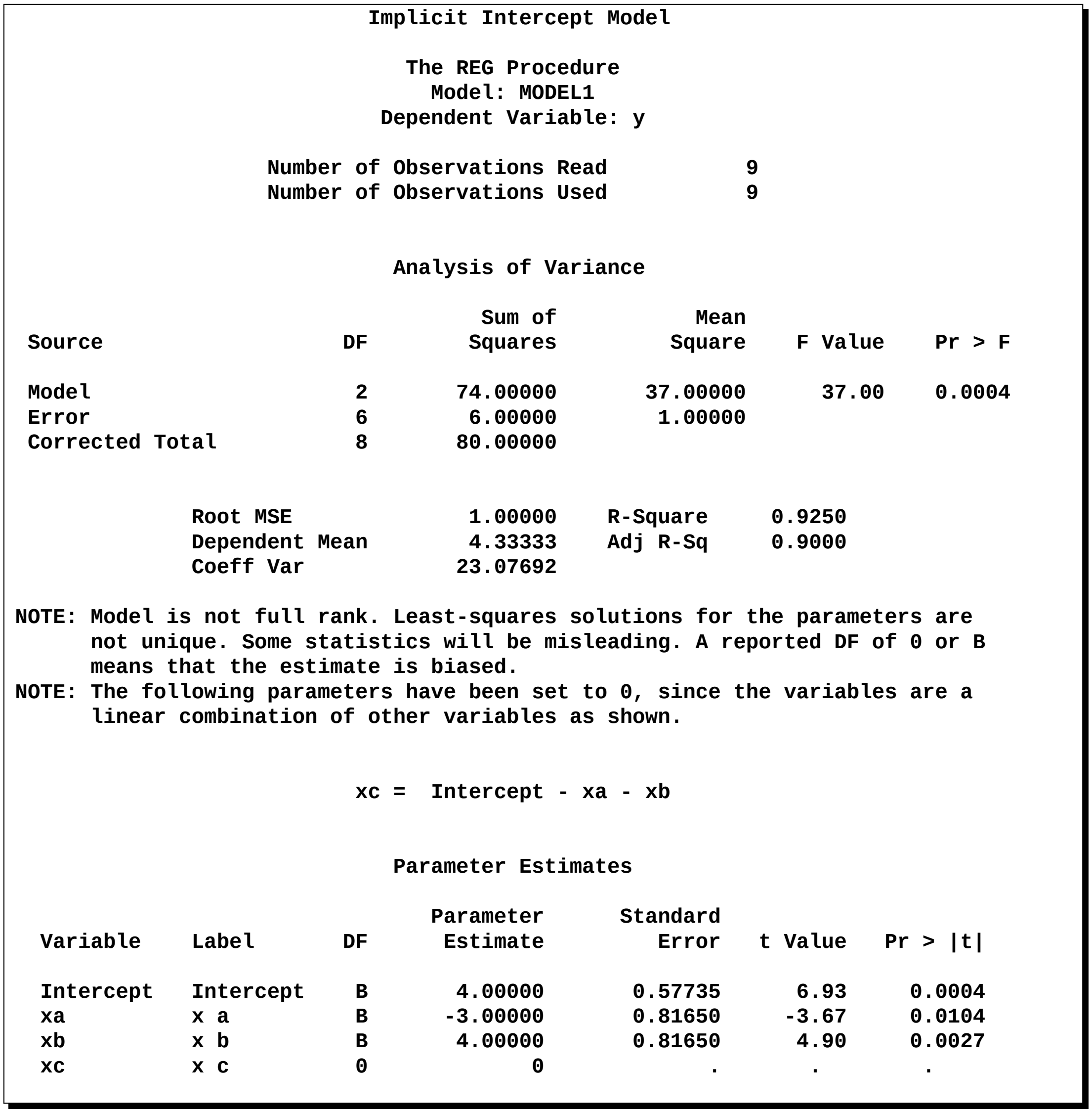
Hypothesis Tests with One-Way ANOVA

One-way ANOVA models are fit with either an explicit or implicit intercept. In implicit intercept models, the ANOVA table of PROC TRANSREG is the correct table for a model with an intercept, and the regression table is the correct table for a model that does not have a separate explicit intercept. The PROC TRANSREG implicit intercept ANOVA table matches the PROC REG table when the NOINT *a-option* is not specified, and the PROC TRANSREG implicit intercept regression table matches the PROC REG table when the NOINT *a-option* is specified. The following code illustrates this relationship. See Figure 75.26 through Figure 75.27 for the results.

```
data oneway;
   input y x $;
   datalines;
0 a
1 a
2 a
7 b
8 b
9 b
3 c
4 c
5 c
;

proc transreg ss2 data=oneway short;
   title ’Implicit Intercept Model’;
   model identity(y) = class(x / zero=none);
   output out=oneway2;
run;

proc reg data=oneway2;
   model y = xa xb xc;         /* Implicit Intercept ANOVA      */
   model y = xa xb xc / noint; /* Implicit Intercept Regression */
run;
```

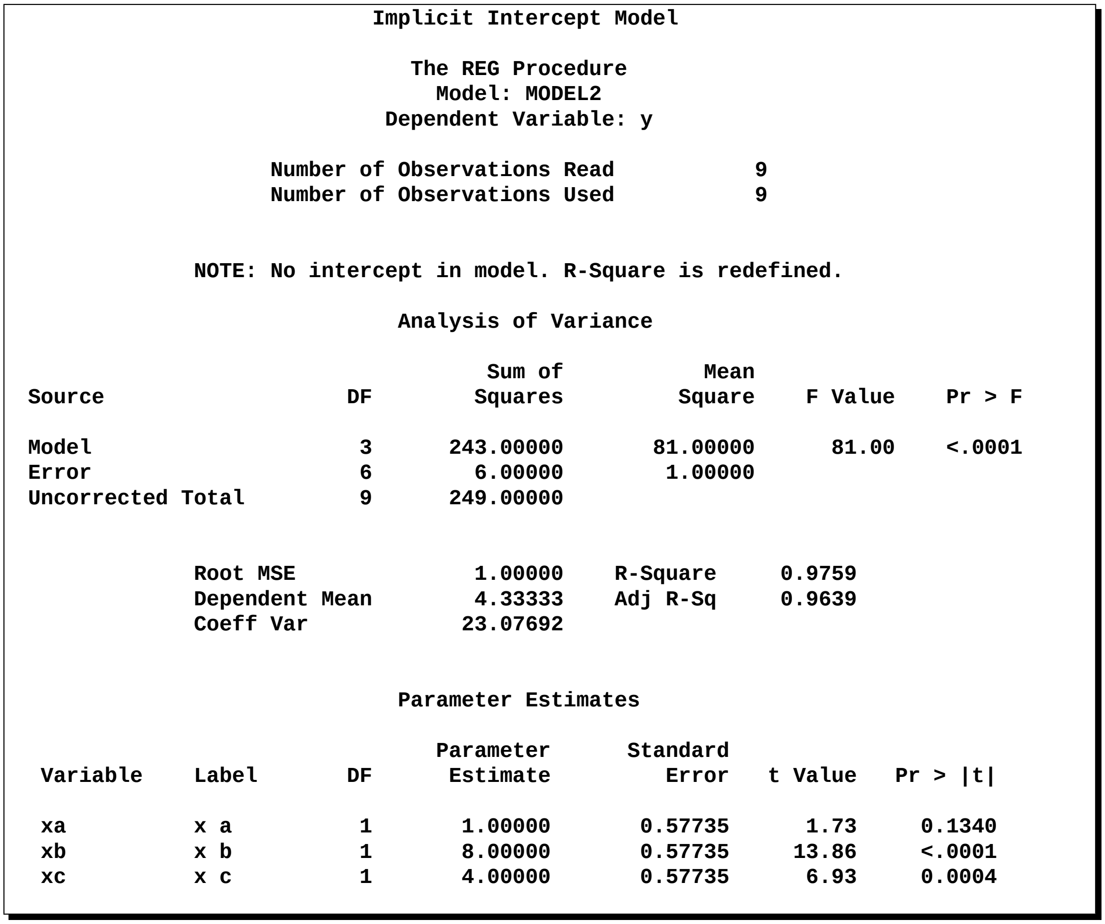
```
                           Implicit Intercept Model

                            The TRANSREG Procedure

                        Dependent Variable Identity(y)


                           Class Level Information

                          Class    Levels    Values

                          x             3    a b c


                  Number of Observations Read             9
                  Number of Observations Used             9
                  Implicit Intercept Model

  Identity(y)
  Algorithm converged.


           The TRANSREG Procedure Hypothesis Tests for Identity(y)


         Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of        Mean
     Source               DF     Squares      Square    F Value    Pr > F

     Model                 2    74.00000    37.00000      37.00    0.0004
     Error                 6     6.00000     1.00000
     Corrected Total       8    80.00000


               Root MSE           1.00000    R-Square    0.9250
               Dependent Mean     4.33333    Adj R-Sq    0.9000
               Coeff Var         23.07692


      Univariate Regression Table Based on the Usual Degrees of Freedom

                                  Type II
                                   Sum of      Mean
Variable       DF    Coefficient  Squares    Square   F Value   Pr > F   Label

Class.xa        1     1.00000000    3.000     3.000      3.00   0.1340   x a
Class.xb        1     8.00000000  192.000   192.000    192.00   <.0001   x b
Class.xc        1     4.00000000   48.000    48.000     48.00   0.0004   x c
```

Figure 75.26. Implicit Intercept Model (TRANSREG Procedure)



```
                         Implicit Intercept Model

                            The REG Procedure
                              Model: MODEL1
                          Dependent Variable: y

                 Number of Observations Read           9
                 Number of Observations Used           9


                           Analysis of Variance

                                      Sum of           Mean
 Source                   DF         Squares         Square    F Value    Pr > F

 Model                     2        74.00000       37.00000      37.00    0.0004
 Error                     6         6.00000        1.00000
 Corrected Total           8        80.00000


              Root MSE              1.00000    R-Square     0.9250
              Dependent Mean        4.33333    Adj R-Sq     0.9000
              Coeff Var            23.07692

NOTE: Model is not full rank. Least-squares solutions for the parameters are
      not unique. Some statistics will be misleading. A reported DF of 0 or B
      means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a
      linear combination of other variables as shown.


                        xc =  Intercept - xa - xb


                           Parameter Estimates

                                     Parameter      Standard
  Variable    Label       DF          Estimate         Error    t Value   Pr > |t|

  Intercept   Intercept    B           4.00000       0.57735       6.93     0.0004
  xa          x a          B          -3.00000       0.81650      -3.67     0.0104
  xb          x b          B           4.00000       0.81650       4.90     0.0027
  xc          x c          0                 0             .          .          .
```

Figure 75.27. Implicit Intercept Model (REG Procedure)



```
                           Implicit Intercept Model

                              The REG Procedure
                                Model: MODEL2
                            Dependent Variable: y

                    Number of Observations Read           9
                    Number of Observations Used           9


              NOTE: No intercept in model. R-Square is redefined.

                             Analysis of Variance

                                   Sum of           Mean
Source                   DF       Squares         Square    F Value    Pr > F

Model                     3     243.00000       81.00000      81.00    <.0001
Error                     6       6.00000        1.00000
Uncorrected Total         9     249.00000


             Root MSE              1.00000    R-Square     0.9759
             Dependent Mean        4.33333    Adj R-Sq     0.9639
             Coeff Var            23.07692


                             Parameter Estimates

                                  Parameter      Standard
 Variable    Label        DF       Estimate         Error    t Value    Pr > |t|

 xa          x a           1        1.00000       0.57735       1.73      0.1340
 xb          x b           1        8.00000       0.57735      13.86      <.0001
 xc          x c           1        4.00000       0.57735       6.93      0.0004
```

Figure 75.27. (continued)

Using the DESIGN Output Option

This example uses PROC TRANSREG and the DESIGN *o-option* to prepare an input data set with classification variables for the LOGISTIC procedure. The DESIGN *o-option* specifies that the goal is design matrix creation, not analysis. When you specify DESIGN, dependent variables are not required. The DEVIATIONS (or EFFECTS) *t-option* requests a deviations-from-means $(1, 0, -1)$ coding of the classification variables, which is the same coding the CATMOD procedure uses. See Figure 75.28. PROC TRANSREG automatically creates a macro variable &_trgind that contains the list of independent variables created. This macro is used in the PROC LOGISTIC MODEL statement. See Figure 75.29. For comparison, the same analysis is also performed with PROC CATMOD. See Figure 75.30.

```
title 'Using PROC TRANSREG to Create a Design Matrix';

data a;
   do y = 1, 2;
      do a = 1 to 4;
```

```
         do b = 1 to 3;
            w = ceil(uniform(1) * 10 + 10);
            output;
         end;
      end;
   end;
run;

proc transreg data=a design;
   model class(a b / deviations);
   id y w;
   output;
run;

proc print;
   title2 ’PROC TRANSREG Output Data Set’;
run;

proc logistic;
   title2 ’PROC LOGISTIC with Classification Variables’;
   freq w;
   model y = &_trgind;
run;

proc catmod data=a;
   title2 ’PROC CATMOD Should Produce the Same Results’;
   model y = a b;
   weight w;
run;
```

```
                  Using PROC TRANSREG to Create a Design Matrix
                          PROC TRANSREG Output Data Set

Obs   _TYPE_   _NAME_   Intercept   a1   a2   a3   b1   b2   a   b   y    w

  1   SCORE      1          1        1    0    0    1    0   1   1   1   12
  2   SCORE      1          1        1    0    0    0    1   1   2   1   20
  3   SCORE      1          1        1    0    0   -1   -1   1   3   1   14
  4   SCORE      1          1        0    1    0    1    0   2   1   1   13
  5   SCORE      1          1        0    1    0    0    1   2   2   1   20
  6   SCORE      1          1        0    1    0   -1   -1   2   3   1   20
  7   SCORE      1          1        0    0    1    1    0   3   1   1   16
  8   SCORE      1          1        0    0    1    0    1   3   2   1   16
  9   SCORE      1          1        0    0    1   -1   -1   3   3   1   11
 10   SCORE      1          1       -1   -1   -1    1    0   4   1   1   11
 11   SCORE      1          1       -1   -1   -1    0    1   4   2   1   19
 12   SCORE      1          1       -1   -1   -1   -1   -1   4   3   1   16
 13   SCORE      2          1        1    0    0    1    0   1   1   2   19
 14   SCORE      2          1        1    0    0    0    1   1   2   2   11
 15   SCORE      2          1        1    0    0   -1   -1   1   3   2   20
 16   SCORE      2          1        0    1    0    1    0   2   1   2   13
 17   SCORE      2          1        0    1    0    0    1   2   2   2   13
 18   SCORE      2          1        0    1    0   -1   -1   2   3   2   17
 19   SCORE      2          1        0    0    1    1    0   3   1   2   20
 20   SCORE      2          1        0    0    1    0    1   3   2   2   13
 21   SCORE      2          1        0    0    1   -1   -1   3   3   2   17
 22   SCORE      2          1       -1   -1   -1    1    0   4   1   2   15
 23   SCORE      2          1       -1   -1   -1    0    1   4   2   2   16
 24   SCORE      2          1       -1   -1   -1   -1   -1   4   3   2   13
```

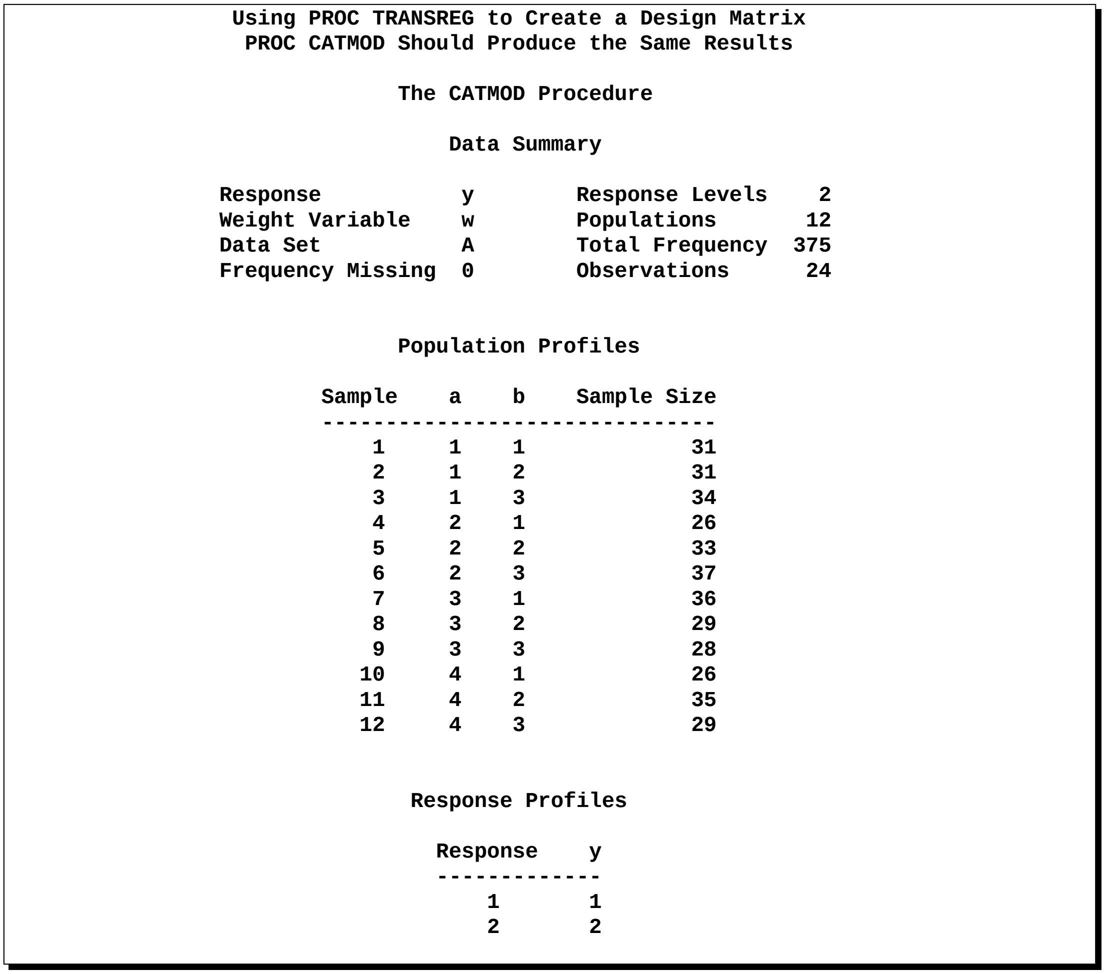
Figure 75.28. The PROC TRANSREG Design Matrix

```
     Using PROC TRANSREG to Create a Design Matrix
      PROC LOGISTIC with Classification Variables

                 The LOGISTIC Procedure

                   Model Information

     Data Set                      WORK.DATA8
     Response Variable             y
     Number of Response Levels     2
     Frequency Variable            w
     Model                         binary logit
     Optimization Technique        Fisher's scoring


        Number of Observations Read          24
        Number of Observations Used          24
        Sum of Frequencies Read             375
        Sum of Frequencies Used             375


                    Response Profile

           Ordered                        Total
             Value            y       Frequency

                 1            1             188
                 2            2             187

              Probability modeled is y=1.


                Model Convergence Status

     Convergence criterion (GCONV=1E-8) satisfied.


                  Model Fit Statistics

                                      Intercept
                       Intercept            and
         Criterion          Only     Covariates

         AIC             521.858        524.378
         SC              525.785        547.939
         -2 Log L        519.858        512.378


        Testing Global Null Hypothesis: BETA=0

Test                 Chi-Square       DF     Pr > ChiSq

Likelihood Ratio         7.4799        5         0.1873
Score                    7.4312        5         0.1905
Wald                     7.3356        5         0.1969
```

Figure 75.29. PROC LOGISTIC Output

```
                  Using PROC TRANSREG to Create a Design Matrix
                   PROC LOGISTIC with Classification Variables

                              The LOGISTIC Procedure

                   Analysis of Maximum Likelihood Estimates

                                     Standard          Wald
      Parameter    DF    Estimate       Error    Chi-Square    Pr > ChiSq

      Intercept     1    -0.00040      0.1044        0.0000        0.9969
      a1            1     -0.0802      0.1791        0.2007        0.6542
      a2            1      0.2001      0.1800        1.2363        0.2662
      a3            1     -0.1350      0.1819        0.5514        0.4578
      b1            1     -0.2392      0.1500        2.5436        0.1107
      b2            1      0.3433      0.1474        5.4223        0.0199


                               Odds Ratio Estimates

                                 Point          95% Wald
                   Effect    Estimate      Confidence Limits

                   a1           0.923       0.650       1.311
                   a2           1.222       0.858       1.738
                   a3           0.874       0.612       1.248
                   b1           0.787       0.587       1.056
                   b2           1.410       1.056       1.882


         Association of Predicted Probabilities and Observed Responses

               Percent Concordant     54.0    Somers' D    0.163
               Percent Discordant     37.8    Gamma        0.177
               Percent Tied            8.2    Tau-a        0.082
               Pairs                 35156    c            0.581
```

Figure 75.29. (continued)



```
                  Using PROC TRANSREG to Create a Design Matrix
                   PROC CATMOD Should Produce the Same Results

                             The CATMOD Procedure

                                 Data Summary

                Response           y        Response Levels    2
                Weight Variable    w        Populations       12
                Data Set           A        Total Frequency  375
                Frequency Missing  0        Observations      24


                             Population Profiles

                        Sample    a    b    Sample Size
                        -------------------------------
                           1      1    1             31
                           2      1    2             31
                           3      1    3             34
                           4      2    1             26
                           5      2    2             33
                           6      2    3             37
                           7      3    1             36
                           8      3    2             29
                           9      3    3             28
                          10      4    1             26
                          11      4    2             35
                          12      4    3             29


                              Response Profiles

                                Response    y
                                -------------
                                    1       1
                                    2       2
```

Figure 75.30. PROC CATMOD Output

```
                 Using PROC TRANSREG to Create a Design Matrix
                  PROC CATMOD Should Produce the Same Results

                             The CATMOD Procedure

                          Maximum Likelihood Analysis

                    Maximum likelihood computations converged.


                     Maximum Likelihood Analysis of Variance

                Source                DF   Chi-Square    Pr > ChiSq
                --------------------------------------------------
                Intercept              1         0.00        0.9969
                a                      3         1.50        0.6823
                b                      2         5.64        0.0597

                Likelihood Ratio       6         2.81        0.8329


                     Analysis of Maximum Likelihood Estimates

                                         Standard        Chi-
           Parameter      Estimate          Error      Square    Pr > ChiSq
           ------------------------------------------------------------
           Intercept      -0.00040         0.1044        0.00        0.9969
           a         1     -0.0802         0.1791        0.20        0.6542
                     2      0.2001         0.1800        1.24        0.2662
                     3     -0.1350         0.1819        0.55        0.4578
           b         1     -0.2392         0.1500        2.54        0.1107
                     2      0.3434         0.1474        5.42        0.0199
```

Figure 75.30. (continued)

Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO

A discrete choice experiment is constructed consisting of four product brands, each available at three different prices, $1.49, $1.99, $2.49. In addition, each choice set contains a constant “other” alternative available at $1.49. In the fifth choice set, price is constant. PROC TRANSREG is used to code the designand the PHREG procedure fits the multinomial logit choice model (not shown). See http://www.sas.com/service/techsup/tnote/tnote_stat.html for more information on discrete choice modeling and the multinomial logit model. Look for the latest “Multinomial Logit, Discrete Choice Modeling” report.

```
title 'Choice Model Coding';

data design;
   array p[4];
   input p1-p4 @@;
   set = _n_;
   do brand = 1 to 4;
      price = p[brand];
```

```
         output;
      end;
      brand = .; price = 1.49; output; /* constant alternative */
      keep set brand price;
      datalines;
   1.49 1.99 1.49 1.99 1.99 1.99 2.49 1.49 1.99 1.49 1.99 1.49
   1.99 1.49 2.49 1.99 1.49 1.49 1.49 1.49 2.49 1.49 1.99 2.49
   1.49 1.49 2.49 2.49 2.49 2.49 1.49 1.49 1.49 2.49 2.49 1.99
   2.49 2.49 2.49 1.49 1.99 2.49 1.49 2.49 2.49 1.99 2.49 2.49
   2.49 1.49 1.49 1.99 1.49 1.99 1.99 1.49 2.49 1.99 1.99 1.99
   1.99 1.99 1.49 2.49 1.99 2.49 1.99 1.99 1.49 2.49 1.99 2.49
   ;

   proc transreg data=design design norestoremissing nozeroconstant;
      model class(brand / zero=none) identity(price);
      output out=coded;
      by set;
   run;

   proc print data=coded(firstobs=21 obs=25);
      var set brand &_trgind;
   run;
```

In the interest of space, only the fifth choice set is displayed in Figure 75.31.

```
                         Choice Model Coding

 Obs    set    brand    brand1    brand2    brand3    brand4    price

  21     5       1         1         0         0         0       1.49
  22     5       2         0         1         0         0       1.49
  23     5       3         0         0         1         0       1.49
  24     5       4         0         0         0         1       1.49
  25     5       .         0         0         0         0       1.49
```

Figure 75.31. The Fifth Choice Set

For the constant alternative (BRAND = .), the brand coding is a row of zeros due to the NORESTOREMISSING *o-option*, and PRICE is a constant $1.49 (instead of 0) due to the NOZEROCONSTANT *a-option*.

The data set was coded by choice set (BY set;). This is a small problem, but with very large problems, it may be necessary to restrict the number of observations that are coded at one time so that the procedure uses less time and memory. Coding by choice set is one option. When coding is performed after the data are merged in, coding by subject and choice set combinations is another option. Alternatively, you can specify DESIGN=n, where n is the number of observations to code at one time. For example, you can specify DESIGN=100 or DESIGN=1000 to process the data set in blocks of 100 or 1000 observations. Specify the NOZEROCONSTANT option to ensure that constant variables within blocks are not zeroed. When you specify DESIGN=n, or perform coding after the data are merged in, specify the dependent variable and any other variables needed for analysis as ID variables.

ANOVA Codings

This set of examples illustrates several different ways to code the same two-way ANOVA model. Figure 75.32 displays the input data set.

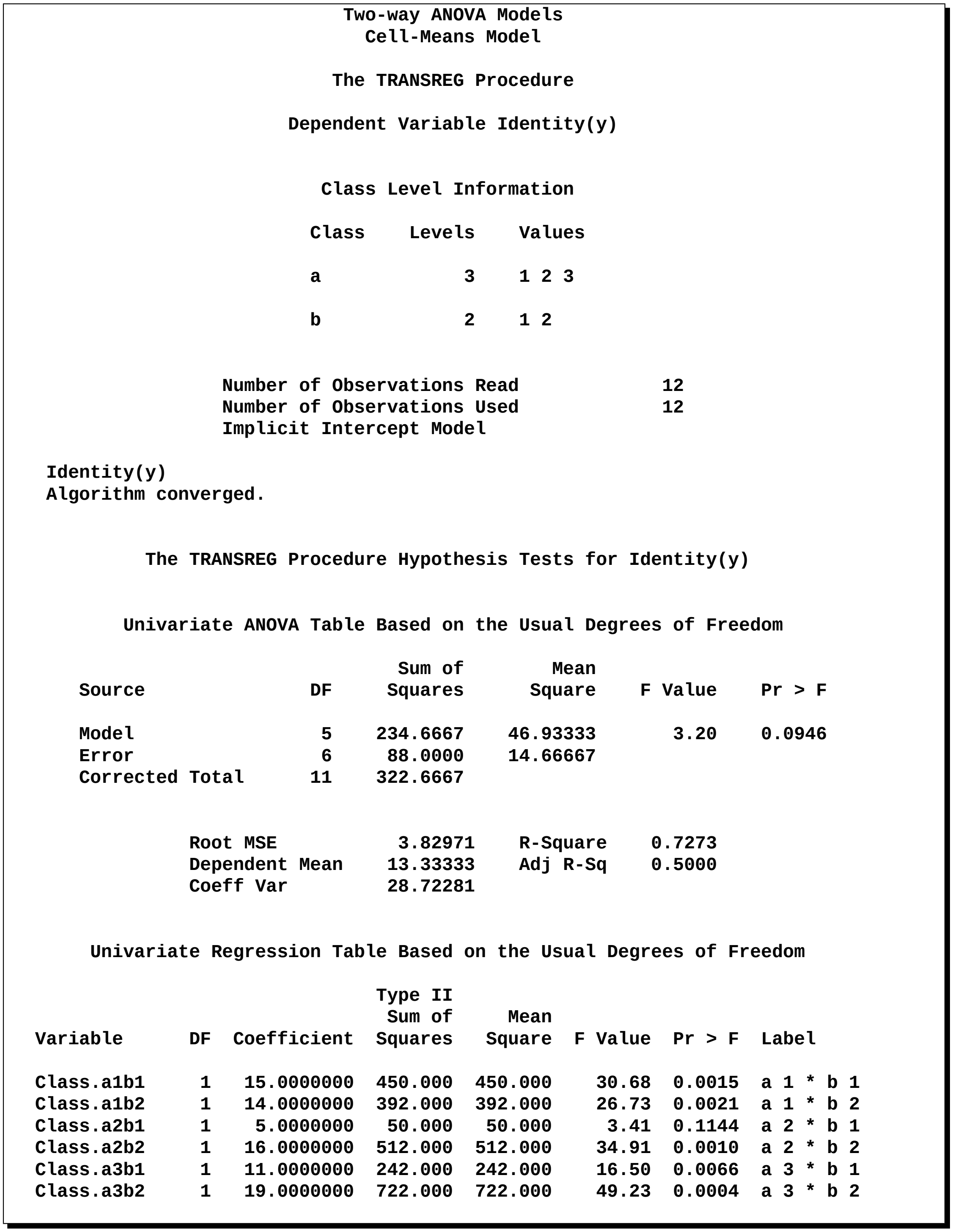
```
title 'Two-way ANOVA Models';

data x;
   input a b @@;
   do i = 1 to 2; input y @@; output; end;
   drop i;
   datalines;
1 1   16 14         1 2   15 13
2 1    1  9         2 2   12 20
3 1   14  8         3 2   18 20
;

proc print label;
run;
```

```
                         Two-way ANOVA Models

                         Obs    a    b     y

                           1    1    1    16
                           2    1    1    14
                           3    1    2    15
                           4    1    2    13
                           5    2    1     1
                           6    2    1     9
                           7    2    2    12
                           8    2    2    20
                           9    3    1    14
                          10    3    1     8
                          11    3    2    18
                          12    3    2    20
```

Figure 75.32. Input Data Set

The following statements fit a cell-means model. See Figure 75.33 and Figure 75.34.

```
proc transreg data=x ss2 short;
   title2 'Cell-Means Model';
   model identity(y) = class(a * b / zero=none);
   output replace;
run;

proc print label;
run;
```



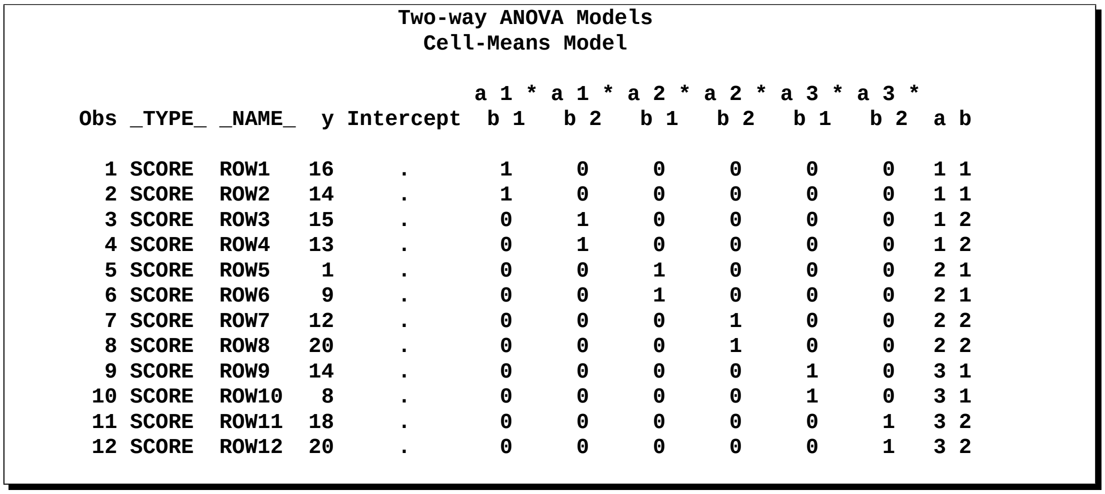
```
                            Two-way ANOVA Models
                              Cell-Means Model

                           The TRANSREG Procedure

                       Dependent Variable Identity(y)


                          Class Level Information

                         Class    Levels    Values

                         a             3    1 2 3

                         b             2    1 2


                 Number of Observations Read             12
                 Number of Observations Used             12
                 Implicit Intercept Model

 Identity(y)
 Algorithm converged.


          The TRANSREG Procedure Hypothesis Tests for Identity(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
    Source               DF     Squares      Square    F Value    Pr > F

    Model                 5    234.6667    46.93333       3.20    0.0946
    Error                 6     88.0000    14.66667
    Corrected Total      11    322.6667


              Root MSE            3.82971    R-Square    0.7273
              Dependent Mean     13.33333    Adj R-Sq    0.5000
              Coeff Var          28.72281


     Univariate Regression Table Based on the Usual Degrees of Freedom

                               Type II
                                Sum of     Mean
Variable      DF  Coefficient  Squares   Square  F Value  Pr > F  Label

Class.a1b1     1   15.0000000  450.000  450.000    30.68  0.0015  a 1 * b 1
Class.a1b2     1   14.0000000  392.000  392.000    26.73  0.0021  a 1 * b 2
Class.a2b1     1    5.0000000   50.000   50.000     3.41  0.1144  a 2 * b 1
Class.a2b2     1   16.0000000  512.000  512.000    34.91  0.0010  a 2 * b 2
Class.a3b1     1   11.0000000  242.000  242.000    16.50  0.0066  a 3 * b 1
Class.a3b2     1   19.0000000  722.000  722.000    49.23  0.0004  a 3 * b 2
```

Figure 75.33. Cell-Means Model

The parameter estimates are

$$
\begin{aligned}
\hat{\mu}_{11} &= \overline{y}_{11} = 15 \\
\hat{\mu}_{12} &= \overline{y}_{12} = 14
\end{aligned}
$$

$$\begin{aligned}
\hat{\mu}_{21} &= \overline{y}_{21} = 5 \\
\hat{\mu}_{22} &= \overline{y}_{22} = 16 \\
\hat{\mu}_{31} &= \overline{y}_{31} = 11 \\
\hat{\mu}_{32} &= \overline{y}_{32} = 19
\end{aligned}$$



```
                         Two-way ANOVA Models
                           Cell-Means Model

                               a 1 * a 1 * a 2 * a 2 * a 3 * a 3 *
Obs _TYPE_ _NAME_   y Intercept  b 1   b 2   b 1   b 2   b 1   b 2  a b

  1 SCORE  ROW1    16     .       1     0     0     0     0     0   1 1
  2 SCORE  ROW2    14     .       1     0     0     0     0     0   1 1
  3 SCORE  ROW3    15     .       0     1     0     0     0     0   1 2
  4 SCORE  ROW4    13     .       0     1     0     0     0     0   1 2
  5 SCORE  ROW5     1     .       0     0     1     0     0     0   2 1
  6 SCORE  ROW6     9     .       0     0     1     0     0     0   2 1
  7 SCORE  ROW7    12     .       0     0     0     1     0     0   2 2
  8 SCORE  ROW8    20     .       0     0     0     1     0     0   2 2
  9 SCORE  ROW9    14     .       0     0     0     0     1     0   3 1
 10 SCORE  ROW10    8     .       0     0     0     0     1     0   3 1
 11 SCORE  ROW11   18     .       0     0     0     0     0     1   3 2
 12 SCORE  ROW12   20     .       0     0     0     0     0     1   3 2
```

Figure 75.34. Cell-Means Model, Design Matrix

The following statements fit a reference cell model. The default reference level is the last cell (3,2). See Figure 75.35 and Figure 75.36.

```
proc transreg data=x ss2 short;
   title2 'Reference Cell Model, (3,2) Reference Cell';
   model identity(y) = class(a | b);
   output replace;
run;

proc print label;
run;
```

```
                            Two-way ANOVA Models
                 Reference Cell Model, (3,2) Reference Cell

                           The TRANSREG Procedure

                       Dependent Variable Identity(y)


                          Class Level Information

                         Class    Levels    Values

                         a             3    1 2 3

                         b             2    1 2


                 Number of Observations Read             12
                 Number of Observations Used             12

 Identity(y)
 Algorithm converged.


          The TRANSREG Procedure Hypothesis Tests for Identity(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
    Source               DF     Squares      Square    F Value    Pr > F

    Model                 5    234.6667    46.93333       3.20    0.0946
    Error                 6     88.0000    14.66667
    Corrected Total      11    322.6667


              Root MSE           3.82971    R-Square    0.7273
              Dependent Mean    13.33333    Adj R-Sq    0.5000
              Coeff Var         28.72281


     Univariate Regression Table Based on the Usual Degrees of Freedom

                               Type II
                                Sum of     Mean
Variable      DF  Coefficient  Squares   Square  F Value  Pr > F  Label

Intercept      1   19.0000000  722.000  722.000    49.23  0.0004  Intercept
Class.a1       1   -5.0000000   25.000   25.000     1.70  0.2395  a 1
Class.a2       1   -3.0000000    9.000    9.000     0.61  0.4632  a 2
Class.b1       1   -8.0000000   64.000   64.000     4.36  0.0817  b 1
Class.a1b1     1    9.0000000   40.500   40.500     2.76  0.1476  a 1 * b 1
Class.a2b1     1   -3.0000000    4.500    4.500     0.31  0.5997  a 2 * b 1
```

Figure 75.35. Reference Cell Model, (3, 2) Reference Cell

The parameter estimates are

$$\begin{aligned}\hat{\mu}_{32} &= \bar{y}_{32} = 19 \\ \hat{\alpha}_1 &= \bar{y}_{12} - \bar{y}_{32} = 14 - 19 = -5\end{aligned}$$

$$\begin{aligned}
\hat{\alpha}_2 &= \overline{y}_{22} - \overline{y}_{32} = 16 - 19 = -3 \\
\hat{\beta}_1 &= \overline{y}_{31} - \overline{y}_{32} = 11 - 19 = -8 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\hat{\mu}_{32} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (19 + -5 + -8) = 9 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\hat{\mu}_{32} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (19 + -3 + -8) = -3
\end{aligned}$$

The structural zeros are

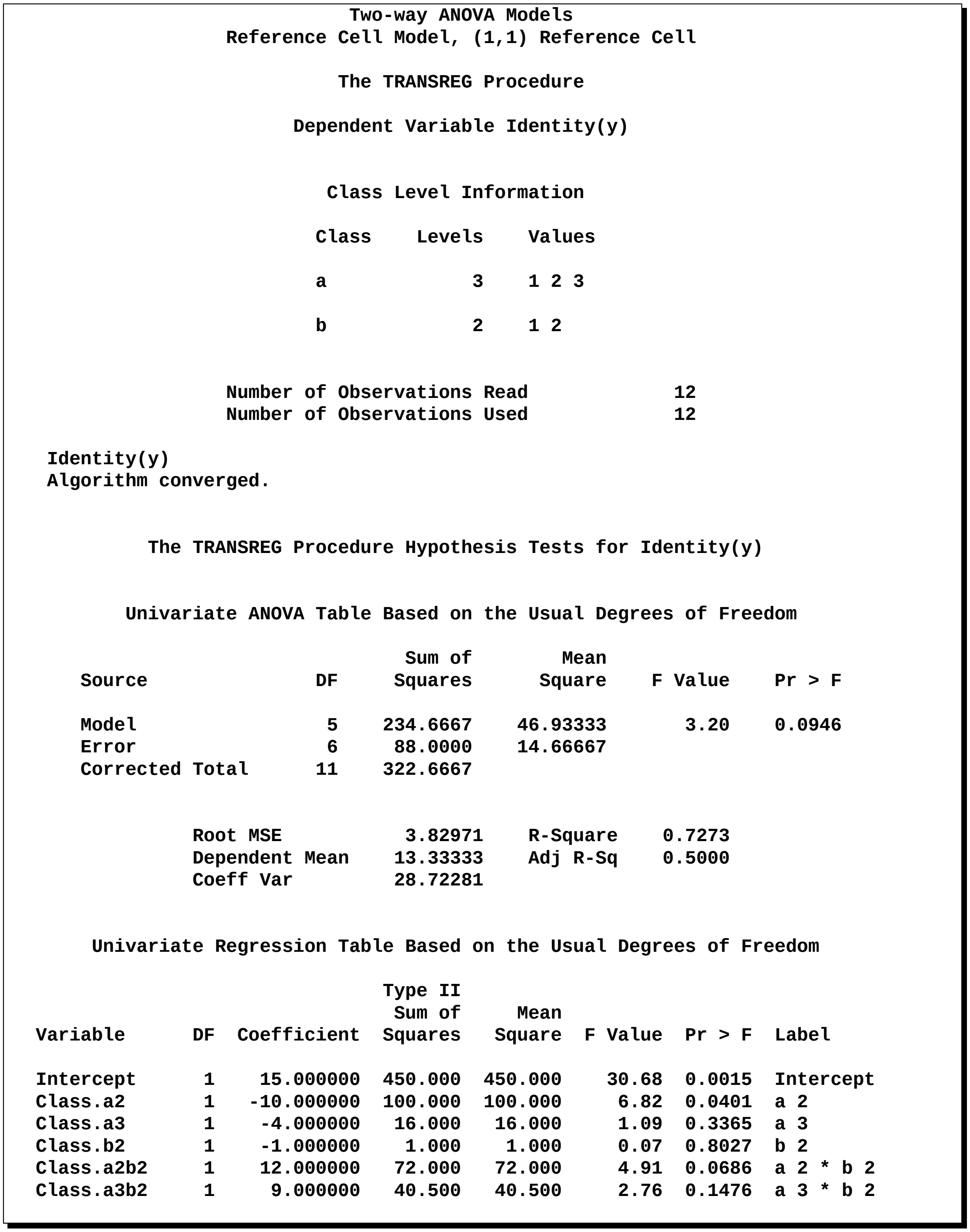
$$\alpha_3 \equiv \beta_2 \equiv \gamma_{12} \equiv \gamma_{22} \equiv \gamma_{31} \equiv \gamma_{32} \equiv 0$$

```
                                Two-way ANOVA Models
                         Reference Cell Model, (3,2) Reference Cell

                                                                 a 1 *   a 2 *
Obs   _TYPE_   _NAME_     y   Intercept   a 1   a 2   b 1     b 1     b 1     a   b

  1   SCORE    ROW1      16       1        1     0     1       1       0      1   1
  2   SCORE    ROW2      14       1        1     0     1       1       0      1   1
  3   SCORE    ROW3      15       1        1     0     0       0       0      1   2
  4   SCORE    ROW4      13       1        1     0     0       0       0      1   2
  5   SCORE    ROW5       1       1        0     1     1       0       1      2   1
  6   SCORE    ROW6       9       1        0     1     1       0       1      2   1
  7   SCORE    ROW7      12       1        0     1     0       0       0      2   2
  8   SCORE    ROW8      20       1        0     1     0       0       0      2   2
  9   SCORE    ROW9      14       1        0     0     1       0       0      3   1
 10   SCORE    ROW10      8       1        0     0     1       0       0      3   1
 11   SCORE    ROW11     18       1        0     0     0       0       0      3   2
 12   SCORE    ROW12     20       1        0     0     0       0       0      3   2
```

Figure 75.36. Reference Cell Model, (3, 2) Reference Cell, Design Matrix

The following statements fit a reference cell model, but this time the reference level is the first cell (1,1). See Figure 75.37 through Figure 75.38.

```
proc transreg data=x ss2 short;
   title2 'Reference Cell Model, (1,1) Reference Cell';
   model identity(y) = class(a | b / zero=first);
   output replace;
run;

proc print label;
run;
```



```
                         Two-way ANOVA Models
                Reference Cell Model, (1,1) Reference Cell

                         The TRANSREG Procedure

                     Dependent Variable Identity(y)


                        Class Level Information

                       Class     Levels    Values

                       a              3    1 2 3

                       b              2    1 2


                Number of Observations Read             12
                Number of Observations Used             12

 Identity(y)
 Algorithm converged.


         The TRANSREG Procedure Hypothesis Tests for Identity(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of        Mean
    Source               DF      Squares      Square    F Value    Pr > F

    Model                 5     234.6667    46.93333       3.20    0.0946
    Error                 6      88.0000    14.66667
    Corrected Total      11     322.6667


             Root MSE            3.82971    R-Square    0.7273
             Dependent Mean     13.33333    Adj R-Sq    0.5000
             Coeff Var          28.72281


      Univariate Regression Table Based on the Usual Degrees of Freedom

                                Type II
                                 Sum of     Mean
Variable      DF  Coefficient   Squares   Square  F Value  Pr > F  Label

Intercept      1    15.000000   450.000  450.000    30.68  0.0015  Intercept
Class.a2       1   -10.000000   100.000  100.000     6.82  0.0401  a 2
Class.a3       1    -4.000000    16.000   16.000     1.09  0.3365  a 3
Class.b2       1    -1.000000     1.000    1.000     0.07  0.8027  b 2
Class.a2b2     1    12.000000    72.000   72.000     4.91  0.0686  a 2 * b 2
Class.a3b2     1     9.000000    40.500   40.500     2.76  0.1476  a 3 * b 2
```

Figure 75.37. Reference Cell Model, (1, 1) Reference Cell

The parameter estimates are

$$\begin{aligned}\hat{\mu}_{11} &= \overline{y}_{11} = 15 \\ \hat{\alpha}_2 &= \overline{y}_{21} - \overline{y}_{11} = 5 - 15 = -10\end{aligned}$$

$$\begin{aligned}
\hat{\alpha}_3 &= \bar{y}_{31} - \bar{y}_{11} = 11 - 15 = -4 \\
\hat{\beta}_2 &= \bar{y}_{12} - \bar{y}_{11} = 14 - 15 = -1 \\
\hat{\gamma}_{22} &= \bar{y}_{22} - (\hat{\mu}_{11} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (15 + -10 + -1) = 12 \\
\hat{\gamma}_{32} &= \bar{y}_{32} - (\hat{\mu}_{11} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (15 + -4 + -1) = 9
\end{aligned}$$

The structural zeros are

$$\alpha_1 \equiv \beta_1 \equiv \gamma_{11} \equiv \gamma_{12} \equiv \gamma_{21} \equiv \gamma_{31} \equiv 0$$

```
                            Two-way ANOVA Models
                  Reference Cell Model, (1,1) Reference Cell

                                                             a 2 *    a 3 *
Obs    _TYPE_    _NAME_     y    Intercept    a 2    a 3    b 2     b 2      b 2     a    b

  1    SCORE     ROW1      16        1         0      0      0       0        0      1    1
  2    SCORE     ROW2      14        1         0      0      0       0        0      1    1
  3    SCORE     ROW3      15        1         0      0      1       0        0      1    2
  4    SCORE     ROW4      13        1         0      0      1       0        0      1    2
  5    SCORE     ROW5       1        1         1      0      0       0        0      2    1
  6    SCORE     ROW6       9        1         1      0      0       0        0      2    1
  7    SCORE     ROW7      12        1         1      0      1       1        0      2    2
  8    SCORE     ROW8      20        1         1      0      1       1        0      2    2
  9    SCORE     ROW9      14        1         0      1      0       0        0      3    1
 10    SCORE     ROW10      8        1         0      1      0       0        0      3    1
 11    SCORE     ROW11     18        1         0      1      1       0        1      3    2
 12    SCORE     ROW12     20        1         0      1      1       0        1      3    2
```

Figure 75.38. Reference Cell Model, (1, 1) Reference Cell, Design Matrix

The following statements fit a deviations-from-means model. The default reference level is the last cell (3,2). This coding is also called effects coding. See Figure 75.39 and Figure 75.40.

```
proc transreg data=x ss2 short;
   title2 'Deviations From Means, (3,2) Reference Cell';
   model identity(y) = class(a | b / deviations);
   output replace;
run;

proc print label;
run;
```

```
                              Two-way ANOVA Models
                     Deviations From Means, (3,2) Reference Cell

                             The TRANSREG Procedure

                         Dependent Variable Identity(y)


                            Class Level Information

                           Class    Levels    Values

                           a             3    1 2 3

                           b             2    1 2


                    Number of Observations Read             12
                    Number of Observations Used             12

 Identity(y)
 Algorithm converged.


          The TRANSREG Procedure Hypothesis Tests for Identity(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of        Mean
    Source              DF       Squares      Square    F Value    Pr > F

    Model                5      234.6667    46.93333       3.20    0.0946
    Error                6       88.0000    14.66667
    Corrected Total     11      322.6667


              Root MSE           3.82971    R-Square    0.7273
              Dependent Mean    13.33333    Adj R-Sq    0.5000
              Coeff Var         28.72281


       Univariate Regression Table Based on the Usual Degrees of Freedom

                                Type II
                                 Sum of     Mean
Variable      DF  Coefficient  Squares    Square  F Value  Pr > F  Label

Intercept      1   13.3333333  2133.33   2133.33   145.45  <.0001  Intercept
Class.a1       1    1.1666667     8.17      8.17     0.56  0.4837  a 1
Class.a2       1   -2.8333333    48.17     48.17     3.28  0.1199  a 2
Class.b1       1   -3.0000000   108.00    108.00     7.36  0.0349  b 1
Class.a1b1     1    3.5000000    73.50     73.50     5.01  0.0665  a 1 * b 1
Class.a2b1     1   -2.5000000    37.50     37.50     2.56  0.1609  a 2 * b 1
```

Figure 75.39. Deviations-From-Means Model, (3, 2) Reference Cell

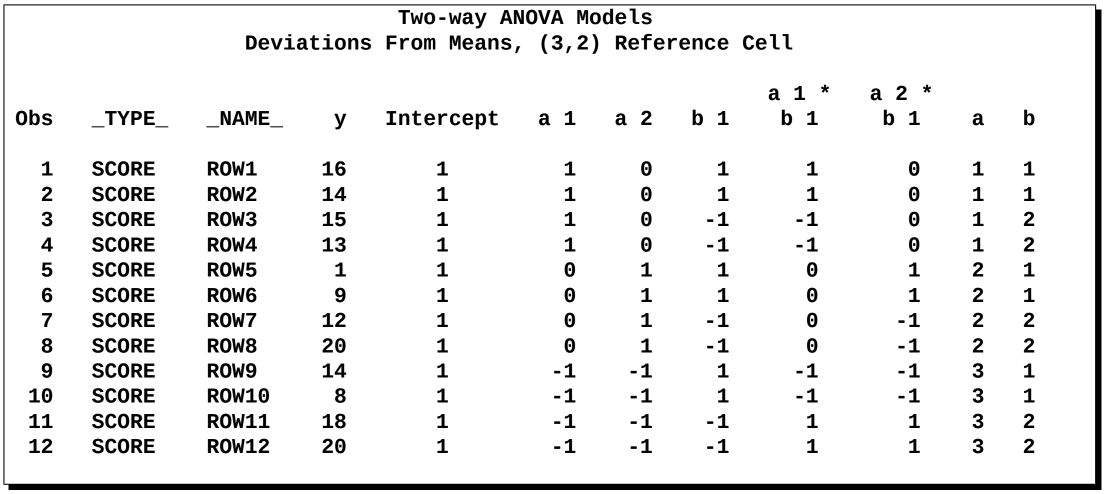
The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.33333 \\
\hat{\alpha}_1 &= (\overline{y}_{11} + \overline{y}_{12})/2 - \overline{y} = (15 + 14)/2 - 13.33333 = 1.16667
\end{aligned}
$$

$$
\begin{aligned}
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5 + 16)/2 - 13.33333 = -2.83333 \\
\hat{\beta}_1 &= (\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31})/3 - \overline{y} = (15 + 5 + 11)/3 - 13.33333 = -3 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (13.33333 + 1.16667 + -3) = 3.5 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (13.33333 + -2.83333 + -3) = -2.5
\end{aligned}
$$

The structural zeros are

$$\alpha_3 \equiv \beta_2 \equiv \gamma_{12} \equiv \gamma_{22} \equiv \gamma_{31} \equiv \gamma_{32} \equiv 0$$



```
                                  Two-way ANOVA Models
                        Deviations From Means, (3,2) Reference Cell

                                                                  a 1 *    a 2 *
Obs    _TYPE_    _NAME_     y    Intercept    a 1    a 2    b 1    b 1      b 1     a    b

  1    SCORE     ROW1      16        1          1      0      1      1        0     1    1
  2    SCORE     ROW2      14        1          1      0      1      1        0     1    1
  3    SCORE     ROW3      15        1          1      0     -1     -1        0     1    2
  4    SCORE     ROW4      13        1          1      0     -1     -1        0     1    2
  5    SCORE     ROW5       1        1          0      1      1      0        1     2    1
  6    SCORE     ROW6       9        1          0      1      1      0        1     2    1
  7    SCORE     ROW7      12        1          0      1     -1      0       -1     2    2
  8    SCORE     ROW8      20        1          0      1     -1      0       -1     2    2
  9    SCORE     ROW9      14        1         -1     -1      1     -1       -1     3    1
 10    SCORE     ROW10      8        1         -1     -1      1     -1       -1     3    1
 11    SCORE     ROW11     18        1         -1     -1     -1      1        1     3    2
 12    SCORE     ROW12     20        1         -1     -1     -1      1        1     3    2
```

Figure 75.40. Deviations-From-Means Model, (3, 2) Reference Cell, Design Matrix

The following statements fit a deviations-from-means model, but this time the reference level is the first cell (1,1). This coding is also called effects coding. See Figure 75.41 through Figure 75.42.

```
proc transreg data=x ss2 short;
   title2 'Deviations From Means, (1,1) Reference Cell';
   model identity(y) = class(a | b / deviations zero=first);
   output replace;
run;

proc print label;
run;
```

```
                                 Two-way ANOVA Models
                        Deviations From Means, (1,1) Reference Cell

                                The TRANSREG Procedure

                            Dependent Variable Identity(y)


                               Class Level Information

                              Class    Levels    Values

                              a             3    1 2 3

                              b             2    1 2


                      Number of Observations Read             12
                      Number of Observations Used             12

  Identity(y)
  Algorithm converged.


            The TRANSREG Procedure Hypothesis Tests for Identity(y)


           Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                      Sum of        Mean
     Source              DF          Squares      Square    F Value    Pr > F

     Model                5         234.6667    46.93333       3.20    0.0946
     Error                6          88.0000    14.66667
     Corrected Total     11         322.6667


               Root MSE           3.82971    R-Square    0.7273
               Dependent Mean    13.33333    Adj R-Sq    0.5000
               Coeff Var         28.72281


         Univariate Regression Table Based on the Usual Degrees of Freedom

                                   Type II
                                    Sum of     Mean
Variable      DF  Coefficient     Squares   Square  F Value  Pr > F  Label

Intercept      1   13.3333333     2133.33  2133.33   145.45  <.0001  Intercept
Class.a2       1   -2.8333333       48.17    48.17     3.28  0.1199  a 2
Class.a3       1    1.6666667       16.67    16.67     1.14  0.3274  a 3
Class.b2       1    3.0000000      108.00   108.00     7.36  0.0349  b 2
Class.a2b2     1    2.5000000       37.50    37.50     2.56  0.1609  a 2 * b 2
Class.a3b2     1    1.0000000        6.00     6.00     0.41  0.5461  a 3 * b 2
```

Figure 75.41. Deviations-From-Means Model, (1, 1) Reference Cell

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.33333 \\
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5 + 16)/2 - 13.33333 = -2.8333
\end{aligned}
$$

$$\begin{aligned}
\hat{\alpha}_3 &= (\overline{y}_{31} + \overline{y}_{32})/2 - \overline{y} = (11 + 19)/2 - 13.33333 = 1.66667 \\
\hat{\beta}_2 &= (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32})/3 - \overline{y} = (14 + 16 + 19)/3 - 13.33333 = 3 \\
\hat{\gamma}_{22} &= \overline{y}_{22} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (13.33333 + -2.8333 + 3) = 2.5 \\
\hat{\gamma}_{32} &= \overline{y}_{32} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (13.33333 + 1.66667 + 3) = 1
\end{aligned}$$

The structural zeros are

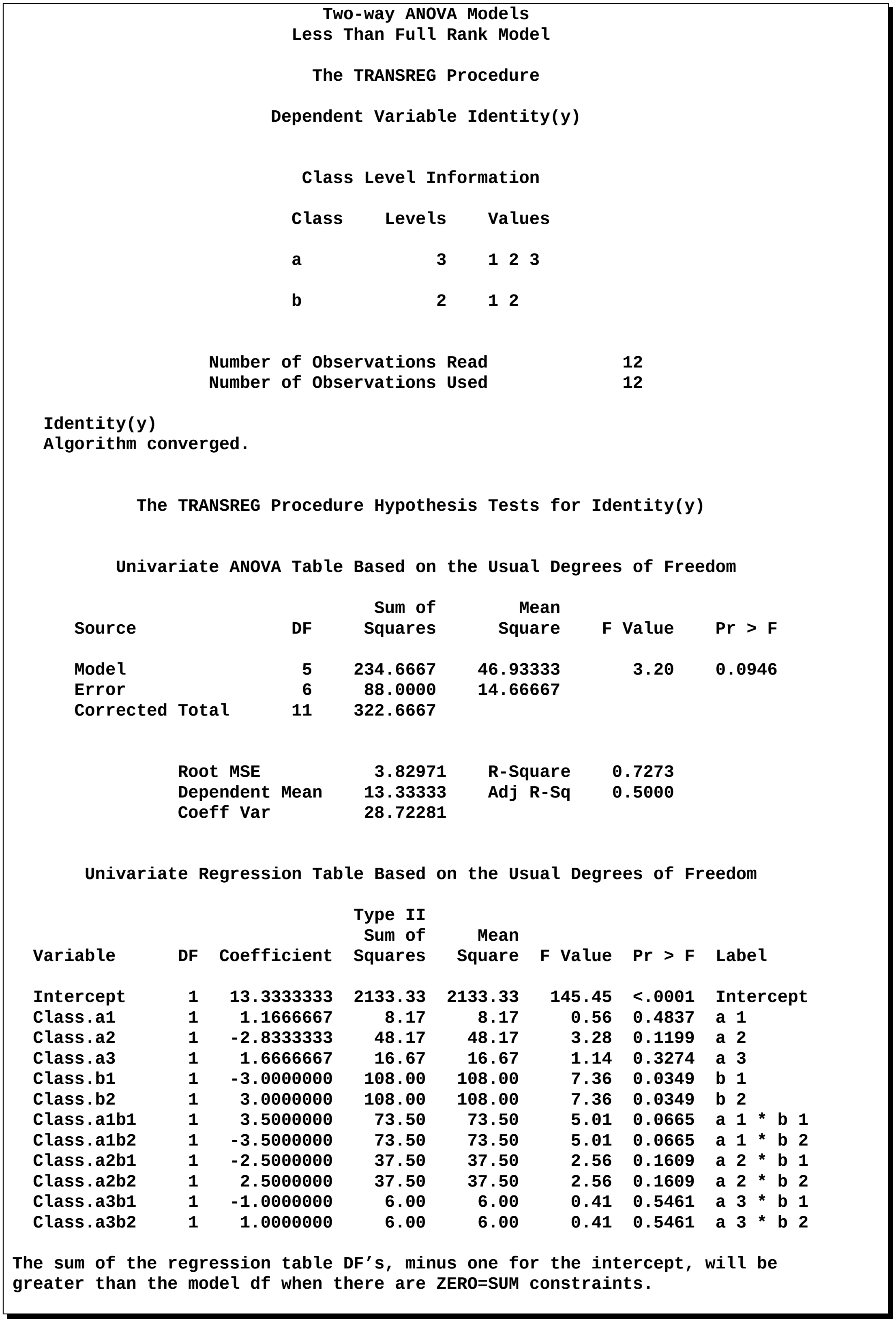
$$\alpha_1 \equiv \beta_1 \equiv \gamma_{11} \equiv \gamma_{12} \equiv \gamma_{21} \equiv \gamma_{31} \equiv 0$$

```
                             Two-way ANOVA Models
                    Deviations From Means, (1,1) Reference Cell

                                                            a 2 *   a 3 *
Obs   _TYPE_   _NAME_    y   Intercept   a 2   a 3   b 2     b 2     b 2    a   b

  1   SCORE    ROW1     16       1        -1    -1    -1      1       1     1   1
  2   SCORE    ROW2     14       1        -1    -1    -1      1       1     1   1
  3   SCORE    ROW3     15       1        -1    -1     1     -1      -1     1   2
  4   SCORE    ROW4     13       1        -1    -1     1     -1      -1     1   2
  5   SCORE    ROW5      1       1         1     0    -1     -1       0     2   1
  6   SCORE    ROW6      9       1         1     0    -1     -1       0     2   1
  7   SCORE    ROW7     12       1         1     0     1      1       0     2   2
  8   SCORE    ROW8     20       1         1     0     1      1       0     2   2
  9   SCORE    ROW9     14       1         0     1    -1      0      -1     3   1
 10   SCORE    ROW10     8       1         0     1    -1      0      -1     3   1
 11   SCORE    ROW11    18       1         0     1     1      0       1     3   2
 12   SCORE    ROW12    20       1         0     1     1      0       1     3   2
```

Figure 75.42. Deviations-From-Means Model, (1, 1) Reference Cell, Design Matrix

The following statements fit a less-than-full-rank model. The parameter estimates are constrained to sum to zero within each effect. See Figure 75.43 and Figure 75.44.

```
proc transreg data=x ss2 short;
   title2 'Less Than Full Rank Model';
   model identity(y) = class(a | b / zero=sum);
   output replace;
run;

proc print label;
run;
```



```
                             Two-way ANOVA Models
                          Less Than Full Rank Model

                            The TRANSREG Procedure

                        Dependent Variable Identity(y)


                           Class Level Information

                          Class    Levels    Values

                          a             3    1 2 3

                          b             2    1 2


                  Number of Observations Read             12
                  Number of Observations Used             12

   Identity(y)
   Algorithm converged.


          The TRANSREG Procedure Hypothesis Tests for Identity(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                   Sum of        Mean
      Source               DF     Squares      Square    F Value    Pr > F

      Model                 5    234.6667    46.93333       3.20    0.0946
      Error                 6     88.0000    14.66667
      Corrected Total      11    322.6667


              Root MSE           3.82971    R-Square    0.7273
              Dependent Mean    13.33333    Adj R-Sq    0.5000
              Coeff Var         28.72281


       Univariate Regression Table Based on the Usual Degrees of Freedom

                                 Type II
                                  Sum of     Mean
  Variable      DF  Coefficient  Squares   Square  F Value  Pr > F  Label

  Intercept      1   13.3333333  2133.33  2133.33   145.45  <.0001  Intercept
  Class.a1       1    1.1666667     8.17     8.17     0.56  0.4837  a 1
  Class.a2       1   -2.8333333    48.17    48.17     3.28  0.1199  a 2
  Class.a3       1    1.6666667    16.67    16.67     1.14  0.3274  a 3
  Class.b1       1   -3.0000000   108.00   108.00     7.36  0.0349  b 1
  Class.b2       1    3.0000000   108.00   108.00     7.36  0.0349  b 2
  Class.a1b1     1    3.5000000    73.50    73.50     5.01  0.0665  a 1 * b 1
  Class.a1b2     1   -3.5000000    73.50    73.50     5.01  0.0665  a 1 * b 2
  Class.a2b1     1   -2.5000000    37.50    37.50     2.56  0.1609  a 2 * b 1
  Class.a2b2     1    2.5000000    37.50    37.50     2.56  0.1609  a 2 * b 2
  Class.a3b1     1   -1.0000000     6.00     6.00     0.41  0.5461  a 3 * b 1
  Class.a3b2     1    1.0000000     6.00     6.00     0.41  0.5461  a 3 * b 2

The sum of the regression table DF's, minus one for the intercept, will be
greater than the model df when there are ZERO=SUM constraints.
```

Figure 75.43. Less-Than-Full-Rank Model

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.33333 \\
\hat{\alpha}_1 &= (\overline{y}_{11} + \overline{y}_{12})/2 - \overline{y} = (15 + 14)/2 - 13.33333 = 1.16667 \\
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5 + 16)/2 - 13.33333 = -2.8333 \\
\hat{\alpha}_3 &= (\overline{y}_{31} + \overline{y}_{32})/2 - \overline{y} = (11 + 19)/2 - 13.33333 = 1.66667 \\
\hat{\beta}_1 &= (\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31})/3 - \overline{y} = (15 + 5 + 11)/3 - 13.33333 = -3 \\
\hat{\beta}_2 &= (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32})/3 - \overline{y} = (14 + 16 + 19)/3 - 13.33333 = 3 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (13.33333 + 1.16667 + -3) = 3.5 \\
\hat{\gamma}_{12} &= \overline{y}_{12} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_2) = 14 - (13.33333 + 1.16667 + 3) = -3.5 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (13.33333 + -2.83333 + -3) = -2.5 \\
\hat{\gamma}_{22} &= \overline{y}_{22} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (13.33333 + -2.8333 + 3) = 2.5 \\
\hat{\gamma}_{31} &= \overline{y}_{31} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_1) = 11 - (13.33333 + 1.66667 + -3) = -1 \\
\hat{\gamma}_{32} &= \overline{y}_{32} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (13.33333 + 1.66667 + 3) = 1
\end{aligned}
$$

The constraints are

$$\alpha_1 + \alpha_2 + \alpha_3 \equiv \beta_1 + \beta_2 \equiv 0$$

$$\gamma_{11} + \gamma_{12} \equiv \gamma_{21} + \gamma_{22} \equiv \gamma_{31} + \gamma_{32} \equiv \gamma_{11} + \gamma_{21} + \gamma_{31} \equiv \gamma_{12} + \gamma_{22} + \gamma_{32} \equiv 0$$

```
                              Two-way ANOVA Models
                           Less Than Full Rank Model


   Obs    _TYPE_    _NAME_     y    Intercept    a 1    a 2    a 3    b 1

     1    SCORE     ROW1      16        1         1      0      0      1
     2    SCORE     ROW2      14        1         1      0      0      1
     3    SCORE     ROW3      15        1         1      0      0      0
     4    SCORE     ROW4      13        1         1      0      0      0
     5    SCORE     ROW5       1        1         0      1      0      1
     6    SCORE     ROW6       9        1         0      1      0      1
     7    SCORE     ROW7      12        1         0      1      0      0
     8    SCORE     ROW8      20        1         0      1      0      0
     9    SCORE     ROW9      14        1         0      0      1      1
    10    SCORE     ROW10      8        1         0      0      1      1
    11    SCORE     ROW11     18        1         0      0      1      0
    12    SCORE     ROW12     20        1         0      0      1      0

                  a 1 *    a 1 *    a 2 *    a 2 *    a 3 *    a 3 *
   Obs    b 2      b 1      b 2      b 1      b 2      b 1      b 2      a    b

     1     0        1        0        0        0        0        0       1    1
     2     0        1        0        0        0        0        0       1    1
     3     1        0        1        0        0        0        0       1    2
     4     1        0        1        0        0        0        0       1    2
     5     0        0        0        1        0        0        0       2    1
     6     0        0        0        1        0        0        0       2    1
     7     1        0        0        0        1        0        0       2    2
     8     1        0        0        0        1        0        0       2    2
     9     0        0        0        0        0        1        0       3    1
    10     0        0        0        0        0        1        0       3    1
    11     1        0        0        0        0        0        1       3    2
    12     1        0        0        0        0        0        1       3    2
```

Figure 75.44. Less-Than-Full-Rank Model, Design Matrix

Centering

You can use transformation options to center and standardize the variables in several ways. For example, this MODEL statement creates three independent variables, x, x^2, and x^3.

```
   model identity(y) = pspline(x);
```

The variables are not centered.

When the CENTER *t-option* is specified, the three independent variables are $x - \bar{x}$, $(x - \bar{x})^2$, and $(x - \bar{x})^3$.

```
   model identity(y) = pspline(x / center);
```

Since operations such as squaring occur after the centering, the resulting variables will not always be centered. The CENTER *t-option* is particularly useful with polynomials since centering before squaring and cubing can help reduce collinearity and numerical problems. For example, if one of your variables is year, with values all greater than 1900. Squaring and cubing without centering first will create variables that are all essentially perfectly correlated.

When the TSTANDARD=CENTER *t-option* is specified, the three independent variables are $x - \bar{x}$, $x^2 - \overline{x^2}$, and $x^3 - \overline{x^3}$.

```
   model identity(y) = pspline(x / tstandard=center);
```

In this case, the variables are squared and cubed and then centered.

Displayed Output

The display options control the amount of displayed output. The displayed output can contain

- an iteration history and convergence status table, by default
- an ANOVA table when the TEST, SS2, or UTILITIES *a-option* is specified
- a regression table when the SS2 *a-option* is specified
- conjoint analysis part-worth utilities when the UTILITIES *a-option* is specified
- model details when the DETAIL *a-option* is specified
- a multivariate ANOVA table when the dependent variable is transformed and the TEST or SS2 *a-option* is specified
- a multivariate regression table when the dependent variable is transformed and it is specified
- liberal and conservative ANOVA, multivariate ANOVA, regression, and multivariate regression tables when there are MONOTONE, UNTIE, or MSPLINE transformations and the TEST or SS2 *a-option* is specified

ODS Table Names

PROC TRANSREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 75.7. ODS Tables Produced in PROC TRANSREG

ODS Table Name	**Description**	**Statement**	**Option**
NObs	ANOVA	MODEL/PROC	TEST/SS2
ClassLevels	ANOVA	MODEL/PROC	TEST/SS2
ANOVA	ANOVA	MODEL/PROC	TEST/SS2
LiberalANOVA	ANOVA, *1	MODEL/PROC	TEST/SS2
ConservANOVA	ANOVA, *1	MODEL/PROC	TEST/SS2
FitStatistics	Fit statistics like R-square	MODEL/PROC	TEST/SS2

Table 75.7. (continued)

ODS Table Name	Description	Statement	Option
LiberalFitStatistics	Fit statistics, *1	MODEL/PROC	TEST/SS2
ConservFitStatistics	Fit statistics, *1	MODEL/PROC	TEST/SS2
MVANOVA	Multivariate ANOVA, *2	MODEL/PROC	TEST/SS2
LiberalMVANOVA	Multivariate ANOVA, *1, *2	MODEL/PROC	TEST/SS2
ConservMVANOVA	Multivariate ANOVA, *1, *2	MODEL/PROC	TEST/SS2
Coef	Regression results	MODEL/PROC	SS2
LiberalCoef	Regression results, *1	MODEL/PROC	SS2
ConservCoef	Regression results, *1	MODEL/PROC	SS2
MVCoef	Multivariate regression results, *2	MODEL/PROC	SS2
LiberalMVCoef	Multivariate regression results, *1, *2	MODEL/PROC	SS2
ConservMVCoef	Multivariate regression results, *1, *2	MODEL/PROC	SS2
Utilities	Conjoint Analysis Utilities	MODEL/PROC	UTILITY
LiberalUtilities	Conjoint Analysis Utilities, *1	MODEL/PROC	UTILITY
ConservUtilities	Conjoint Analysis Utilities, *1	MODEL/PROC	UTILITY
BoxCox	Box-Cox Transformation Results	MODEL	BOXCOX
Equation	Linear Dependency Equation		less-than-full-rank model
Details	Model Details	MODEL/PROC	DETAIL
Univariate	Univariate Iteration History	MODEL/PROC	METHOD=UNIVARIATE
MORALS	MORALS Iteration History	MODEL/PROC	METHOD=MORALS
CANALS	CANALS Iteration History	MODEL/PROC	METHOD=CANALS
Redundancy	Redundancy Iteration History	MODEL/PROC	METHOD=REDUNDANCY
TestIterations	Hypothesis Test Iterations Iteration History	MODEL/PROC	SS2
ConvergenceStatus	Convergence Status		default
Footnotes	Iteration History Footnotes		default
SplineCoef	Spline coefficients	MODEL	SPLINE/MSPLINE

*1. Liberal and conservative test tables are produced when a MONOTONE, UNTIE, or MSPLINE, transformation is requested.

*2. Multivariate tables are produced when the dependent variable is iteratively transformed.

Examples

Example 75.1. Using Splines and Knots

This example illustrates some properties of splines. *Splines* are curves, which are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n - 1$ derivatives that agree at the points where they join. The abscissa values of the join points are called *knots*. The term "spline" is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Knots give the curve freedom to bend to more closely follow the data. Refer to Smith (1979) for an excellent introduction to splines.

In this example, an artificial data set is created with a variable Y that is a discontinuous function of X. See the first plot in Output 75.1.7. Notice that the function has four unconnected parts, each of which is a curve. Notice too that there is an overall quadratic trend, that is, ignoring the shapes of the individual curves, at first the Y values tend to decrease as X increases, then Y values tend to increase.

The first PROC TRANSREG analysis fits a linear regression model. The predicted values of Y given X are output and plotted to form the linear regression line. The R^2 for the linear regression is 0.10061, and it can be seen from the second plot in Output 75.1.7 that the linear regression model is not appropriate for these data. The following statements create the data set and perform the first PROC TRANSREG analysis. These statements produce Output 75.1.1.

```
title 'An Illustration of Splines and Knots';

* Create in Y a discontinuous function of X.
*
* Store copies of X in V1-V7 for use in PROC GPLOT.
* These variables are only necessary so that each
* plot can have its own x-axis label while putting
* four plots on a page.;

data A;
   array V[7] V1-V7;
   X=-0.000001;
   do I=0 to 199;
      if mod(I,50)=0 then do;
         C=((X/2)-5)**2;
         if I=150 then C=C+5;
         Y=C;
         end;
      X=X+0.1;
      Y=Y-sin(X-C);
      do J=1 to 7;
         V[J]=X;
         end;
      output;
      end;
run;

* Each of the PROC TRANSREG steps fits a
* different spline model to the data set created
* previously.  The TRANSREG steps build up a data set with
* various regression functions.  All of the functions
* are then plotted with the final PROC GPLOT step.
*
* The OUTPUT statements add new predicted values
* variables to the data set, while the ID statements
* save all of the previously created variables that
* are needed for the plots.;

proc transreg data=A;
   model identity(Y) = identity(X);
   title2 'A Linear Regression Function';
   output out=A pprefix=Linear;
   id V1-V7;
run;
```

Output 75.1.1. Fitting a Linear Regression Model with PROC TRANSREG

```
                   An Illustration of Splines and Knots
                       A Linear Regression Function

                          The TRANSREG Procedure

     TRANSREG Univariate Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.00000    0.00000     0.10061                 Converged

Algorithm converged.
```

The second PROC TRANSREG analysis finds a degree two spline transformation with no knots, which is a quadratic polynomial. The spline is a weighted sum of a single constant, a single straight line, and a single quadratic curve. The R^2 increases from 0.10061, which is the linear fit value from before, to 0.40720. It can be seen from the third plot in Output 75.1.7 that the quadratic regression function does not fit any of the individual curves well, but it does follow the overall trend in the data. Since the overall trend is quadratic, a degree three spline with no knots (not shown) increases R^2 by only a small amount. The following statements perform the quadratic analysis and produce Output 75.1.2.

```
proc transreg data=A;
```

```
   model identity(Y)=spline(X / degree=2);
   title2 'A Quadratic Polynomial Regression Function';
   output out=A pprefix=Quad;
   id V1-V7 LinearY;
run;
```

Output 75.1.2. Fitting a Quadratic Polynomial

```
                     An Illustration of Splines and Knots
                  A Quadratic Polynomial Regression Function

                            The TRANSREG Procedure

         TRANSREG MORALS Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.82127    2.77121     0.10061
        2    0.00000    0.00000     0.40720      0.30659    Converged

Algorithm converged.
```

The next step uses the default degree of three, for a piecewise cubic polynomial, and requests knots at the known break points, X=5, 10, and 15. This requests a spline that is continuous, has continuous first and second derivatives, and has a third derivative that is discontinuous at 5, 10, and 15. The spline is a weighted sum of a single constant, a single straight line, a single quadratic curve, a cubic curve for the portion of X less than 5, a different cubic curve for the portion of X between 5 and 10, a different cubic curve for the portion of X between 10 and 15, and another cubic curve for the portion of X greater than 15. The new R^2 is 0.61730, and it can be seen from the fourth plot (in Output 75.1.7) that the spline is less smooth than the quadratic polynomial and it follows the data more closely than the quadratic polynomial. The following statements perform this analysis and produce Output 75.1.3:

```
proc transreg data=A;
   model identity(Y) = spline(X / knots=5 10 15);
   title2 'A Cubic Spline Regression Function';
   title3 'The Third Derivative is Discontinuous at X=5, 10, 15';
   output out=A pprefix=Cub1;
   id V1-V7 LinearY QuadY;
run;
```

Output 75.1.3. Fitting a Piecewise Cubic Polynomial

```
                     An Illustration of Splines and Knots
                      A Cubic Spline Regression Function
                 The Third Derivative is Discontinuous at X=5, 10, 15

                            The TRANSREG Procedure

           TRANSREG MORALS Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.85367    3.88449     0.10061
        2    0.00000    0.00000     0.61730      0.51670    Converged

Algorithm converged.
```

The same model could be fit with a DATA step and PROC REG, as follows. (The output from the following code is not displayed.)

```
data B;             /* A is the data set used for transreg */
   set a(keep=X Y);
   X1=X;                        /* X                      */
   X2=X**2;                     /* X squared              */
   X3=X**3;                     /* X cubed                */
   X4=(X> 5)*((X-5)**3);        /* change in X**3 after  5 */
   X5=(X>10)*((X-10)**3);       /* change in X**3 after 10 */
   X6=(X>15)*((X-15)**3);       /* change in X**3 after 15 */
run;

proc reg;
   model Y=X1-X6;
run;
```

In the next step each knot is repeated three times, so the first, second, and third derivatives are discontinuous at X=5, 10, and 15, but the spline is required to be continuous at the knots. The spline is a weighted sum of the following.

- a single constant
- a line for the portion of X less than 5
- a quadratic curve for the portion of X less than 5
- a cubic curve for the portion of X less than 5
- a different line for the portion of X between 5 and 10
- a different quadratic curve for the portion of X between 5 and 10
- a different cubic curve for the portion of X between 5 and 10
- a different line for the portion of X between 10 and 15
- a different quadratic curve for the portion of X between 10 and 15
- a different cubic curve for the portion of X between 10 and 15

- another line for the portion of X greater than 15
- another quadratic curve for the portion of X greater than 15
- and another cubic curve for the portion of X greater than 15

The spline is continuous since there is not a separate constant in the formula for the spline for each knot. Now the R^2 is 0.95542, and the spline closely follows the data, except at the knots. The following statements perform this analysis and produce Output 75.1.4:

```
proc transreg data=A;
   model identity(y) = spline(x / knots=5 5 5 10 10 10 15 15 15);
   title3 'First - Third Derivatives Discontinuous at X=5, 10, 15';
   output out=A pprefix=Cub3;
   id V1-V7 LinearY QuadY Cub1Y;
run;
```

Output 75.1.4. Piecewise Polynomial with Discontinuous Derivatives

```
                       An Illustration of Splines and Knots
                        A Cubic Spline Regression Function
              First - Third Derivatives Discontinuous at X=5, 10, 15

                              The TRANSREG Procedure

            TRANSREG MORALS Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                   Criterion
      Number     Change     Change    R-Square          Change    Note
   -------------------------------------------------------------------------
           1    0.92492    3.50038     0.10061
           2    0.00000    0.00000     0.95542         0.85481    Converged

   Algorithm converged.
```

The same model could be fit with a DATA step and PROC REG, as follows. (The output from the following code is not displayed.)

```
data B;
   set a(keep=X Y);
   X1=X;                          /* X                      */
   X2=X**2;                       /* X squared              */
   X3=X**3;                       /* X cubed                */
   X4=(X>5)   * (X- 5);           /* change in X    after  5 */
   X5=(X>10)  * (X-10);           /* change in X    after 10 */
   X6=(X>15)  * (X-15);           /* change in X    after 15 */
   X7=(X>5)   * ((X-5)**2);       /* change in X**2 after  5 */
   X8=(X>10)  * ((X-10)**2);      /* change in X**2 after 10 */
   X9=(X>15)  * ((X-15)**2);      /* change in X**2 after 15 */
   X10=(X>5)  * ((X-5)**3);       /* change in X**3 after  5 */
   X11=(X>10) * ((X-10)**3);      /* change in X**3 after 10 */
   X12=(X>15) * ((X-15)**3);    /* change in X**3 after 15 */
run;

proc reg;
   model Y=X1-X12;
run;
```

When the knots are repeated four times in the next step, the spline function is discontinuous at the knots and follows the data even more closely, with an R^2 of 0.99254. In this step, each separate curve is approximated by a cubic polynomial (with no knots within the separate polynomials). The following statements perform this analysis and produce Output 75.1.5:

```
proc transreg data=A;
   model identity(Y) = spline(X / knots=5 5 5 5 10 10 10 10 15 15 15 15);
   title3 'Discontinuous Function and Derivatives';
   output out=A pprefix=Cub4;
   id V1-V7 LinearY QuadY Cub1Y Cub3Y;
run;
```

Output 75.1.5. Discontinuous Function and Derivatives

```
                    An Illustration of Splines and Knots
                     A Cubic Spline Regression Function
                   Discontinuous Function and Derivatives

                           The TRANSREG Procedure

         TRANSREG MORALS Algorithm Iteration History for Identity(Y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.90271    3.29184     0.10061
        2    0.00000    0.00000     0.99254      0.89193    Converged

Algorithm converged.
```

To solve this problem with a DATA step and PROC REG, you would need to create all of the variables in the preceding DATA step (the B data set for the piecewise polynomial with discontinuous third derivatives), plus the following three variables.

```
X13=(X >  5);   /* intercept change after  5 */
X14=(X > 10);   /* intercept change after 10 */
X15=(X > 15);   /* intercept change after 15 */
```

The last two steps use the NKNOTS= *t-option* to specify the number of knots but not their location. NKNOTS=4 places knots at the quintiles while NKNOTS=9 places knots at the deciles. The spline and its first two derivatives are continuous. The R^2 values are 0.74450 and 0.95256. Even though the knots are placed in the wrong places, the spline can closely follow the data with NKNOTS=9. The following statements produce Output 75.1.6.

```
proc transreg data=A;
   model identity(Y) = spline(X / nknots=4);
   title3 'Four Knots';
   output out=A pprefix=Cub4k;
   id V1-V7 LinearY QuadY Cub1Y Cub3Y Cub4Y;
run;

proc transreg data=A;
   model identity(Y) = spline(X / nknots=9);
   title3 'Nine Knots';
   output out=A pprefix=Cub9k;
   id V1-V7 LinearY QuadY Cub1Y Cub3Y Cub4Y Cub4kY;
run;
```

Output 75.1.6. Specifying Number of Knots instead of Knot Location

```
                       An Illustration of Splines and Knots
                        A Cubic Spline Regression Function
                                    Four Knots

                              The TRANSREG Procedure

                TRANSREG MORALS Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.90305    4.46027     0.10061
           2    0.00000    0.00000     0.74450      0.64389    Converged

   Algorithm converged.
```

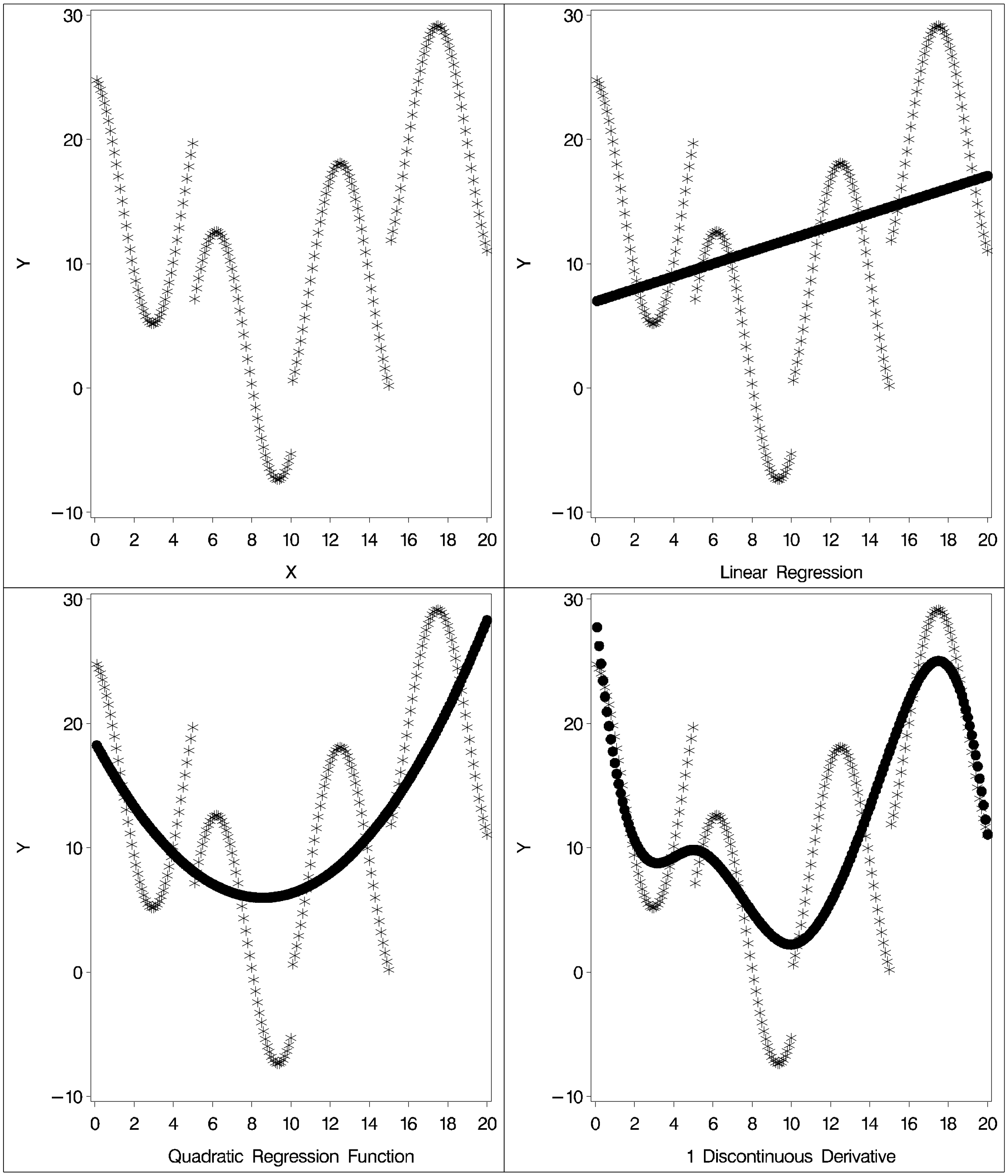
Output 75.1.6. (continued)

```
                       An Illustration of Splines and Knots
                        A Cubic Spline Regression Function
                                    Nine Knots

                              The TRANSREG Procedure

                TRANSREG MORALS Algorithm Iteration History for Identity(Y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.94832    3.03488     0.10061
           2    0.00000    0.00000     0.95256      0.85196    Converged

   Algorithm converged.
```

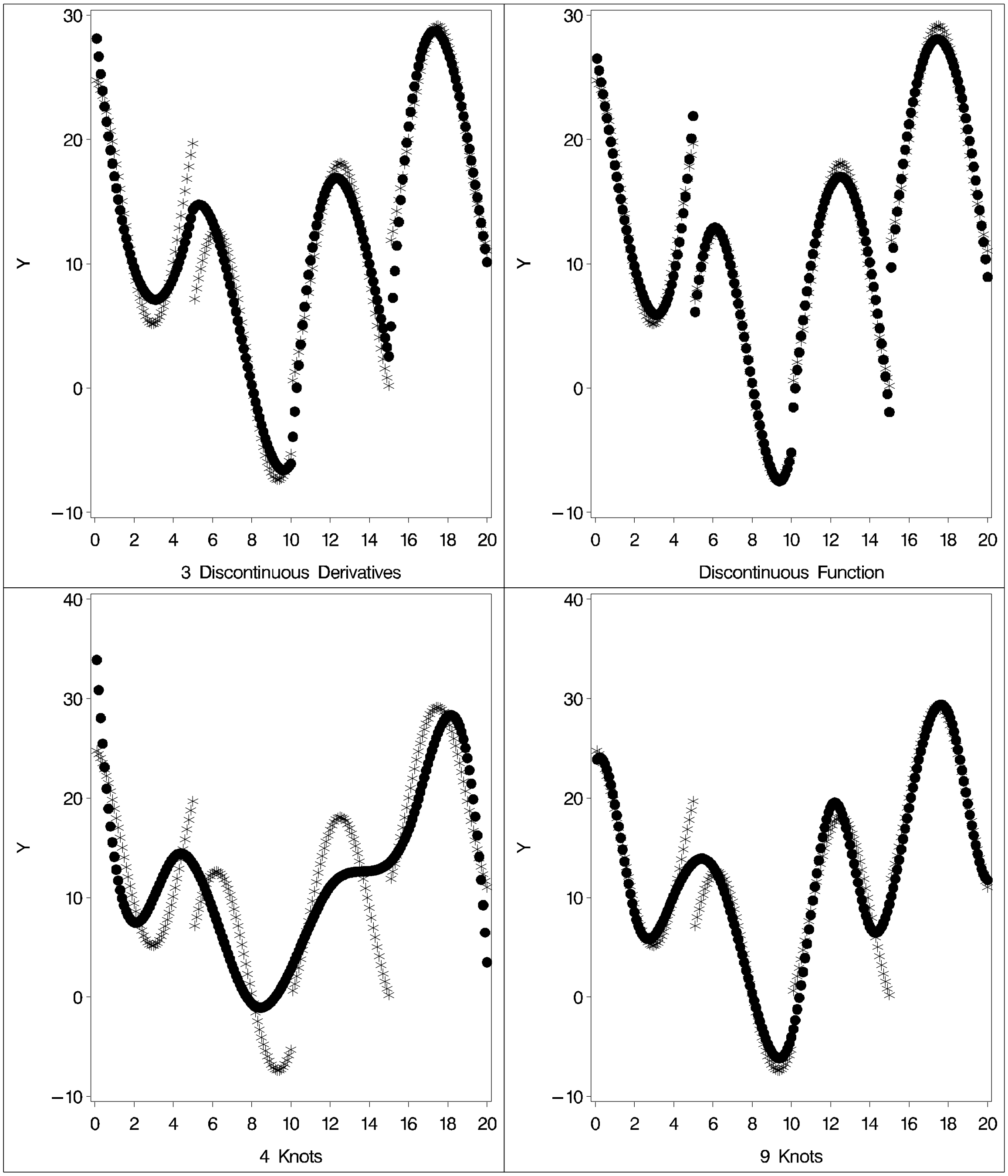
The following statements produce plots that show the data and fit at each step of the analysis. These statements produce Output 75.1.7.

```
goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot data=A;
   title;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   plot Y*X=1              /        &opts name=’tregdis1’;
   plot Y*V1=1 linearY*X=2 /overlay &opts name=’tregdis2’;
   plot Y*V2=1 quadY  *X=2 /overlay &opts name=’tregdis3’;
   plot Y*V3=1 cub1Y  *X=2 /overlay &opts name=’tregdis4’;
   plot Y*V4=1 cub3Y  *X=2 /overlay &opts name=’tregdis5’;
   plot Y*V5=1 cub4Y  *X=2 /overlay &opts name=’tregdis6’;
   plot Y*V6=1 cub4kY *X=2 /overlay &opts name=’tregdis7’;
   plot Y*V7=1 cub9kY *X=2 /overlay &opts name=’tregdis8’;
   symbol1 color=blue   v=star i=none;
   symbol2 color=yellow v=dot  i=none;
   label V1      = ’Linear Regression’
         V2      = ’Quadratic Regression Function’
         V3      = ’1 Discontinuous Derivative’
         V4      = ’3 Discontinuous Derivatives’
         V5      = ’Discontinuous Function’
         V6      = ’4 Knots’
         V7      = ’9 Knots’
         Y       = ’Y’ LinearY = ’Y’ QuadY  = ’Y’ Cub1Y  = ’Y’
         Cub3Y   = ’Y’ Cub4Y   = ’Y’ Cub4kY = ’Y’ Cub9kY = ’Y’;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:tregdis1 2:tregdis3 3:tregdis2 4:tregdis4;
   treplay 1:tregdis5 2:tregdis7 3:tregdis6 4:tregdis8;
run; quit;
```

Output 75.1.7. Plots Summarizing Analysis for Spline Example



Output 75.1.7. (continued)



These next steps show how to find optimal spline transformations of variables in one data set and apply the same transformations to variables in another data set. These steps produce two artificial data sets, in which the variable Y is a linear function of nonlinear transformations of the variables X, W, and Z.

```
title2 'Scoring Spline Variables';

data x;
   do i = 1 to 5000;
      w = normal(7);
      x = normal(7);
      z = normal(7);
      y = w * w + log(5 + x) + sin(z) + normal(7);
      output;
      end;
run;

data z;
   do i = 1 to 5000;
      w = normal(1);
      x = normal(1);
      z = normal(1);
      y = w * w + log(5 + x) + sin(z) + normal(1);
      output;
      end;
run;
```

First, you run PROC TRANSREG asking for spline transformations of the three independent variables. You must use the EXKNOTS= *t-option*, because you need to use the same knots, both interior and exterior, with both data sets. By default the exterior knots will be different if the minima and maxima are different in the two data sets, so you will get the wrong results if you do not specify the EXKNOTS= *t-option* with values less than the minima and greater than the maxima of the two Y variables.

```
ods output splinecoef=c;
proc transreg data=x dum det ss2;
   model ide(y) = spl(w x z / knots=-1.5 to 1.5 by 0.5 exknots=-5 5);
   output out=d;
run;
```

The nonprinting “SplineCoef” table is output to a SAS data set. This data set contains the coefficients used to get the spline transformations and can be used to transform variables in other data sets. These coefficients are also in the details table. However, in the “SplineCoef” table they are in a form directly suitable for use with PROC SCORE.

This next step reads a different input data set and generates an output data set with the B-spline basis for each of the variables. Note that the same interior and exterior knots are used in both the previous and the next steps.

```
proc transreg data=z design;
   model bspl(w x z / knots=-1.5 to 1.5 by 0.5 exknots=-5 5);
   output out=b;
run;
```

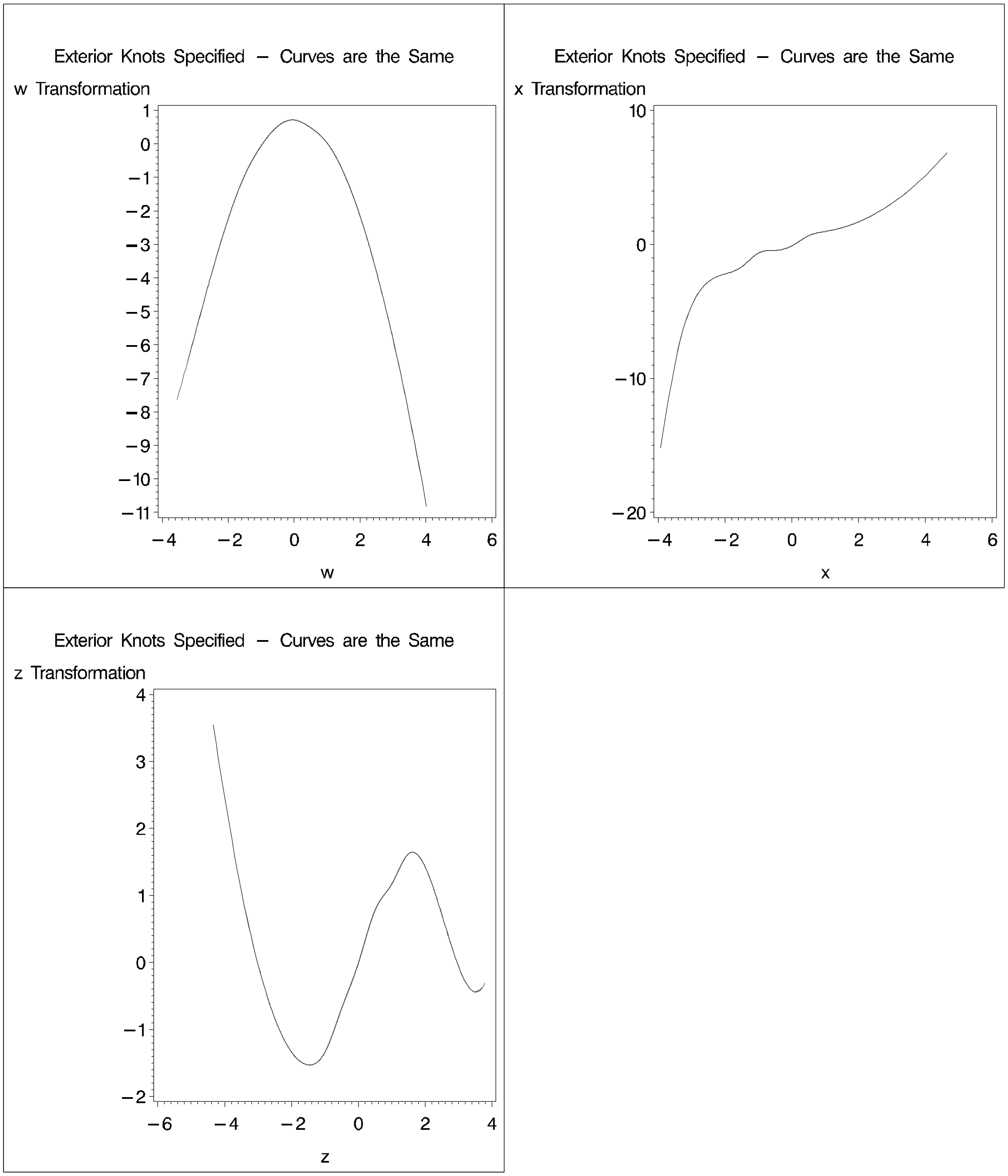
These next three steps score the B-spline bases created in the previous step using the coefficients generated in the first PROC TRANSREG step. PROC SCORE is run once for each SPLINE variable.

```
proc score data=b score=c out=o1(rename=(spline=bw w=nw));
   var w:;
run;

proc score data=b score=c out=o2(rename=(spline=bx x=nx));
   var x:;
run;

proc score data=b score=c out=o3(rename=(spline=bz z=nz));
   var z:;
run;
```

The next steps merge the three transformations with the original data and plot the results. The plots in Output 75.1.8 show that in fact the two transformations for each variable, original and scored, are the same function. Furthermore, PROC TRANSREG found the functional forms that were used to generate the data: quadratic for W, log for X, and sine for Z.

```
goptions goutmode=replace nodisplay;
data all;
   merge d(keep=w x z tw tx tz) o1(keep=nw bw)
         o2(keep=nx bx) o3(keep=nz bz);
run;

proc gplot data=all;
   title3 ’Exterior Knots Specified - Curves are the Same’;
   symbol1 color=blue v=none i=smooths;
   symbol2 color=red  v=none i=smooths;
   plot tw * w = 1 bw * nw = 2 / overlay name=’tregspl1’;
   plot tx * x = 1 bx * nx = 2 / overlay name=’tregspl2’;
   plot tz * z = 1 bz * nz = 2 / overlay name=’tregspl3’;
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:tregspl1 2:tregspl3 3:tregspl2;
run; quit;
```

Output 75.1.8. Scoring Spline Variables Example



Example 75.2. Nonmetric Conjoint Analysis of Tire Data

This example uses PROC TRANSREG to perform a nonmetric conjoint analysis of tire preference data. Conjoint analysis decomposes rank ordered evaluation judgments of products or services into components based on qualitative product attributes.

For each level of each attribute of interest, a numerical “part-worth utility” value is computed. The sum of the part-worth utilities for each product is an estimate of the utility for that product. The goal is to compute part-worth utilities such that the product utilities are as similar as possible to the original rank ordering. (This example is a greatly simplified introductory example.)

The stimuli for the experiment are 18 hypothetical tires. The stimuli represent different brands (Goodstone, Pirogi, Machismo),* prices ($69.99, $74.99, $79.99), expected tread life (50,000, 60,000, 70,000), and road hazard insurance plans (Yes, No). There are $3 \times 3 \times 3 \times 2 = 54$ possible combinations. From these, 18 combinations are selected that form an efficient experimental design for a main effects model. The combinations are then ranked from 1 (most preferred) to 18 (least preferred). In this simple example, there is one set of rankings. A real conjoint study would have many more.

First, the FORMAT procedure is used to specify the meanings of the factor levels, which are entered as numbers in the DATA step along with the ranks. PROC TRANSREG is used to perform the conjoint analysis. A maximum of 50 iterations is requested. The specification Monotone(Rank / Reflect) in the MODEL statement requests that the dependent variable Rank should be monotonically transformed and reflected so that positive utilities mean high preference. The variables Brand, Price, Life, and Hazard are designated as CLASS variables, and the part-worth utilities are constrained by ZERO=SUM to sum to zero within each factor. The UTILITIES *a-option* displays the conjoint analysis results.

The Importance column of the Utilities Table shows that price is the most important attribute in determining preference (57%), followed by expected tread life (18%), brand (15%), and road hazard insurance (10%). Looking at the Utilities Table for the maximum part-worth utility within each attribute, you see from the results that the most preferred combination is Pirogi brand tires, at $69.99, with a 70,000 mile expected tread life, and road hazard insurance. This product is not actually in the data set. The sum of the part-worth utilities for this combination is

$$20.64 = 9.50 + 1.90 + 5.87 + 2.41 + 0.96$$

The following statements produce Output 75.2.1:

```
title 'Nonmetric Conjoint Analysis of Ranks';

proc format;
   value BrandF
              1 = 'Goodstone'
              2 = 'Pirogi   '
              3 = 'Machismo ';
   value PriceF
              1 = '$69.99'
              2 = '$74.99'
              3 = '$79.99';
```

* In real conjoint experiments, real brand names are used.

```
   value LifeF
              1 = ’50,000’
              2 = ’60,000’
              3 = ’70,000’;
   value HazardF
              1 = ’Yes’
              2 = ’No ’;
run;

data Tires;
   input Brand Price Life Hazard Rank;
   format Brand BrandF9. Price PriceF9. Life LifeF6. Hazard HazardF3.;
   datalines;
1 1 2 1  3
1 1 3 2  2
1 2 1 2 14
1 2 2 2 10
1 3 1 1 17
1 3 3 1 12
2 1 1 2  7
2 1 3 2  1
2 2 1 1  8
2 2 3 1  5
2 3 2 1 13
2 3 2 2 16
3 1 1 1  6
3 1 2 1  4
3 2 2 2 15
3 2 3 1  9
3 3 1 2 18
3 3 3 2 11
;

proc transreg maxiter=50 utilities short;
   ods select ConvergenceStatus FitStatistics Utilities;
   model monotone(Rank / reflect) =
         class(Brand Price Life Hazard / zero=sum);
   output ireplace predicted;
run;

proc print label;
   var Rank TRank PRank Brand Price Life Hazard;
   label PRank = ’Predicted Ranks’;
run;
```

Output 75.2.1. Simple Conjoint Analysis

```
                    Nonmetric Conjoint Analysis of Ranks

                           The TRANSREG Procedure

 Monotone(Rank)
 Algorithm converged.


         The TRANSREG Procedure Hypothesis Tests for Monotone(Rank)


              Root MSE           0.49759    R-Square    0.9949
              Dependent Mean     9.50000    Adj R-Sq    0.9913
              Coeff Var          5.23783


           Utilities Table Based on the Usual Degrees of Freedom

                                          Importance
                              Standard    (% Utility
Label              Utility       Error        Range)    Variable

Intercept           9.5000     0.11728                  Intercept

Brand Goodstone    -1.1718     0.16586        15.463    Class.BrandGoodstone
Brand Pirogi        1.8980     0.16586                  Class.BrandPirogi
Brand Machismo     -0.7262     0.16586                  Class.BrandMachismo

Price $69.99        5.8732     0.16586        56.517    Class.Price_69_99
Price $74.99       -0.5261     0.16586                  Class.Price_74_99
Price $79.99       -5.3471     0.16586                  Class.Price_79_99

Life 50,000        -1.2350     0.16586        18.361    Class.Life50_000
Life 60,000        -1.1751     0.16586                  Class.Life60_000
Life 70,000         2.4101     0.16586                  Class.Life70_000

Hazard Yes          0.9588     0.11728         9.659    Class.HazardYes
Hazard No          -0.9588     0.11728                  Class.HazardNo

The standard errors are not adjusted for the fact that the dependent
variable was transformed and so are generally liberal (too small).
```

Output 75.2.1. (continued)

```
                     Nonmetric Conjoint Analysis of Ranks

                       Rank       Predicted
 Obs  Rank  Transformation     Ranks    Brand      Price      Life    Hazard

   1    3       14.4462      13.9851   Goodstone  $69.99     60,000   Yes
   2    2       15.6844      15.6527   Goodstone  $69.99     70,000   No
   3   14        5.7229       5.6083   Goodstone  $74.99     50,000   No
   4   10        5.7229       5.6682   Goodstone  $74.99     60,000   No
   5   17        2.6699       2.7049   Goodstone  $79.99     50,000   Yes
   6   12        5.7229       6.3500   Goodstone  $79.99     70,000   Yes
   7    7       14.4462      15.0774   Pirogi     $69.99     50,000   No
   8    1       18.7699      18.7225   Pirogi     $69.99     70,000   No
   9    8       11.1143      10.5957   Pirogi     $74.99     50,000   Yes
  10    5       14.4462      14.2408   Pirogi     $74.99     70,000   Yes
  11   13        5.7229       5.8346   Pirogi     $79.99     60,000   Yes
  12   16        3.8884       3.9170   Pirogi     $79.99     60,000   No
  13    6       14.4462      14.3708   Machismo   $69.99     50,000   Yes
  14    4       14.4462      14.4307   Machismo   $69.99     60,000   Yes
  15   15        5.7229       6.1139   Machismo   $74.99     60,000   No
  16    9       11.1143      11.6166   Machismo   $74.99     70,000   Yes
  17   18        1.1905       1.2330   Machismo   $79.99     50,000   No
  18   11        5.7229       4.8780   Machismo   $79.99     70,000   No
```

Example 75.3. Metric Conjoint Analysis of Tire Data

This example, which is more detailed than the previous one, uses PROC TRANSREG to perform a metric conjoint analysis of tire preference data. Conjoint analysis can be used to decompose preference ratings of products or services into components based on qualitative product attributes. For each level of each attribute of interest, a numerical “part-worth utility” value is computed. The sum of the part-worth utilities for each product is an estimate of the utility for that product. The goal is to compute part-worth utilities such that the product utilities are as similar as possible to the original ratings. Metric conjoint analysis, as shown in this example, fits an ordinary linear model directly to data assumed to be measured on an interval scale. Nonmetric conjoint analysis, as shown in Example 75.2, finds an optimal monotonic transformation of original data before fitting an ordinary linear model to the transformed data.

This example has three parts. In the first part, an experimental design is created. In the second part, a DATA step creates descriptions of the stimuli for the experiment. The third part of the example performs the conjoint analyses.

The stimuli for the experiment are 18 hypothetical tires. The stimuli represent different brands (Goodstone, Pirogi, Machismo),[*] prices ($69.99, $74.99, $79.99), expected tread life (50,000, 60,000, 70,000), and road hazard insurance plans (Yes, No).

For a conjoint study such as this, you need to create an experimental design with 3 three-level factors, 1 two-level factor, and 18 combinations or *runs*. The easiest way to get this design is with the %MktEx autocall macro. The %MktEx macro requires

[*] In real conjoint experiments, real brand names would be used

you to specify the number of levels of each of the four factors, followed by N=18, the number of runs. Specifying a random number seed, while not strictly necessary, helps ensure that the design is reproducible. The %MktLab macro assigns the actual factor names instead of the default names X1, X2, and so on, and it assigns formats to the factor levels. The %MktEval macro helps you evaluate the design. It shows how correlated or independent the factors are, how often each factor level appears in the design, how often each pair occurs for every factor pair, and how often each product profile or run occurs in the design. See Kuhfeld (2003) for more information on these tools and their use in conjoint and choice modeling.

```
title ’Tire Study, Experimental Design’;

proc format;
   value BrandF
              1 = ’Goodstone’
              2 = ’Pirogi   ’
              3 = ’Machismo ’;
   value PriceF
              1 = ’$69.99’
              2 = ’$74.99’
              3 = ’$79.99’;
   value LifeF
              1 = ’50,000’
              2 = ’60,000’
              3 = ’70,000’;
   value HazardF
              1 = ’Yes’
              2 = ’No ’;(persist
run;

%mktex(3 3 3 2, n=18, seed=448)
%mktlab(vars=Brand Price Life Hazard, out=sasuser.TireDesign,
        statements=format Brand BrandF9. Price PriceF9.
                   Life LifeF6. Hazard HazardF3.)
%mkteval;

proc print data=sasuser.TireDesign;
run;
```

The %MktEx macro output displayed in Output 75.3.1 shows you that the design is 100% efficient, which means it is orthogonal and balanced. The %MktEval macro output displayed in Output 75.3.2 shows you that all of the factors are uncorrelated or orthogonal, the design is balanced (each level occurs once), and every pair of factor levels occurs equally often (again showing that the design is orthogonal). The *n*-way frequencies show that each product profile occurs once (there are no duplicates). The design is shown in Output 75.3.3. The design is automatically randomized (the profiles were sorted into a random order and the original levels are randomly reassigned). Orthogonality, balance, randomization, and other design concepts are discussed in detail in Kuhfeld (2003).

Output 75.3.1. Tire Study, Design Efficiency

```
                         Tire Study, Experimental Design

                             Algorithm Search History

                                    Current          Best
            Design    Row,Col  D-Efficiency  D-Efficiency  Notes
            ----------------------------------------------------------
                 1      Start      100.0000      100.0000  Tab
                 1        End      100.0000




                         Tire Study, Experimental Design

                               The OPTEX Procedure

                             Class Level Information

                            Class  Levels    -Values-

                               x1       3    1 2 3
                               x2       3    1 2 3
                               x3       3    1 2 3
                               x4       2    1 2



                         Tire Study, Experimental Design

                               The OPTEX Procedure

                                                                   Average
                                                                  Prediction
     Design                                                        Standard
     Number     D-Efficiency     A-Efficiency     G-Efficiency       Error
     ------------------------------------------------------------------------
          1       100.0000         100.0000         100.0000          0.6667
```

Output 75.3.2. Tire Study, Design Evaluation

```
                Canonical Correlations Between the Factors
             There are 0 Canonical Correlations Greater Than 0.316

                          Brand    Price    Life    Hazard

                Brand     1        0        0        0
                Price     0        1        0        0
                Life      0        0        1        0
                Hazard    0        0        0        1



                          Summary of Frequencies
             There are 0 Canonical Correlations Greater Than 0.316

                              Frequencies

              Brand           6 6 6
              Price           6 6 6
              Life            6 6 6
              Hazard          9 9
              Brand Price     2 2 2 2 2 2 2 2 2
              Brand Life      2 2 2 2 2 2 2 2 2
              Brand Hazard    3 3 3 3 3 3
              Price Life      2 2 2 2 2 2 2 2 2
              Price Hazard    3 3 3 3 3 3
              Life Hazard     3 3 3 3 3 3
              N-Way           1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Output 75.3.3. Tire Study, Design

```
               Obs    Brand        Price        Life       Hazard

                 1    Pirogi       $79.99       50,000      No
                 2    Machismo     $79.99       70,000      No
                 3    Machismo     $74.99       50,000      Yes
                 4    Machismo     $74.99       50,000      No
                 5    Goodstone    $74.99       70,000      Yes
                 6    Pirogi       $69.99       70,000      Yes
                 7    Goodstone    $69.99       50,000      Yes
                 8    Machismo     $69.99       60,000      Yes
                 9    Pirogi       $74.99       60,000      Yes
                10    Pirogi       $74.99       60,000      No
                11    Goodstone    $79.99       60,000      No
                12    Goodstone    $69.99       50,000      No
                13    Pirogi       $79.99       50,000      Yes
                14    Goodstone    $74.99       70,000      No
                15    Machismo     $69.99       60,000      No
                16    Machismo     $79.99       70,000      Yes
                17    Pirogi       $69.99       70,000      No
                18    Goodstone    $79.99       60,000      Yes
```

Next, the questionnaires are printed, and subjects are given the questionnaires and are asked to rate the tires.

The following statements produce Output 75.3.4. This output is abbreviated; the statements produce stimuli for all combinations.

```
data _null_;
   title;
   set sasuser.TireDesign;
   file print;
   if mod(_n_,4) eq 1 then do;
      put _page_;
      put +55 'Subject ________';
      end;
   length hazardstring $ 7.;
   if put(hazard, hazardf3.) = 'Yes'
      then hazardstring = 'with';
      else hazardstring = 'without';

   s = 3 + (_n_ >= 10);
   put // _n_ +(-1) ') For your next tire purchase, '
          'how likely are you to buy this product?'
       // +s Brand 'brand tires at ' Price +(-1) ','
       /  +s 'with a ' Life 'tread life guarantee, '
       /  +s 'and ' hazardstring 'road hazard insurance.'
       // +s 'Definitely Would                 Definitely Would'
       /  +s 'Not Purchase                             Purchase'
       // +s '1     2     3     4     5     6     7     8     9 ';
run;
```

Output 75.3.4. Conjoint Analysis, Stimuli Descriptions

```
                                                       Subject ________


1) For your next tire purchase, how likely are you to buy this product?

   Pirogi brand tires at $79.99,
   with a 50,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


2) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $79.99,
   with a 70,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


3) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $74.99,
   with a 50,000 tread life guarantee,
   and with road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


4) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $74.99,
   with a 50,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9
```

The third part of the example performs the conjoint analyses. The DATA step reads the data. Only the ratings are entered, one row per subject. Real conjoint studies have many more subjects than five. The TRANSPOSE procedure transposes this (5×18) data set into an (18×5) data set that can be merged with the factor level data set sasuser.TireDesign. The next DATA step does the merge. The PRINT procedure displays the input data set.

PROC TRANSREG fits the five individual conjoint models, one for each subject. The UTILITIES *a-option* displays the conjoint analysis results. The SHORT *a-option* suppresses the iteration histories, OUTTEST=Utils creates an output data set with all of the conjoint results, and the SEPARATORS= option requests that the labels constructed for each category contain two blanks between the variable name and the level value. The ODS select statement is used to limit the displayed output. The MODEL statement specifies IDENTITY for the ratings, which specifies a metric conjoint analysis —the ratings are not transformed. The variables Brand, Price, Life, and Hazard are designated as CLASS variables, and the part-worth utilities are constrained to sum to zero within each factor.

The following statements produce Output 75.3.5:

```
title 'Tire Study, Data Entry, Preprocessing';

data Results;
   input (c1-c18) (1.);
   datalines;
233279766526376493
124467885349168274
262189456534275794
184396375364187754
133379775526267493
;

*---Create an Object by Subject Data Matrix---;
proc transpose data=Results out=Results(drop=_name_) prefix=Subj;
run;

*---Merge the Factor Levels With the Data Matrix---;
data Both;
   merge sasuser.TireDesign Results;
run;

*---Print Input Data Set---;
proc print;
   title2 'Data Set for Conjoint Analysis';
run;

*---Fit Each Subject Individually---;
proc transreg data=Both utilities short outtest=Utils separators='  ';
   ods select TestsNote FitStatistics Utilities;
   title2 'Individual Conjoint Analyses';
   model identity(Subj1-Subj5) =
         class(Brand Price Life Hazard / zero=sum);
run;
```

The output contains two tables per subject, one with overall fit statistics and one with the conjoint analysis results.

Output 75.3.5. Conjoint Analysis

```
                      Tire Study, Data Entry, Preprocessing
                          Data Set for Conjoint Analysis

Obs  Brand      Price      Life     Hazard  Subj1  Subj2  Subj3  Subj4  Subj5

  1  Pirogi     $79.99     50,000    No       2      1      2      1      1
  2  Machismo   $79.99     70,000    No       3      2      6      8      3
  3  Machismo   $74.99     50,000    Yes      3      4      2      4      3
  4  Machismo   $74.99     50,000    No       2      4      1      3      3
  5  Goodstone  $74.99     70,000    Yes      7      6      8      9      7
  6  Pirogi     $69.99     70,000    Yes      9      7      9      6      9
  7  Goodstone  $69.99     50,000    Yes      7      8      4      3      7
  8  Machismo   $69.99     60,000    Yes      6      8      5      7      7
  9  Pirogi     $74.99     60,000    Yes      6      5      6      5      5
 10  Pirogi     $74.99     60,000    No       5      3      5      3      5
 11  Goodstone  $79.99     60,000    No       2      4      3      6      2
 12  Goodstone  $69.99     50,000    No       6      9      4      4      6
 13  Pirogi     $79.99     50,000    Yes      3      1      2      1      2
 14  Goodstone  $74.99     70,000    No       7      6      7      8      6
 15  Machismo   $69.99     60,000    No       6      8      5      7      7
 16  Machismo   $79.99     70,000    Yes      4      2      7      7      4
 17  Pirogi     $69.99     70,000    No       9      7      9      5      9
 18  Goodstone  $79.99     60,000    Yes      3      4      4      4      3
```

Output 75.3.5. (continued)

```
                    Tire Study, Data Entry, Preprocessing
                         Individual Conjoint Analyses

                            The TRANSREG Procedure

         The TRANSREG Procedure Hypothesis Tests for Identity(Subj1)


               Root MSE           0.44721    R-Square    0.9783
               Dependent Mean     5.00000    Adj R-Sq    0.9630
               Coeff Var          8.94427


            Utilities Table Based on the Usual Degrees of Freedom

                                           Importance
                               Standard    (% Utility
Label               Utility       Error        Range)    Variable

Intercept            5.0000     0.10541                  Intercept

Brand  Goodstone     0.3333     0.14907        17.857    Class.BrandGoodstone
Brand  Pirogi        0.6667     0.14907                  Class.BrandPirogi
Brand  Machismo     -1.0000     0.14907                  Class.BrandMachismo

Price  $69.99        2.1667     0.14907        46.429    Class.Price_69_99
Price  $74.99        0.0000     0.14907                  Class.Price_74_99
Price  $79.99       -2.1667     0.14907                  Class.Price_79_99

Life  50,000        -1.1667     0.14907        28.571    Class.Life50_000
Life  60,000        -0.3333     0.14907                  Class.Life60_000
Life  70,000         1.5000     0.14907                  Class.Life70_000

Hazard  Yes          0.3333     0.10541         7.143    Class.HazardYes
Hazard  No          -0.3333     0.10541                  Class.HazardNo
```

Output 75.3.5. (continued)

```
                      Tire Study, Data Entry, Preprocessing
                           Individual Conjoint Analyses

                              The TRANSREG Procedure

            The TRANSREG Procedure Hypothesis Tests for Identity(Subj2)


               Root MSE            0.50553    R-Square    0.9770
               Dependent Mean      4.94444    Adj R-Sq    0.9608
               Coeff Var          10.22410


              Utilities Table Based on the Usual Degrees of Freedom

                                                  Importance
                                      Standard    (% Utility
Label                   Utility          Error        Range)    Variable

Intercept                4.9444        0.11915                  Intercept

Brand  Goodstone         1.2222        0.16851        25.161    Class.BrandGoodstone
Brand  Pirogi           -0.9444        0.16851                  Class.BrandPirogi
Brand  Machismo         -0.2778        0.16851                  Class.BrandMachismo

Price  $69.99            2.8889        0.16851        63.871    Class.Price_69_99
Price  $74.99           -0.2778        0.16851                  Class.Price_74_99
Price  $79.99           -2.6111        0.16851                  Class.Price_79_99

Life  50,000            -0.4444        0.16851         9.677    Class.Life50_000
Life  60,000             0.3889        0.16851                  Class.Life60_000
Life  70,000             0.0556        0.16851                  Class.Life70_000

Hazard  Yes              0.0556        0.11915         1.290    Class.HazardYes
Hazard  No              -0.0556        0.11915                  Class.HazardNo
```

Output 75.3.5. (continued)

```
                      Tire Study, Data Entry, Preprocessing
                           Individual Conjoint Analyses

                              The TRANSREG Procedure

           The TRANSREG Procedure Hypothesis Tests for Identity(Subj3)


                 Root MSE           0.50553    R-Square    0.9747
                 Dependent Mean     4.94444    Adj R-Sq    0.9570
                 Coeff Var         10.22410


              Utilities Table Based on the Usual Degrees of Freedom

                                                  Importance
                                       Standard    (% Utility
Label                   Utility        Error         Range)    Variable

Intercept                4.9444     0.11915                    Intercept

Brand  Goodstone         0.0556     0.16851          13.125    Class.BrandGoodstone
Brand  Pirogi            0.5556     0.16851                    Class.BrandPirogi
Brand  Machismo         -0.6111     0.16851                    Class.BrandMachismo

Price  $69.99            1.0556     0.16851          22.500    Class.Price_69_99
Price  $74.99           -0.1111     0.16851                    Class.Price_74_99
Price  $79.99           -0.9444     0.16851                    Class.Price_79_99

Life  50,000            -2.4444     0.16851          58.125    Class.Life50_000
Life  60,000            -0.2778     0.16851                    Class.Life60_000
Life  70,000             2.7222     0.16851                    Class.Life70_000

Hazard  Yes              0.2778     0.11915           6.250    Class.HazardYes
Hazard  No              -0.2778     0.11915                    Class.HazardNo
```

Output 75.3.5. (continued)

```
                    Tire Study, Data Entry, Preprocessing
                         Individual Conjoint Analyses

                            The TRANSREG Procedure

         The TRANSREG Procedure Hypothesis Tests for Identity(Subj4)


               Root MSE            0.92496    R-Square     0.9099
               Dependent Mean      5.05556    Adj R-Sq     0.8468
               Coeff Var          18.29596


            Utilities Table Based on the Usual Degrees of Freedom

                                              Importance
                                    Standard  (% Utility
Label                    Utility       Error      Range)    Variable

Intercept                 5.0556     0.21802                Intercept

Brand  Goodstone          0.6111     0.30832      31.469    Class.BrandGoodstone
Brand  Pirogi            -1.5556     0.30832                Class.BrandPirogi
Brand  Machismo           0.9444     0.30832                Class.BrandMachismo

Price  $69.99             0.2778     0.30832      10.490    Class.Price_69_99
Price  $74.99             0.2778     0.30832                Class.Price_74_99
Price  $79.99            -0.5556     0.30832                Class.Price_79_99

Life  50,000             -2.3889     0.30832      56.643    Class.Life50_000
Life  60,000              0.2778     0.30832                Class.Life60_000
Life  70,000              2.1111     0.30832                Class.Life70_000

Hazard  Yes               0.0556     0.21802       1.399    Class.HazardYes
Hazard  No               -0.0556     0.21802                Class.HazardNo
```

Output 75.3.5. (continued)

```
                      Tire Study, Data Entry, Preprocessing
                           Individual Conjoint Analyses

                              The TRANSREG Procedure

         The TRANSREG Procedure Hypothesis Tests for Identity(Subj5)


               Root MSE           0.34960    R-Square    0.9879
               Dependent Mean     4.94444    Adj R-Sq    0.9794
               Coeff Var          7.07062


            Utilities Table Based on the Usual Degrees of Freedom

                                                 Importance
                                       Standard  (% Utility
Label                   Utility        Error       Range)    Variable

Intercept                4.9444     0.08240                  Intercept

Brand  Goodstone         0.2222     0.11653          7.500   Class.BrandGoodstone
Brand  Pirogi            0.2222     0.11653                  Class.BrandPirogi
Brand  Machismo         -0.4444     0.11653                  Class.BrandMachismo

Price  $69.99            2.5556     0.11653         56.250   Class.Price_69_99
Price  $74.99           -0.1111     0.11653                  Class.Price_74_99
Price  $79.99           -2.4444     0.11653                  Class.Price_79_99

Life  50,000            -1.2778     0.11653         30.000   Class.Life50_000
Life  60,000            -0.1111     0.11653                  Class.Life60_000
Life  70,000             1.3889     0.11653                  Class.Life70_000

Hazard  Yes              0.2778     0.08240          6.250   Class.HazardYes
Hazard  No              -0.2778     0.08240                  Class.HazardNo
```

These following statements summarize the results. Three tables are displayed: all of the importance values, the average importance, and the part-worth utilities. The first DATA step selects the importance information from the Utils data set. The final assignment statement stores just the variable name from the label relying on the fact that the separator is two blanks. PROC TRANSPOSE creates the data set of importances, one row per subject, and PROC PRINT displays the results. The MEANS procedure displays the average importance of each attribute across the subjects. The next DATA step selects the part-worth utilities information from the Utils data set. PROC TRANSPOSE creates the data set of utilities, one row per subject, and PROC PRINT displays the results.

```
*---Gather the Importance Values---;
data Importance;
   set Utils(keep=_depvar_ Importance Label);
   if n(Importance);
   label = substr(label, 1, index(label, ’  ’));
run;

proc transpose out=Importance2(drop=_:);
   by _depvar_;
   id Label;
run;

proc print;
   title2 ’Importance Values’;
run;

proc means;
   title2 ’Average Importance’;
run;

*---Gather the Part-Worth Utilites---;
data Utilities;
   set Utils(keep=_depvar_ Coefficient Label);
   if n(Coefficient);
run;

proc transpose out=Utilities2(drop=_:);
   by _depvar_;
   id Label;
   idlabel Label;
run;

proc print label;
   title2 ’Utilities’;
run;
```

Output 75.3.6. Summary of Conjoint Analysis Results

```
                  Tire Study, Data Entry, Preprocessing
                            Importance Values

          Obs     Brand      Price       Life      Hazard

           1     17.8571    46.4286    28.5714    7.14286
           2     25.1613    63.8710     9.6774    1.29032
           3     13.1250    22.5000    58.1250    6.25000
           4     31.4685    10.4895    56.6434    1.39860
           5      7.5000    56.2500    30.0000    6.25000
```

Output 75.3.6. (continued)

```
                      Tire Study, Data Entry, Preprocessing
                                Average Importance

                               The MEANS Procedure

Variable    N            Mean         Std Dev         Minimum         Maximum
-----------------------------------------------------------------------------
Brand       5      19.0223929       9.5065111       7.5000000      31.4685315
Price       5      39.9078099      22.6510962      10.4895105      63.8709677
Life        5      36.6034409      20.6028215       9.6774194      58.1250000
Hazard      5       4.4663562       2.8733577       1.2903226       7.1428571
-----------------------------------------------------------------------------
```

Output 75.3.6. (continued)

```
                      Tire Study, Data Entry, Preprocessing
                                    Utilities

                        Brand        Brand        Brand       Price       Price
 Obs    Intercept     Goodstone     Pirogi     Machismo     $69.99      $74.99

  1      5.00000       0.33333      0.66667    -1.00000     2.16667     0.00000
  2      4.94444       1.22222     -0.94444    -0.27778     2.88889    -0.27778
  3      4.94444       0.05556      0.55556    -0.61111     1.05556    -0.11111
  4      5.05556       0.61111     -1.55556     0.94444     0.27778     0.27778
  5      4.94444       0.22222      0.22222    -0.44444     2.55556    -0.11111

          Price        Life         Life         Life       Hazard      Hazard
 Obs     $79.99       50,000       60,000       70,000       Yes          No

  1     -2.16667     -1.16667     -0.33333     1.50000    0.33333    -0.33333
  2     -2.61111     -0.44444      0.38889     0.05556    0.05556    -0.05556
  3     -0.94444     -2.44444     -0.27778     2.72222    0.27778    -0.27778
  4     -0.55556     -2.38889      0.27778     2.11111    0.05556    -0.05556
  5     -2.44444     -1.27778     -0.11111     1.38889    0.27778    -0.27778
```

Based on the importance values, price is the most important attribute for some of the respondents, but expected tread life is most important for others. On the average, price is most important followed closely by expected tread life. Brand and road hazard insurance are less important. Both Goodstone and Pirogi are the most preferred brands by some of the respondents. All respondents preferred a lower price over a higher price, a longer tread life, and road hazard insurance.

Example 75.4. Transformation Regression of Exhaust Emissions Data

In this example, the MORALS algorithm is applied to data from an experiment in which nitrogen oxide emissions from a single cylinder engine are measured for various combinations of fuel, compression ratio, and equivalence ratio. The data are provided by Brinkman (1981).

The equivalence ratio and nitrogen oxide variables are continuous and numeric, so spline transformations of these variables are requested. Each spline is degree three with nine knots (one at each decile) in order to allow PROC TRANSREG a great deal of freedom in finding transformations. The compression ratio variable has only five discrete values, so an optimal scoring is requested. The character variable Fuel is nominal, so it is designated as a classification variable. No monotonicity constraints are placed on any of the transformations. Observations with missing values are excluded with the NOMISS *a-option*.

The squared multiple correlation for the initial model is less than 0.25. PROC TRANSREG increases the R^2 to over 0.95 by transforming the variables. The transformation plots show how each variable is transformed. The transformation of compression ratio (TCpRatio) is nearly linear. The transformation of equivalence ratio (TEqRatio) is nearly parabolic. It can be seen from this plot that the optimal transformation of equivalence ratio is nearly uncorrelated with the original scoring. This suggests that the large increase in R^2 is due to this transformation. The transformation of nitrogen oxide (TNOx) is something like a log transformation.

These results suggest the parametric model

$$\begin{aligned}\log(\mathrm{NOX}) \quad &= \quad b_0 + b_1 \times \mathsf{EqRatio} + b_2 \times \mathsf{EqRatio}^2 + b_3 \times \mathsf{CpRatio} \\ &\quad + \sum_j b_j \mathrm{class}_j(\mathsf{Fuel}) + \mathrm{error}\ .\end{aligned}$$

You can perform this analysis with PROC TRANSREG using the following MODEL statement:

```
model log(NOx)= psp(EqRatio / deg=2) identity(CpRatio)
                class(Fuel / zero=first);
```

The LOG transformation computes the natural log. The PSPLINE expansion expands EqRatio into a linear term, EqRatio, and a squared term, $\mathsf{EqRatio}^2$. A linear transformation of CpRatio and a dummy variable expansion of Fuel is requested with the first level as the reference level. These should provide a good parametric operationalization of the optimal transformations. The final model has an R^2 of 0.91 (smaller than before since the model uses fewer degrees of freedom, but still quite good).

The following statements produce Output 75.4.1 through Output 75.4.2:

```
title 'Gasoline Example';

data Gas;
   input Fuel :$8. CpRatio EqRatio NOx @@;
   label Fuel    = 'Fuel'
         CpRatio = 'Compression Ratio (CR)'
         EqRatio = ’Equivalence Ratio (PHI)’
         NOx     = ’Nitrogen Oxide (NOx)’;
   datalines;
Ethanol  12.0 0.907 3.741 Ethanol  12.0 0.761 2.295
Ethanol  12.0 1.108 1.498 Ethanol  12.0 1.016 2.881
Ethanol  12.0 1.189 0.760 Ethanol   9.0 1.001 3.120
Ethanol   9.0 1.231 0.638 Ethanol   9.0 1.123 1.170
Ethanol  12.0 1.042 2.358 Ethanol  12.0 1.215 0.606
Ethanol  12.0 0.930 3.669 Ethanol  12.0 1.152 1.000
Ethanol  15.0 1.138 0.981 Ethanol  18.0 0.601 1.192
Ethanol   7.5 0.696 0.926 Ethanol  12.0 0.686 1.590
Ethanol  12.0 1.072 1.806 Ethanol  15.0 1.074 1.962
Ethanol  15.0 0.934 4.028 Ethanol   9.0 0.808 3.148
Ethanol   9.0 1.071 1.836 Ethanol   7.5 1.009 2.845
Ethanol   7.5 1.142 1.013 Ethanol  18.0 1.229 0.414
Ethanol  18.0 1.175 0.812 Ethanol  15.0 0.568 0.374
Ethanol  15.0 0.977 3.623 Ethanol   7.5 0.767 1.869
Ethanol   7.5 1.006 2.836 Ethanol   9.0 0.893 3.567
Ethanol  15.0 1.152 0.866 Ethanol  15.0 0.693 1.369
Ethanol  15.0 1.232 0.542 Ethanol  15.0 1.036 2.739
Ethanol  15.0 1.125 1.200 Ethanol   9.0 1.081 1.719
Ethanol   9.0 0.868 3.423 Ethanol   7.5 0.762 1.634
Ethanol   7.5 1.144 1.021 Ethanol   7.5 1.045 2.157
Ethanol  18.0 0.797 3.361 Ethanol  18.0 1.115 1.390
Ethanol  18.0 1.070 1.947 Ethanol  18.0 1.219 0.962
Ethanol   9.0 0.637 0.571 Ethanol   9.0 0.733 2.219
Ethanol   9.0 0.715 1.419 Ethanol   9.0 0.872 3.519
Ethanol   7.5 0.765 1.732 Ethanol   7.5 0.878 3.206
Ethanol   7.5 0.811 2.471 Ethanol  15.0 0.676 1.777
Ethanol  18.0 1.045 2.571 Ethanol  18.0 0.968 3.952
Ethanol  15.0 0.846 3.931 Ethanol  15.0 0.684 1.587
Ethanol   7.5 0.729 1.397 Ethanol   7.5 0.911 3.536
Ethanol   7.5 0.808 2.202 Ethanol   7.5 1.168 0.756
Indolene  7.5 0.831 4.818 Indolene  7.5 1.045 2.849
Indolene  7.5 1.021 3.275 Indolene  7.5 0.970 4.691
Indolene  7.5 0.825 4.255 Indolene  7.5 0.891 5.064
Indolene  7.5 0.710 2.118 Indolene  7.5 0.801 4.602
Indolene  7.5 1.074 2.286 Indolene  7.5 1.148 0.970
Indolene  7.5 1.000 3.965 Indolene  7.5 0.928 5.344
Indolene  7.5 0.767 3.834 Ethanol   7.5 0.749 1.620
Ethanol   7.5 0.892 3.656 Ethanol   7.5 1.002 2.964
82rongas  7.5 0.873 6.021 82rongas  7.5 0.987 4.467
82rongas  7.5 1.030 3.046 82rongas  7.5 1.101 1.596
82rongas  7.5 1.173 0.835 82rongas  7.5 0.931 5.498
82rongas  7.5 0.822 5.470 82rongas  7.5 0.749 4.084
82rongas  7.5 0.625 0.716 94%Eth    7.5 0.818 2.382
94%Eth    7.5 1.128 1.004 94%Eth    7.5 1.191 0.623
94%Eth    7.5 1.132 1.030 94%Eth    7.5 0.993 2.593
94%Eth    7.5 0.866 2.699 94%Eth    7.5 0.910 3.177
94%Eth   12.0 1.139 1.151 94%Eth   12.0 1.267 0.474
94%Eth   12.0 1.017 2.814 94%Eth   12.0 0.954 3.308
94%Eth   12.0 0.861 3.031 94%Eth   12.0 1.034 2.537
94%Eth   12.0 0.781 2.403 94%Eth   12.0 1.058 2.412
94%Eth   12.0 0.884 2.452 94%Eth   12.0 0.766 1.857
```

```
94%Eth    7.5 1.193 0.657 94%Eth    7.5 0.885 2.969
94%Eth    7.5 0.915 2.670 Ethanol  18.0 0.812 3.760
Ethanol  18.0 1.230 0.672 Ethanol  18.0 0.804 3.677
Ethanol  18.0 0.712  .    Ethanol  12.0 0.813 3.517
Ethanol  12.0 1.002 3.290 Ethanol   9.0 0.696 1.139
Ethanol   9.0 1.199 0.727 Ethanol   9.0 1.030 2.581
Ethanol  15.0 0.602 0.923 Ethanol  15.0 0.694 1.527
Ethanol  15.0 0.816 3.388 Ethanol  15.0 0.896  .
Ethanol  15.0 1.037 2.085 Ethanol  15.0 1.181 0.966
Ethanol   7.5 0.899 3.488 Ethanol   7.5 1.227 0.754
Indolene  7.5 0.701 1.990 Indolene  7.5 0.807 5.199
Indolene  7.5 0.902 5.283 Indolene  7.5 0.997 3.752
Indolene  7.5 1.224 0.537 Indolene  7.5 1.089 1.640
Ethanol   9.0 1.180 0.797 Ethanol   7.5 0.795 2.064
Ethanol  18.0 0.990 3.732 Ethanol  18.0 1.201 0.586
Methanol  7.5 0.975 2.941 Methanol  7.5 1.089 1.467
Methanol  7.5 1.150 0.934 Methanol  7.5 1.212 0.722
Methanol  7.5 0.859 2.397 Methanol  7.5 0.751 1.461
Methanol  7.5 0.720 1.235 Methanol  7.5 1.090 1.347
Methanol  7.5 0.616 0.344 Gasohol   7.5 0.712 2.209
Gasohol   7.5 0.771 4.497 Gasohol   7.5 0.959 4.958
Gasohol   7.5 1.042 2.723 Gasohol   7.5 1.125 1.244
Gasohol   7.5 1.097 1.562 Gasohol   7.5 0.984 4.468
Gasohol   7.5 0.928 5.307 Gasohol   7.5 0.889 5.425
Gasohol   7.5 0.827 5.330 Gasohol   7.5 0.674 1.448
Gasohol   7.5 1.031 3.164 Methanol  7.5 0.871 3.113
Methanol  7.5 1.026 2.551 Methanol  7.5 0.598 0.204
Indolene  7.5 0.973 5.055 Indolene  7.5 0.980 4.937
Indolene  7.5 0.665 1.561 Ethanol   7.5 0.629 0.561
Ethanol   9.0 0.608 0.563 Ethanol  12.0 0.584 0.678
Ethanol  15.0 0.562 0.370 Ethanol  18.0 0.535 0.530
94%Eth    7.5 0.674 0.900 Gasohol   7.5 0.645 1.207
Ethanol  18.0 0.655 1.900 94%Eth    7.5 1.022 2.787
94%Eth    7.5 0.790 2.645 94%Eth    7.5 0.720 1.475
94%Eth    7.5 1.075 2.147
;

*---Fit the Nonparametric Model---;
proc transreg data=Gas dummy test nomiss;
   model spline(NOx / nknots=9)=spline(EqRatio / nknots=9)
                       opscore(CpRatio) class(Fuel / zero=first);
   title2 ’Iteratively Estimate NOx, CPRATIO and EQRATIO’;
   output out=Results;
run;

*---Plot the Results---;
goptions goutmode=replace nodisplay;
%let opts = haxis=axis2 vaxis=axis1 frame cframe=ligr;
* Depending on your goptions, these plot options may work better:
* %let opts = haxis=axis2 vaxis=axis1 frame;

proc gplot data=Results;
   title;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   symbol1 color=blue v=dot i=none;
   plot TCpRatio*CpRatio / &opts name='tregex1';
   plot TEqRatio*EqRatio / &opts name='tregex2';
   plot TNOx*NOx         / &opts name='tregex3';
run; quit;

goptions display;
proc greplay nofs tc=sashelp.templt template=l2r2;
   igout gseg;
   treplay 1:tregex1 2:tregex3 3:tregex2;
run; quit;


*-Fit the Parametric Model Suggested by the Nonparametric Analysis-;
proc transreg data=Gas dummy ss2 short nomiss;
   title 'Gasoline Example';
   title2 'Now fit log(NOx) = b0 + b1*EqRatio + b2*EqRatio**2 +';
   title3 'b3*CpRatio + Sum b(j)*Fuel(j) + Error';
   model log(NOx)= pspline(EqRatio / deg=2) identity(CpRatio)
                   class(Fuel / zero=first);
   output out=Results2;
run;
```

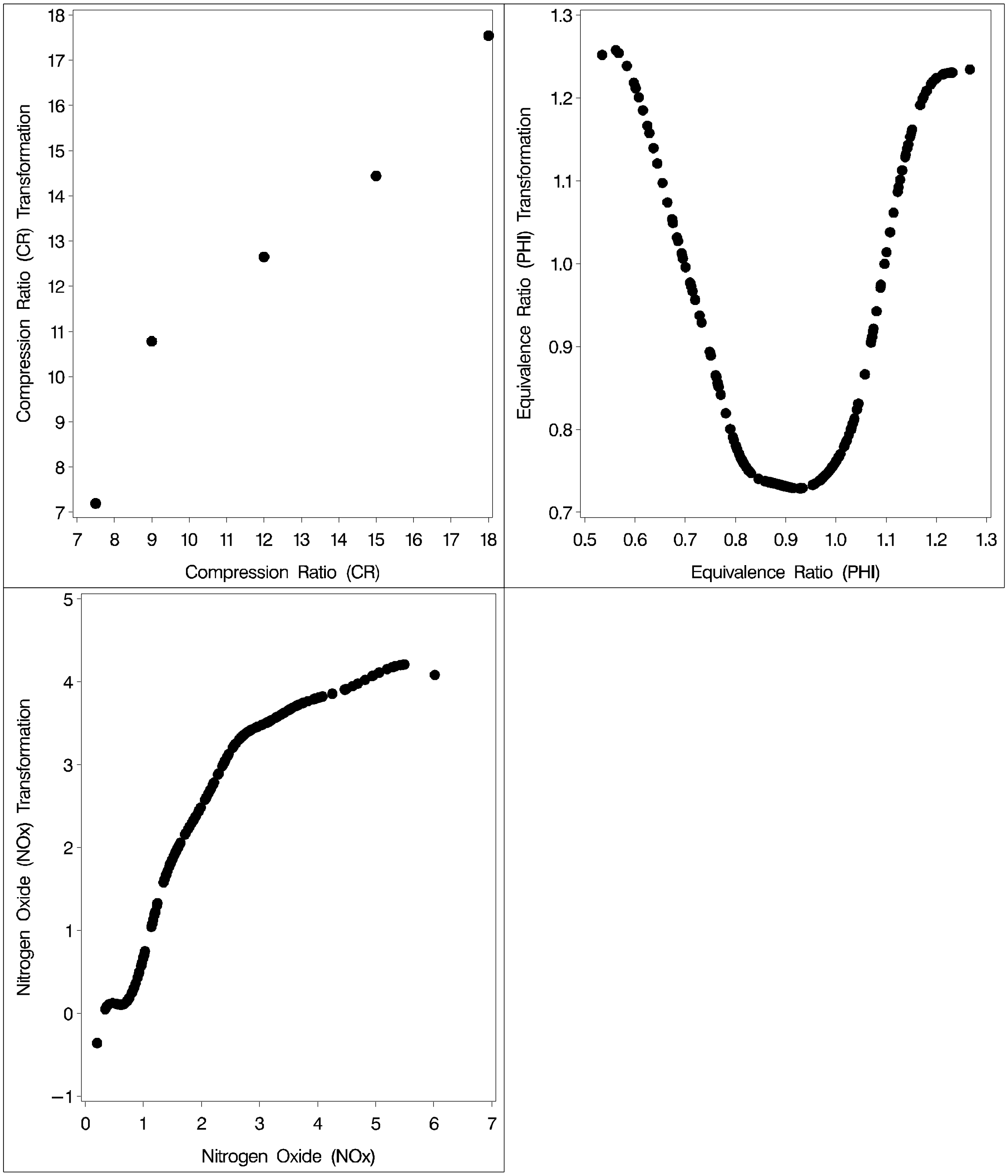
Output 75.4.1. Transformation Regression Example: The Nonparametric Model

```
                             Gasoline Example
              Iteratively Estimate NOx, CPRATIO and EQRATIO

                          The TRANSREG Procedure

                      Dependent Variable Spline(NOx)
                           Nitrogen Oxide (NOx)


                         Class Level Information

   Class    Levels    Values

   Fuel          6    82rongas 94%Eth Ethanol Gasohol Indolene Methanol


                Number of Observations Read            171
                Number of Observations Used            169


       TRANSREG MORALS Algorithm Iteration History for Spline(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.48074    3.86778     0.24597
        1    0.00000    0.00000     0.95865      0.71267    Converged

Algorithm converged.


         The TRANSREG Procedure Hypothesis Tests for Spline(NOx)
                           Nitrogen Oxide (NOx)


      Univariate ANOVA Table Based on the Usual Degrees of Freedom

                              Sum of        Mean
 Source               DF     Squares      Square    F Value    Liberal p

 Model                21    326.0946    15.52831     162.27    >= <.0001
 Error               147     14.0674     0.09570
 Corrected Total     168    340.1619

 The above statistics are not adjusted for the fact that the dependent
 variable was transformed and so are generally liberal.


             Root MSE           0.30935    R-Square    0.9586
             Dependent Mean     2.34593    Adj R-Sq    0.9527
             Coeff Var         13.18661
```

Output 75.4.1. (continued)

```
                                Gasoline Example
                 Iteratively Estimate NOx, CPRATIO and EQRATIO

                            The TRANSREG Procedure

    Adjusted Multivariate ANOVA Table Based on the Usual Degrees of Freedom

          Dependent Variable Scoring Parameters=12   S=12   M=4   N=67

 Statistic                    Value    F Value    Num DF    Den DF           p

 Wilks' Lambda             0.041355       2.05       252      1455    <= <.0001
 Pillai's Trace            0.958645       0.61       252      1764    <= 1.0000
 Hotelling-Lawley Trace    23.18089      12.35       252    945.01    <= <.0001
 Roy's Greatest Root       23.18089     162.27        21       147    >= <.0001

The Wilks' Lambda, Pillai's Trace, and Hotelling-Lawley Trace statistics are a
conservative adjustment of the normal statistics.  Roy's Greatest Root is
liberal.  These statistics are normally defined in terms of the squared
canonical correlations which are the eigenvalues of the matrix H*inv(H+E).
Here the R-Square is used for the first eigenvalue and all other eigenvalues
are set to zero since only one linear combination is used.  Degrees of freedom
are computed assuming all linear combinations contribute to the Lambda and
Trace statistics, so the F tests for those statistics are conservative.  The p
values for the liberal and conservative statistics provide approximate lower
and upper bounds on p.  A liberal test statistic with conservative degrees of
freedom and a conservative test statistic with liberal degrees of freedom yield
at best an approximate p value, which is indicated by a "~" before the p value.
```

Output 75.4.2. Plots of Compression Ratio, Equivalence Ratio, and Nitrogen Oxide



Output 75.4.3. Transformation Regression Example: The Parametric Model

```
                                Gasoline Example
              Now fit log(NOx) = b0 + b1*EqRatio + b2*EqRatio**2 +
                     b3*CpRatio + Sum b(j)*Fuel(j) + Error

                             The TRANSREG Procedure

                          Dependent Variable Log(NOx)
                              Nitrogen Oxide (NOx)


                            Class Level Information

      Class    Levels    Values

      Fuel          6    82rongas 94%Eth Ethanol Gasohol Indolene Methanol


                   Number of Observations Read           171
                   Number of Observations Used           169

   Log(NOx)
   Algorithm converged.


              The TRANSREG Procedure Hypothesis Tests for Log(NOx)
                              Nitrogen Oxide (NOx)


          Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                   Sum of         Mean
      Source               DF     Squares       Square    F Value    Pr > F

      Model                 8    79.33838     9.917298     213.09    <.0001
      Error               160     7.44659     0.046541
      Corrected Total     168    86.78498


                Root MSE            0.21573    R-Square     0.9142
                Dependent Mean      0.63130    Adj R-Sq     0.9099
                Coeff Var          34.17294


       Univariate Regression Table Based on the Usual Degrees of Freedom

                                   Type II
                                    Sum of    Mean
Variable            DF Coefficient Squares  Square F Value Pr > F Label

Intercept            1  -14.586532 49.9469 49.9469 1073.18 <.0001 Intercept
Pspline.EqRatio_1    1   35.102914 62.7478 62.7478 1348.22 <.0001 Equivalence
                                                                  Ratio (PHI) 1
Pspline.EqRatio_2    1  -19.386468 64.6430 64.6430 1388.94 <.0001 Equivalence
                                                                  Ratio (PHI) 2
Identity(CpRatio)    1    0.032058  1.4445  1.4445   31.04 <.0001 Compression
                                                                  Ratio (CR)
Class.Fuel94_Eth     1   -0.449583  1.3158  1.3158   28.27 <.0001 Fuel 94%Eth
Class.FuelEthanol    1   -0.414242  1.2560  1.2560   26.99 <.0001 Fuel Ethanol
Class.FuelGasohol    1   -0.016719  0.0015  0.0015    0.03 0.8584 Fuel Gasohol
Class.FuelIndolene   1    0.001572  0.0000  0.0000    0.00 0.9853 Fuel Indolene
Class.FuelMethanol   1   -0.580133  1.7219  1.7219   37.00 <.0001 Fuel Methanol
```

Example 75.5. Preference Mapping of Cars Data

This example uses PROC TRANSREG to perform a preference mapping (PREFMAP) analysis (Carroll 1972) of car preference data after a PROC PRINQUAL principal component analysis. The PREFMAP analysis is a response surface regression that locates ideal points for each dependent variable in a space defined by the independent variables.

The data are ratings obtained from 25 judges of their preference for each of 17 automobiles. The ratings were made on a zero (very weak preference) to nine (very strong preference) scale. These judgments were made in 1980 about that year's products. There are two character variables that indicate the manufacturer and model of the automobile. The data set also contains three ratings: miles per gallon (MPG), projected reliability (Reliability), and quality of the ride (Ride). These ratings are on a one (bad) to five (good) scale. PROC PRINQUAL creates an OUT= data set containing standardized principal component scores (Prin1 and Prin2), along with the ID variables MODEL, MPG, Reliability, and Ride.

The first PROC TRANSREG step fits univariate regression models for MPG and Reliability. All variables are designated IDENTITY. A vector drawn in the plot of Prin1 and Prin2 from the origin to the point defined by an attribute's regression coefficients approximately shows how the cars differ on that attribute. Refer to Carroll (1972) for more information. The Prin1 and Prin2 columns of the TResult1 OUT= data set contain the car coordinates (_Type_='SCORE' observations) and endpoints of the MPG and Reliability vectors (_Type_='M COEFFI' observations).

The second PROC TRANSREG step fits a univariate regression model with Ride designated IDENTIY, and Prin1 and Prin2 designated POINT. The POINT expansion creates an additional independent variable _ISSQ_, which contains the sum of Prin1 squared and Prin2 squared. The OUT= data set TResult2 contains no _Type_='SCORE' observations, only ideal point (_Type_='M POINT') coordinates for Ride. The coordinates of both the vectors and the ideal points are output by specifying COORDINATES in the OUTPUT statement in PROC TRANSREG.

A vector model is used for MPG and Reliability because perfectly efficient and reliable cars do not exist in the data set. The ideal points for MPG and Reliability are far removed from the plot of the cars. It is more likely that an ideal point for quality of the ride is in the plot, so an ideal point model is used for the ride variable. Refer to Carroll (1972) and Schiffman, Reynolds, and Young (1981) for discussions of the vector model and point models (including the EPOINT and QPOINT versions of the point model that are not used in this example).

The final DATA step combines the two output data sets and creates a data set suitable for the %PLOTIT macro. (For information on the %PLOTIT macro, see Appendix B, "Using the %PLOTIT Macro.") The plot contains one point per car and one point for each of the three ratings. The %PLOTIT macro options specify the input data set, how to handle anti-ideal points (described later), and where to draw horizontal and vertical reference lines. The DATATYPE= option specifies that the input data set contains results of a PREFMAP vector model and a PREFMAP ideal point model.

This instructs the macro to draw vectors to _Type_='M COEFFI' observations and circles around _Type_='M POINT' observations.

An unreliable to reliable direction extends from the left and slightly below the origin to the right and slightly above the origin. The Japanese and European Cars are rated, on the average, as more reliable. A low MPG to good MPG direction extends from the top left of the plot to the bottom right. The smaller cars, on the average, get better gas mileage. The ideal point for Ride is in the top, just right of the center of the plot. Cars near the Ride ideal point tend to have a better ride than cars far away. It can be seen from the iteration history tables that none of these ratings perfectly fits the model, so all of the interpretations are approximate.

The Ride point is a "negative-negative" ideal point. The point models assume that small ratings mean the object (car) is similar to the rating name and large ratings imply dissimilarity to the rating name. Because the opposite scoring is used, the interpretation of the Ride point must be reversed to a negative ideal point (bad ride). However, the coefficient for the _ISSQ_ variable is negative, so the interpretation is reversed again, back to the original interpretation. Anti-ideal points are taken care of in the %PLOTIT macro. Specify ANTIIDEA=1 when large values are positive or ideal and ANTIIDEA=-1 when small values are positive or ideal.

The following statements produce Output 75.5.1 through Output 75.5.2:

```
title 'Preference Ratings for Automobiles Manufactured in 1980';
data CarPreferences;
   input Make $ 1-10 Model $ 12-22 @25 (Judge1-Judge25) (1.)
         MPG Reliability Ride;
   datalines;
Cadillac   Eldorado     8007990491240508971093809 3 2 4
Chevrolet  Chevette     0051200423451043003515698 5 3 2
Chevrolet  Citation     4053305814161643544747795 4 1 5
Chevrolet  Malibu       6027400723121345545668658 3 3 4
Ford       Fairmont     2024006715021443530648655 3 3 4
Ford       Mustang      5007197705021101850657555 3 2 2
Ford       Pinto        0021000303030201500514078 4 1 1
Honda      Accord       5956897609699952998975078 5 5 3
Honda      Civic        4836709507488852567765075 5 5 3
Lincoln    Continental  7008990592230409962091909 2 4 5
Plymouth   Gran Fury    7006000434101107333458708 2 1 5
Plymouth   Horizon      3005005635461302444675655 4 3 3
Plymouth   Volare       4005003614021602754476555 2 1 3
Pontiac    Firebird     0107895613201206958265907 1 1 5
Volkswagen Dasher       4858696508877795377895000 5 3 4
Volkswagen Rabbit       4858509709695795487885000 5 4 3
Volvo      DL           9989998909999987989919000 4 5 5
;

*---Compute Coordinates for a 2-Dimensional Scatter Plot of Cars---;
proc prinqual data=CarPreferences out=PResults(drop=Judge1-Judge25)
              n=2 replace standard scores;
   id Model MPG Reliability Ride;
   transform identity(Judge1-Judge25);
   title2 'Multidimensional Preference (MDPREF) Analysis';
run;

*---Compute Endpoints for MPG and Reliability Vectors---;
proc transreg data=PResults;
   Model identity(MPG Reliability)=identity(Prin1 Prin2);
   output tstandard=center coordinates replace out=TResult1;
   id Model;
   title2 'Preference Mapping (PREFMAP) Analysis';
run;

*---Compute Ride Ideal Point Coordinates---;
proc transreg data=PResults;
   Model identity(Ride)=point(Prin1 Prin2);
   output tstandard=center coordinates replace noscores out=TResult2;
   id Model;
run;

proc print; run;

*---Combine Data Sets and Plot the Results---;
data plot;
   title3 'Plot of Automobiles and Ratings';
   set Tresult1 Tresult2;
run;

%plotit(data=plot, datatype=vector ideal, antiidea=1, href=0, vref=0);
```

Output 75.5.1. Preference Ratings Example Output

```
          Preference Ratings for Automobiles Manufactured in 1980
               Multidimensional Preference (MDPREF) Analysis

                           The PRINQUAL Procedure

                  PRINQUAL MTV Algorithm Iteration History

Iteration    Average    Maximum     Proportion    Criterion
   Number     Change     Change    of Variance       Change    Note
----------------------------------------------------------------------------
        1    0.00000    0.00000        0.66946                 Converged

 Algorithm converged.
WARNING: The number of observations is less than or equal to the number of
                                 variables.
               WARNING: Multiple optimal solutions may exist.
```

Output 75.5.1. (continued)

```
         Preference Ratings for Automobiles Manufactured in 1980
                  Preference Mapping (PREFMAP) Analysis

                          The TRANSREG Procedure

    TRANSREG Univariate Algorithm Iteration History for Identity(MPG)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.00000    0.00000     0.57197                 Converged

Algorithm converged.
```

Output 75.5.1. (continued)

```
         Preference Ratings for Automobiles Manufactured in 1980
                  Preference Mapping (PREFMAP) Analysis

                          The TRANSREG Procedure

TRANSREG Univariate Algorithm Iteration History for Identity(Reliability)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.00000    0.00000     0.50859                 Converged

Algorithm converged.
```

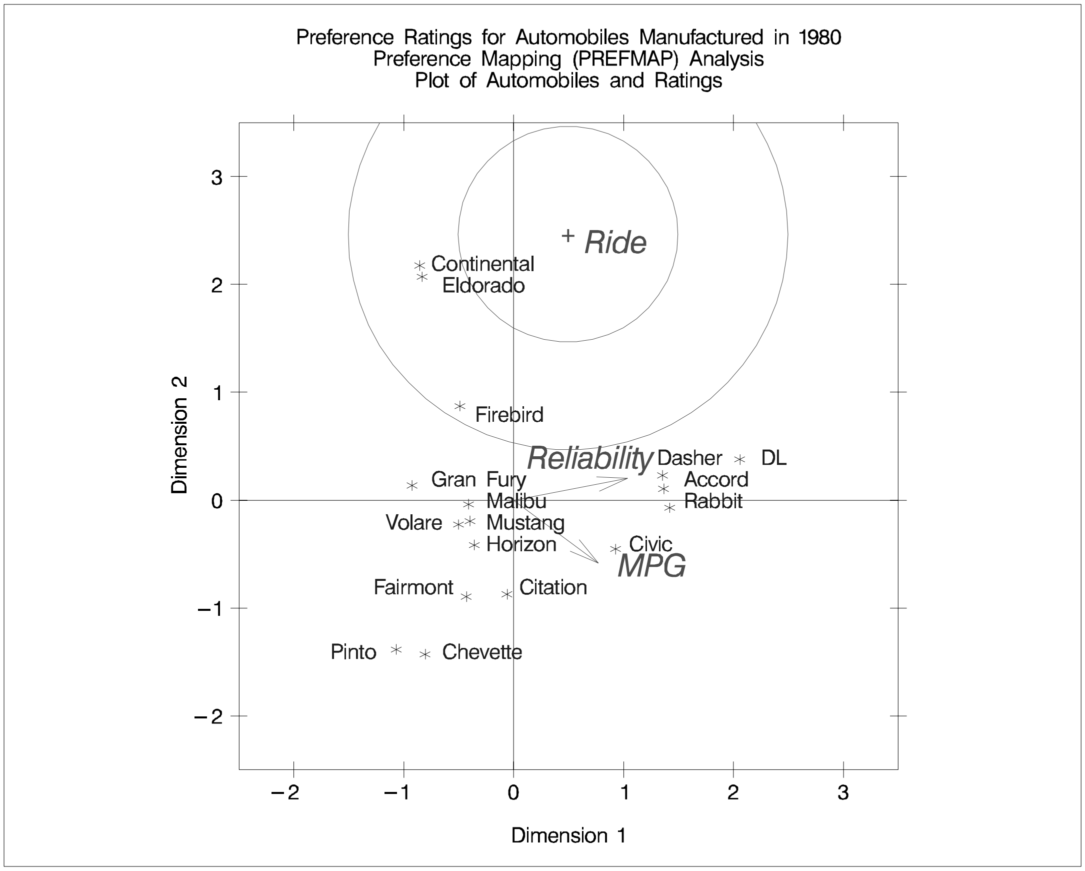
Output 75.5.1. (continued)

```
         Preference Ratings for Automobiles Manufactured in 1980
                  Preference Mapping (PREFMAP) Analysis

                          The TRANSREG Procedure

    TRANSREG Univariate Algorithm Iteration History for Identity(Ride)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        1    0.00000    0.00000     0.37797                 Converged

Algorithm converged.
```

Output 75.5.1. (continued)

```
                 Preference Ratings for Automobiles Manufactured in 1980
                          Preference Mapping (PREFMAP) Analysis

Obs   _TYPE_    _NAME_   Ride   Intercept    Prin1      Prin2     _ISSQ_    Model

 1    M POINT   Ride      .         .       0.49461    2.46539   -0.17448   Ride
```

Output 75.5.2. Preference Ratings for Automobiles Manufactured in 1980



Example 75.6. Box Cox

This example illustrates finding a Box-Cox transformation (see the “Box-Cox Transformations” section on page 4595) of some artificial data. Data were generated from the model

$$y = e^{x+\epsilon}$$

where $\epsilon \sim \mathrm{N}(0, 1)$. The transformed data can be fit with a linear model

$$\log(y) = x + \epsilon$$

These statements produce Output 75.6.1.

```
title 'Basic Box-Cox Example';

data x;
   do x = 1 to 8 by 0.025;
      y = exp(x + normal(7));
      output;
      end;
   run;

proc transreg data=x ss2 details;
   title2 'Defaults';
   model boxcox(y) = identity(x);
   run;
```

Output 75.6.1. Basic Box-Cox Example, Default Output

```
                              Basic Box-Cox Example                              88
                                     Defaults

                              The TRANSREG Procedure

                            Transformation Information
                                  for BoxCox(y)

                         Lambda      R-Square    Log Like

                          -3.00          0.03    -4601.01
                          -2.75          0.04    -4266.08
                          -2.50          0.04    -3934.11
                          -2.25          0.05    -3605.75
                          -2.00          0.06    -3281.88
                          -1.75          0.07    -2963.74
                          -1.50          0.10    -2653.14
                          -1.25          0.14    -2352.72
                          -1.00          0.21    -2066.32
                          -0.75          0.34    -1799.25
                          -0.50          0.52    -1558.55
                          -0.25          0.71    -1360.28
                           0.00 +        0.79    -1275.31 <
                           0.25          0.70    -1382.62
                           0.50          0.51    -1589.03
                           0.75          0.34    -1834.53
                           1.00          0.22    -2105.88
                           1.25          0.15    -2397.35
                           1.50          0.11    -2704.64
                           1.75          0.08    -3024.24
                           2.00          0.06    -3353.38
                           2.25          0.05    -3689.91
                           2.50          0.04    -4032.18
                           2.75          0.03    -4378.97
                           3.00          0.03    -4729.37

                       < - Best Lambda
                       * - Confidence Interval
                       + - Convenient Lambda
```

PROC TRANSREG correctly selects the log transformation $\lambda = 0$, with a narrow confidence interval. The maximum of the log likelihood function is flagged with the less-than sign (<), and the convenient power parameter of $\lambda = 0$ in the confidence interval is flagged by the plus sign (+). The rest of the output is shown next in Output 75.6.2.

Output 75.6.2. Basic Box-Cox Example, Default Output

```
                             Basic Box-Cox Example                            89
                                    Defaults

                             The TRANSREG Procedure

                          Dependent Variable BoxCox(y)


                   Number of Observations Read            281
                   Number of Observations Used            281


         TRANSREG Univariate Algorithm Iteration History for BoxCox(y)

   Iteration    Average    Maximum                Criterion
      Number     Change     Change    R-Square       Change    Note
   -------------------------------------------------------------------------
           1    0.00000    0.00000     0.79064                 Converged

   Algorithm converged.


                     Model Statement Specification Details

 Type  DF Variable    Description     Value

 Dep    1 BoxCox(y)   Lambda Used     0
                      Lambda          0
                      Log Likelihood  -1275.3
                      Conv. Lambda    0
                      Conv. Lambda LL -1275.3
                      CI Limit        -1277.2
                      Alpha           0.05

 Ind    1 Identity(x) DF              1


             The TRANSREG Procedure Hypothesis Tests for BoxCox(y)


         Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
    Source               DF     Squares      Square    F Value    Liberal p

    Model                 1    1145.884    1145.884    1053.66    >= <.0001
    Error               279     303.421       1.088
    Corrected Total     280    1449.305

    The above statistics are not adjusted for the fact that the dependent
    variable was transformed and so are generally liberal.


                Root MSE           1.04285    R-Square    0.7906
                Dependent Mean     4.49653    Adj R-Sq    0.7899
                Coeff Var         23.19225    Lambda      0.0000


       Univariate Regression Table Based on the Usual Degrees of Freedom

                                     Type II
                                      Sum of       Mean
Variable        DF    Coefficient    Squares     Square    F Value    Liberal p

Intercept        1     0.01551366       0.01       0.01       0.01    >= 0.9185
Identity(x)      1     0.99578183    1145.88    1145.88    1053.66    >= <.0001

The above statistics are not adjusted for the fact that the dependent variable
was transformed and so are generally liberal.
```

This next example uses several options. The LAMBDA= option specifies power parameters sparsely from -2 to -0.5 and 0.5 to 2 just to get the general shape of the log likelihood function in that region. Between -0.5 and 0.5, more power parameters are tried. The CONVENIENT option is specified so that if a power parameter like $\lambda = 1$ or $\lambda = 0$ is found in the confidence interval, it will be used instead of the optimal power parameter. PARAMETER=2 is specified to add 2 to each y before performing the transformations. ALPHA=0.00001 specifies a wide confidence interval.

These statements produce Output 75.6.3.

```
proc transreg data=x ss2 details;
   title2 'Several Options Demonstrated';
   model boxcox(y / lambda=-2 -1 -0.5 to 0.5 by 0.05 1 2
                    convenient
                    parameter=2
                    alpha=0.00001)
       = identity(x);
   run;
```

Output 75.6.3. Basic Box-Cox Example, Several Options Demonstrated

```
                      Basic Box-Cox Example                           90
                   Several Options Demonstrated

                      The TRANSREG Procedure

                    Transformation Information
                          for BoxCox(y)

                 Lambda      R-Square    Log Like

                 -2.000          0.22    -2583.73
                 -1.000          0.45    -1779.35
                 -0.500          0.67    -1439.82
                 -0.450          0.70    -1410.51
                 -0.400          0.72    -1382.74
                 -0.350          0.74    -1356.92
                 -0.300          0.76    -1333.59
                 -0.250          0.77    -1313.42
                 -0.200          0.79    -1297.21
                 -0.150          0.79    -1285.83 *
                 -0.100          0.80    -1280.09 <
                 -0.050          0.80    -1280.63 *
                  0.000 +        0.79    -1287.71 *
                  0.050          0.78    -1301.19
                  0.100          0.76    -1320.56
                  0.150          0.74    -1345.09
                  0.200          0.72    -1373.99
                  0.250          0.69    -1406.51
                  0.300          0.65    -1442.02
                  0.350          0.62    -1480.02
                  0.400          0.58    -1520.13
                  0.450          0.54    -1562.05
                  0.500          0.50    -1605.57
                  1.000          0.22    -2105.88
                  2.000          0.06    -3320.36

               < - Best Lambda
               * - Confidence Interval
               + - Convenient Lambda
```

The results show that the optimal power parameter is -0.1 but 0 is in the confidence interval, hence a log transformation is chosen. The rest of the output is shown next in Output 75.6.4.

Output 75.6.4. Basic Box-Cox Example, Several Options Demonstrated

```
                            Basic Box-Cox Example                            91
                         Several Options Demonstrated

                            The TRANSREG Procedure

                         Dependent Variable BoxCox(y)


                  Number of Observations Read           281
                  Number of Observations Used           281


        TRANSREG Univariate Algorithm Iteration History for BoxCox(y)

  Iteration    Average    Maximum                Criterion
     Number     Change     Change    R-Square       Change    Note
  -------------------------------------------------------------------------
          1    0.00000    0.00000     0.79238                 Converged

  Algorithm converged.


                    Model Statement Specification Details

Type  DF Variable    Description     Value

Dep    1 BoxCox(y)   Lambda Used     0
                     Lambda          -0.1
                     Log Likelihood  -1280.1
                     Conv. Lambda    0
                     Conv. Lambda LL -1287.7
                     CI Limit        -1289.9
                     Alpha           0.00001
                     Parameter       2
                     Options         Convenient Lambda Used

Ind    1 Identity(x) DF              1
```

Output 75.6.4. (continued)

```
                              Basic Box-Cox Example                                92
                           Several Options Demonstrated

                              The TRANSREG Procedure

              The TRANSREG Procedure Hypothesis Tests for BoxCox(y)


          Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                   Sum of        Mean
    Source                DF      Squares      Square    F Value    Liberal p

    Model                  1      999.438    999.4381    1064.82    >= <.0001
    Error                279      261.868      0.9386
    Corrected Total      280     1261.306

    The above statistics are not adjusted for the fact that the dependent
    variable was transformed and so are generally liberal.


                Root MSE           0.96881    R-Square    0.7924
                Dependent Mean     4.61429    Adj R-Sq    0.7916
                Coeff Var         20.99591    Lambda      0.0000


        Univariate Regression Table Based on the Usual Degrees of Freedom

                                       Type II
                                        Sum of       Mean
Variable        DF    Coefficient    Squares      Square    F Value    Liberal p

Intercept        1     0.42939328      8.746       8.746       9.32    >= 0.0025
Identity(x)      1     0.92997620    999.438     999.438    1064.82    >= <.0001

The above statistics are not adjusted for the fact that the dependent variable
was transformed and so are generally liberal.
```

The next part of this example shows how to make graphical displays of the Box-Cox transformation results. Plots include the log likelihood function with the confidence interval, root mean squared error as a function of the power parameter, R^2 as a function of the power parameter, the Box-Cox transformation of the variable y, the original scatter plot based on the untransformed data, and the new scatter plot based on the transformed data. Also, a condensed version of the log likelihood table with the confidence interval is printed. Here are the data.

```
   title h=1 'Box-Cox Graphical Displays';

   data x;
      input y x @@;
      datalines;
  10.0  3.0  72.6  8.3  59.7  8.1  20.1  4.8  90.1  9.8   1.1  0.9
  78.2  8.5  87.4  9.0   9.5  3.4   0.1  1.4   0.1  1.1  42.5  5.1
  57.0  7.5   9.9  1.9   0.5  1.0 121.1  9.9  37.5  5.9  49.5  6.7
   8.3  1.8   0.6  1.8  53.0  6.7 112.8 10.0  40.7  6.4   5.1  2.4
  73.3  9.5 122.4  9.9  87.2  9.4 121.2  9.9  23.1  4.3   7.1  3.5
  12.4  3.3   5.6  2.7 113.0  9.6 110.5 10.0   3.1  1.5  52.4  7.9
  80.4  8.1   0.6  1.6 115.1  9.1  15.9  3.1  56.5  7.3  85.4  9.8
  32.5  5.8  43.0  6.2   0.1  0.8  21.8  5.2  15.2  3.5   5.2  3.0
   0.2  0.8  73.5  8.2   4.9  3.2   0.2  0.3  69.0  9.2   3.6  3.5
   0.2  0.9 101.3  9.9  10.0  3.7  16.9  3.0  11.2  5.0   0.2  0.4
  80.8  9.4  24.9  5.7 113.5  9.7   6.2  2.1  12.5  3.2   4.8  1.8
  80.1  8.3  26.4  4.8  13.4  3.8  99.8  9.7  44.1  6.2  15.3  3.8
   2.2  1.5  10.3  2.7  13.8  4.7  38.6  4.5  79.1  9.8  33.6  5.8
   9.1  4.5  89.3  9.1   5.5  2.6  20.0  4.8   2.9  2.9  82.9  8.4
   7.0  3.5  14.5  2.9  16.0  3.7  29.3  6.1  48.9  6.3   1.6  1.9
  34.7  6.2  33.5  6.5  26.0  5.6  12.7  3.1   0.1  0.3  15.4  4.2
   2.6  1.8  58.6  7.9  81.2  8.1  37.2  6.9
   ;
```

The TRANSREG procedure is run to find the Box-Cox transformation. The lambda list is -2 TO 2 BY 0.01, which produces 401 lambdas. This many power parameters makes a nice graphical display with plenty of detail around the confidence interval. However, 401 values is a lot to print, so for this reason, the usual Box-Cox transformation information table is excluded from the printed output. Instead, it is output to a SAS data set using ODS so a sample of it can be printed. Just the confidence interval and the rows corresponding to power parameters that are multiples of 0.5 are printed. Null labels are provided for the columns that need to be printed without headers. The details table is also output to a SAS data set using ODS, since it contains information that will be incorporated into some of the plots. These statements produce Output 75.6.5.

```
   * Fit Box-Cox model, output results to output data sets;
   ods output boxcox=b details=d;
   ods exclude boxcox;
   proc transreg details data=x;
      model boxcox(y / convenient lambda=-2 to 2 by 0.01) = identity(x);
      output out=trans;
      run;

   proc print noobs label data=b(drop=rmse);
      title2 'Confidence Interval';
      where ci ne ' ' or abs(lambda - round(lambda, 0.5)) < 1e-6;
      label convenient = '00'x ci = '00'x;
      run;
```

Output 75.6.5. Box-Cox Graphical Displays

```
                          Box-Cox Graphical Displays                          93

                            The TRANSREG Procedure

        TRANSREG Univariate Algorithm Iteration History for BoxCox(y)

  Iteration    Average    Maximum                Criterion
     Number     Change     Change    R-Square       Change    Note
  -------------------------------------------------------------------------
          1    0.00000    0.00000     0.95396                 Converged

  Algorithm converged.


                    Model Statement Specification Details

Type  DF Variable    Description     Value

Dep    1 BoxCox(y)   Lambda Used     0.5
                     Lambda          0.46
                     Log Likelihood  -167.0
                     Conv. Lambda    0.5
                     Conv. Lambda LL -168.3
                     CI Limit        -169.0
                     Alpha           0.05
                     Options         Convenient Lambda Used

Ind    1 Identity(x) DF              1
```

Output 75.6.5. (continued)

```
                          Box-Cox Graphical Displays                          94
                             Confidence Interval

           Dependent      Lambda         R-Square    Log Like

           BoxCox(y)       -2.00             0.14    -1030.56
           BoxCox(y)       -1.50             0.17     -810.50
           BoxCox(y)       -1.00             0.22     -602.53
           BoxCox(y)       -0.50             0.39     -415.56
           BoxCox(y)        0.00             0.78     -257.92
           BoxCox(y)        0.41             0.95     -168.40    *
           BoxCox(y)        0.42             0.95     -167.86    *
           BoxCox(y)        0.43             0.95     -167.46    *
           BoxCox(y)        0.44             0.95     -167.19    *
           BoxCox(y)        0.45             0.95     -167.05    *
           BoxCox(y)        0.46             0.95     -167.04    <
           BoxCox(y)        0.47             0.95     -167.16    *
           BoxCox(y)        0.48             0.95     -167.41    *
           BoxCox(y)        0.49             0.95     -167.79    *
           BoxCox(y)        0.50    +        0.95     -168.28    *
           BoxCox(y)        0.51             0.95     -168.89    *
           BoxCox(y)        1.00             0.89     -253.09
           BoxCox(y)        1.50             0.79     -345.35
           BoxCox(y)        2.00             0.70     -435.01
```

These next steps extract information from the Box-Cox transformation and details tables and store the information in macro variables. The confidence interval limit from the details table provides a vertical axis reference line for the log likelihood plot. The convenient power parameter (’Lambda Used’) is extracted from the footnote. The confidence interval is extracted from the confidence interval observations of the Box-Cox transformation table and will be used in the footnote and for horizontal axis reference lines in the log likelihood plot.

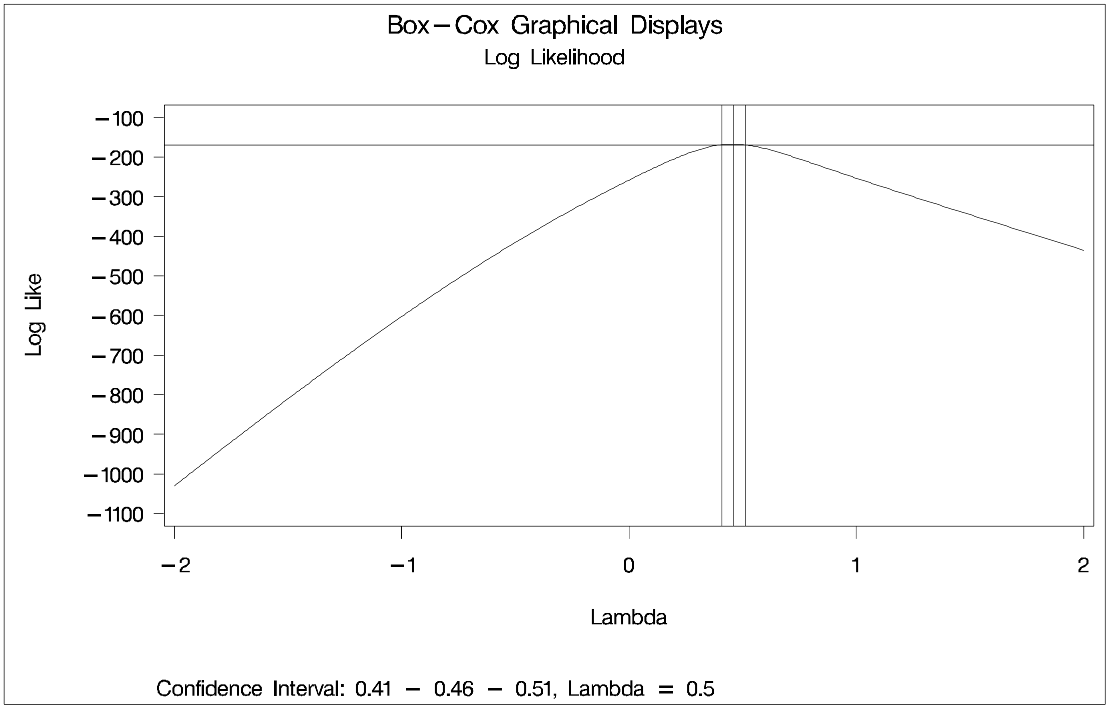
```
* Store values for reference lines;
data _null_;
   set d;
   if description = ’CI Limit’
      then call symput(’vref’,   formattedvalue);
   if description = ’Lambda Used’
      then call symput(’lambda’, formattedvalue);
   run;

data _null_;
   set b end=eof;
   where ci ne ’ ’;
   if _n_ = 1
      then call symput(’href1’, compress(put(lambda, best12.)));
   if ci  = ’<’
      then call symput(’href2’, compress(put(lambda, best12.)));
   if eof
      then call symput(’href3’, compress(put(lambda, best12.)));
   run;
```

These steps plot the log likelihood, root mean square error and R^2. The input data set is the Box-Cox transformation table, which was output using ODS. These statements produce Output 75.6.6.

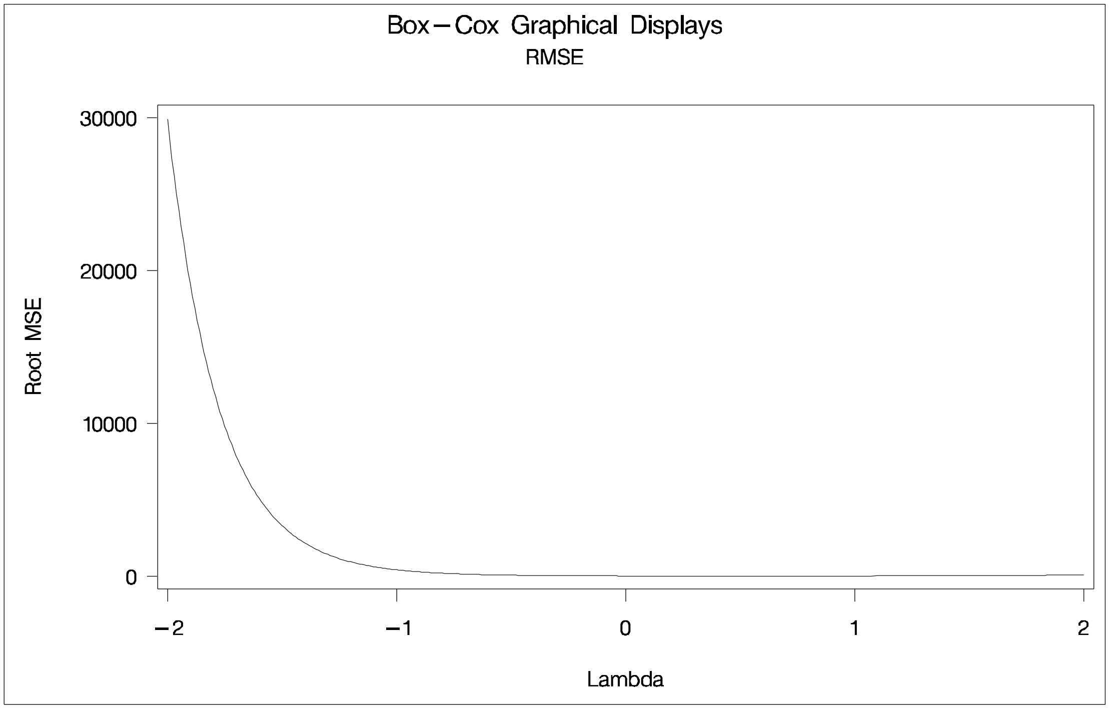
```
* Plot log likelihood, confidence interval;
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot data=b;
   title2 ’Log Likelihood’;
   plot loglike * lambda / vref=&vref href=&href1 &href2 &href3
                           vaxis=axis1 haxis=axis2 frame cframe=ligr;
   footnote "Confidence Interval: &href1 - &href2 - &href3, "
            "Lambda = &lambda";
   symbol v=none i=spline c=blue;
   run;

   footnote;
   title2 ’RMSE’;
   plot rmse * lambda / vaxis=axis1 haxis=axis2 frame cframe=ligr;
   run;

   title2 ’R-Square’;
   plot rsquare * lambda / vaxis=axis1 haxis=axis2 frame cframe=ligr;
```

```
axis1 order=(0 to 1 by 0.1) label=(angle=90 rotate=0) minor=none;
run; quit;
```

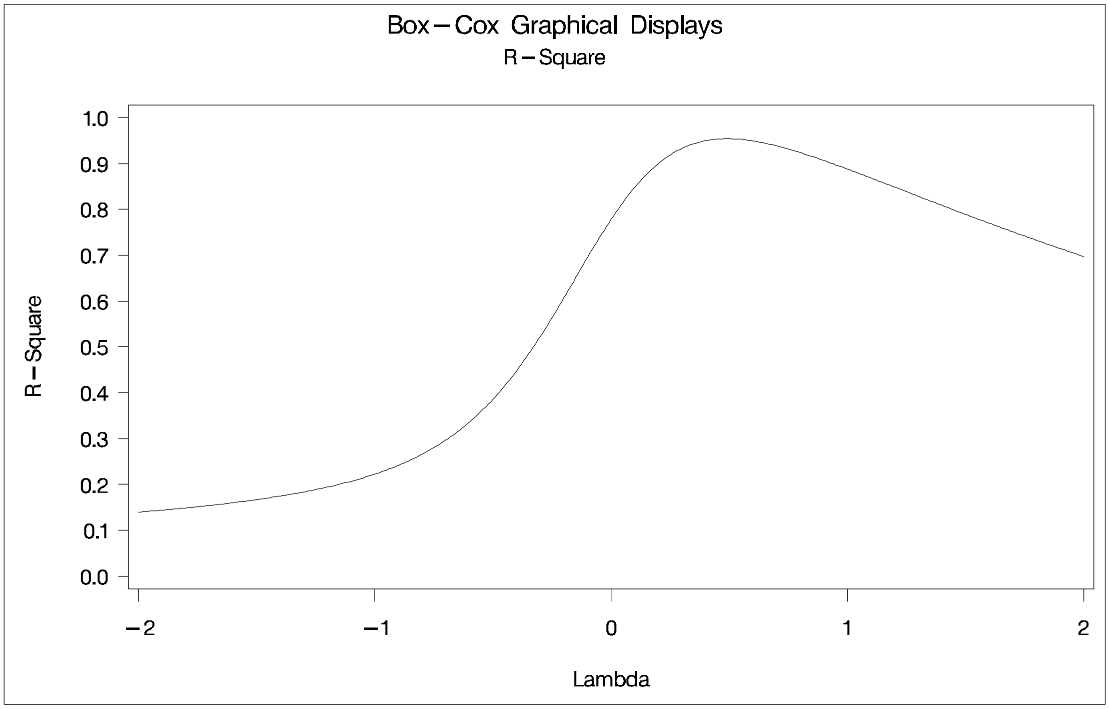
Output 75.6.6. Box-Cox Graphical Displays



Output 75.6.6. (continued)



Output 75.6.6. (continued)

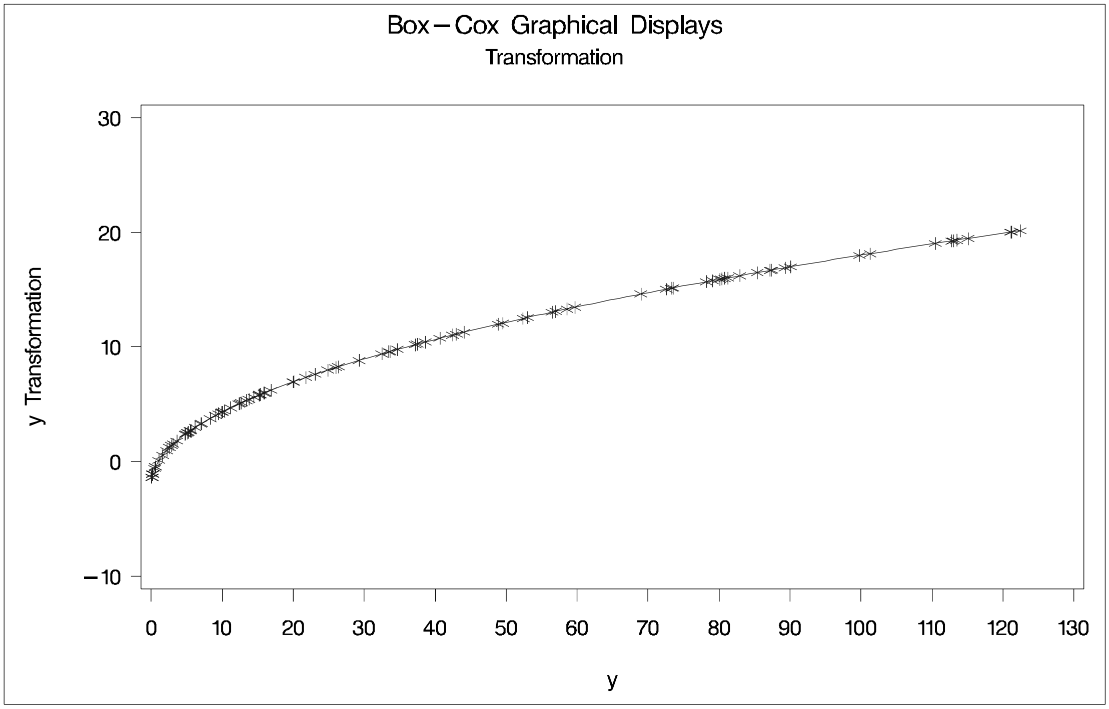


The optimal power parameter is 0.46, but since 0.5 is in the confidence interval, and since the CONVENIENT option was specified, the procedure chooses a square root transformation.

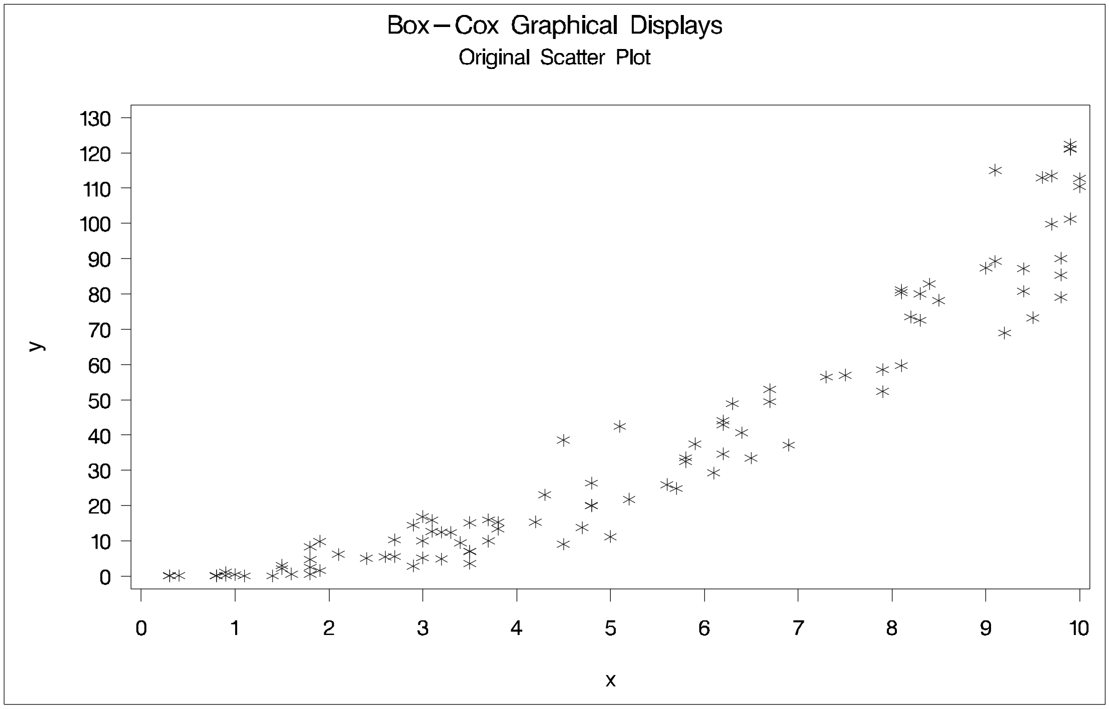
The next steps plot the transformation of Y, the original scatter plot based on the untransformed data, and the new scatter plot based on the transformed data. The results are shown in Output 75.6.7. The input data set is the ordinary output data set from PROC TRANSREG. The transformation of the variable Y by default is Ty.

```
axis1 label=(angle=90 rotate=0) minor=none;
axis2 minor=none;
proc gplot data=trans;
   title2 'Transformation';
   symbol i=splines v=star c=blue;
   plot ty * y / vaxis=axis1 haxis=axis2 frame cframe=ligr;
   run;

   title2 'Original Scatter Plot';
   symbol i=none v=star c=blue;
   plot y * x / vaxis=axis1 haxis=axis2 frame cframe=ligr;
   run;

   title2 'Transformed Scatter Plot';
   symbol i=none v=star c=blue;
   plot ty * x / vaxis=axis1 haxis=axis2 frame cframe=ligr;
   run; quit;
```

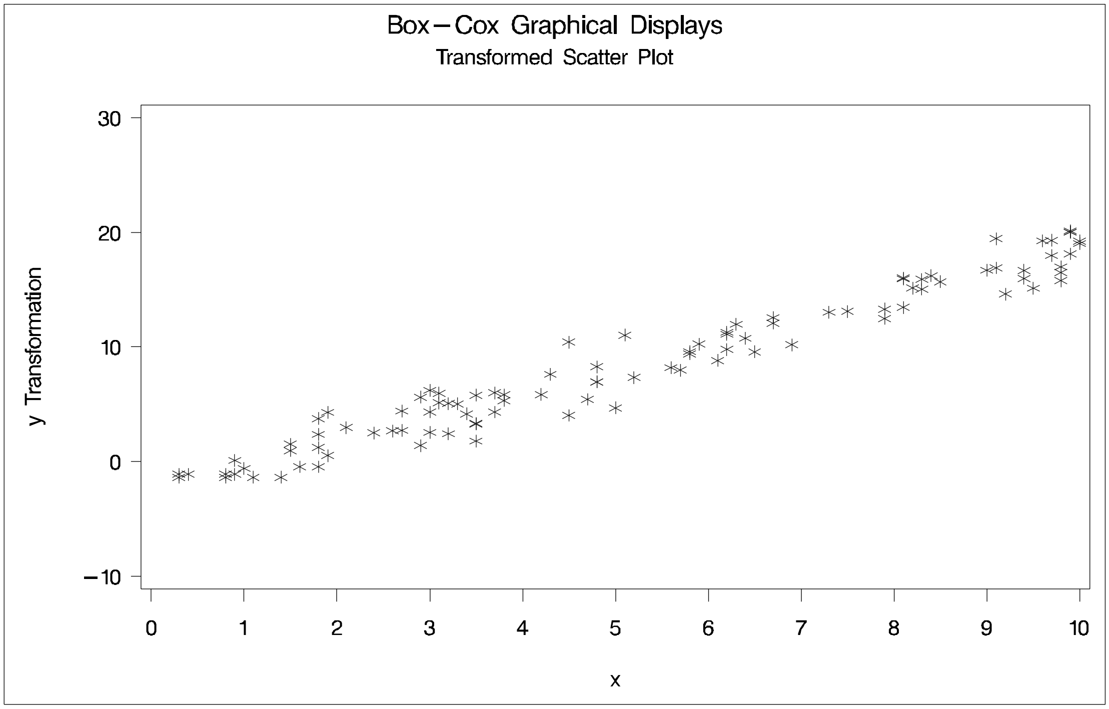
Output 75.6.7. Box-Cox Graphical Displays



Output 75.6.7. (continued)



Output 75.6.7. (continued)



The square root transformation makes the scatter plot essentially linear.

References

de Boor, C. (1978), *A Practical Guide to Splines*, New York: Springer Verlag.

Box, G.E.P. and Cox, D.R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistics Society*, B-26, 211–252.

Breiman, L. and Friedman, J.H. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," (with discussion), *Journal of the American Statistical Association*, 77, 580–619.

Brinkman, N.D. (1981), "Ethanol Fuel—A Single-Cylinder Engine Study of Efficiency and Exhaust Emissions," *Society of Automotive Engineers Transactions*, 90, 1410–1424.

van der Burg, E. and de Leeuw, J. (1983), "Non-linear Canonical Correlation," *British Journal of Mathematical and Statistical Psychology*, 36, 54–80.

Carroll, J.D. (1972), "Individual Differences and Multidimensional Scaling," in R.N. Shepard, A.K. Romney, and S.B. Nerlove (eds.), *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, New York: Seminar Press.

Draper, N.R. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons, Inc.

Fisher, R. (1938), *Statistical Methods for Research Workers (10th Edition)*, Edinburgh: Oliver and Boyd Press.

Gabriel, K.R. (1981), "Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis," *Interpreting Multivariate Data*, ed. Barnett, V., London: John Wiley & Sons.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons, Inc.

Goodnight, J.H. (1978), *SAS Technical Report R-106, The Sweep Operator: Its Importance in Statistical Computing*, Cary NC: SAS Institute Inc.

Green, P.E. and Wind, Y. (1975), "New Way to Measure Consumers' Judgements," *Harvard Business Review*, July–August, 107–117.

Hastie, T. and Tibshirani, R. (1986), "Generalized Additive Models," *Statistical Science*, 3, 297–318.

Israels, A.Z. (1984), "Redundancy Analysis for Qualitative Variables," *Psychometrika*, 49, 331–346.

Khuri, A.I. and Cornell, J.A. (1987), *Response Surfaces*, New York: Marcel Dekker.

Kruskal, J.B. (1964), "Multidimensional Scaling By Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, 29, 1–27.

Kuhfeld, W.F. (2003), "Marketing Research Methods in SAS," [http://support.sas.com/techsup/tnote/tnote_stat.html#market].

de Leeuw, J., Young, F.W., and Takane, Y. (1976), "Additive Structure in Qualitative Data: An Alternating Least Squares Approach with Optimal Scaling Features," *Psychometrika*, 41, 471–503.

de Leeuw, J. (1986), "Regression with Optimal Scaling of the Dependent Variable," Department of Data Theory, The Netherlands: The University of Leiden.

Meyers, R.H. (1976), *Response Surface Methodology*, Blacksburg, VA: Virginia Polytechnic Institute and State University.

van Rijckeveorsel, J. (1982), "Canonical Analysis with B-Splines," in H. Caussinus, P. Ettinger, and R. Tomassone (ed.), *COMPUSTAT 1982*, Part I, Wein, Physica Verlag.

SAS Institute Inc. (1993), SAS Technical Report R-109, *Conjoint Analysis Examples*, Cary, NC: SAS Institute Inc.

Schiffman, S.S., Reynolds, M.L., and Young, F.W. (1981), *Introduction to Multidimensional Scaling*, New York: Academic Press.

Siegel, S. (1956), *Nonparametric Statistics*, New York: McGraw-Hill.

Smith, P.L. (1979), "Splines as a Useful and Convenient Statistical Tool," *The American Statistician*, 33, 57–62.

Stewart, D. and Love, W. (1968), “A General Canonical Correlation Index,” *Psychological Bulletin*, 70, 160–163.

Winsberg, S. and Ramsay, J.O. (1980), “Monotonic Transformations to Additivity Using Splines,” *Biometrika*, 67, 669–674.

Young, F.W. (1981), “Quantitative Analysis of Qualitative Data,” *Psychometrika*, 46, 357–388.

Young, F.W., de Leeuw, J., and Takane, Y. (1976), “Regression with Qualitative and Quantitative Variables: An Alternating Least Squares Approach with Optimal Scaling Features,” *Psychometrika*, 41, 505–529.

Chapter 76
The TREE Procedure

Chapter Contents

Chapter 76
The TREE Procedure

Overview

The TREE procedure produces a tree diagram, also known as a *dendrogram* or *phenogram*, using a data set created by the CLUSTER or VARCLUS procedure. The CLUSTER and VARCLUS procedures create output data sets that contain the results of hierarchical clustering as a tree structure. The TREE procedure uses the output data set to produce a diagram of the tree structure in the style of Johnson(1967), with the root at the top. Alternatively, the diagram can be oriented horizontally, with the root at the left. Any numeric variable in the output data set can be used to specify the heights of the clusters. PROC TREE can also create an output data set containing a variable to indicate the disjoint clusters at a specified level in the tree.

Tree diagrams are discussed in the context of cluster analysis by Duran and Odell (1974), Hartigan (1975), and Everitt (1980). Knuth (1973) provides a general treatment of tree diagrams in computer programming.

The literature on tree diagrams contains a mixture of botanical and genealogical terminology. The objects that are clustered are *leaves*. The cluster containing all objects is the *root*. A cluster containing at least two objects but not all of them is a *branch*. The general term for leaves, branches, and roots is *node*. If a cluster A is the union of clusters B and C, then A is the *parent* of B and C, and B and C are *children* of A. A leaf is thus a node with no children, and a root is a node with no parent. If every cluster has at most two children, the tree diagram is a *binary tree*. The CLUSTER procedure always produces binary trees. The VARCLUS procedure can produce tree diagrams with clusters that have many children.

Getting Started

The TREE procedure creates tree diagrams from a SAS data set containing the tree structure. You can create this type of data set with the CLUSTER or VARCLUS procedure.

In the following example, the VARCLUS procedure is used to divide a set of variables into hierarchical clusters and to create the SAS data set containing the tree structure. The TREE procedure then generates the tree diagrams.

The following data, from Hand, et al. (1994), represent the amount of protein consumed from nine food groups for each of 25 European countries. The nine food groups are red meat (RedMeat), white meat (WhiteMeat), eggs (Eggs), milk (Milk), fish (Fish), cereal (Cereal), starch (Starch), nuts (Nuts), and fruits and vegetables (FruVeg).

The following SAS statements create the data set Protein:

```
data Protein;
   input Country $15. RedMeat WhiteMeat Eggs Milk
      Fish Cereal Starch Nuts FruVeg;
   datalines;
Albania        10.1  1.4  0.5   8.9  0.2  42.3  0.6  5.5  1.7
Austria         8.9 14.0  4.3  19.9  2.1  28.0  3.6  1.3  4.3
Belgium        13.5  9.3  4.1  17.5  4.5  26.6  5.7  2.1  4.0
Bulgaria        7.8  6.0  1.6   8.3  1.2  56.7  1.1  3.7  4.2
Czechoslovakia  9.7 11.4  2.8  12.5  2.0  34.3  5.0  1.1  4.0
Denmark        10.6 10.8  3.7  25.0  9.9  21.9  4.8  0.7  2.4
E Germany       8.4 11.6  3.7  11.1  5.4  24.6  6.5  0.8  3.6
Finland         9.5  4.9  2.7  33.7  5.8  26.3  5.1  1.0  1.4
France         18.0  9.9  3.3  19.5  5.7  28.1  4.8  2.4  6.5
Greece         10.2  3.0  2.8  17.6  5.9  41.7  2.2  7.8  6.5
Hungary         5.3 12.4  2.9   9.7  0.3  40.1  4.0  5.4  4.2
Ireland        13.9 10.0  4.7  25.8  2.2  24.0  6.2  1.6  2.9
Italy           9.0  5.1  2.9  13.7  3.4  36.8  2.1  4.3  6.7
Netherlands     9.5 13.6  3.6  23.4  2.5  22.4  4.2  1.8  3.7
Norway          9.4  4.7  2.7  23.3  9.7  23.0  4.6  1.6  2.7
Poland          6.9 10.2  2.7  19.3  3.0  36.1  5.9  2.0  6.6
Portugal        6.2  3.7  1.1   4.9 14.2  27.0  5.9  4.7  7.9
Romania         6.2  6.3  1.5  11.1  1.0  49.6  3.1  5.3  2.8
Spain           7.1  3.4  3.1   8.6  7.0  29.2  5.7  5.9  7.2
Sweden          9.9  7.8  3.5   4.7  7.5  19.5  3.7  1.4  2.0
Switzerland    13.1 10.1  3.1  23.8  2.3  25.6  2.8  2.4  4.9
UK             17.4  5.7  4.7  20.6  4.3  24.3  4.7  3.4  3.3
USSR            9.3  4.6  2.1  16.6  3.0  43.6  6.4  3.4  2.9
W Germany      11.4 12.5  4.1  18.8  3.4  18.6  5.2  1.5  3.8
Yugoslavia      4.4  5.0  1.2   9.5  0.6  55.9  3.0  5.7  3.2
;
run;
```

The data set Protein contains the character variable Country and the nine numeric variables representing the food groups. The $15. in the INPUT statement specifies that the variable Country is a character variable with a length of 15.

The following statements cluster the variables in the data set Protein. The OUTTREE= option creates an output SAS data set named Tree to contain the tree structure. The CENTROID option specifies the centroid clustering method, and the MAXCLUSTERS= option specifies that the largest number of clusters desired is four. The NOPRINT option suppresses the display of the output. The VAR statement specifies that all numeric variables (RedMeat—FruVeg) are used by the procedure.

```
proc varclus data=Protein outtree=Tree
             centroid maxclusters=4 noprint;
   var RedMeat--FruVeg;
run;
```

The output data set Tree, created by the OUTTREE= option in the previous statements, contains the following variables:

NAME the name of the cluster

PARENT the parent of the cluster

NCL the number of clusters

VAREXP the amount of variance explained by the cluster

PROPOR the proportion of variance explained by the clusters at the current level of the tree diagram

MINPRO the minimum proportion of variance explained by a cluster

MAXEIGEN the maximum second eigenvalue of a cluster

The following statements produce a tree diagram of the clusters created by PROC VARCLUS:

```
proc tree data=tree ;
proc tree data=tree lineprinter;
```

PROC TREE is invoked twice. In the first invocation, the tree diagram is presented using the default high resolution graphical output. In the second invocation, the LINEPRINTER option specifies line printer output.

Figure 76.1 displays the default high resolution graphics version of the tree diagram.

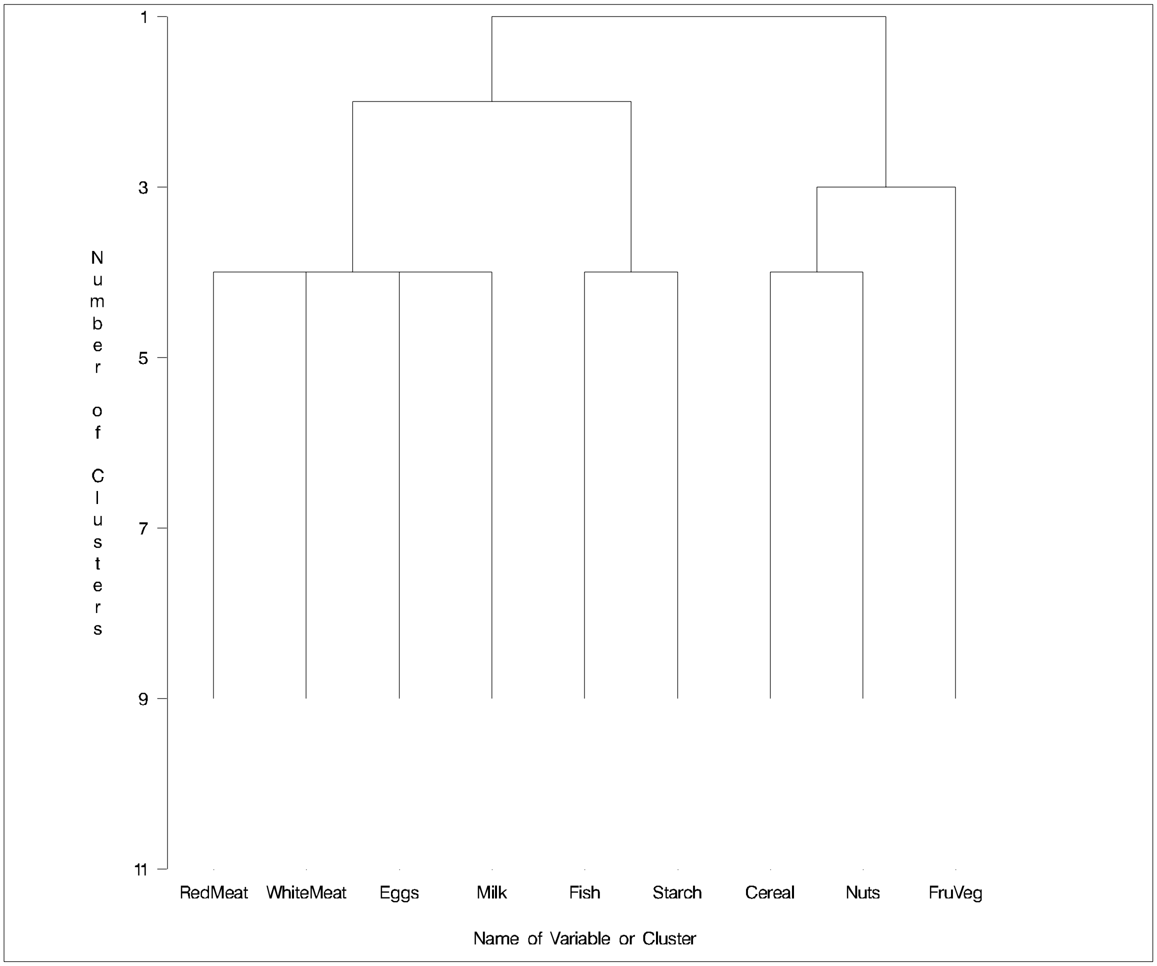
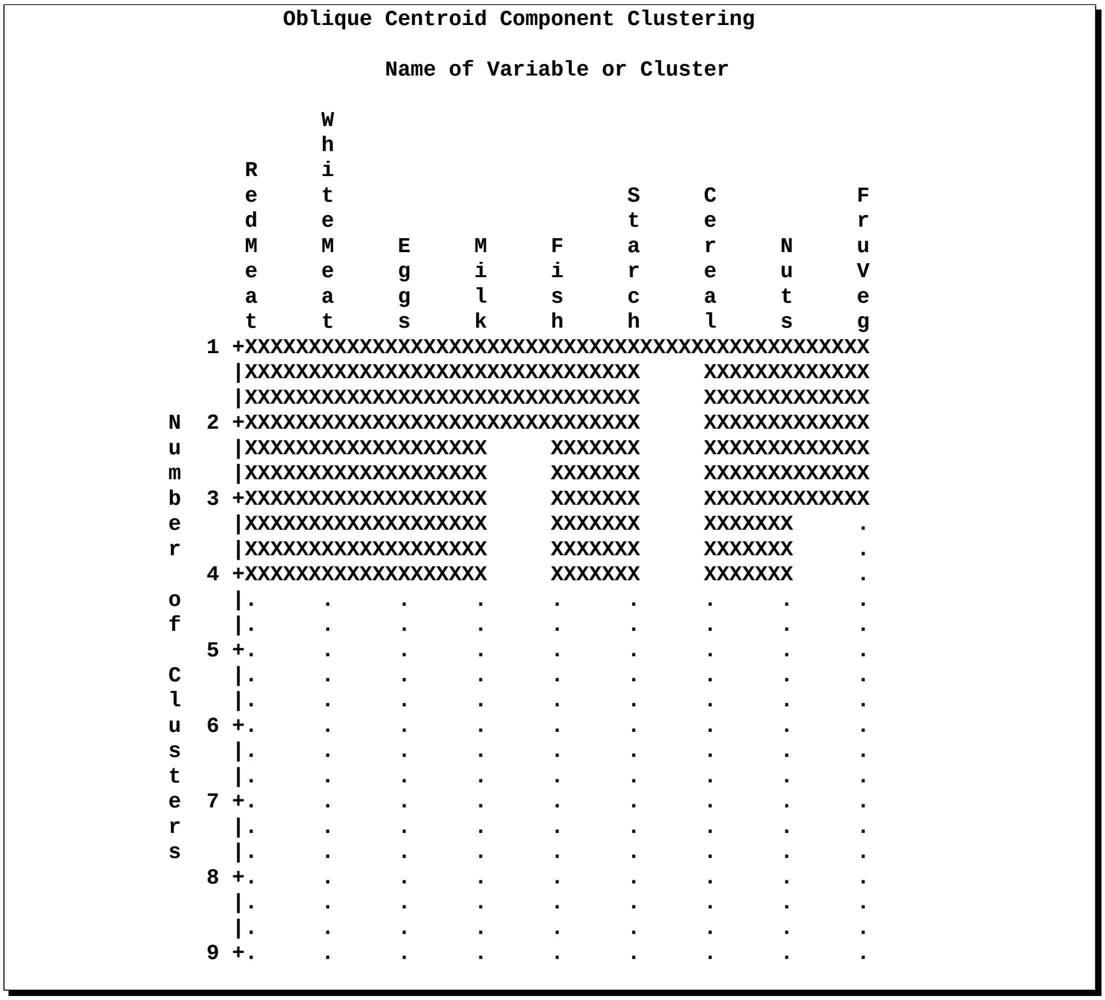


Figure 76.1. High Resolution Tree Diagram from PROC TREE

Figure 76.2 displays the same information as Figure 76.1, using line printer output.



```
         Oblique Centroid Component Clustering

                 Name of Variable or Cluster

            W
            h
      R     i
      e     t                       S     C           F
      d     e                       t     e           r
      M     M     E     M     F     a     r     N     u
      e     e     g     i     i     r     e     u     V
      a     a     g     l     s     c     a     t     e
      t     t     s     k     h     h     l     s     g
   1 +XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
     |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     XXXXXXXXXXXXX
     |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     XXXXXXXXXXXXX
N  2 +XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX     XXXXXXXXXXXXX
u    |XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXXXXXXXX
m    |XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXXXXXXXX
b  3 +XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXXXXXXXX
e    |XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXX     .
r    |XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXX     .
   4 +XXXXXXXXXXXXXXXXXXX     XXXXXXX     XXXXXXX     .
o    |.     .     .     .     .     .     .     .     .
f    |.     .     .     .     .     .     .     .     .
   5 +.     .     .     .     .     .     .     .     .
C    |.     .     .     .     .     .     .     .     .
l    |.     .     .     .     .     .     .     .     .
u  6 +.     .     .     .     .     .     .     .     .
s    |.     .     .     .     .     .     .     .     .
t    |.     .     .     .     .     .     .     .     .
e  7 +.     .     .     .     .     .     .     .     .
r    |.     .     .     .     .     .     .     .     .
s    |.     .     .     .     .     .     .     .     .
   8 +.     .     .     .     .     .     .     .     .
     |.     .     .     .     .     .     .     .     .
     |.     .     .     .     .     .     .     .     .
   9 +.     .     .     .     .     .     .     .     .
```

Figure 76.2. Line Printer Graphics Version of the Tree Diagram

In both figures, the name of the cluster is displayed on the horizontal axis and the number of clusters is displayed on the vertical or height axis.

As you look up from the bottom of the figures, clusters are progressively joined until a single, all-encompassing cluster is formed at the top (or root) of the diagram. Clusters exist at each level of the diagram. For example, at the level where the diagram indicates three clusters, the clusters are as follows:

- Cluster 1: RedMeat WhiteMeat Eggs Milk
- Cluster 2: Fish Starch
- Cluster 3: Cereal Nuts FruVeg

As you proceed up the diagram one level, the number of clusters is two. The clusters are

- Cluster 1: RedMeat WhiteMeat Eggs Milk Fish Starch
- Cluster 2: Cereal Nuts FruVeg

The following statements illustrate how you can specify the numeric variable defining the height of each node (cluster) in the tree. First, the AXIS1 statement is defined. The ORDER= option specifies the data values in the order in which they are to appear on the axis.

Next, the TREE procedure is invoked. The HORIZONTAL option orients the tree diagram horizontally. The HAXIS option specifies that the AXIS1 statement be used to customize the appearance of the horizontal axis. The HEIGHT statement specifies the variable _PROPOR_ (the proportion of variance explained) as the height variable.

```
axis1 order=(0 to 1 by 0.2);
proc tree data=Tree horizontal haxis=axis1;
   height _PROPOR_;
run;
```

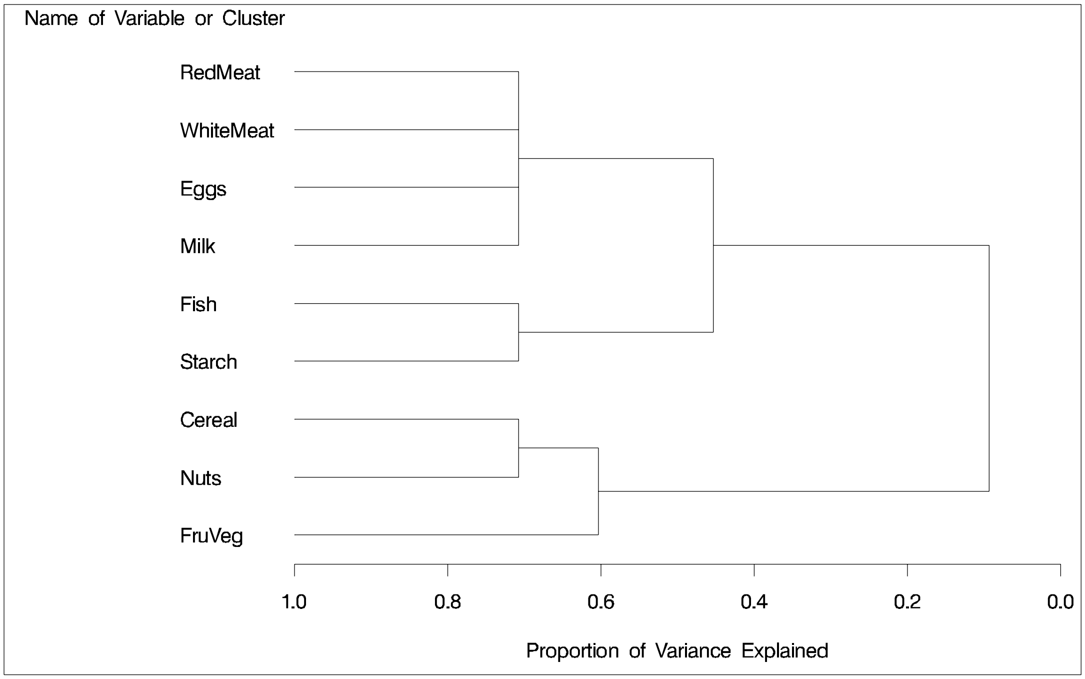


Figure 76.3. Horizontal Tree Diagram Using _PROPOR_ as the HEIGHT Variable

Figure 76.3 displays the tree diagram oriented horizontally, using the variable _PROPOR_ as the height variable. As you look from left to right in the diagram, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the right (or root) of the diagram.

Clusters exist at each level of the diagram, represented by horizontal line segments. Each vertical line segment represents a point where leaves and branches are connected into progressively larger clusters.

For example, three clusters are formed at the left-most point along the axis where three horizontal line segments exist. At that point, where a vertical line segment connects the Cereal-Nuts and FruVeg clusters, the proportion of variance explained is about 0.6 (_PROPOR_ = 0.6). At the next clustering level the variables Fish and Starch are clustered with variables RedMeat through Milk, resulting in a total of two clusters. The proportion of variance explained is about 0.45 at that point.

Syntax

The TREE procedure is invoked by the following statements:

```
PROC TREE < options > ;
    NAME variables ;
    HEIGHT variable ;
    PARENT variables ;
    BY variables ;
    COPY variables ;
    FREQ variable ;
    ID variable ;
```

If the input data set has been created by CLUSTER or VARCLUS, the only statement required is the PROC TREE statement. The BY, COPY, FREQ, HEIGHT, ID, NAME, and PARENT statements are described after the PROC TREE statement.

PROC TREE Statement

PROC TREE < *options* > **;**

The PROC TREE statement starts the TREE procedure.

The options that can appear in the PROC TREE statement are summarized in the following table.

Table 76.1. PROC TREE Statement Options

Task	**Options**	**Effect**
Specify data sets	DATA=	specifies the input data set
	DOCK=	does not count small clusters in OUT= data set
	LEVEL=	defines disjoint cluster in OUT= data set
	NCLUSTERS=	specifies the number of clusters in OUT= data set
	OUT=	specifies the output data set

Table 76.1. (continued)

Task	Options	Effect
	ROOT=	displays the root of a subtree
Specify cluster heights	HEIGHT=	specifies the variable for the height axis
	DISSIMILAR	specifies that large values are far apart
	SIMILAR	specifies that small values are close together
Display horizontal trees	HORIZONTAL	specifies that the height axis is horizontal
Control sort order	DESCENDING	reverses SORT order
	SORT	sorts children by HEIGHT variable
Control displayed output	LIST	displays all nodes in the tree
	NOPRINT	suppresses display of the tree
	LINEPRINTER	displays tree using line printer style graphics
High resolution graphics	INC=	specifies the increment between tick values
	MAXHEIGHT=	specifies the maximum value on axis
	MINHEIGHT=	specifies the minimum value on axis
	NTICK=	specifies the number of tick intervals
	CFRAME=	specifies the color of the frame
	DESCRIPTION=	specifies the catalog description
	GOUT=	specifies the catalog name
	HAXIS=	customizes horizontal axis
	HORDISPLAY=	displays a horizontal tree with leaves on the right
	HPAGES=	specifies the number of pages to expand tree horizontally
	LINES=	specifies the line color and thickness, dots at the nodes
	NAME=	specifies the name of graph in the catalog
	VAXIS=	customizes vertical axis
	VPAGES=	specifies the number of pages to expand tree vertically
Line printer graphics	INC=	specifies the increment between tick values
	MAXHEIGHT=	specifies the maximum value on axis
	MINHEIGHT=	specifies the minimum value on axis
	NTICK=	specifies the number of tick intervals
	PAGES=	specifies the number of pages
	POS=	specifies the number of column positions
	SPACES=	specifies the number of spaces between objects
	TICKPOS=	specifies the number of column positions between ticks
	FILLCHAR=	specifies the fill character between unjoined leaves
	JOINCHAR=	specifies the character to display between joined leaves

Table 76.1. (continued)

Task	Options	Effect
	LEAFCHAR=	specifies the character to represent clusters with no children
	TREECHAR=	specifies the character to represent clusters with children

CFRAME=*color*

specifies a color for the frame, which is the rectangle bounded by the axes.

DATA=*SAS-data-set*

specifies the input data set defining the tree. If you omit the DATA= option, the most recently created SAS data set is used.

DESCENDING
DES

reverses the sorting order for the SORT option.

DESCRIPTION=*entry-description*

specifies a description for the graph in the GOUT= catalog. The default is “Proc Tree Graph Output.”

DISSIMILAR
DIS

implies that the values of the HEIGHT variable are dissimilarities; that is, a large height value means that the clusters are very dissimilar or far apart.

If neither the SIMILAR nor the DISSIMILAR option is specified, PROC TREE attempts to infer from the data whether the height values are similarities or dissimilarities. If PROC TREE cannot tell this from the data, it issues an error message and does not display a tree diagram.

DOCK=*n*

causes observations in the OUT= data set assigned to output clusters with a frequency of n or less to be given missing values for the output variables CLUSTER and CLUSNAME. If the NCLUSTERS= option is also specified, DOCK= also prevents clusters with a frequency of n or less from being counted toward the number of clusters requested by the NCLUSTERS= option. By default, DOCK=0.

FILLCHAR=*’c’*
FC=*’c’*

specifies the character to display between leaves that are not joined into a cluster. The character should be enclosed in single quotes. The default is a blank. The LINEPRINTER option must also be specified.

GOUT=*<libref.>member-name*

specifies the catalog in which the generated graph is stored. The default is WORK.GSEG.

HAXIS=AXIS*n*

specifies the AXISn statement used to customize the appearance of the horizontal axis.

HEIGHT=*name*
H=*name*

specifies certain conventional variables to be used for the height axis of the tree diagram. For many situations, the only option you need is the HEIGHT= option. Valid values for *name* and their meanings are as follows:

HEIGHT | H	specifies the _HEIGHT_ variable.
LENGTH | L	defines the height of each node as its path length from the root. This can also be interpreted as the number of ancestors of the node.
MODE | M	specifies the _MODE_ variable.
NCL | N	specifies the _NCL_ (number of clusters) variable.
RSQ | R	specifies the _RSQ_ variable.

See also the "HEIGHT Statement" section on page 4755, which can specify any variable in the input data set to be used for the height axis. In rare cases, you may need to specify either the DISSIMILAR option or the SIMILAR option.

HORDISPLAY=RIGHT

specifies that the graph is to be oriented horizontally, with the leaf nodes on the right side, when the HORIZONTAL option is also specified. By default, the leaf nodes are on the left side.

HORIZONTAL
HOR

orients the tree diagram with the height axis horizontal and the root at the left. The leaf nodes are on the side specified in the HORDISPLAY= option. If you do not specify the HORIZONTAL option, the height axis is vertical, with the root at the top. When the tree takes up more than one page and is viewed on a screen, horizontal orientation can make the tree diagram considerably easier to read.

HPAGES=*n1*

specifies that the original graph is to be enlarged to cover $n1$ pages. If you also specify the VPAGES=$n2$ option, the original graph is enlarged to cover $n1 \times n2$ graphs. For example, if HPAGES=2 and VPAGES=3, then the original graph is generated followed by $2 \times 3 = 6$ more graphs. In these six graphs, the original is enlarged by a factor of 2 in the horizontal direction and by a factor of 3 in the vertical direction. The graphs are generated in left-to-right and top-to-bottom order.

INC=*n*

specifies the increment between tick values on the height axis. If the HEIGHT variable is _NCL_, the default is usually 1, although a different value can be specified for consistency with other options. For any other HEIGHT variable, the default is some power of 10 times 1, 2, 2.5, or 5.

JOINCHAR=*'c'*
JC=*'c'*

specifies the character to display between leaves that are joined into a cluster. The character should be enclosed in single quotes. The default is X. The LINEPRINTER option must also be specified.

LEAFCHAR=*'c'*
LC=*'c'*

specifies a character to represent clusters having no children. The character should be enclosed in single quotes. The default is a period. The LINEPRINTER option must also be specified.

LEVEL=*n*

specifies the level of the tree defining disjoint clusters for the OUT= data set. The LEVEL= option also causes only clusters between the root and a height of n to be displayed. The clusters in the output data set are those that exist at a height of n on the tree diagram. For example, if the HEIGHT variable is _NCL_ (number of clusters) and LEVEL=5 is specified, then the OUT= data set contains five disjoint clusters. If the HEIGHT variable is _RSQ_ (R^2) and LEVEL=0.9 is specified, then the OUT= data set contains the smallest number of clusters that yields an R^2 of at least 0.9.

LINEPRINTER

specifies that the generated report is to be displayed using line printer graphics.

LINES=(<COLOR=*color***><WIDTH=***n***><DOTS>)**

enables you to specify both the color and the thickness of the lines. In addition, a dot can be drawn at each leaf node. Note that if the frame and the lines are specified to be the same color, PROC TREE selects a different color for the lines.

LIST

lists all the nodes in the tree, displaying the height, parent, and children of each node.

MAXHEIGHT=*n*
MAXH=*n*

specifies the maximum value displayed on the height axis.

MINHEIGHT=*n*
MINH=*n*

specifies the minimum value displayed on the height axis.

NAME=*name*

specifies the entry name for the generated graph in the GOUT= catalog. Note that each time another graph is generated with the same name, the name is modified by appending a number to make it unique.

NCLUSTERS=*n*
NCL=*n*
N=*n*

specifies the number of clusters desired in the OUT= data set. The number of clusters obtained may not equal the number specified if (1) there are fewer than n leaves in the tree, (2) there are more than n unconnected trees in the data set, (3) a multi-way tree does not contain a level with the specified number of clusters, or (4) the DOCK= option eliminates too many clusters.

The NCLUSTERS= option uses the _NCL_ variable to determine the order in which the clusters are formed. If there is no _NCL_ variable, the height variable (as determined by the HEIGHT statement or HEIGHT= option) is used instead.

NTICK=*n*

specifies the number of tick intervals on the height axis. The default depends on the values of other options.

NOPRINT

suppresses the display of the tree. Specify the NOPRINT option if you want only to create an OUT= data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output data set that contains one observation for each object in the tree or subtree being processed and variables called CLUSTER and CLUSNAME showing cluster membership at any specified level in the tree. If you specify the OUT= option, you must also specify either the NCLUSTERS= or LEVEL= option in order to define the output partition level. If you want to create a permanent SAS data set, you must specify a two-level name (refer to "SAS Data Files" in *SAS Language Reference: Concepts*).

PAGES=*n*

specifies the number of pages over which the tree diagram (from root to leaves) is to extend. The default is 1. The LINEPRINTER option must also be specified.

POS=*n*

specifies the number of column positions on the height axis. The default depends on the value of the PAGES= option, the orientation of the tree diagram, and the values specified by the PAGESIZE= and LINESIZE= options. The LINEPRINTER option must also be specified.

ROOT=*'name'*

specifies the value of the NAME variable for the root of a subtree to be displayed if you do not want to display the entire tree. If you also specify the OUT= option, the output data set contains only objects belonging to the subtree specified by the ROOT= option.

SIMILAR
SIM

implies that the values of the HEIGHT variable are similarities; that is, a large height value means that the clusters are very similar or close together.

If neither the SIMILAR nor the DISSIMILAR option is specified, PROC TREE attempts to infer from the data whether the height values are similarities or dissimilarities. If PROC TREE cannot tell this from the data, it issues an error message and does not display a tree diagram.

SORT

sorts the children of each node by the HEIGHT variable, in the order of cluster formation. See the DESCENDING option on page 4750.

SPACES=*s*
S=*s*

specifies the number of spaces between objects on the output. The default depends on the number of objects, the orientation of the tree diagram, and the values specified by the PAGESIZE= and LINESIZE= options. The LINEPRINTER option must also be specified.

TICKPOS=*n*

specifies the number of column positions per tick interval on the height axis. The default value is usually between 5 and 10, although a different value can be specified for consistency with other options.

TREECHAR=*'c'*
TC=*'c'*

specifies a character to represent clusters with children. The character should be enclosed in single quotes. The default is X. The LINEPRINTER option must also be specified.

VAXIS=AXIS*n*

specifies that the AXISn statement be used to customize the appearance of the vertical axis.

VPAGES=*n2*

specifies that the original graph is to be enlarged to cover $n2$ pages. If you also specify the HPAGES=$n1$ option, the original graph is enlarged to cover $n1 \times n2$ pages. For example, if HPAGES=2 and VPAGES=3, then the original graph is generated followed by $2 \times 3 = 6$ more graphs. In these six graphs, the original is enlarged by a factor of 2 in the horizontal direction and by a factor of 3 in the vertical direction. The graphs are generated in left-to-right and top-to-bottom order.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TREE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TREE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

COPY Statement

COPY *variables* **;**

The COPY statement specifies one or more character or numeric variables to be copied to the OUT= data set.

FREQ Statement

FREQ *variables* **;**

The FREQ statement specifies one numeric variable that tells how many clustering observations belong to the cluster. If the FREQ statement is omitted, PROC TREE looks for a variable called _FREQ_ to specify the number of observations per cluster. If neither the FREQ statement nor the _FREQ_ variable is present, each leaf is assumed to represent one clustering observation, and the frequency for each internal node is found by summing the frequencies of its children.

HEIGHT Statement

HEIGHT *variable* **;**

The HEIGHT statement specifies the name of a numeric variable to define the height of each node (cluster) in the tree. The height variable can also be specified by the HEIGHT= option in the PROC TREE statement. If both the HEIGHT statement and the HEIGHT= option are omitted, PROC TREE looks for a variable called _HEIGHT_. If the data set does not contain _HEIGHT_, PROC TREE looks for a variable called _NCL_. If _NCL_ is not found either, the height of each node is defined to be its path length from the root.

ID Statement

ID *variables* **;**

The ID variable is used to identify the objects (leaves) in the tree on the output. The ID variable can be a character or numeric variable of any length. If the ID statement is omitted, the variable in the NAME statement is used instead. If both the ID and NAME statements are omitted, PROC TREE looks for a variable called _NAME_. If the _NAME_ variable is not found in the data set, PROC TREE issues an error message and stops. The ID variable is copied to the OUT= data set.

NAME Statement

NAME *variables* **;**

The NAME statement specifies a character or numeric variable identifying the node represented by each observation. The NAME variable and the PARENT variable jointly define the tree structure. If the NAME statement is omitted, PROC TREE looks for a variable called _NAME_. If the _NAME_ variable is not found in the data set, PROC TREE issues an error message and stops.

PARENT Statement

PARENT *variables* **;**

The PARENT statement specifies a character or numeric variable identifying the node in the tree that is the parent of each observation. The PARENT variable must have the same formatted length as the NAME variable. If the PARENT statement is omitted, PROC TREE looks for a variable called _PARENT_. If the _PARENT_ variable is not found in the data set, PROC TREE issues an error message and stops.

Details

Missing Values

An observation with a missing value for the NAME variable is omitted from processing. If the PARENT variable has a missing value but the NAME variable is present, the observation is treated as the root of a tree. A data set can contain several roots and, hence, several trees.

Missing values of the HEIGHT variable are set to upper or lower bounds determined from the nonmissing values under the assumption that the heights are monotonic with respect to the tree structure.

Missing values of the FREQ variable are inferred from nonmissing values where possible; otherwise, they are treated as zero.

Output Data Set

The OUT= data set contains one observation for each leaf in the tree or subtree being processed. The variables are as follows:

- the BY variables, if any
- the ID variable, or the NAME variable if the ID statement is not used
- the COPY variables
- a numeric variable CLUSTER taking values from 1 to c, where c is the number of disjoint clusters. The cluster to which the first observation belongs is given the number 1, the cluster to which the next observation belongs that does not belong to cluster 1 is given the number 2, and so on.

- a character variable CLUSNAME giving the value of the NAME variable of the cluster to which the observation belongs

The CLUSTER and CLUSNAME variables are missing if the corresponding leaf has a nonpositive frequency.

Displayed Output

The displayed output from the TREE procedure includes the following:

- the names of the objects in the tree
- the height axis
- the tree diagram. A high-resolution graphics tree diagram is produced on the graphics device. The leaves are displayed at the bottom of the graph. Horizontal lines connect the leaves into branches, while the topmost horizontal line indicates the root.

  If the LINEPRINTER option is specified, the root (the cluster containing all the objects) is indicated by a solid line of the character specified by the TREECHAR= option (the default character is 'X'). At each level of the tree, clusters are shown by unbroken lines of the TREECHAR= symbol with the FILLCHAR= symbol (the default is a blank) separating the clusters. The LEAFCHAR= symbol (the default character is a period) represents single-member clusters.

By default, the tree diagram is oriented with the height axis vertical and the object names at the top of the diagram. If the HORIZONTAL option is specified, then the height axis is horizontal and the object names are on the left.

ODS Table Names

PROC TREE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, "Using the Output Delivery System."

Table 76.2. ODS Tables Produced in PROC TREE

ODS Table Name	Description	Statement	Option
Tree	Line-printer plot of the tree	PROC	LINEPRINTER
TreeListing	Line-printer listing of all nodes in the tree	PROC	LIST

Examples

Example 76.1. Mammals' Teeth

The following data give the numbers of different kinds of teeth for a variety of mammals. The mammals are clustered by average linkage using the CLUSTER procedure (Output 76.1.1). The PROC TREE statement uses the average-linkage distance as the height axis, which is the default, and creates a horizontal high-resolution graphics tree (Output 76.1.2).

```
data teeth;
   title 'Mammals'' Teeth';
   input mammal $ 1-16 @21 (v1-v8) (1.);
   label V1='Right Top Incisors'
         V2='Right Bottom Incisors'
         V3='Right Top Canines'
         V4='Right Bottom Canines'
         V5='Right Top Premolars'
         V6='Right Bottom Premolars'
         V7='Right Top Molars'
         V8='Right Bottom Molars';
   datalines;
Brown Bat           23113333
Mole                32103333
Silver Hair Bat     23112333
Pigmy Bat           23112233
House Bat           23111233
Red Bat             13112233
Pika                21002233
Rabbit              21003233
Beaver              11002133
Groundhog           11002133
Gray Squirrel       11001133
House Mouse         11000033
Porcupine           11001133
Wolf                33114423
Bear                33114423
Raccoon             33114432
Marten              33114412
Weasel              33113312
Wolverine           33114412
Badger              33113312
River Otter         33114312
Sea Otter           32113312
Jaguar              33113211
Cougar              33113211
Fur Seal            32114411
Sea Lion            32114411
Grey Seal           32113322
Elephant Seal       21114411
Reindeer            04103333
Elk                 04103333
Deer                04003333
```

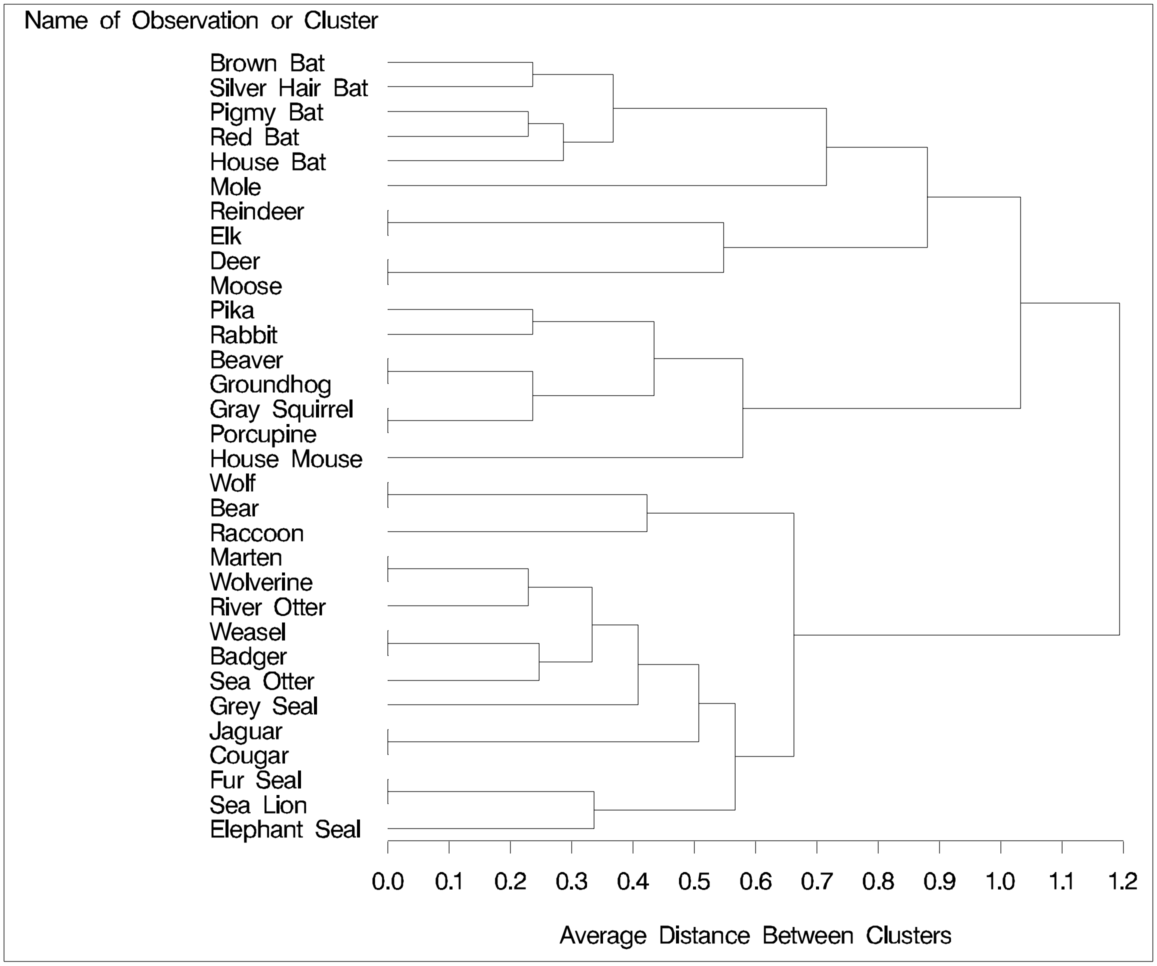
```
Moose                04003333
;
options pagesize=60 linesize=110;

proc cluster method=average std pseudo noeigen outtree=tree;
   id mammal;
   var v1-v8;
run;

proc tree graphics horizontal;
run;
```

Output 76.1.1 displays the information on how the clusters are joined. For example, the cluster history shows that the observations Wolf and Bear form cluster 29, which is merged with Raccoon to form cluster 11.

Output 76.1.1. Output from PROC CLUSTER

```
                                     Mammals' Teeth

                                  The CLUSTER Procedure
                              Average Linkage Cluster Analysis

          The data have been standardized to mean 0 and variance 1
          Root-Mean-Square Total-Sample Standard Deviation =        1
          Root-Mean-Square Distance Between Observations   =        4


                                     Cluster History
                                                                      Norm   T
                                                                       RMS   i
 NCL   ----------Clusters Joined----------     FREQ     PSF   PST2     Dist   e

  31   Beaver             Groundhog               2       .      .        0   T
  30   Gray Squirrel      Porcupine               2       .      .        0   T
  29   Wolf               Bear                    2       .      .        0   T
  28   Marten             Wolverine               2       .      .        0   T
  27   Weasel             Badger                  2       .      .        0   T
  26   Jaguar             Cougar                  2       .      .        0   T
  25   Fur Seal           Sea Lion                2       .      .        0   T
  24   Reindeer           Elk                     2       .      .        0   T
  23   Deer               Moose                   2       .      .        0
  22   Pigmy Bat          Red Bat                 2     281      .   0.2289
  21   CL28               River Otter             3     139      .   0.2292
  20   CL31               CL30                    4    83.2      .   0.2357   T
  19   Brown Bat          Silver Hair Bat         2    76.7      .   0.2357   T
  18   Pika               Rabbit                  2    73.2      .   0.2357
  17   CL27               Sea Otter               3    67.4      .   0.2462
  16   CL22               House Bat               3    62.9    1.7   0.2859
  15   CL21               CL17                    6    47.4    6.8   0.3328
  14   CL25               Elephant Seal           3    45.0      .   0.3362
  13   CL19               CL16                    5    40.8    3.5   0.3672
  12   CL15               Grey Seal               7    38.9    2.8   0.4078
  11   CL29               Raccoon                 3    38.0      .    0.423
  10   CL18               CL20                    6    34.5   10.3   0.4339
   9   CL12               CL26                    9    30.0    7.3   0.5071
   8   CL24               CL23                    4    28.7      .   0.5473
   7   CL9                CL14                   12    25.7    7.0   0.5668
   6   CL10               House Mouse             7    28.3    4.1   0.5792
   5   CL11               CL7                    15    26.8    6.9   0.6621
   4   CL13               Mole                    6    31.9    7.2   0.7156
   3   CL4                CL8                    10    31.0   12.7   0.8799
   2   CL3                CL6                    17    27.8   16.1   1.0316
   1   CL2                CL5                    32       .   27.8   1.1938
```

Output 76.1.2. PROC TREE High-Resolution Graphics



As you look from left-to-right in the diagram in Output 76.1.2, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the right (or root) of the diagram. Clusters exist at each level of the diagram, and every vertical line connects leaves and branches into progressively larger clusters. For example, the five bats form a cluster at the 0.6 level, while the next cluster consists only of the mole. The observations Reindeer, Elk, Deer, and Moose form the next cluster at the 0.6 level, the mammals Pika through House Mouse are in the fourth cluster, The observations Wolf, Bear, and Raccoon form the fifth cluster, while the last cluster contains the observations Marten through Elephant Seal.

The following statements create the same tree with line printer graphics in a vertical orientation; the tree is displayed in Output 76.1.3.

```
proc tree lineprinter;
run;
```

Output 76.1.3. PROC TREE with the LINEPRINTER Option

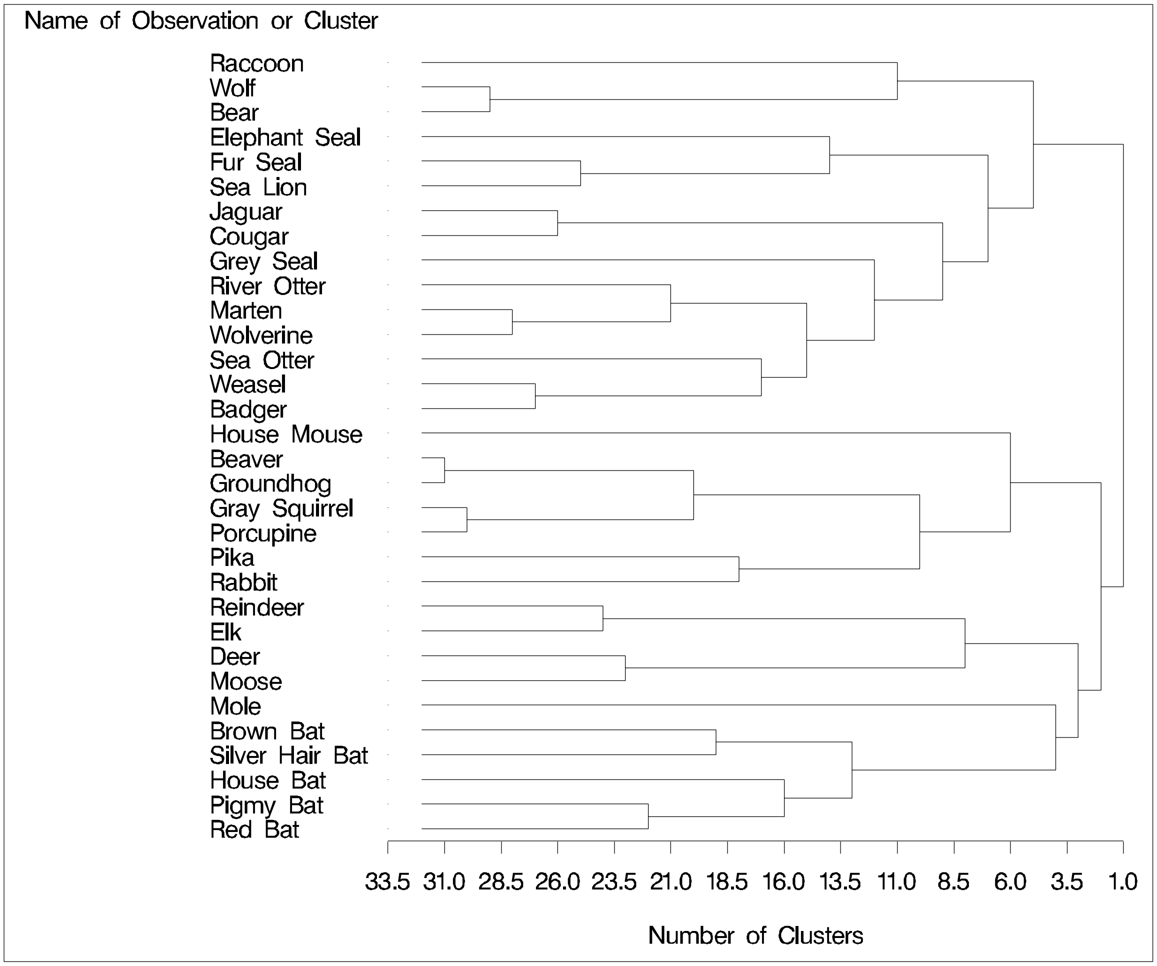
```
                    Average Linkage Cluster Analysis

                        Name of Observation or Cluster

          S
          i
          l                         G                                 E
          v                         r                                 l
          e                         a   H           R                 e
          r                         y   o           i                 p
        B   P   H                 G   P u         W v     S G         h
        r H i   o   R             r S o s         o e     e r     F S a
        o a g R u   e             o q r e     R   l r     a e     u e n
        w i m e s   i         R B u u c       a M v   W B   y J C r a t
        n r y d e   n     M   a e n i u M     c a e O e a O   a o
                  M d   D o P b a d r p o W B c r r t a d t S g u S L S
        B B B B B o e E e o i b v h r i u o e o t i t s g t e u g e i e
        a a a a a l e l e s k i e o e n s l a o e n e e e e a a a a o a
        t t t t t e r k r e a t r g l e e f r n n e r l r r l r r l n l
A  1.5 +
v      |
e      |
r      |
a      |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
g      |XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
e    1 +XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       |XXXXXXXXXXXXXXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
D      |XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
i      |XXXXXXXXXXX XXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
s      |XXXXXXXXX . XXXXXXX XXXXXXXXXXXXX XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
t      |XXXXXXXXX . XXXXXXX XXXXXXXXXXXXX XXXXX XXXXXXXXXXXXXXXXXXXXXXX
a  0.5 +XXXXXXXXX . XXX XXX XXXXXXXXXXX . XXXXX XXXXXXXXXXXXXXXX XXXXX
n      |XXXXXXXXX . XXX XXX XXXXXXXXXXX . XXXXX XXXXXXXXXXXXX XXX XXXXX
c      |XXXXXXXXX . XXX XXX XXX XXXXXXX . XXX . XXXXXXXXXXX . XXX XXXXX
e      |XXX XXXXX . XXX XXX XXX XXXXXXX . XXX . XXXXX XXXXX . XXX XXX .
       |. . . . . . XXX XXX . . XXX XXX . XXX . XXX . XXX . . XXX XXX .
B      |. . . . . . XXX XXX . . XXX XXX . XXX . XXX . XXX . . XXX XXX .
e    0 +. . . . . . XXX XXX . . XXX XXX . XXX . XXX . XXX . . XXX XXX .
t
w
```

As you look up from the bottom of the diagram, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the top (or root) of the diagram. Clusters exist at each level of the diagram. For example, the unbroken line of Xs at the left-most side of the 0.6 level indicates that the five bats have formed a cluster. The next cluster is represented by a period because it contains only one mammal, Mole. Reindeer, Elk, Deer, and Moose form the next cluster, indicated by Xs again. The mammals Pika through House Mouse are in the fourth cluster. The observations Wolf, Bear, and Raccoon form the fifth cluster, while the last cluster contains the observations Marten through Elephant Seal.

The next statement sorts the clusters at each branch in order of formation and uses the number of clusters as the height axis. The resulting tree is displayed in Output 76.1.4.

```
proc tree sort height=n horizontal;
run;
```

Output 76.1.4. PROC TREE with SORT and HEIGHT= Options



Because the CLUSTER procedure always produces binary trees, the number of internal (root and branch) nodes in the tree is one less than the number of leaves. Therefore 31 clusters are formed from the 32 mammals in the input data set. These are represented by the 31 vertical line segments in the tree diagram, each at a different value along the horizontal axis.

As you examine the tree from left to right, the first vertical line segment is where Beaver and Groundhog are clustered and the number of clusters is 31. The next cluster is formed from Gray Squirrel and Porcupine. The third contains Wolf and Bear. Note how the tree graphically displays the clustering order information that was presented in tabular form by the CLUSTER procedure in Output 76.1.1.

The same clusters as in Output 76.1.2 and Output 76.1.3 can be seen at the six-cluster level of the tree diagram in Output 76.1.4, although the SORT and HEIGHT= options make them appear in a different order.

The following statements create these six clusters and display them in Output 76.1.5. The PROC TREE statement produces no output but creates an output data set indicating the cluster to which each observation belongs at the six-cluster level in the tree.

```
proc tree noprint out=part nclusters=6;
   id mammal;
   copy v1-v8;
proc sort;
   by cluster;
proc print label uniform;
   id mammal;
   var v1-v8;
   format v1-v8 1.;
   by cluster;
run;
```

Output 76.1.5. PROC TREE OUT= Data Set

```
---------------------------------- CLUSTER=1 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top      Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Beaver                1           1           0          0
         Groundhog             1           1           0          0
         Gray Squirrel         1           1           0          0
         Porcupine             1           1           0          0
         Pika                  2           1           0          0
         Rabbit                2           1           0          0
         House Mouse           1           1           0          0

                                           Right      Right     Right
                            Right Top     Bottom       Top      Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Beaver                 2            1          3         3
         Groundhog              2            1          3         3
         Gray Squirrel          1            1          3         3
         Porcupine              1            1          3         3
         Pika                   2            2          3         3
         Rabbit                 3            2          3         3
         House Mouse            0            0          3         3


---------------------------------- CLUSTER=2 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top      Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Wolf                  3           3           1          1
         Bear                  3           3           1          1
         Raccoon               3           3           1          1

                                           Right      Right     Right
                            Right Top     Bottom       Top      Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Wolf                   4            4          2         3
         Bear                   4            4          2         3
         Raccoon                4            4          3         2
```

```
---------------------------------- CLUSTER=3 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top      Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Marten                3           3           1          1
         Wolverine             3           3           1          1
         Weasel                3           3           1          1
         Badger                3           3           1          1
         Jaguar                3           3           1          1
         Cougar                3           3           1          1
         Fur Seal              3           2           1          1
         Sea Lion              3           2           1          1
         River Otter           3           3           1          1
         Sea Otter             3           2           1          1
         Elephant Seal         2           1           1          1
         Grey Seal             3           2           1          1

                                           Right      Right     Right
                            Right Top     Bottom      Top       Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Marten                 4            4          1         2
         Wolverine              4            4          1         2
         Weasel                 3            3          1         2
         Badger                 3            3          1         2
         Jaguar                 3            2          1         1
         Cougar                 3            2          1         1
         Fur Seal               4            4          1         1
         Sea Lion               4            4          1         1
         River Otter            4            3          1         2
         Sea Otter              3            3          1         2
         Elephant Seal          4            4          1         1
         Grey Seal              3            3          2         2
```

```
---------------------------------- CLUSTER=4 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top       Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Reindeer              0           4           1          0
         Elk                   0           4           1          0
         Deer                  0           4           0          0
         Moose                 0           4           0          0

                                           Right      Right     Right
                            Right Top     Bottom       Top      Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Reindeer               3            3          3         3
         Elk                    3            3          3         3
         Deer                   3            3          3         3
         Moose                  3            3          3         3


---------------------------------- CLUSTER=5 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top       Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Pigmy Bat             2           3           1          1
         Red Bat               1           3           1          1
         Brown Bat             2           3           1          1
         Silver Hair Bat       2           3           1          1
         House Bat             2           3           1          1

                                           Right      Right     Right
                            Right Top     Bottom       Top      Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Pigmy Bat              2            2          3         3
         Red Bat                2            2          3         3
         Brown Bat              3            3          3         3
         Silver Hair Bat        2            3          3         3
         House Bat              1            2          3         3
```

```
---------------------------------- CLUSTER=6 -----------------------------------

                             Right       Right       Right      Right
                              Top        Bottom       Top       Bottom
         mammal             Incisors    Incisors    Canines    Canines

         Mole                  3           2           1          0

                                           Right      Right     Right
                            Right Top     Bottom       Top      Bottom
         mammal             Premolars    Premolars    Molars    Molars

         Mole                   3            3          3         3
```

Example 76.2. Iris Data

Fisher’s (1936) iris data gives sepal and petal dimensions for three different species of iris. The data are clustered by kth-nearest-neighbor density linkage using the CLUSTER procedure with K=8. Observations are identified by species (Setosa, Versicolor or Virginica) in the tree diagram, which is oriented with the height axis horizontal. The following statements produce Output 76.2.1 and Output 76.2.2.

```
proc format;
   value specname
      1=’Setosa    ’
      2=’Versicolor’
      3=’Virginica ’;
run;

data iris;
   title ’Fisher (1936) Iris Data’;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength=’Sepal Length in mm.’
         SepalWidth =’Sepal Width in mm.’
         PetalLength=’Petal Length in mm.’
         PetalWidth =’Petal Width in mm.’;
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
```

```
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
proc cluster data=iris method=twostage print=10
             outtree=tree k=8 noeigen;
   var SepalLength SepalWidth PetalLength PetalWidth;
   copy Species;
   id Species;
run;

options pagesize=60 linesize=110;

proc tree data=tree horizontal lineprinter pages=1 maxh=10;
   id species;
run;
```

The PAGES=1 option specifies that the tree diagram extends over one page from tree to root. Since the HORIZONTAL option is also specified, the horizontal extent of the diagram is one page. The number of vertical pages required for the diagram is dictated by the number of leaves in the tree.

The MAXH=10 limits the values displayed on the height axis to a maximum of 10. This prunes the tree diagram so that only the portion from the leaves to level 10 is displayed. You can see this pruning effect in Output 76.2.2.

Output 76.2.1. Clustering of Fisher's Iris Data

```
                                   Fisher (1936) Iris Data

                                    The CLUSTER Procedure
                             Two-Stage Density Linkage Clustering

                                           K = 8
                   Root-Mean-Square Total-Sample Standard Deviation = 10.69224


                                       Cluster History
                                                   Normalized       Maximum Density      T
                                                       Fusion        in Each Cluster      i
 NCL    ----Clusters Joined-----       FREQ        Density      Lesser      Greater    e

  10    CL11          Versicolor         48         0.2879      0.1479       8.3678
   9    CL13          Virginica          46         0.2802      0.2005       3.5156
   8    CL10          Virginica          49         0.2699      0.1372       8.3678
   7    CL8           Versicolor         50         0.2586      0.1372       8.3678
   6    CL9           Virginica          47         0.1412      0.0832       3.5156
   5    CL6           Virginica          48          0.107      0.0605       3.5156
   4    CL5           Virginica          49         0.0969      0.0541       3.5156
   3    CL4           Virginica          50         0.0715      0.0370       3.5156
   2    CL3           CL7               100         2.6277      3.5156       8.3678
                              3 modal clusters have been formed.
```

Output 76.2.2. Horizontal Tree for Fisher’s Iris Data

```
                                   Two-Stage Density Linkage Clustering

                                                Cluster Fusion Density

              0        1        2        3        4        5        6        7        8        9       10
              +--------+--------+--------+--------+--------+--------+--------+--------+--------+--------+
S   Virginica XX.........................................................................................
p             XX
e   Virginica XX.........................................................................................
c             XX
i   Virginica XXXX.......................................................................................
e             XXXX
s   Virginica XXXX.......................................................................................
              XXXX
    Virginica XXXXXX.....................................................................................
              XXXXXX
   Versicolor XXXXXXX....................................................................................
              XXXXXXX
    Virginica XXXXXXXX...................................................................................
              XXXXXXXX
    Virginica XXXXXXXXX..................................................................................
              XXXXXXXXX
    Virginica XXXXXXXXXXXXXXX............................................................................
              XXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXX...........................................................................
              XXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXX.........................................................................
              XXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXX.......................................................................
              XXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXX......................................................................
              XXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXX.....................................................................
              XXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXX.....................................................................
              XXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXX.....................................................................
              XXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXX....................................................................
              XXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXX....................................................................
              XXXXXXXXXXXXXXXXXXXXXXX
   Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXX................................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXX................................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
              XXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXXX...............................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXXX................................................................
              XXXXXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
              XXXXXXXXXXXXXXXXXXXXXXX
    Virginica XXXXXXXXXXXXXXXXXXXXXXX....................................................................
              XXXXXXXXXXXXXXXXXXXXXX
```

```
      Versicolor XXXXXXXXXXXXXXXXXXXXXX.....................................................................
                 XXXXXXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXXXXXX.....................................................................
                 XXXXXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXXXXX......................................................................
                 XXXXXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXXXXX......................................................................
                 XXXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXXX........................................................................
                 XXXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXXX........................................................................
                 XXXXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXXXX.........................................................................
                 XXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXX...........................................................................
                 XXXXXXXXXXXXXXXX
       Virginica XXXXXXXXXXXXXXXX...........................................................................
                 XXXXXXXXXXX
       Virginica XXXXXXXXXXX................................................................................
                 XXXXXXXXXXX
       Virginica XXXXXXXXXXX................................................................................
                 XXXXXXXXX
       Virginica XXXXXXXXX..................................................................................
                 XXXXXXXX
       Virginica XXXXXXXX...................................................................................
                 XXXXXX
       Virginica XXXXXX.....................................................................................
                 XXXXXX
       Virginica XXXXXX.....................................................................................
                 XXXXXX
       Virginica XXXXXX.....................................................................................
                 XXXXXX
       Virginica XXXXXX.....................................................................................
                 XXXXX
       Virginica XXXXX......................................................................................
                 XX
       Virginica XX.........................................................................................
                 XX
       Virginica XX.........................................................................................
                 X
       Virginica XXX........................................................................................
                 XXX
      Versicolor XXXX.......................................................................................
                 XXXX
      Versicolor XXXXXXXXXXX................................................................................
                 XXXXXXXXXXX
      Versicolor XXXXXXXXXXXXX..............................................................................
                 XXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXX............................................................................
                 XXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXX...........................................................................
                 XXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXX..........................................................................
                 XXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXX.........................................................................
                 XXXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXXXX.......................................................................
                 XXXXXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXXXXXX.....................................................................
                 XXXXXXXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXXXXXXXXX..................................................................
                 XXXXXXXXXXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXXXXXXXXX..................................................................
                 XXXXXXXXXXXXXXXXXXXXXXXXX
      Versicolor XXXXXXXXXXXXXXXXXXXXXXXXX..................................................................
                 XXXXXXXXXXXXXXXXXXXXXXXXX
```

```
 Virginica XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.............................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...........................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.............................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.........................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.........................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...............................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..........................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
           XXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXX.................................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...........................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX........................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX........................................................
           XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...........................................................
           XXXXXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXXXXX....................................................................
           XXXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXXX.......................................................................
           XXXXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXXXX........................................................................
           XXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXX..........................................................................
           XXXXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXXXX..........................................................................
           XXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXX.............................................................................
           XXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXX..............................................................................
           XXXXXXXXXX
```

```
      Versicolor XXXXXXXXXX................................................................................
                 XXXX
      Versicolor XXXX......................................................................................
                 XXXX
      Versicolor XXXX......................................................................................
                 XXX
      Versicolor XXX.......................................................................................

          Setosa XXXXXXXXXXXXXXXX..........................................................................
                 XXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXX................................................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...........................................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..............................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..........................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.........................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..................................
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX.
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
          Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                 XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX..........................................
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX............................................
       XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX........................................................
       XXXXXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXXXXX.....................................................................
       XXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXX........................................................................
       XXXXXXXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXXXXXXX........................................................................
       XXXXXXXXXXXXXX
Setosa XXXXXXXXXXXXXX.............................................................................
       XXXXXXXXX
Setosa XXXXXXXXX..................................................................................
       XXXXX
Setosa XXXXX......................................................................................
       XXXX
Setosa XXXX.......................................................................................
```

References

Duran, B.S. and Odell, P.L. (1974), *Cluster Analysis,* New York: Springer-Verlag.

Everitt, B.S. (1980), *Cluster Analysis,* Second Edition, London: Heineman Educational Books Ltd.

Fisher, R.A. (1936), “The Use of Multiple Measurements in Taxonomic Problems,” *Annals of Eugenics*, 7, 179–188.

Hand, D.J.; Daly, F.; Lunn, A.D.; McConway, K.J.; and Ostrowski E. (1994), *A Handbook of Small Data Sets,* London: Chapman & Hall, 297–298.

Hartigan, J.A. (1975), *Clustering Algorithms,* New York: John Wiley & Sons, Inc.

Johnson, S.C. (1967), “Hierarchical Clustering Schemes,” *Psychometrika*, 32, 241–254.

Knuth, D.E. (1973), *The Art of Computer Programming, Volume 1, Fundamental Algorithms*, Reading, MA: Addison-Wesley Publishing Co., Inc.

Chapter 77
The TTEST Procedure

Chapter Contents

Chapter 77
The TTEST Procedure

Overview

The TTEST procedure performs t tests for one sample, two samples, and paired observations. The one-sample t test compares the mean of the sample to a given number. The two-sample t test compares the mean of the first sample minus the mean of the second sample to a given number. The paired observations t test compares the mean of the differences in the observations to a given number.

For one-sample tests, PROC TTEST computes the sample mean of the variable and compares it with a given number. Paired comparisons use the one sample process on the differences between the observations. Paired comparisons can be made between many pairs of variables with one call to PROC TTEST. For group comparisons, PROC TTEST computes sample means for each of two groups of observations and tests the hypothesis that the population means differ by a given amount. This latter analysis can be considered a special case of a one-way analysis of variance with two levels of classification.

The underlying assumption of the t test in all three cases is that the observations are random samples drawn from normally distributed populations. This assumption can be checked using the UNIVARIATE procedure; if the normality assumptions for the t test are not satisfied, you should analyze your data using the NPAR1WAY procedure. The two populations of a group comparison must also be independent. If they are not independent, you should question the validity of a paired comparison.

PROC TTEST computes the group comparison t statistic based on the assumption that the variances of the two groups are equal. It also computes an approximate t based on the assumption that the variances are unequal (the Behrens-Fisher problem). The degrees of freedom and probability level are given for each; Satterthwaite's (1946) approximation is used to compute the degrees of freedom associated with the approximate t. In addition, you can request the Cochran and Cox (1950) approximation of the probability level for the approximate t. The folded form of the F statistic is computed to test for equality of the two variances (Steel and Torrie 1980).

FREQ and WEIGHT statements are available. Data can be input in the form of observations or summary statistics. Summary statistics and their confidence intervals, and differences of means are output. For two-sample tests, the pooled-variance and a test for equality of variances are also produced.

Getting Started

One-Sample *t* Test

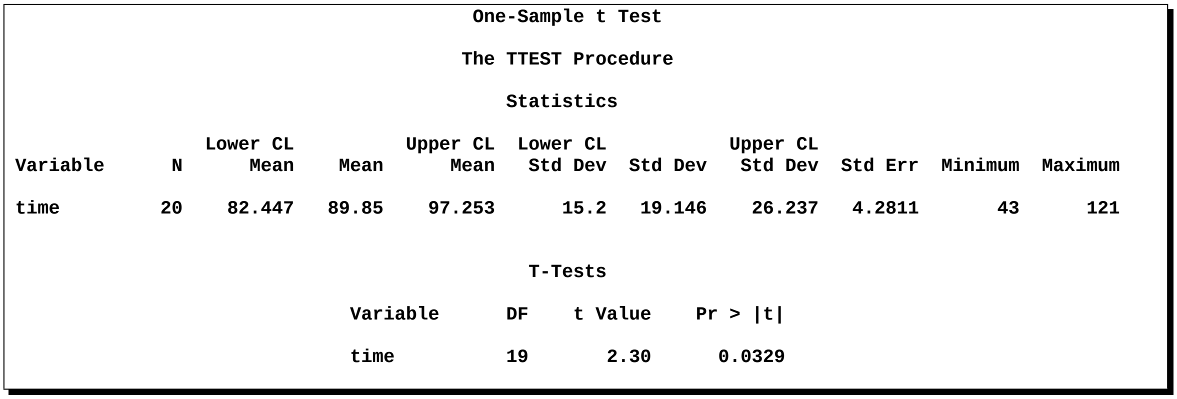
A one-sample t test can be used to compare a sample mean to a given value. This example, taken from Huntsberger and Billingsley (1989, p. 290), tests whether the mean length of a certain type of court case is 80 days using 20 randomly chosen cases. The data are read by the following DATA step:

```
title 'One-Sample t Test';
data time;
   input time @@;
   datalines;
 43  90  84  87  116   95  86   99   93  92
121  71  66  98   79  102  60  112  105  98
;
run;
```

The only variable in the data set, time, is assumed to be normally distributed. The trailing at signs (@@) indicate that there is more than one observation on a line. The following code invokes PROC TTEST for a one-sample t test:

```
proc ttest h0=80 alpha=0.1;
   var time;
run;
```

The VAR statement indicates that the time variable is being studied, while the H0= option specifies that the mean of the time variable should be compared to the value 80 rather than the default null hypothesis of 0. This ALPHA= option requests 10% confidence intervals rather than the default 5% confidence intervals. The output is displayed in Figure 77.1.



```
                                     One-Sample t Test

                                    The TTEST Procedure

                                        Statistics

                        Lower CL            Upper CL  Lower CL            Upper CL
Variable          N         Mean      Mean      Mean   Std Dev   Std Dev   Std Dev   Std Err   Minimum   Maximum

time             20       82.447     89.85    97.253      15.2    19.146    26.237    4.2811        43       121


                                          T-Tests

                              Variable       DF    t Value    Pr > |t|

                              time           19       2.30      0.0329
```

Figure 77.1. One-Sample t Test Results

Summary statistics appear at the top of the output. The sample size (N), the mean and its confidence bounds (Lower CL Mean and Upper CL Mean), the standard deviation and its confidence bounds (Lower CL Std Dev and Upper CL Std Dev), and the standard error are displayed with the minimum and maximum values of the time variable. The test statistic, the degrees of freedom, and the p-value for the t test are displayed next; at the 10% α-level, this test indicates that the mean length of the court cases are significantly different from 80 days $(t = 2.30, p = 0.0329)$.

Comparing Group Means

If you want to compare values obtained from two different groups, and if the groups are independent of each other and the data are normally distributed in each group, then a group t test can be used. Examples of such group comparisons include

- test scores for two third-grade classes, where one of the classes receives tutoring
- fuel efficiency readings of two automobile nameplates, where each nameplate uses the same fuel
- sunburn scores for two sunblock lotions, each applied to a different group of people
- political attitude scores of males and females

In the following example, the golf scores for males and females in a physical education class are compared. The sample sizes from each population are equal, but this is not required for further analysis. The data are read by the following statements:

```
title 'Comparing Group Means';
data scores;
   input Gender $ Score @@;
   datalines;
f 75  f 76  f 80  f 77  f 80  f 77  f 73
m 82  m 80  m 85  m 85  m 78  m 87  m 82
;
run;
```

The dollar sign ($) following Gender in the INPUT statement indicates that Gender is a character variable. The trailing at signs (@@) enable the procedure to read more than one observation per line.

You can use a group t test to determine if the mean golf score for the men in the class differs significantly from the mean score for the women. If you also suspect that the distributions of the golf scores of males and females have unequal variances, then submitting the following statements invokes PROC TTEST with options to deal with the unequal variance case.

```
proc ttest cochran ci=equal umpu;
   class Gender;
   var Score;
run;
```

The CLASS statement contains the variable that distinguishes the groups being compared, and the VAR statement specifies the response variable to be used in calculations. The COCHRAN option produces p-values for the unequal variance situation using the Cochran and Cox(1950) approximation. Equal tailed and uniformly most powerful unbiased (UMPU) confidence intervals for σ are requested by the CI= option. Output from these statements is displayed in Figure 77.2 through Figure 77.4.

```
                                Comparing Group Means

                                 The TTEST Procedure

                                     Statistics

                                                                            UMPU
                           Lower CL            Upper CL  Lower CL  Lower CL
Variable  Gender        N      Mean      Mean      Mean   Std Dev   Std Dev   Std Dev

Score     f             7    74.504    76.857    79.211    1.6399    1.5634    2.5448
Score     m             7    79.804    82.714    85.625     2.028    1.9335    3.1472
Score     Diff (1-2)          -9.19    -5.857    -2.524    2.0522    2.0019    2.8619

                                     Statistics

                               UMPU
                           Upper CL    Upper CL
    Variable  Gender        Std Dev     Std Dev     Std Err     Minimum     Maximum

    Score     f              5.2219      5.6039      0.9619          73          80
    Score     m              6.4579      6.9303      1.1895          78          87
    Score     Diff (1-2)     4.5727      4.7242      1.5298
```

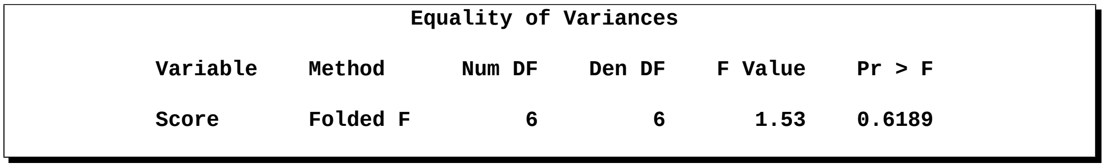
Figure 77.2. Simple Statistics

Simple statistics for the two populations being compared, as well as for the difference of the means between the populations, are displayed in Figure 77.2. The Variable column denotes the response variable, while the Class column indicates the population corresponding to the statistics in that row. The sample size (N) for each population, the sample means (Mean), and lower and upper confidence bounds for the means (Lower CL Mean and Upper CL Mean) are displayed next. The standard deviations (Std Dev) are displayed as well, with equal tailed confidence bounds in the Lower CL Std Dev and Upper CL Std Dev columns and UMPU confidence bounds in the UMPU Upper CL Std Dev and UMPU Lower CL Std Dev columns. In addition, standard error of the mean and the minimum and maximum data values are displayed.

```
                                    T-Tests

       Variable    Method           Variances      DF    t Value    Pr > |t|

       Score       Pooled           Equal          12      -3.83      0.0024
       Score       Satterthwaite    Unequal      11.5      -3.83      0.0026
       Score       Cochran          Unequal         6      -3.83      0.0087
```

Figure 77.3. t Tests

The test statistics, associated degrees of freedom, and p-values are displayed in Figure 77.3. The Method column denotes which t test is being used for that row, and the Variances column indicates what assumption about variances is being made. The pooled test assumes that the two populations have equal variances and uses degrees of freedom $n_1 + n_2 - 2$, where n_1 and n_2 are the sample sizes for the two populations. The remaining two tests do not assume that the populations have equal variances. The Satterthwaite test uses the Satterthwaite approximation for degrees of freedom, while the Cochran test uses the Cochran and Cox approximation for the p-value.



```
                        Equality of Variances

 Variable     Method       Num DF     Den DF     F Value     Pr > F

 Score        Folded F          6          6        1.53     0.6189
```

Figure 77.4. Tests of Equality of Variances

Examine the output in Figure 77.4 to determine which t test is appropriate. The "Equality of Variances" test results show that the assumption of equal variances is reasonable for these data (the Folded F statistic $F' = 1.53$, with $p = 0.6189$). If the assumption of normality is also reasonable, the appropriate test is the usual pooled t test, which shows that the average golf scores for men and women are significantly different $(t = -3.83, p = 0.0024)$. If the assumption of equality of variances is not reasonable, then either the Satterthwaite or the Cochran test should be used.

The assumption of normality can be checked using PROC UNIVARIATE; if the assumption of normality is not reasonable, you should analyze the data with the nonparametric Wilcoxon Rank Sum test using PROC NPAR1WAY.

Syntax

The following statements are available in PROC TTEST.

```
PROC TTEST < options > ;
    CLASS variable ;
    PAIRED variables ;
    BY variables ;
    VAR variables ;
    FREQ variable ;
    WEIGHT variable ;
```

No statement can be used more than once. There is no restriction on the order of the statements after the PROC statement.

PROC TTEST Statement

PROC TTEST < *options* > **;**

The following options can appear in the PROC TTEST statement.

ALPHA=*p*

specifies that confidence intervals are to be $100(1-p)$% confidence intervals, where $0 < p < 1$. By default, PROC TTEST uses ALPHA=0.05. If p is 0 or less, or 1 or more, an error message is printed.

CI=EQUAL
CI=UMPU
CI=NONE

specifies whether a confidence interval is displayed for σ and, if so, what kind. The CI=EQUAL option specifies an equal tailed confidence interval, and it is the default. The CI=UMPU option specifies an interval based on the uniformly most powerful unbiased test of $H_0{:}\, \sigma = \sigma_0$. The CI=NONE option requests that no confidence interval be displayed for σ. The values EQUAL and UMPU together request that both types of confidence intervals be displayed. If the value NONE is specified with one or both of the values EQUAL and UMPU, NONE takes precedence. For more information, see the "Confidence Interval Estimation" section on page 4785.

COCHRAN

requests the Cochran and Cox (1950) approximation of the probability level of the approximate t statistic for the unequal variances situation.

DATA=*SAS-data-set*

names the SAS data set for the procedure to use. By default, PROC TTEST uses the most recently created SAS data set. The input data set can contain summary statistics of the observations instead of the observations themselves. The number, mean, and standard deviation of the observations are required for each BY group (one sample and paired differences) or for each class within each BY group (two samples). For more information on the DATA= option, see the "Input Data Set of Statistics" section on page 4783.

H0=*m*

requests tests against m instead of 0 in all three situations (one-sample, two-sample, and paired observation t tests). By default, PROC TTEST uses H0=0.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TTEST to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TTEST procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the *SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

A CLASS statement giving the name of the classification (or grouping) variable must accompany the PROC TTEST statement in the two independent sample cases. It should be omitted for the one sample or paired comparison situations. If it is used without the VAR statement, all numeric variables in the input data set (except those appearing in the CLASS, BY, FREQ, or WEIGHT statement) are included in the analysis.

The class variable must have two, and only two, levels. PROC TTEST divides the observations into the two groups for the t test using the levels of this variable. You can use either a numeric or a character variable in the CLASS statement.

Class levels are determined from the formatted values of the CLASS variable. Thus, you can use formats to define group levels. Refer to the discussions of the FORMAT procedure, the FORMAT statement, formats, and informats in *SAS Language Reference: Dictionary*.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC TTEST treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If the value is not an integer, only the integer portion is used. If the frequency value is less than 1 or is missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1. The FREQ statement cannot be used if the DATA= data set contains statistics instead of the original observations.

PAIRED Statement

PAIRED *PairLists* **;**

The *PairLists* in the PAIRED statement identifies the variables to be compared in paired comparisons. You can use one or more *PairLists*. Variables or lists of variables are separated by an asterisk (*) or a colon (:). The asterisk requests comparisons between each variable on the left with each variable on the right. The colon requests comparisons between the first variable on the left and the first on the right, the second on the left and the second on the right, and so forth. The number of variables on the left must equal the number on the right when the colon is used. The differences are calculated by taking the variable on the left minus the variable on the right for both the asterisk and colon. A pair formed by a variable with itself is ignored. Use the PAIRED statement only for paired comparisons. The CLASS and VAR statements cannot be used with the PAIRED statement.

Examples of the use of the asterisk and the colon are shown in the following table.

These PAIRED statements...	yield these comparisons
`PAIRED A*B;`	A-B
`PAIRED A*B C*D;`	A-B and C-D
`PAIRED (A B)*(C D);`	A-C, A-D, B-C, and B-D
`PAIRED (A B)*(C B);`	A-C, A-B, and B-C
`PAIRED (A1-A2)*(B1-B2);`	A1-B1, A1-B2, A2-B1, and A2-B2
`PAIRED (A1-A2):(B1-B2);`	A1-B1 and A2-B2

VAR Statement

VAR *variables* **;**

The VAR statement names the variables to be used in the analyses. One-sample comparisons are conducted when the VAR statement is used without the CLASS statement, while group comparisons are conducted when the VAR statement is used with a CLASS statement. If the VAR statement is omitted, all numeric variables in the input data set (except a numeric variable appearing in the BY, CLASS, FREQ, or WEIGHT statement) are included in the analysis. The VAR statement can be used with one- and two-sample t tests and cannot be used with the PAIRED statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement weights each observation in the input data set by the value of the WEIGHT variable. The values of the WEIGHT variable can be nonintegral, and they are not truncated. Observations with negative, zero, or missing values for the WEIGHT variable are not used in the analyses. Each observation is assigned a weight of 1 when the WEIGHT statement is not used. The WEIGHT statement cannot be used with an input data set of summary statistics.

Details

Input Data Set of Statistics

PROC TTEST accepts data containing either observation values or summary statistics. It assumes that the DATA= data set contains statistics if it contains a character variable with name _TYPE_ or _STAT_. The TTEST procedure expects this character variable to contain the names of statistics. If both _TYPE_ and _STAT_ variables exist and are of type character, PROC TTEST expects _TYPE_ to contain the names of statistics including 'N', 'MEAN', and 'STD' for each BY group (or for each class within each BY group for two-sample t tests). If no 'N', 'MEAN', or 'STD' statistics exist, an error message is printed.

FREQ, WEIGHT, and PAIRED statements cannot be used with input data sets of statistics. BY, CLASS, and VAR statements are the same regardless of data set type. For paired comparisons, see the _DIF_ values for the _TYPE_=T observations in output produced by the OUTSTATS= option in the PROC COMPARE statement (refer to the *SAS Procedures Guide*).

Missing Values

An observation is omitted from the calculations if it has a missing value for either the CLASS variable, a PAIRED variable, or the variable to be tested. If more than one variable is listed in the VAR statement, a missing value in one variable does not eliminate the observation from the analysis of other nonmissing variables.

Computational Methods

The t Statistic

The form of the t statistic used varies with the type of test being performed.

- To compare an individual mean with a sample of size n to a value m, use

  $$t = \frac{\bar{x} - m}{s/\sqrt{n}}$$

  where $\bar{x}$ is the sample mean of the observations and s^2 is the sample variance of the observations.

- To compare n paired differences to a value m, use

  $$t = \frac{\bar{d} - m}{s_d/\sqrt{n}}$$

  where $\bar{d}$ is the sample mean of the paired differences and s_d^2 is the sample variance of the paired differences.

- To compare means from two independent samples with n_1 and n_2 observations to a value m, use

  $$t = \frac{(\bar{x}_1 - \bar{x}_2) - m}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

  where s^2 is the pooled variance

  $$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

  and s_1^2 and s_2^2 are the sample variances of the two groups. The use of this t statistic depends on the assumption that $\sigma_1^2 = \sigma_2^2$, where σ_1^2 and σ_2^2 are the population variances of the two groups.

The Folded Form F Statistic

The folded form of the F statistic, F', tests the hypothesis that the variances are equal, where

$$F' = \frac{\max(s_1^2, s_2^2)}{\min(s_1^2, s_2^2)}$$

A test of F' is a two-tailed F test because you do not specify which variance you expect to be larger. The p-value gives the probability of a greater F value under the null hypothesis that $\sigma_1^2 = \sigma_2^2$.

The Approximate t Statistic

Under the assumption of unequal variances, the approximate t statistic is computed as

$$t' = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{w_1 + w_2}}$$

where

$$w_1 = \frac{s_1^2}{n_1}, \quad w_2 = \frac{s_2^2}{n_2}$$

The Cochran and Cox Approximation

The Cochran and Cox (1950) approximation of the probability level of the approximate t statistic is the value of p such that

$$t' = \frac{w_1 t_1 + w_2 t_2}{w_1 + w_2}$$

where t_1 and t_2 are the critical values of the t distribution corresponding to a significance level of p and sample sizes of n_1 and n_2, respectively. The number of degrees of freedom is undefined when $n_1 \neq n_2$. In general, the Cochran and Cox test tends to be conservative (Lee and Gurland 1975).

Satterthwaite's Approximation

The formula for Satterthwaite's (1946) approximation for the degrees of freedom for the approximate t statistic is:

$$df = \frac{(w_1 + w_2)^2}{\left(\frac{w_1^2}{n_1 - 1} + \frac{w_2^2}{n_2 - 1} \right)}$$

Refer to Steel and Torrie (1980) or Freund, Littell, and Spector (1986) for more information.

Confidence Interval Estimation

The form of the confidence interval varies with the statistic for which it is computed. In the following confidence intervals involving means, $t_{1-\frac{\alpha}{2},n-1}$ is the $100(1 - \frac{\alpha}{2})\%$ quantile of the t distribution with $n - 1$ degrees of freedom. The confidence interval for

- an individual mean from a sample of size n compared to a value m is given by

  $$(\bar{x} - m) \pm t_{1-\frac{\alpha}{2},n-1} \frac{s}{\sqrt{n}}$$

  where $\bar{x}$ is the sample mean of the observations and s^2 is the sample variance of the observations

- paired differences with a sample of size n differences compared to a value m is given by

  $$(\bar{d} - m) \pm t_{1-\frac{\alpha}{2},n-1} \frac{s_d}{\sqrt{n}}$$

  where $\bar{d}$ and s_d^2 are the sample mean and sample variance of the paired differences, respectively

- the difference of two means from independent samples with n_1 and n_2 observations compared to a value m is given by

  $$((\bar{x}_1 - \bar{x}_2) - m) \pm t_{1-\frac{\alpha}{2},n_1+n_2-2} s \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

  where s^2 is the pooled variance

  $$s^2 = \frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}$$

  and where s_1^2 and s_2^2 are the sample variances of the two groups. The use of this confidence interval depends on the assumption that $\sigma_1^2 = \sigma_2^2$, where σ_1^2 and σ_2^2 are the population variances of the two groups.

The distribution of the estimated standard deviation of a mean is not symmetric, so alternative methods of estimating confidence intervals are possible. PROC TTEST computes two estimates. For both methods, the data are assumed to have a normal distribution with mean μ and variance σ^2, both unknown. The methods are as follows:

- The default method, an equal-tails confidence interval, puts an equal amount of area ($\frac{\alpha}{2}$) in each tail of the chi-square distribution. An equal tails test of $H_0\colon \sigma = \sigma_0$ has acceptance region

  $$\left\{ \chi^2_{\frac{\alpha}{2},n-1} \leq \frac{(n-1)S^2}{\sigma_0^2} \leq \chi^2_{\frac{1-\alpha}{2},n-1} \right\}$$

  which can be algebraically manipulated to give the following $100(1 - \alpha)\%$ confidence interval for σ^2:

  $$\left(\frac{(n-1)S^2}{\chi^2_{1-\frac{\alpha}{2},n-1}}, \frac{(n-1)S^2}{\chi^2_{\frac{\alpha}{2},n-1}} \right)$$

  In order to obtain a confidence interval for σ, the square root of each side is taken, leading to the following $100(1 - \alpha)\%$ confidence interval:

  $$\left(\sqrt{\frac{(n-1)S^2}{\chi^2_{1-\frac{\alpha}{2},n-1}}}, \sqrt{\frac{(n-1)S^2}{\chi^2_{\frac{\alpha}{2},n-1}}} \right)$$

- The second method yields a confidence interval derived from the uniformly most powerful unbiased test of $H_0\colon \sigma = \sigma_0$ (Lehmann 1986). This test has acceptance region

  $$\left\{ c_1 \leq \frac{(n-1)S^2}{\sigma_0^2} \leq c_2 \right\}$$

  where the critical values c_1 and c_2 satisfy

  $$\int_{c_1}^{c_2} f_n(y)dy = 1 - \alpha$$

and

$$\int_{c_1}^{c_2} y f_n(y) dy = n(1-\alpha)$$

where $f_n(y)$ is the chi-squared distribution with n degrees of freedom. This acceptance region can be algebraically manipulated to arrive at

$$P\left\{\frac{(n-1)S^2}{c_2} \le \sigma^2 \le \frac{(n-1)S^2}{c_1}\right\} = 1-\alpha$$

where c_1 and c_2 solve the preceding two integrals. To find the area in each tail of the chi-square distribution to which these two critical values correspond, solve $c_1 = \chi^2_{1-\alpha_2,n-1}$ and $c_2 = \chi^2_{\alpha_1,n-1}$ for α_1 and α_2; the resulting α_1 and α_2 sum to α. Hence, a $100(1-\alpha)\%$ confidence interval for σ^2 is given by

$$\left(\frac{(n-1)S^2}{\chi^2_{1-\alpha_2,n-1}}, \frac{(n-1)S^2}{\chi^2_{\alpha_1,n-1}}\right)$$

In order to obtain a $100(1-\alpha)\%$ confidence interval for σ, the square root is taken of both terms, yielding

$$\left(\sqrt{\frac{(n-1)S^2}{\chi^2_{1-\alpha_2,n-1}}}, \sqrt{\frac{(n-1)S^2}{\chi^2_{\alpha_1,n-1}}}\right)$$

Displayed Output

For each variable in the analysis, the TTEST procedure displays the following summary statistics for each group:

- the name of the dependent variable
- the levels of the classification variable
- N, the number of nonmissing values
- Lower CL Mean, the lower confidence bound for the mean
- the Mean or average
- Upper CL Mean, the upper confidence bound for the mean
- Lower CL Std Dev, the lower confidence bound for the standard deviation
- Std Dev, the standard deviation
- Upper CL Std Dev, the upper confidence bound for the standard deviation
- Std Err, the standard error of the mean
- the Minimum value, if the line size allows
- the Maximum value, if the line size allows
- upper and lower UMPU confidence bounds for the standard deviation, displayed if the CI=UMPU option is specified in the PROC TTEST statement

Next, the results of several t tests are given. For one-sample and paired observations t tests, the TTEST procedure displays

- t Value, the t statistic for testing the null hypothesis that the mean of the group is zero
- DF, the degrees of freedom
- Pr > |t|, the probability of a greater absolute value of t under the null hypothesis. This is the two-tailed significance probability.

To compute the one-tailed significance probability, first determine whether large values of t are significant or small values are. Let p denote the significance probability for the two-tailed test. If large values of t are significant, then the one-tailed probability is $p/2$ if $t \geq 0$, and is $1 - p/2$ if $t < 0$. If small values of t are significant, then the one-tailed probability is $1 - p/2$ if $t \geq 0$, and is $p/2$ if $t < 0$.

For two-sample t tests, the TTEST procedure displays all the items in the following list. You need to decide whether equal or unequal variances are appropriate for your data.

- Under the assumption of unequal variances, the TTEST procedure displays results using Satterthwaite's method. If the COCHRAN option is specified, the results for the Cochran and Cox approximation are also displayed.
    - t Value, an approximate t statistic for testing the null hypothesis that the means of the two groups are equal
    - DF, the approximate degrees of freedom
    - Pr > |t|, the probability of a greater absolute value of t under the null hypothesis. This is the two-tailed significance probability. The one-tailed probability is computed the same way as in a one-sample t test.
- Under the assumption of equal variances, the TTEST procedure displays results obtained by pooling the group variances.
    - t Value, the t statistic for testing the null hypothesis that the means of the two groups are equal
    - DF, the degrees of freedom
    - Pr > |t|, the probability of a greater absolute value of t under the null hypothesis. This is the two-tailed significance probability. The one-tailed probability is computed the same way as in a one-sample t test.
- PROC TTEST then gives the results of the test of equality of variances:
    - the F' (folded) statistic (see the "The Folded Form F Statistic" section on page 4784)
    - Num DF and Den DF, the numerator and denominator degrees of freedom in each group
    - Pr > F, the probability of a greater F' value. This is the two-tailed significance probability.

ODS Table Names

PROC TTEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 77.1. ODS Tables Produced in PROC TTEST

ODS Table Name	Description	Statement
Equality	Tests for equality of variance	CLASS statement
Statistics	Univariate summary statistics	by default
TTests	t-tests	by default

Examples

Example 77.1. Comparing Group Means Using Input Data Set of Summary Statistics

The following example, taken from Huntsberger and Billingsley (1989), compares two grazing methods using 32 steer. Half of the steer are allowed to graze continuously while the other half are subjected to controlled grazing time. The researchers want to know if these two grazing methods impact weight gain differently. The data are read by the following DATA step.

```
title 'Group Comparison Using Input Data Set of Summary Statistics';
data graze;
   length GrazeType $ 10;
   input GrazeType $ WtGain @@;
   datalines;
controlled  45   controlled  62
controlled  96   controlled 128
controlled 120   controlled  99
controlled  28   controlled  50
controlled 109   controlled 115
controlled  39   controlled  96
controlled  87   controlled 100
controlled  76   controlled  80
continuous  94   continuous  12
continuous  26   continuous  89
continuous  88   continuous  96
continuous  85   continuous 130
continuous  75   continuous  54
continuous 112   continuous  69
continuous 104   continuous  95
continuous  53   continuous  21
;
run;
```

The variable GrazeType denotes the grazing method: ‘controlled’ is controlled grazing and ‘continuous’ is continuous grazing. The dollar sign ($) following GrazeType makes it a character variable, and the trailing at signs (@@) tell the procedure that there is more than one observation per line. The MEANS procedure is invoked to create a data set of summary statistics with the following statements:

```
proc sort;
   by GrazeType;
proc means data=graze noprint;
   var WtGain;
   by GrazeType;
   output out=newgraze;
run;
```

The NOPRINT option eliminates all output from the MEANS procedure. The VAR statement tells PROC MEANS to compute summary statistics for the WtGain variable, and the BY statement requests a separate set of summary statistics for each level of GrazeType. The OUTPUT OUT= statement tells PROC MEANS to put the summary statistics into a data set called newgraze so that it may be used in subsequent procedures. This new data set is displayed in Output 77.1.1 by using PROC PRINT as follows:

```
proc print data=newgraze;
run;
```

The _STAT_ variable contains the names of the statistics, and the GrazeType variable indicates which group the statistic is from.

Output 77.1.1. Output Data Set of Summary Statistics

```
          Group Comparison Using Input Data Set of Summary Statistics

           Obs    GrazeType     _TYPE_    _FREQ_    _STAT_     WtGain

             1    continuous       0        16       N         16.000
             2    continuous       0        16       MIN       12.000
             3    continuous       0        16       MAX      130.000
             4    continuous       0        16       MEAN      75.188
             5    continuous       0        16       STD       33.812
             6    controlled       0        16       N         16.000
             7    controlled       0        16       MIN       28.000
             8    controlled       0        16       MAX      128.000
             9    controlled       0        16       MEAN      83.125
            10    controlled       0        16       STD       30.535
```

The following code invokes PROC TTEST using the newgraze data set, as denoted by the DATA= option.

```
proc ttest data=newgraze;
   class GrazeType;
   var WtGain;
run;
```

The CLASS statement contains the variable that distinguishes between the groups being compared, in this case GrazeType. The summary statistics and confidence intervals are displayed first, as shown in Output 77.1.2.

Output 77.1.2. Summary Statistics

```
                          The TTEST Procedure

                               Statistics

                               Lower CL          Upper CL  Lower CL
Variable  GrazeType         N      Mean    Mean      Mean   Std Dev   Std Dev

WtGain    continuous       16    57.171  75.188    93.204         .    33.812
WtGain    controlled       16    66.854  83.125    99.396         .    30.535
WtGain    Diff (1-2)              -31.2  -7.938    15.323    25.743    32.215

                              Statistics

                             Upper CL
     Variable  GrazeType      Std Dev    Std Err    Minimum    Maximum

     WtGain    continuous           .     8.4529         12        130
     WtGain    controlled           .     7.6337         28        128
     WtGain    Diff (1-2)      43.061      11.39
```

In Output 77.1.2, the Variable column states the variable used in computations and the Class column specifies the group for which the statistics are computed. For each class, the sample size, mean, standard deviation and standard error, and maximum and minimum values are displayed. The confidence bounds for the mean are also displayed; however, since summary statistics are used as input, the confidence bounds for the standard deviation of the groups are not calculated.

Output 77.1.3. t Tests

```
                                 T-Tests

   Variable    Method           Variances      DF    t Value    Pr > |t|

   WtGain      Pooled           Equal          30      -0.70      0.4912
   WtGain      Satterthwaite    Unequal      29.7      -0.70      0.4913


                          Equality of Variances

       Variable    Method      Num DF    Den DF    F Value    Pr > F

       WtGain      Folded F        15        15       1.23    0.6981
```

Output 77.1.3 shows the results of tests for equal group means and equal variances. A group test statistic for the equality of means is reported for equal and unequal variances. Before deciding which test is appropriate, you should look at the test for equality of variances; this test does not indicate a significant difference in the two variances $(F' = 1.23, p = 0.6981)$, so the pooled t statistic should be used. Based on the pooled statistic, the two grazing methods are not significantly different $(t = 0.70, p = 0.4912)$. Note that this test assumes that the observations in both data sets are normally distributed; this assumption can be checked in PROC UNIVARIATE using the raw data.

Example 77.2. One-Sample Comparison Using the FREQ Statement

This example examines children's reading skills. The data consist of Degree of Reading Power (DRP) test scores from 44 third-grade children and are taken from Moore (1995, p. 337). Their scores are given in the following DATA step.

```
title 'One-Mean Comparison Using FREQ Statement';
data read;
   input score count @@;
   datalines;
40 2   47 2   52 2   26 1   19 2
25 2   35 4   39 1   26 1   48 1
14 2   22 1   42 1   34 2   33 2
18 1   15 1   29 1   41 2   44 1
51 1   43 1   27 2   46 2   28 1
49 1   31 1   28 1   54 1   45 1
;
run;
```

The following statements invoke the TTEST procedure to test if the mean test score is equal to 30. The count variable contains the frequency of occurrence of each test score; this is specified in the FREQ statement.

```
proc ttest data=read h0=30;
   var score;
   freq count;
run;
```

The output, shown in Output 77.2.1, contains the results.

Output 77.2.1. TTEST Results

```
                               One-Mean Comparison Using FREQ Statement

                                         The TTEST Procedure

                                             Statistics

                         Lower CL              Upper CL  Lower CL             Upper CL
Variable          N        Mean       Mean       Mean     Std Dev   Std Dev   Std Dev   Std Err   Minimum   Maximum

score            44      31.449     34.864     38.278     9.2788     11.23    14.229     1.693        14        54


                                               T-Tests

                              Variable       DF    t Value    Pr > |t|

                              score          43       2.87      0.0063
```

The SAS log states that 30 observations and two variables have been read. However, the sample size given in the TTEST output is N=44. This is due to specifying the count variable in the FREQ statement. The test is significant $(t = 2.87, p = 0.0063)$ at the 5% level, thus you can conclude that the mean test score is different from 30.

Example 77.3. Paired Comparisons

When it is not feasible to assume that two groups of data are independent, and a natural pairing of the data exists, it is advantageous to use an analysis that takes the correlation into account. Utilizing this correlation results in higher power to detect existing differences between the means. The differences between paired observations are assumed to be normally distributed. Some examples of this natural pairing are

- pre- and post-test scores for a student receiving tutoring
- fuel efficiency readings of two fuel types observed on the same automobile
- sunburn scores for two sunblock lotions, one applied to the individual's right arm, one to the left arm
- political attitude scores of husbands and wives

In this example, taken from *SUGI Supplemental Library User's Guide, Version 5 Edition*, a stimulus is being examined to determine its effect on systolic blood pressure. Twelve men participate in the study. Their systolic blood pressure is measured both before and after the stimulus is applied. The following statements input the data:

```
title 'Paired Comparison';
data pressure;
      input SBPbefore SBPafter @@;
      datalines;
120 128   124 131   130 131   118 127
140 132   128 125   140 141   135 137
126 118   130 132   126 129   127 135
;
run;
```

The variables SBPbefore and SBPafter denote the systolic blood pressure before and after the stimulus, respectively.

The statements to perform the test follow.

```
proc ttest;
   paired SBPbefore*SBPafter;
run;
```

The PAIRED statement is used to test whether the mean change in systolic blood pressure is significantly different from zero. The output is displayed in Output 77.3.1.

Output 77.3.1. TTEST Results

```
                                          Paired Comparison

                                         The TTEST Procedure

                                              Statistics

                                 Lower CL             Upper CL  Lower CL             Upper CL
Difference                  N      Mean      Mean       Mean   Std Dev   Std Dev    Std Dev   Std Err   Minimum   Maximum

SBPbefore - SBPafter       12    -5.536    -1.833     1.8698    4.1288    5.8284     9.8958    1.6825        -9         8


                                               T-Tests

                             Difference                 DF     t Value    Pr > |t|

                             SBPbefore - SBPafter       11       -1.09      0.2992
```

The variables SBPbefore and SBPafter are the paired variables with a sample size of 12. The summary statistics of the difference are displayed (mean, standard deviation, and standard error) along with their confidence limits. The minimum and maximum differences are also displayed. The t test is not significant ($t = -1.09$, $p = 0.2992$), indicating that the stimuli did not significantly affect systolic blood pressure.

Note that this test of hypothesis assumes that the differences are normally distributed. This assumption can be investigated using PROC UNIVARIATE with the NORMAL option. If the assumption is not satisfied, PROC NPAR1WAY should be used.

References

Best, D.I. and Rayner, C.W. (1987), “Welch’s Approximate Solution for the Behren’s-Fisher Problem,” *Technometrics*, 29, 205–210.

Cochran, W.G. and Cox, G.M. (1950), *Experimental Designs*, New York: John Wiley & Sons, Inc.

Freund, R.J., Littell, R.C., and Spector, P.C. (1986), *SAS System for Linear Models, 1986 Edition*, Cary, NC: SAS Institute Inc.

Huntsberger, David V. and Billingsley, Patrick P. (1989), *Elements of Statistical Inference*, Dubuque, Iowa: Wm. C. Brown Publishers.

Moore, David S. (1995), *The Basic Practice of Statistics*, New York: W. H. Freeman and Company.

Lee, A.F.S. and Gurland, J. (1975), “Size and Power of Tests for Equality of Means of Two Normal Populations with Unequal Variances,” *Journal of the American Statistical Association*, 70, 933–941.

Lehmann, E. L. (1986), *Testing Statistical Hypostheses*, New York: John Wiley & Sons.

Posten, H.O., Yeh, Y.Y., and Owen, D.B. (1982), “Robustness of the Two-Sample t Test Under Violations of the Homogeneity of Variance Assumption,” *Communications in Statistics*, 11, 109–126.

Ramsey, P.H. (1980), “Exact Type I Error Rates for Robustness of Student’s t Test with Unequal Variances,” *Journal of Educational Statistics*, 5, 337–349.

Robinson, G.K. (1976), “Properties of Student’s t and of the Behrens-Fisher Solution to the Two Mean Problem,” *Annals of Statistics*, 4, 963–971.

Satterthwaite, F.W. (1946), “An Approximate Distribution of Estimates of Variance Components,” *Biometrics Bulletin*, 2, 110–114.

Scheffe, H. (1970), “Practical Solutions of the Behrens-Fisher Problem,” *Journal of the American Statistical Association*, 65, 1501–1508.

SAS Institute Inc, (1986), *SUGI Supplemental Library User’s Guide, Version 5 Edition*. Cary, NC: SAS Institute Inc.

Steel, R.G.D. and Torrie, J.H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill Book Company.

Wang, Y.Y. (1971), “Probabilities of the Type I Error of the Welch Tests for the Behren’s-Fisher Problem,” *Journal of the American Statistical Association*, 66, 605–608.

Yuen, K.K. (1974), “The Two-Sample Trimmed t for Unequal Population Variances,” *Biometrika*, 61, 165–170.

Chapter 78
The VARCLUS Procedure

Chapter Contents

Chapter 78
The VARCLUS Procedure

Overview

The VARCLUS procedure divides a set of numeric variables into disjoint or hierarchical clusters. Associated with each cluster is a linear combination of the variables in the cluster. This linear combination can be either the first principal component (the default) or the centroid component (if you specify the CENTROID option). The first principal component is a weighted average of the variables that explains as much variance as possible. See Chapter 58, "The PRINCOMP Procedure," for further details. Centroid components are unweighted averages of either the standardized variables (the default) or the raw variables (if you specify the COVARIANCE option). PROC VARCLUS tries to maximize the variance that is explained by the cluster components, summed over all the clusters.

The cluster components are oblique, not orthogonal, even when the cluster components are first principal components. In an ordinary principal component analysis, all components are computed from the same variables, and the first principal component is orthogonal to the second principal component and to each other principal component. In PROC VARCLUS, each cluster component is computed from a different set of variables than all the other cluster components. The first principal component of one cluster may be correlated with the first principal component of another cluster. Hence, PROC VARCLUS is a type of oblique component analysis.

As in principal component analysis, either the correlation or the covariance matrix can be analyzed. If correlations are used, all variables are treated as equally important. If covariances are used, variables with larger variances have more importance in the analysis.

PROC VARCLUS creates an output data set that can be used with the SCORE procedure to compute component scores for each cluster. A second output data set can be used by the TREE procedure to draw a tree diagram of hierarchical clusters.

The VARCLUS procedure can be used as a variable-reduction method. A large set of variables can often be replaced by the set of cluster components with little loss of information. A given number of cluster components does not generally explain as much variance as the same number of principal components on the full set of variables, but the cluster components are usually easier to interpret than the principal components, even if the latter are rotated.

For example, an educational test might contain fifty items. PROC VARCLUS can be used to divide the items into, say, five clusters. Each cluster can then be treated as a subtest, with the subtest scores given by the cluster components. If the cluster components are centroid components of the covariance matrix, each subtest score is simply the sum of the item scores for that cluster.

The VARCLUS algorithm is both divisive and iterative. By default, PROC VARCLUS begins with all variables in a single cluster. It then repeats the following steps:

1. A cluster is chosen for splitting. Depending on the options specified, the selected cluster has either the smallest percentage of variation explained by its cluster component (using the PROPORTION= option) or the largest eigenvalue associated with the second principal component (using the MAXEIGEN= option).
2. The chosen cluster is split into two clusters by finding the first two principal components, performing an orthoblique rotation (raw quartimax rotation on the eigenvectors; Harris and Kaiser, 1964), and assigning each variable to the rotated component with which it has the higher squared correlation.
3. Variables are iteratively reassigned to clusters to try to maximize the variance accounted for by the cluster components. You can require the reassignment algorithms to maintain a hierarchical structure for the clusters.

The procedure stops splitting when either:

- the maximum number of clusters as specified by the MAXCLUSTERS= option is reached, or
- each cluster satisfies the stopping criteria specified by the PROPORTION= (percentage of variation explained) and/or the MAXEIGEN= (second eigenvalue) options.

By default, VARCLUS stops splitting when each cluster has only one eigenvalue greater than one, thus satisfying the most popular criterion for determining the sufficiency of a single underlying dimension.

The iterative reassignment of variables to clusters proceeds in two phases. The first is a nearest component sorting (NCS) phase, similar in principle to the nearest centroid sorting algorithms described by Anderberg (1973). In each iteration, the cluster components are computed, and each variable is assigned to the component with which it has the highest squared correlation. The second phase involves a search algorithm in which each variable is tested to see if assigning it to a different cluster increases the amount of variance explained. If a variable is reassigned during the search phase, the components of the two clusters involved are recomputed before the next variable is tested. The NCS phase is much faster than the search phase but is more likely to be trapped by a local optimum.

If principal components are used, the NCS phase is an alternating least-squares method and converges rapidly. The search phase can be very time consuming for a large number of variables. But if the default initialization method is used, the search phase is rarely able to substantially improve the results of the NCS phase, so the search takes few iterations. If random initialization is used, the NCS phase may be trapped by a local optimum from which the search phase can escape.

If centroid components are used, the NCS phase is not an alternating least-squares method and may not increase the amount of variance explained; therefore, it is limited, by default, to one iteration.

You can have VARCLUS do the clustering hierarchically by restricting the reassignment of variables such that the clusters maintain a tree structure. In this case, when a cluster is split, a variable in one of the two resulting clusters can be reassigned to the other cluster resulting from the split but not to a cluster that is not part of the original cluster (the one that is split).

Getting Started

This example demonstrates how you can use the VARCLUS procedure to create hierarchical, unidimensional clusters of variables.

The following data, from Hand, et al. (1994), represent amounts of protein consumed from nine food groups for each of 25 European countries. The nine food groups are red meat (RedMeat), white meat (WhiteMeat), eggs (Eggs), milk (Milk), fish (Fish), cereal (Cereal), starch (Starch), nuts (Nuts), and fruits and vegetables (FruitVeg).

Suppose you want to simplify interpretation of the data by reducing the number of variables to a smaller set of variable cluster components. You can use the VARCLUS procedure for this type of variable reduction.

The following DATA step creates the SAS data set Protein:

```
data Protein;
   input Country $18. RedMeat WhiteMeat Eggs Milk
      Fish Cereal Starch Nuts FruitVeg;
   datalines;
Albania        10.1  1.4  0.5   8.9  0.2  42.3  0.6  5.5  1.7
Austria         8.9 14.0  4.3  19.9  2.1  28.0  3.6  1.3  4.3
Belgium        13.5  9.3  4.1  17.5  4.5  26.6  5.7  2.1  4.0
Bulgaria        7.8  6.0  1.6   8.3  1.2  56.7  1.1  3.7  4.2
Czechoslovakia  9.7 11.4  2.8  12.5  2.0  34.3  5.0  1.1  4.0
Denmark        10.6 10.8  3.7  25.0  9.9  21.9  4.8  0.7  2.4
E Germany       8.4 11.6  3.7  11.1  5.4  24.6  6.5  0.8  3.6
Finland         9.5  4.9  2.7  33.7  5.8  26.3  5.1  1.0  1.4
France         18.0  9.9  3.3  19.5  5.7  28.1  4.8  2.4  6.5
Greece         10.2  3.0  2.8  17.6  5.9  41.7  2.2  7.8  6.5
Hungary         5.3 12.4  2.9   9.7  0.3  40.1  4.0  5.4  4.2
Ireland        13.9 10.0  4.7  25.8  2.2  24.0  6.2  1.6  2.9
Italy           9.0  5.1  2.9  13.7  3.4  36.8  2.1  4.3  6.7
Netherlands     9.5 13.6  3.6  23.4  2.5  22.4  4.2  1.8  3.7
Norway          9.4  4.7  2.7  23.3  9.7  23.0  4.6  1.6  2.7
Poland          6.9 10.2  2.7  19.3  3.0  36.1  5.9  2.0  6.6
Portugal        6.2  3.7  1.1   4.9 14.2  27.0  5.9  4.7  7.9
Romania         6.2  6.3  1.5  11.1  1.0  49.6  3.1  5.3  2.8
Spain           7.1  3.4  3.1   8.6  7.0  29.2  5.7  5.9  7.2
Sweden          9.9  7.8  3.5   4.7  7.5  19.5  3.7  1.4  2.0
Switzerland    13.1 10.1  3.1  23.8  2.3  25.6  2.8  2.4  4.9
```

```
UK               17.4  5.7  4.7  20.6  4.3  24.3  4.7  3.4  3.3
USSR              9.3  4.6  2.1  16.6  3.0  43.6  6.4  3.4  2.9
W Germany        11.4 12.5  4.1  18.8  3.4  18.6  5.2  1.5  3.8
Yugoslavia        4.4  5.0  1.2   9.5  0.6  55.9  3.0  5.7  3.2
;
```

The data set Protein contains the character variable Country and the nine numeric variables representing the food groups. The $18. in the INPUT statement specifies that the variable Country is a character variable with a length of 18.

The following statements create the variable clusters.

```
proc varclus data=Protein outtree=tree centroid maxclusters=4;
   var RedMeat--FruitVeg;
run;
```

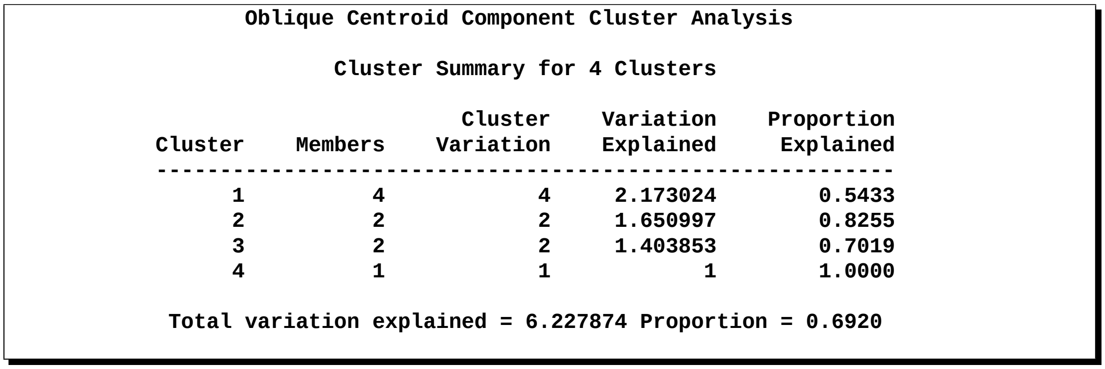
The DATA= option specifies the SAS data set Protein as input. The OUTTREE= option creates the output SAS data set Tree to contain the tree structure information. When you specify this option, you are implicitly requiring the clusters to be hierarchical rather than disjoint.

The CENTROID option specifies the centroid method of clustering. This means that the calculated cluster components are the unweighted averages of the standardized variables. The MAXCLUSTERS=4 option specifies that no more than four clusters be computed.

The VAR statement lists the numeric variables (RedMeat–FruitVeg) to be used in the analysis.

The results of this analysis are displayed in the following figures.

Although PROC VARCLUS displays output for each step in the clustering process, the following figures display only the final analysis for four clusters. Figure 78.1 displays the final cluster summary.



```
                  Oblique Centroid Component Cluster Analysis

                         Cluster Summary for 4 Clusters

                                 Cluster    Variation    Proportion
           Cluster    Members    Variation    Explained     Explained
           ----------------------------------------------------------
                 1          4            4     2.173024        0.5433
                 2          2            2     1.650997        0.8255
                 3          2            2     1.403853        0.7019
                 4          1            1            1        1.0000

            Total variation explained = 6.227874 Proportion = 0.6920
```

Figure 78.1. Final Cluster Summary from the VARCLUS Procedure

For each cluster, Figure 78.1 displays the number of variables in the cluster, the cluster variation, the total explained variation, and the proportion of the total variance explained by the variables in the cluster. The variance explained by the variables in a cluster is similar to the variance explained by a factor in common factor analysis, but it includes contributions only from the variables in the cluster rather than from all variables.

The line labeled “Total variation explained” in Figure 78.1 gives the sum of the explained variation over all clusters. The final “Proportion” represents the total explained variation divided by the sum of cluster variation. This value, 0.6920, indicates that about 69% of the total variation in the data can be accounted for by the four cluster components.

Figure 78.2 shows how the variables are clustered. The first cluster represents animal protein (RedMeat, WhiteMeat, Eggs, and Milk), the second cluster contains the variables Cereal and Nuts, the third cluster is composed of the variables Fish and Starch, and the last cluster contains the single variable representing fruits and vegetables (FruitVeg).

```
                  Oblique Centroid Component Cluster Analysis

                                       R-squared with
          4 Clusters                 ------------------
                                          Own        Next     1-R**2
          Cluster       Variable      Cluster     Closest      Ratio
          -------------------------------------------------------
          Cluster 1     RedMeat        0.4375      0.1518     0.6631
                        WhiteMeat      0.6302      0.3331     0.5545
                        Eggs           0.7024      0.4902     0.5837
                        Milk           0.4288      0.2721     0.7847
          -------------------------------------------------------
          Cluster 2     Cereal         0.8255      0.3983     0.2900
                        Nuts           0.8255      0.5901     0.4257
          -------------------------------------------------------
          Cluster 3     Fish           0.7019      0.1365     0.3452
                        Starch         0.7019      0.3075     0.4304
          -------------------------------------------------------
          Cluster 4     FruitVeg       1.0000      0.0578     0.0000
```

Figure 78.2. R-square Values from the VARCLUS Procedure

Figure 78.2 also displays the R^2 value of each variable with its own cluster and the R^2 value with its nearest cluster. The R^2 value for a variable with the nearest cluster should be low if the clusters are well separated. The last column displays the ratio of $(1 - R^2_{own})/(1 - R^2_{nearest})$ for each variable. Small values of this ratio indicate good clustering.

Figure 78.3 displays the cluster structure and the intercluster correlations. The structure table displays the correlation of each variable with each cluster component. The table of intercorrelations contains the correlations between the cluster components.

```
                   Oblique Centroid Component Cluster Analysis

                                Cluster Structure

       Cluster                1              2              3              4
       -----------------------------------------------------------------
       RedMeat          0.66145       -0.38959        0.06450       -0.34109
       WhiteMeat        0.79385       -0.57715        0.04760       -0.06132
       Eggs             0.83811       -0.70012        0.30902       -0.04552
       Milk             0.65483       -0.52163        0.16805       -0.26096
       Fish            -0.08108       -0.36947        0.83781        0.26614
       Cereal          -0.58070        0.90857       -0.63111        0.04655
       Starch           0.41593       -0.55448        0.83781        0.08441
       Nuts            -0.76817        0.90857       -0.37089        0.37497
       FruitVeg        -0.24045        0.23197        0.20920        1.00000


                           Inter-Cluster Correlations

        Cluster               1              2              3              4

        1               1.00000       -0.74230        0.19984       -0.24045
        2              -0.74230        1.00000       -0.55141        0.23197
        3               0.19984       -0.55141        1.00000        0.20920
        4              -0.24045        0.23197        0.20920        1.00000
```

Figure 78.3. Cluster Correlations and Intercorrelations

PROC VARCLUS next displays the summary table of statistics for the cluster history (Figure 78.4). The first three columns give the number of clusters, the total variation explained by clusters, and the proportion of variation explained by clusters.

As displayed in Figure 78.4, when the number of allowable clusters is two, the total variation explained is 3.9607, and the cumulative proportion of variation explained by two clusters is 0.4401. When the number of clusters increases to three, the proportion of explained variance increases to 0.5880. When four clusters are computed, the explained variation is 0.6920.

```
                   Oblique Centroid Component Cluster Analysis

                  Total     Proportion        Minimum      Minimum         Maximum
  Number      Variation   of Variation     Proportion    R-squared   1-R**2 Ratio
      of      Explained      Explained      Explained        for a          for a
Clusters    by Clusters    by Clusters   by a Cluster     Variable       Variable
--------------------------------------------------------------------------------
       1       0.732343         0.0814         0.0814       0.0875
       2       3.960717         0.4401         0.3743       0.1007         1.0213
       3       5.291887         0.5880         0.5433       0.3928         0.7978
       4       6.227874         0.6920         0.5433       0.4288         0.7847
```

Figure 78.4. Final Cluster Summary Table from the VARCLUS Procedure

Figure 78.4 also displays the minimum proportion of variance explained by a cluster, the minimum R^2 for a variable, and the maximum $(1 - R^2)$ ratio for a variable. The last quantity is the ratio of the value $1 - R^2$ for a variable's own cluster to the value $1 - R^2$ for its nearest cluster.

The following statements produce a tree diagram of the cluster structure created by PROC VARCLUS. The AXIS1 statement suppresses the label for the vertical axis, which would otherwise be “Name of Variable or Cluster”.

```
axis1 label=none;
proc tree data=tree horizontal vaxis=axis1;
height _propor_;
run;
```

Next, the TREE procedure is invoked using the SAS data set TREE, created by the OUTTREE= option in the preceding PROC VARCLUS statement. The HORIZONTAL option orients the tree diagram horizontally. The VAXIS option associates the vertical axis with the the AXIS1 statement. The HEIGHT statement specifies the use of the variable _PROPOR_ (the proportion of variance explained) as the height variable.

Figure 78.5 shows how the clusters are created. The ordered variable names are displayed on the vertical axis. The horizontal axis displays the proportion of variance explained at each clustering level.

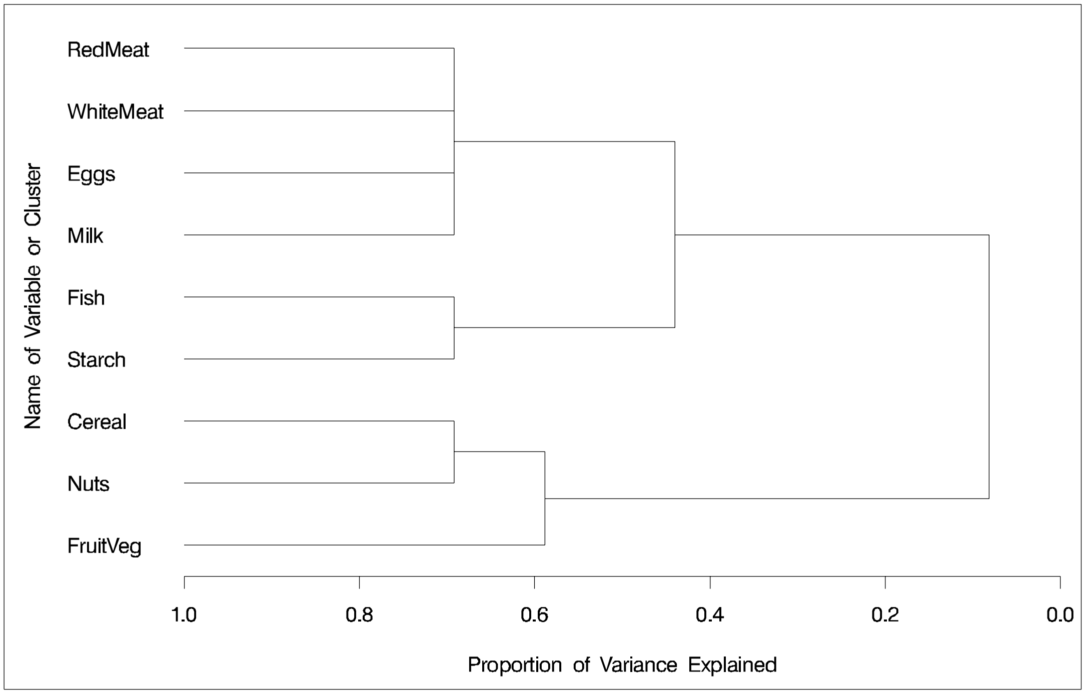


Figure 78.5. Horizontal Tree Diagram from PROC TREE

As you look from left to right in the diagram, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the right (or root) of the diagram. Clusters exist at each level of the diagram, and every vertical line connects leaves and branches into progressively larger clusters.

For example, when the variables are formed into three clusters, one cluster contains the variables RedMeat, WhiteMeat, Eggs, and Milk; the second cluster contains the variables Fish and Starch; the third cluster contains the variables Cereal, Nuts, and FruitVeg. The proportion of variance explained at that level is 0.5880 (from Figure 78.4). At the next stage of clustering, the third cluster is split as the variable FruitVeg forms the fourth cluster; the proportion of variance explained is 0.6920.

Syntax

The following statements are available in PROC VARCLUS.

PROC VARCLUS < *options* > **;**
VAR *variables* **;**
SEED *variables* **;**
PARTIAL *variables* **;**
WEIGHT *variables* **;**
FREQ *variables* **;**
BY *variables* **;**

Usually you need only the VAR statement in addition to the PROC VARCLUS statement. The following sections give detailed syntax information for each of the statements, beginning with the PROC VARCLUS statement. The remaining statements are listed in alphabetical order.

PROC VARCLUS Statement

PROC VARCLUS < *options* >**;**

The PROC VARCLUS statement starts the VARCLUS procedure. By default, VARCLUS clusters the numeric variables in the most recently created SAS data set, starting with one cluster and splitting clusters until all clusters have at most one eigenvalue greater than one.

VARCLUS chooses a cluster to split based on two options: MAXEIGEN=, and PROPORTION=.

1. If you specify *either* or *both* of these two options, then *only* the specified options affect the choice of the cluster to split.
2. If you specify *neither* of these options, the criterion for choice of cluster to split depends on the CENTROID option:

   (a) If you specify CENTROID, VARCLUS splits the cluster with the smallest percentage of variation explained by its cluster component, as if you had specified the PROPORTION= option.

   (b) If you do not specify CENTROID, VARCLUS splits the cluster with the largest eigenvalue associated with the second principal component, as if you had specified the MAXEIGEN= option.

The final number of clusters is controlled by three options: MAXCLUSTERS=, MAXEIGEN=, and PROPORTION=.

1. If you specify *any* of these three options, then *only* the options you specify affect the final number of clusters.
2. If you specify *none* of these options, VARCLUS continues to split clusters until the default splitting criterion is satisfied. The default splitting criterion depends on the CENTROID option:

    (a) If you specify CENTROID, the default splitting criterion is PROPORTION=0.75.

    (b) If you do not specify CENTROID, splitting is based on the MAXEIGEN= criterion, with a default depending on the COVARIANCE option:

    i. for analyzing a correlation matrix (no COVARIANCE option), the defaut value for MAXEIGEN= is one.

    ii. for analyzing a covariance matrix (using the COVARIANCE option), the default value for MAXEIGEN= is the average variance of the variables being clustered.

VARCLUS continues to split clusters until any of the following conditions holds:

- the number of cluster equals the value specified for MAXCLUSTERS=.
- no cluster qualifies for splitting according to the MAXEIGEN= or PROPORTION= criteria.
- a cluster was chosen for splitting, but after iteratively reassigning variables to clusters, one of the cluster has no members.

Table 78.1 summarizes some of the options available in the PROC VARCLUS statement.

Table 78.1. Options Available in the PROC VARCLUS Statement

Task	Options
Specify data sets	DATA= OUTSTAT= OUTTREE=
Determine the number of clusters	MAXCLUSTERS= MINCLUSTERS= MAXEIGEN= PROPORTION=
Specify cluster formation	CENTROID COVARIANCE HIERARCHY INITIAL= MAXITER= MAXSEARCH= MULTIPLEGROUP RANDOM=
Control output	CORR NOPRINT SHORT SIMPLE SUMMARY TRACE
Omit intercept	NOINT
Specify divisor for variances	VARDEF=

The following list gives details on these options. The list is in alphabetical order.

CENTROID

uses centroid components rather than principal components. You should specify centroid components if you want the cluster components to be unweighted averages of the standardized variables (the default) or the unstandardized variables (if you specify the COVARIANCE option). It is possible to obtain locally optimal clusterings in which a variable is not assigned to the cluster component with which it has the highest squared correlation. You cannot specify both the CENTROID and MAXEIGEN= options.

CORR
C

displays the correlation matrix.

COVARIANCE
COV

analyzes the covariance matrix instead of the correlation matrix. The COVARIANCE option causes variables with a large variance to have more effect on the cluster components than variables with a small variance.

DATA=*SAS-data-set*

specifies the input data set to be analyzed. The data set can be an ordinary SAS data set or TYPE=CORR, UCORR, COV, UCOV, FACTOR, or SSCP. If you do not specify the DATA= option, the most recently created SAS data set is used. See Appendix A, "Special SAS Data Sets," for more information on types of SAS data sets.

HIERARCHY
HI

requires the clusters at different levels to maintain a hierarchical structure. To draw a tree diagram, use the OUTTREE= option and the TREE procedure.

INITIAL=GROUP
INITIAL=INPUT
INITIAL=RANDOM
INITIAL=SEED

specifies the method for initializing the clusters. If the INITIAL= option is omitted and the MINCLUSTERS= option is greater than 1, the initial cluster components are obtained by extracting the required number of principal components and performing an orthoblique rotation (raw quartimax rotation on the eigenvectors; Harris and Kaiser, 1964). The following list describes the values for the INITIAL= option:

GROUP obtains the cluster membership of each variable from an observation in the DATA= data set where the _TYPE_ variable has a value of "GROUP". In this observation, the variables to be clustered must each have an integer value ranging from one to the number of clusters. You can use this option only if the DATA= data set is a TYPE=CORR, UCORR, COV, UCOV, or FACTOR data set. You can use a data set created either by a previous run of PROC VARCLUS or in a DATA step.

INPUT obtains scoring coefficients for the cluster components from observations in the DATA= data set where the _TYPE_ variable has a value of "SCORE". You can use this option only if the DATA= data set is a TYPE=CORR, UCORR, COV, UCOV, or FACTOR data set, You can use scoring coefficients from the FACTOR procedure or a previous run of PROC VARCLUS, or you can enter other coefficients in a DATA step.

RANDOM assigns variables randomly to clusters.

SEED initializes each cluster component to be one of the variables named in the SEED statement. Each variable listed in the SEED statement becomes the sole member of a cluster, and the other variables are initially unassigned. If you do not specify the SEED statement, the first MINCLUSTERS= variables in the VAR statement are used as seeds.

MAXCLUSTERS=*n*
MAXC=*n*

specifies the largest number of clusters desired. The default value is the number of variables. VARCLUS stops splitting clusters after the number of clusters reaches the value of the MAXCLUSTERS= option, regardless of what other splitting options are specified.

MAXEIGEN=*n*

specifies that when choosing a cluster to split, VARCLUS should choose the cluster with the largest second eigenvalue, provided that its second eigenvalue is greater than the MAXEIGEN= value. The MAXEIGEN= option cannot be used with the CENTROID or MULTIPLEGROUP options.

If you do not specify MAXEIGEN=, then:

- If you specify PROPORTION=, CENTROID, or MULTIPLEGROUP, cluster splitting does not depend on the second eigenvalue.
- Otherwise, if you specify MAXCLUSTERS=, the default value for MAXEIGEN= is zero.
- Otherwise, the default value for MAXEIGEN= is either 1.0 if the correlation matrix is analyzed, or the average variance if the COVARIANCE option is specified.

If you specify both MAXEIGEN= and MAXCLUSTERS=, the number of clusters will never exceed the value of the MAXCLUSTERS= option.

If you specify both MAXEIGEN= and PROPORTION=, VARCLUS first looks for a cluster to split based on the MAXEIGEN= criterion. If no cluster meets that criterion, VARCLUS then looks for a cluster to split based on the PROPORTION= criterion.

MAXITER=*n*

specifies the maximum number of iterations during the NCS phase. The default value is 1 if you specify the CENTROID option; the default is 10 otherwise.

MAXSEARCH=*n*

specifies the maximum number of iterations during the search phase. The default is 1000 divide by the number of variables.

MINCLUSTERS=*n*
MINC=*n*

specifies the smallest number of clusters desired. The default value is 2 for INITIAL=RANDOM or INITIAL=SEED; otherwise, VARCLUS begins with one cluster and tries to split it in accordance with the PROPORTION= or MAXEIGEN= options.

MULTIPLEGROUP
MG

performs a multiple group component analysis (Harman 1976). You specify which variables belong to which clusters. No clusters are split, and no variables are reassigned to a different cluster. The input data set must be TYPE=CORR, UCORR, COV, UCOV, FACTOR or SSCP and must contain an observation with _TYPE_="GROUP" defining the variable groups. Specifying the MULTIPLEGROUP option is equivalent to specifying all of the following options: INITIAL=GROUP, MINC=1, MAXITER=0, MAXSEARCH=0, PROPORTION=0, and MAXEIGEN=large number.

NOINT

requests that no intercept be used; covariances or correlations are not corrected for the mean. If you specify the NOINT option, the OUTSTAT= data set is TYPE=UCORR.

NOPRINT

suppresses displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 14, "Using the Output Delivery System."

OUTSTAT=*SAS-data-set*

creates an output data set to contain statistics including means, standard deviations, correlations, cluster scoring coefficients, and the cluster structure. If you want to create a permanent SAS data set, you must specify a two-level name. The OUTSTAT= data set is TYPE=UCORR if the NOINT option is specified. For more information on permanent SAS data sets, refer to "SAS Files" and "DATA Step Concepts" in *SAS Language Reference: Concepts*. For information on types of SAS data sets, see Appendix A, "Special SAS Data Sets,".

OUTTREE=*SAS-data-set*

creates an output data set to contain information on the tree structure that can be used by the TREE procedure to display a tree diagram. The OUTTREE= option implies the HIERARCHY option. See Example 78.1 for use of the OUTTREE= option. If you want to create a permanent SAS data set, you must specify a two-level name. For more information on permanent SAS data sets, refer to "SAS Files" and "DATA Step Concepts" in *SAS Language Reference: Concepts*.

PROPORTION=*n*
PERCENT=*n*

specifies that when choosing a cluster to split, VARCLUS should choose the cluster with the smallest proportion of variation explained, provided that its proportion of variation explained is less than the PROPORTION= value. Values greater than 1.0 are considered to be percentages, so PROPORTION=0.75 and PERCENT=75 are equivalent.

However, if you specify both MAXEIGEN= and PROPORTION=, VARCLUS first looks for a cluster to split based on the MAXEIGEN= criterion. If no cluster meets that criterion, VARCLUS then looks for a cluster to split based on the PROPORTION= criterion.

If you do not specify PROPORTION= then:

- If you specify MAXEIGEN=, cluster splitting does not depend on the proportion of variation explained.
- Otherwise, if you specify CENTROID and MAXCLUSTERS=, the default value for PROPORTION= is one.
- Otherwise, if you specify CENTROID, without MAXCLUSTERS=, the default value is PROPORTION=0.75 or PERCENT=75.
- Otherwise, cluster splitting does not depend on the proportion of variation explained.

If you specify both PROPORTION= and MAXCLUSTERS=, the number of clusters will never exceed the value of the MAXCLUSTERS= option.

RANDOM=*n*

specifies a positive integer as a starting value for use with REPLACE=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence.

SHORT

suppresses display of the cluster structure, scoring coefficient, and intercluster correlation matrices.

SIMPLE
S

displays means and standard deviations.

SUMMARY

suppresses all default displayed output except the final summary table.

TRACE

lists the cluster to which each variable is assigned during the iterations.

VARDEF=DF
VARDEF=N
VARDEF=WDF
VARDEF=WEIGHT | WGT

specifies the divisor to be used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table.

Value	**Divisor**	**Formula**
DF	degrees of freedom	$n - i$
N	number of observations	n
WDF	sum of weights minus one	$(\sum_j w_j) - 1$
WEIGHT | WGT	sum of weights	$\sum_j w_j$

In the preceding table, $i = 0$ if the NOINT option is specified, and $i = 1$ otherwise.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC VARCLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the VARCLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure.

For more information on the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in your data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than 1, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered equal to the sum of the FREQ variable.

PARTIAL Statement

PARTIAL *variable* **;**

If you want to base the clustering on partial correlations, list the variables to be partialled out in the PARTIAL statement.

SEED Statement

SEED *variables* **;**

The SEED statement specifies variables to be used as seeds to initialize the clusters. It is not necessary to use INITIAL=SEED if the SEED statement is present, but if any other INITIAL= option is specified, the SEED statement is ignored.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the variables to be clustered. If you do not specify the VAR statement and do not specify TYPE=SSCP, all numeric variables not listed in other statements (except the SEED statement) are processed. The default VAR variable list does not include the variable INTERCEPT if the DATA= data set is TYPE=SSCP. If the variable INTERCEPT is explicitly specified in the VAR statement with a TYPE=SSCP data set, the NOINT option is enabled.

WEIGHT Statement

WEIGHT *variables* **;**

If you want to specify relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. The WEIGHT variable can take nonintegral values. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

Details

Missing Values

Observations containing missing values are omitted from the analysis.

Using PROC VARCLUS

Default options for PROC VARCLUS often provide satisfactory results. If you want to change the final number of clusters, use the MAXCLUSTERS=, MAXEIGEN=, or PROPORTION= options. The MAXEIGEN= and PROPORTION= options usually produce similar results but occasionally cause different clusters to be selected for splitting. The MAXEIGEN= option tends to choose clusters with a large number of variables, while the PROPORTION= option is more likely to select a cluster with a small number of variables.

Execution time

PROC VARCLUS usually requires more computer time than principal factor analysis, but it can be faster than some of the iterative factoring methods. If you have more than 30 variables, you may want to reduce execution time by one or more of the following methods:

- Specify the MINCLUSTERS= and MAXCLUSTERS= options if you know how many clusters you want.
- Specify the HIERARCHY option.
- Specify the SEED statement if you have some prior knowledge of what clusters to expect.

If computer time is not a limiting factor, you may want to try one of the following methods to obtain a better solution:

- If the clustering algorithm has not converged, specify larger values for MAXITER= and MAXSEARCH=.
- Try several factoring and rotation methods with PROC FACTOR to use as input to PROC VARCLUS.
- Run PROC VARCLUS several times, specifying INITIAL=RANDOM.

Output Data Sets

OUTSTAT= Data Set

The OUTSTAT= data set is TYPE=CORR, and it can be used as input to the SCORE procedure or a subsequent run of PROC VARCLUS. The variables it contains are

- BY variables
- _NCL_, a numeric variable giving the number of clusters
- _TYPE_, a character variable indicating the type of statistic the observation contains
- _NAME_, a character variable containing a variable name or a cluster name, which is of the form CLUSn where n is the number of the cluster
- the variables that are clustered

The values of the _TYPE_ variable are listed in the following table.

Table 78.2. _TYPE_ Value and Statistic

TYPE	**Contents**
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations, produced when the NOINT option is specified
N	number of observations
CORR	correlations
UCORR	uncorrected correlation matrix, produced when the NOINT option is specified
MEMBERS	number of members in each cluster
VAREXP	variance explained by each cluster
PROPOR	proportion of variance explained by each cluster
GROUP	number of the cluster to which each variable belongs
RSQUARED	squared multiple correlation of each variable with its cluster component
SCORE	standardized scoring coefficients
USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables, produced when the NOINT option is specified
STRUCTUR	cluster structure
CCORR	correlations between cluster components

The observations with _TYPE_=“MEAN”, “STD”, “N”, and “CORR” have missing values for the _NCL_ variable. All other values of the _TYPE_ variable are repeated for each cluster solution, with different solutions distinguished by the value of the _NCL_ variable. If you want to specify the OUTSTAT= data set with the SCORE procedure, you can use a DATA step to select observations with the _NCL_ variable missing or equal to the desired number of clusters.

```
data Coef2;
   set Coef;
   if _ncl_ = . or _ncl_ = 3;
   drop _ncl_;
run;

proc score data=NewScore score=Coef2; run;
```

PROC SCORE standardizes the new data by subtracting the original variable means that are stored in the _TYPE_='MEAN' observations, and dividing by the original variable standard deviations from the _TYPE_='STD' observations. Then PROC SCORE multiplies the standardized variables by the coefficients from the _TYPE_='SCORE' observations to get the cluster scores.

OUTTREE= Data Set

The OUTTREE= data set contains one observation for each variable clustered plus one observation for each cluster of two or more variables, that is, one observation for each node of the cluster tree. The total number of output observations is between n and $2n - 1$, where n is the number of variables clustered.

The variables in the OUTTREE= data set are

- BY variables, if any
- _NAME_, a character variable giving the name of the node. If the node is a cluster, the name is CLUSn where n is the number of the cluster. If the node is a single variable, the variable name is used.
- _PARENT_, a character variable giving the value of _NAME_ of the parent of the node. If the node is the root of the tree, _PARENT_ is blank.
- _LABEL_, a character variable giving the label of the node. If the node is a cluster, the label is CLUSn where n is the number of the cluster. If the node is a single variable, the variable label is used.
- _NCL_, the number of clusters.
- _VAREXP_, the total variance explained by the clusters at the current level of the tree.
- _PROPOR_, the total proportion of variance explained by the clusters at the current level of the tree.
- _MINPRO_, the minimum proportion of variance explained by a cluster component.
- _MAXEIG_, the maximum second eigenvalue of a cluster.

Computational Resources

Let

$$\begin{aligned} n &= \text{number of observations} \\ v &= \text{number of variables} \\ c &= \text{number of clusters} \end{aligned}$$

It is assumed that, at each stage of clustering, the clusters all contain the same number of variables.

Time

The time required for PROC VARCLUS to analyze a given data set varies greatly depending on the number of clusters requested, the number of iterations in both the alternating least-squares and search phases, and whether centroid or principal components are used.

The time required to compute the correlation matrix is roughly proportional to nv^2.

Default cluster initialization requires time roughly proportional to v^3. Any other method of initialization requires time roughly proportional to cv^2.

In the alternating least-squares phase, each iteration requires time roughly proportional to cv^2 if centroid components are used or

$$\left(c + 5\frac{v}{c^2}\right) v^2$$

if principal components are used.

In the search phase, each iteration requires time roughly proportional to v^3/c if centroid components are used or v^4/c^2 if principal components are used. The HIERARCHY option speeds up each iteration after the first split by as much as $c/2$.

Memory

The amount of memory, in bytes, needed by PROC VARCLUS is approximately

$$v^2 + 2vc + 20v + 15c$$

Interpreting VARCLUS Procedure Output

Because PROC VARCLUS is a type of oblique component analysis, its output is similar to the output from the FACTOR procedure for oblique rotations. The scoring coefficients have the same meaning in both PROC VARCLUS and PROC FACTOR; they are coefficients applied to the standardized variables to compute component scores. The cluster structure is analogous to the factor structure containing the correlations between each variable and each cluster component. A cluster pattern is not displayed because it would be the same as the cluster structure, except that zeros would appear in the same places in which zeros appear in the scoring coefficients. The intercluster correlations are analogous to interfactor correlations; they are the correlations among cluster components.

PROC VARCLUS also displays a cluster summary and a cluster listing. The cluster summary gives the number of variables in each cluster and the variation explained by the cluster component. The latter is similar to the variation explained by a factor but includes contributions from only the variables in that cluster rather than from all variables, as in PROC FACTOR. The proportion of variance explained is obtained by dividing the variance explained by the total variance of variables in the cluster. If the cluster contains two or more variables and the CENTROID option is not used, the second largest eigenvalue of the cluster is also displayed.

The cluster listing gives the variables in each cluster. Two squared correlations are calculated for each cluster. The column labeled “Own Cluster” gives the squared correlation of the variable with its own cluster component. This value should be higher than the squared correlation with any other cluster unless an iteration limit has been exceeded or the CENTROID option has been used. The larger the squared correlation is, the better. The column labeled “Next Closest” contains the next highest squared correlation of the variable with a cluster component. This value is low if the clusters are well separated. The column headed “1–R**2 Ratio” gives the ratio of one minus the “Own Cluster” R^2 to one minus the “Next Closest” R^2. A small “1–R**2 Ratio” indicates a good clustering.

Displayed Output

The following items are displayed for each cluster solution unless the NOPRINT or SUMMARY option is specified. The CLUSTER SUMMARY table includes

- the Cluster number
- Members, the number of members in the cluster
- Cluster Variation of the variables in the cluster
- Variation Explained by the cluster component. This statistic is based only on the variables in the cluster rather than on all variables.
- Proportion Explained, the result of dividing the variation explained by the cluster variation
- Second Eigenvalue, the second largest eigenvalue of the cluster. This is displayed if the cluster contains more than one variable and the CENTROID option is not specified

PROC VARCLUS also displays

- Total variation explained, the sum across clusters of the variation explained by each cluster
- Proportion, the total explained variation divided by the total variation of all the variables

The cluster listing includes

- Variable, the variables in each cluster
- R-squared with Own Cluster, the squared correlation of the variable with its own cluster component; and R-squared with Next Closest, the next highest squared correlation of the variable with a cluster component. Own Cluster values should be higher than the R^2 with any other cluster unless an iteration limit is exceeded or you specify the CENTROID option. Next Closest should be a low value if the clusters are well separated.
- 1−R**2 Ratio, the ratio of one minus the value in the Own Cluster column to one minus the value in the Next Closest column. The occurrence of low ratios indicates well-separated clusters.

If the SHORT option is not specified, PROC VARCLUS also displays

- Standardized Scoring Coefficients, standardized regression coefficients for predicting cluster components from variables
- Cluster Structure, the correlations between each variable and each cluster component
- Inter-Cluster Correlations, the correlations between the cluster components

If the analysis includes partitions for two or more numbers of clusters, a final summary table is displayed. Each row of the table corresponds to one partition. The columns include

- Number of Clusters
- Total Variation Explained by Clusters
- Proportion of Variation Explained by Clusters
- Minimum Proportion (of variation) Explained by a Cluster
- Maximum Second Eigenvalue in a Cluster
- Minimum R-squared for a Variable
- Maximum 1−R**2 Ratio for a Variable

ODS Table Names

PROC VARCLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 78.3. ODS Tables Produced in PROC VARCLUS

ODS Table Name	Description	Option
ClusterQuality	Cluster quality	default
ClusterStructure	Cluster structure	default
ClusterSummary	Cluster Summary	default
ConvergenceStatus	Convergence status	default
Corr	Correlations	CORR
DataOptSummary	Data and options summary table	default
InterClusterCorr	Inter-cluster correlations	default
IterHistory	Iteration history	TRACE
RSquare	Cluster Rsq	default
SimpleStatistics	Simple statistics	SIMPLE
StdScoreCoef	Standardized scoring coefficients	default

Example

Example 78.1. Correlations among Physical Variables

The following data are correlations among eight physical variables as given by Harman (1976). The first PROC VARCLUS run clusters on the basis of principal components, the second run clusters on the basis of centroid components. The third analysis is hierarchical, and the TREE procedure is used to display a tree diagram. The results of the analyses follow.

```
data phys8(type=corr);
   title ’Eight Physical Measurements on 305 School Girls’;
   title2 ’Harman: Modern Factor Analysis, 3rd Ed, p22’;
   label height=’Height’      arm_span=’Arm Span’
         forearm=’Length of Forearm’
         low_leg=’Length of Lower Leg’
         weight=’Weight’      bit_diam=’Bitrochanteric Diameter’
         girth=’Chest Girth’  width=’Chest Width’;
   input _name_ $ 1-8
         (height arm_span forearm low_leg weight bit_diam
          girth width)(7.);
   _type_=’corr’;
   datalines;
height  1.0    .846   .805   .859   .473   .398   .301   .382
arm_span.846   1.0    .881   .826   .376   .326   .277   .415
forearm .805   .881   1.0    .801   .380   .319   .237   .345
low_leg .859   .826   .801   1.0    .436   .329   .327   .365
weight  .473   .376   .380   .436   1.0    .762   .730   .629
bit_diam.398   .326   .319   .329   .762   1.0    .583   .577
girth   .301   .277   .237   .327   .730   .583   1.0    .539
width   .382   .415   .345   .365   .629   .577   .539   1.0
;

proc varclus data=phys8;
run;
```

The PROC VARCLUS statement invokes the procedure. By default, PROC VARCLUS clusters on the basis of principal components.

Output 78.1.1. Principal Cluster Components: Cluster Summary

```
                Eight Physical Measurements on 305 School Girls
                  Harman: Modern Factor Analysis, 3rd Ed, p22

                  Oblique Principal Component Cluster Analysis

                          Cluster Summary for 1 Cluster

                               Cluster    Variation    Proportion        Second
    Cluster    Members    Variation    Explained      Explained    Eigenvalue
    ------------------------------------------------------------------------
          1          8            8      4.67288         0.5841        1.7710

            Total variation explained = 4.67288 Proportion = 0.5841

Cluster 1 will be split.


                         Cluster Summary for 2 Clusters

                               Cluster    Variation    Proportion        Second
    Cluster    Members    Variation    Explained      Explained    Eigenvalue
    ------------------------------------------------------------------------
          1          4            4     3.509218         0.8773        0.2361
          2          4            4     2.917284         0.7293        0.4764

            Total variation explained = 6.426502 Proportion = 0.8033


                               R-squared with
  2 Clusters                 -----------------
                                  Own      Next    1-R**2   Variable
  Cluster      Variable    Cluster   Closest     Ratio   Label
  --------------------------------------------------------------------------
  Cluster 1    height        0.8777    0.2088    0.1545   Height
               arm_span      0.9002    0.1658    0.1196   Arm Span
               forearm       0.8661    0.1413    0.1560   Length of Forearm
               low_leg       0.8652    0.1829    0.1650   Length of Lower Leg
  --------------------------------------------------------------------------
  Cluster 2    weight        0.8477    0.1974    0.1898   Weight
               bit_diam      0.7386    0.1341    0.3019   Bitrochanteric Diameter
               girth         0.6981    0.0929    0.3328   Chest Girth
               width         0.6329    0.1619    0.4380   Chest Width

No cluster meets the criterion for splitting.
```

As displayed in Output 78.1.1, the cluster component (by default, the first principal component) explains 58.41% of the total variation in the 8 variables.

The cluster is split because the second eigenvalue is greater than 1 (the default value of the MAXEIGEN option).

The two resulting cluster components explain 80.33% of the variation in the original variables. The cluster summary table shows that the variables height, arm_span, forearm, and low_leg have been assigned to the first cluster; and that the variables weight, bit_diam, girth, and width have been assigned to the second cluster.

Output 78.1.2. Standard Scoring Coefficients and Cluster Structure Table

```
                  Oblique Principal Component Cluster Analysis

                       Standardized Scoring Coefficients

       Cluster                                            1             2
       ----------------------------------------------------------------
       height        Height                        0.266977      0.000000
       arm_span      Arm Span                      0.270377      0.000000
       forearm       Length of Forearm             0.265194      0.000000
       low_leg       Length of Lower Leg           0.265057      0.000000
       weight        Weight                        0.000000      0.315597
       bit_diam      Bitrochanteric Diameter       0.000000      0.294591
       girth         Chest Girth                   0.000000      0.286407
       width         Chest Width                   0.000000      0.272710


                               Cluster Structure

       Cluster                                            1             2
       ----------------------------------------------------------------
       height        Height                        0.936881      0.456908
       arm_span      Arm Span                      0.948813      0.407210
       forearm       Length of Forearm             0.930624      0.375865
       low_leg       Length of Lower Leg           0.930142      0.427715
       weight        Weight                        0.444281      0.920686
       bit_diam      Bitrochanteric Diameter       0.366201      0.859404
       girth         Chest Girth                   0.304779      0.835529
       width         Chest Width                   0.402430      0.795572
```

The standardized scoring coefficients in Output 78.1.2 show that each cluster component has similar scores for each of its associated variables. This suggests that the principal cluster component solution should be similar to the centroid cluster component solution, which follows in the next PROC VARCLUS run.

The cluster structure table displays high correlations between the variables and their own cluster component. The correlations between the variables and the opposite cluster component are all moderate.

Output 78.1.3. Inter-Cluster Correlations

```
                  Oblique Principal Component Cluster Analysis

                          Inter-Cluster Correlations

                      Cluster             1             2

                      1             1.00000       0.44513
                      2             0.44513       1.00000
```

The intercluster correlation table shows that the cluster components are moderately correlated with $\rho = 0.44513$.

In the following statements, the CENTROID option in the PROC VARCLUS statement specifies that cluster centroids be used as the basis for clustering.

```
   proc varclus data=phys8 centroid;
   run;
```

Output 78.1.4. Centroid Cluster Components: Cluster Summary

```
                          Oblique Centroid Component Cluster Analysis

                                 Cluster Summary for 1 Cluster

                                     Cluster    Variation    Proportion
                   Cluster    Members    Variation    Explained     Explained
                   ----------------------------------------------------------
                         1          8            8        4.631        0.5789

                     Total variation explained = 4.631 Proportion = 0.5789


                                Cluster Summary for 2 Clusters

                                     Cluster    Variation    Proportion
                   Cluster    Members    Variation    Explained     Explained
                   ----------------------------------------------------------
                         1          4            4        3.509        0.8773
                         2          4            4         2.91        0.7275

                     Total variation explained = 6.419 Proportion = 0.8024


                                   R-squared with
       2 Clusters                ------------------
                                    Own       Next     1-R**2    Variable
       Cluster       Variable    Cluster    Closest     Ratio    Label
       --------------------------------------------------------------------------------
       Cluster 1     height       0.8778     0.2075    0.1543    Height
                     arm_span     0.8994     0.1669    0.1208    Arm Span
                     forearm      0.8663     0.1410    0.1557    Length of Forearm
                     low_leg      0.8658     0.1824    0.1641    Length of Lower Leg
       --------------------------------------------------------------------------------
       Cluster 2     weight       0.8368     0.1975    0.2033    Weight
                     bit_diam     0.7335     0.1341    0.3078    Bitrochanteric Diameter
                     girth        0.6988     0.0929    0.3321    Chest Girth
                     width        0.6473     0.1618    0.4207    Chest Width
```

The first cluster component, which, in the centroid method, is an unweighted sum of the standardized variables, explains 57.89% of the variation in the data. This value is near the maximum possible variance explained, 58.41%, which is attained by the first principal component (Output 78.1.1).

The centroid clustering algorithm splits the variables into the same two clusters created in the principal component method. Recall that this outcome was suggested by the similar standardized scoring coefficients in the principal cluster component solution.

The default behavior in the centroid method is to split any cluster with less than 75% of the total cluster variance explained by the centroid component. In the next step, the second cluster, with a component that explains only 72.75% of the total variation of the cluster, is split.

In the R-squared table for two clusters, the width variable has a weaker relation to its cluster than any other variable; in the three cluster solution this variable is in a cluster of its own.

Output 78.1.5. Standardized Scoring Coefficients

```
                     Oblique Centroid Component Cluster Analysis

                          Standardized Scoring Coefficients

        Cluster                                             1             2
        -----------------------------------------------------------------
        height        Height                         0.266918      0.000000
        arm_span      Arm Span                       0.266918      0.000000
        forearm       Length of Forearm              0.266918      0.000000
        low_leg       Length of Lower Leg            0.266918      0.000000
        weight        Weight                         0.000000      0.293105
        bit_diam      Bitrochanteric Diameter        0.000000      0.293105
        girth         Chest Girth                    0.000000      0.293105
        width         Chest Width                    0.000000      0.293105
```

Each cluster component (Output 78.1.5) is an unweighted average of the cluster's standardized variables. Thus, the coefficients for each of the cluster's associated variables are identical in the centroid cluster component solution.

Output 78.1.6. Cluster Summary for Three Clusters

```
                     Oblique Centroid Component Cluster Analysis

                            Cluster Summary for 3 Clusters

                                  Cluster     Variation     Proportion
            Cluster    Members    Variation    Explained      Explained
            ----------------------------------------------------------
                  1          4            4        3.509         0.8773
                  2          3            3     2.383333         0.7944
                  3          1            1            1         1.0000

             Total variation explained = 6.892333 Proportion = 0.8615


                              R-squared with
3 Clusters                 ------------------
                                 Own       Next    1-R**2    Variable
Cluster       Variable       Cluster    Closest     Ratio    Label
---------------------------------------------------------------------------------
Cluster 1     height          0.8778     0.1921    0.1513    Height
              arm_span        0.8994     0.1722    0.1215    Arm Span
              forearm         0.8663     0.1225    0.1524    Length of Forearm
              low_leg         0.8658     0.1668    0.1611    Length of Lower Leg
---------------------------------------------------------------------------------
Cluster 2     weight          0.8685     0.3956    0.2175    Weight
              bit_diam        0.7691     0.3329    0.3461    Bitrochanteric Diameter
              girth           0.7482     0.2905    0.3548    Chest Girth
---------------------------------------------------------------------------------
Cluster 3     width           1.0000     0.4259    0.0000    Chest Width
```

The centroid method stops at the three cluster solution. As displayed in Output 78.1.6 and Output 78.1.7, the three centroid components account for 86.15% of the variability in the eight variables, and all cluster components account for at least 79.44% of the total variation in the corresponding cluster. Additionally, the smallest squared correlation between the variables and their own cluster component is 0.7482.

Output 78.1.7. Cluster Quality Table

```
                          Oblique Centroid Component Cluster Analysis

                           Total      Proportion         Minimum      Minimum          Maximum
          Number       Variation    of Variation      Proportion    R-squared     1-R**2 Ratio
              of       Explained       Explained       Explained        for a            for a
        Clusters     by Clusters     by Clusters    by a Cluster     Variable         Variable
        --------------------------------------------------------------------------------------
               1        4.631000          0.5789          0.5789       0.4306
               2        6.419000          0.8024          0.7275       0.6473           0.4207
               3        6.892333          0.8615          0.7944       0.7482           0.3548
```

Note that, if the proportion option were set to a value between 0.5789 (the proportion of variance explained in the 1-cluster solution) and 0.7275 (the minimum proportion of variance explained in the 2-cluster solution), PROC VARCLUS would stop at a two cluster solution, and the centroid solution would find the same clusters as the principal components solution.

In the following statements, the MAXC= option computes all clustering solutions, from one to eight clusters. The SUMMARY option suppresses all output except the final cluster quality table, and the OUTTREE= option saves the results of the analysis to an output data set and forces the clusters to be hierarchical. The TREE procedure is invoked to produce a graphical display of the clusters.

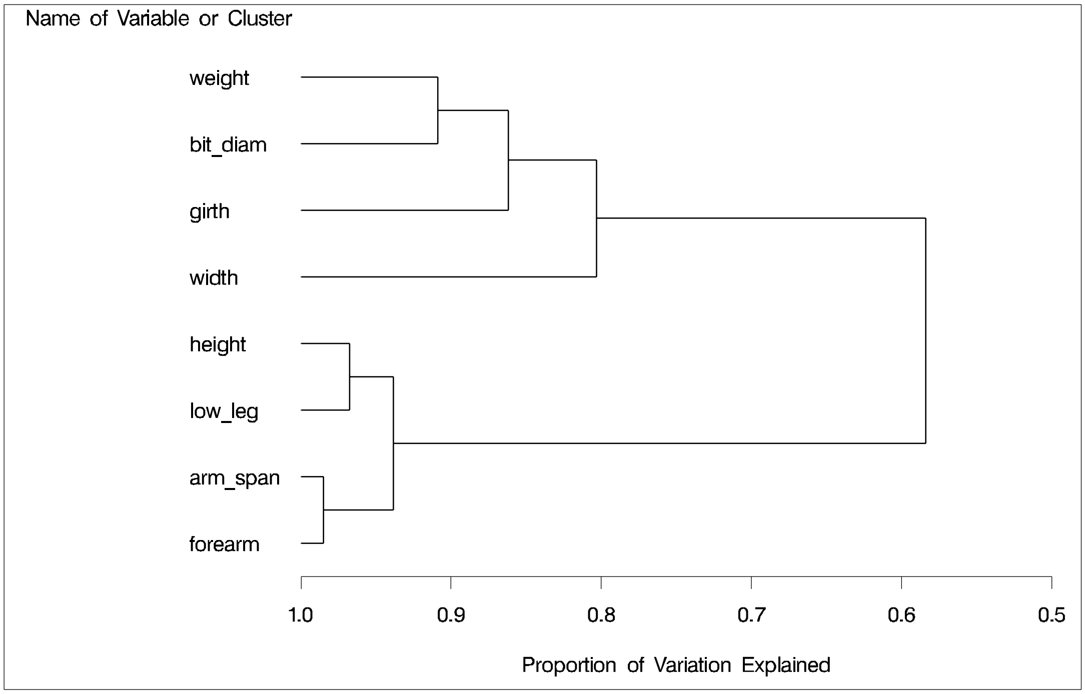
```
proc varclus data=phys8 maxc=8 summary outtree=tree;
run;

goptions ftext=swiss;
axis2 label=(justify=left);
axis1 order=(0.5 to 1.0 by 0.1);
proc tree horizontal vaxis=axis2 haxis=axis1 lines=(width=2);
   height _propor_;
   id _label_;
run;
```

Output 78.1.8. Hierarchical Clusters and the SUMMARY Option

```
                         Oblique Principal Component Cluster Analysis

                 Total    Proportion       Minimum       Maximum     Minimum        Maximum
  Number     Variation  of Variation    Proportion        Second   R-squared   1-R**2 Ratio
      of     Explained     Explained     Explained    Eigenvalue       for a          for a
Clusters   by Clusters   by Clusters  by a Cluster  in a Cluster    Variable       Variable
-------------------------------------------------------------------------------------------
       1      4.672880        0.5841        0.5841      1.770983      0.3810
       2      6.426502        0.8033        0.7293      0.476418      0.6329         0.4380
       3      6.895347        0.8619        0.7954      0.418369      0.7421         0.3634
       4      7.271218        0.9089        0.8773      0.238000      0.8652         0.2548
       5      7.509218        0.9387        0.8773      0.236135      0.8652         0.1665
       6      7.740000        0.9675        0.9295      0.141000      0.9295         0.2560
       7      7.881000        0.9851        0.9405      0.119000      0.9405         0.2093
       8      8.000000        1.0000        1.0000      0.000000      1.0000         0.0000
```

The principal component method first separates the variables into the same two clusters that were created in the first PROC VARCLUS run. Note that, in creating the third cluster, the principal component method identifies the variable width. This is the same variable that is put into its own cluster in the preceding centroid method example.

Output 78.1.9. TREE Diagram from PROC TREE



The tree diagram in Output 78.1.9 displays the cluster hierarchy. It is clear from the diagram that there are two, or possibly three, clusters present. However, the MAXC=8 option forces PROC VARCLUS to split the clusters until each variable is in its own cluster.

References

Anderberg, M.R. (1973), *Cluster Analysis for Applications*, New York: Academic Press, Inc.

Harris, C.W.; and Kaiser, H.F. (1964), “Oblique factor analytic solutions by orthogonal transformation,” *Psychometrika*, 32, 363–379.

Harman, H.H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Hand, D.J.; Daly, F.; Lunn, A.D.; McConway, K.J.; and Ostrowski E. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall, 297–298.

Chapter 79
The VARCOMP Procedure

Chapter Contents

Chapter 79
The VARCOMP Procedure

Overview

The VARCOMP procedure handles general linear models that have random effects. Random effects are classification effects with levels that are assumed to be randomly selected from an infinite population of possible levels. PROC VARCOMP estimates the contribution of each of the random effects to the variance of the dependent variable.

A single MODEL statement specifies the dependent variables and the effects: main effects, interactions, and nested effects. The effects must be composed of class variables; no continuous variables are allowed on the right side of the equal sign.

You can specify certain effects as fixed (nonrandom) by putting them first in the MODEL statement and indicating the number of fixed effects with the FIXED= option. An intercept is always fitted and assumed fixed. Except for the effects specified as fixed, all other effects are assumed to be random, and their contribution to the model can be thought of as an observation from a distribution that is normally and independently distributed.

The dependent variables are grouped based on the similarity of their missing values. Each group of dependent variables is then analyzed separately. The columns of the design matrix **X** are formed in the same order in which the effects are specified in the MODEL statement. No reparameterization is done. Thus, the columns of **X** contain only 0s and 1s.

You can specify four methods of estimation in the PROC VARCOMP statement using the METHOD= option. They are TYPE1 (based on computation of Type I sum of squares for each effect), MIVQUE0, Maximum Likelihood (METHOD=ML), and Restricted Maximum Likelihood (METHOD=REML).

Getting Started

Analyzing the Cure Rate of Rubber

This example, using data from Hicks (1973), concerns an experiment to determine the sources of variability in cure rates of rubber. The goal of the experiment was to find out if the different laboratories contributed more to the variance of cure rates than did the different batches of raw materials. This information would be useful in trying to control the cure rate of the final product because it would provide insights into the sources of the variability in cure rates. The rubber used was cured at three temperatures, which were taken to be fixed. Three laboratories were chosen at random, and three different batches of raw material were tested at each combination of temperature and laboratory. The following statements read the data into the SAS data set Cure.

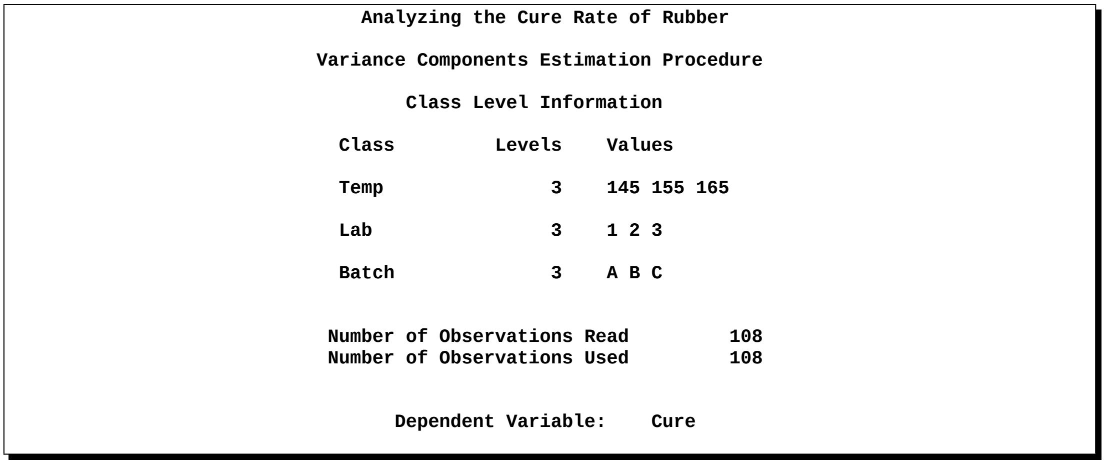
```
title ’Analyzing the Cure Rate of Rubber’;
data Cure;
   input Lab Temp Batch $ Cure @@;
   datalines;
1 145 A 18.6   1 145 A 17.0   1 145 A 18.7   1 145 A 18.7
1 145 B 14.5   1 145 B 15.8   1 145 B 16.5   1 145 B 17.6
1 145 C 21.1   1 145 C 20.8   1 145 C 21.8   1 145 C 21.0
1 155 A  9.5   1 155 A  9.4   1 155 A  9.5   1 155 A 10.0
1 155 B  7.8   1 155 B  8.3   1 155 B  8.9   1 155 B  9.1
1 155 C 11.2   1 155 C 10.0   1 155 C 11.5   1 155 C 11.1
1 165 A  5.4   1 165 A  5.3   1 165 A  5.7   1 165 A  5.3
1 165 B  5.2   1 165 B  4.9   1 165 B  4.3   1 165 B  5.2
1 165 C  6.3   1 165 C  6.4   1 165 C  5.8   1 165 C  5.6
2 145 A 20.0   2 145 A 20.1   2 145 A 19.4   2 145 A 20.0
2 145 B 18.4   2 145 B 18.1   2 145 B 16.5   2 145 B 16.7
2 145 C 22.5   2 145 C 22.7   2 145 C 21.5   2 145 C 21.3
2 155 A 11.4   2 155 A 11.5   2 155 A 11.4   2 155 A 11.5
2 155 B 10.8   2 155 B 11.1   2 155 B  9.5   2 155 B  9.7
2 155 C 13.3   2 155 C 14.0   2 155 C 12.0   2 155 C 11.5
2 165 A  6.8   2 165 A  6.9   2 165 A  6.0   2 165 A  5.7
2 165 B  6.0   2 165 B  6.1   2 165 B  5.0   2 165 B  5.2
2 165 C  7.7   2 165 C  8.0   2 165 C  6.6   2 165 C  6.3
3 145 A 19.7   3 145 A 18.3   3 145 A 16.8   3 145 A 17.1
3 145 B 16.3   3 145 B 16.7   3 145 B 14.4   3 145 B 15.2
3 145 C 22.7   3 145 C 21.9   3 145 C 19.3   3 145 C 19.3
3 155 A  9.3   3 155 A 10.2   3 155 A  9.8   3 155 A  9.5
3 155 B  9.1   3 155 B  9.2   3 155 B  8.0   3 155 B  9.0
3 155 C 11.3   3 155 C 11.0   3 155 C 10.9   3 155 C 11.4
3 165 A  6.7   3 165 A  6.0   3 165 A  5.0   3 165 A  4.8
3 165 B  5.7   3 165 B  5.5   3 165 B  4.6   3 165 B  5.4
3 165 C  6.6   3 165 C  6.5   3 165 C  5.9   3 165 C  5.8
;
```

The variables Lab, Temp, and Batch contain levels of laboratory, temperature, and batch, respectively. The Cure variable contains the response values.

The following SAS statements perform a restricted maximum-likelihood variance component analysis.

```
proc varcomp method=reml;
   class Temp Lab Batch;
   model Cure=Temp|Lab Batch(Lab Temp) / fixed=1;
run;
```

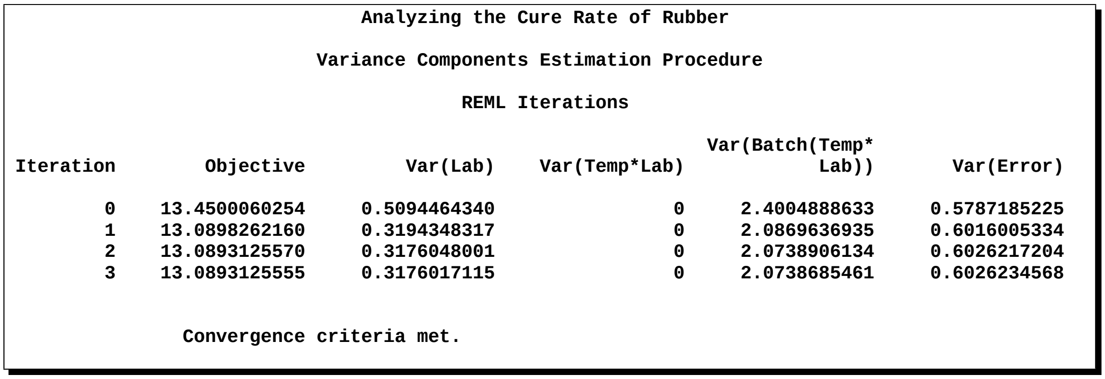
The FIXED=1 option indicates that the first factor, Temp, is fixed. The effect specification Temp|Lab is equivalent to putting the three terms Temp, Lab, and Temp*Lab in the model. Batch(Lab Temp) is equivalent to putting Batch(Temp*Lab) in the MODEL statement. The results of this analysis are displayed in Figure 79.1 through Figure 79.4.



```
                     Analyzing the Cure Rate of Rubber

                  Variance Components Estimation Procedure

                          Class Level Information

                   Class         Levels    Values

                   Temp               3    145 155 165

                   Lab                3    1 2 3

                   Batch              3    A B C


                  Number of Observations Read         108
                  Number of Observations Used         108


                        Dependent Variable:    Cure
```

Figure 79.1. Class Level Information

Figure 79.1 provides information about the variables used in the analysis and the number of observations and specifies the dependent variable.



```
                     Analyzing the Cure Rate of Rubber

                  Variance Components Estimation Procedure

                              REML Iterations

                                                           Var(Batch(Temp*
Iteration       Objective         Var(Lab)    Var(Temp*Lab)            Lab))        Var(Error)

        0    13.4500060254     0.5094464340                0     2.4004888633     0.5787185225
        1    13.0898262160     0.3194348317                0     2.0869636935     0.6016005334
        2    13.0893125570     0.3176048001                0     2.0738906134     0.6026217204
        3    13.0893125555     0.3176017115                0     2.0738685461     0.6026234568


               Convergence criteria met.
```

Figure 79.2. Iteration History

The “REML Iterations” table, shown in Figure 79.2, displays the iteration history, which includes the value of the objective function associated with REML and the values of the variance components at each iteration.

```
                       Analyzing the Cure Rate of Rubber

                    Variance Components Estimation Procedure

                                REML Estimates

                   Variance Component        Estimate

                   Var(Lab)                   0.31760
                   Var(Temp*Lab)                    0
                   Var(Batch(Temp*Lab))       2.07387
                   Var(Error)                 0.60262
```

Figure 79.3. REML Estimates

Figure 79.3 displays the REML estimates of the variance components.

```
                                Analyzing the Cure Rate of Rubber

                             Variance Components Estimation Procedure

                            Asymptotic Covariance Matrix of Estimates

                             Var(Lab)         Var(Temp*Lab)  Var(Batch(Temp*Lab))           Var(Error)

Var(Lab)                      0.32452                     0              -0.04998           1.0259E-12
Var(Temp*Lab)                       0                     0                     0                    0
Var(Batch(Temp*Lab))         -0.04998                     0               0.45042           -0.0022417
Var(Error)                 1.0259E-12                     0            -0.0022417            0.0089668
```

Figure 79.4. Covariance Matrix for REML Estimates

The “Asymptotic Covariance Matrix of Estimates” table in Figure 79.4 displays the asymptotic covariance matrix of the REML estimates.

The results of the analysis show that the variance attributable to Batch(Temp*Lab) (with a variance component of 2.0739) is considerably larger than the variance attributable to Lab (0.3176). Therefore, attempts to reduce the variability of cure rates should concentrate on improving the homogeneity of the batches of raw material used rather than standardizing the practices or equipment within the laboratories. Also, note that since the Batch(Temp*Lab) variance is considerably larger than the experimental error (Var(Error)=0.6026), the Batch(Temp*Lab) variability plays an important part in the overall variability of the cure rates.

Syntax

The following statements are available in PROC VARCOMP.

> **PROC VARCOMP** < *options* > **;**
> **CLASS** *variables* **;**
> **MODEL** *dependent* = < *effects* > < / *options* > **;**
> **BY** *variables* **;**

Only one MODEL statement is allowed. The BY, CLASS, and MODEL statements are described after the PROC VARCOMP statement.

PROC VARCOMP Statement

PROC VARCOMP < *options* >**;**

This statement invokes the VARCOMP procedure. You can specify the following options in the PROC VARCOMP statement.

DATA=*SAS-data-set*
specifies the input SAS data set to use. If this option is omitted, the most recently created SAS data set is used.

EPSILON=*number*
specifies the convergence value of the objective function for METHOD=ML or METHOD=REML. By default, EPSILON=1E−8.

MAXITER=*number*
specifies the maximum number of iterations for METHOD=ML or METHOD=REML. By default, MAXITER=50.

METHOD=TYPE1 | MIVQUE0 | ML | REML
specifies which of the four methods (TYPE1, MIVQUE0, ML, or REML) you want to use. By default, METHOD= MIVQUE0. For more information see the "Computational Methods" section on page 4838.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC VARCOMP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives.

- Sort the data using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the VARCOMP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables using the DATASETS procedure (in base SAS software).

For more information on the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information on the DATASETS procedure, see the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables to be used in the analysis. All effects in the MODEL statement must be composed of effects that appear in the CLASS statement. Class variables can be either numeric or character; if they are character, only the first 16 characters are used.

Numeric class variables are not restricted to integers since a variable's format determines the levels. For more information, see the discussion of the FORMAT statement in *SAS Language Reference: Dictionary*.

MODEL Statement

MODEL *dependent* = < *effects* > < / *option* > **;**

The MODEL statement gives the dependent variables and independent effects. If you specify more than one dependent variable, a separate analysis is performed for each one. The independent effects are limited to main effects, interactions, and nested effects; no continuous effects are allowed. All independent effects must be composed of effects that appear in the CLASS statement. Effects are specified in the VARCOMP procedure in the same way as described for the ANOVA procedure. Only one MODEL statement is allowed.

Only one option is available in the MODEL statement.

FIXED=*n*

tells the VARCOMP procedure that the first *n* effects in the MODEL statement are fixed effects. The remaining effects are assumed to be random. By default, PROC VARCOMP assumes that all effects are random in the model. Keep in mind that if you use bar notation and, for example, specify Y=A|B / FIXED=2, then A*B is considered a random effect.

Details

Missing Values

If an observation has a missing value for any variable used in the independent effects, then the analyses of all dependent variables omit this observation. An observation is deleted from the analysis of a given dependent variable if the observation's value for that dependent variable is missing. Note that a missing value in one dependent variable does not eliminate an observation from the analysis of the other dependent variables.

During processing, PROC VARCOMP groups the dependent variables on their missing values across observations so that sums of squares and cross products can be computed in the most efficient manner.

Fixed and Random Effects

Central to the idea of variance components models is the idea of fixed and random effects. Each effect in a variance components model must be classified as either a fixed or a random effect. Fixed effects arise when the levels of an effect constitute the entire population about which you are interested. For example, if a plant scientist is comparing the yields of three varieties of soybeans, then Variety would be a fixed effect, providing that the scientist was concerned about making inferences on only these three varieties of soybeans. Similarly, if an industrial experiment focused on the effectiveness of two brands of a machine, Machine would be a fixed effect only if the experimenter's interest did not go beyond the two machine brands.

On the other hand, an effect is classified as a random effect when you want to make inferences on an entire population, and the levels in your experiment represent only a sample from that population. Psychologists comparing test results between different groups of subjects would consider Subject as a random effect. Depending on the psychologists' particular interest, the Group effect might be either fixed or random. For example, if the groups are based on the sex of the subject, then Sex would be a fixed effect. But if the psychologists are interested in the variability in test scores due to different teachers, then they might choose a random sample of teachers as being representative of the total population of teachers, and Teacher would be a random effect. Note that, in the soybean example presented earlier, if the scientists are interested in making inferences on the entire population of soybean varieties and randomly choose three varieties for testing, then Variety would be a random effect.

If all the effects in a model (except for the intercept) are considered random effects, then the model is called a *random effects model*; likewise, a model with only fixed effects is called a *fixed-effects model*. The more common case, where some factors are fixed and others are random, is called a *mixed model*. In PROC VARCOMP, by default, effects are assumed to be random. You specify which effects are fixed by using the FIXED= option in the MODEL statement. In general, if an interaction or nested effect contains any effect that is random, then the interaction or nested effect should be considered as a random effect as well.

In the linear model, each level of a fixed effect contributes a fixed amount to the expected value of the dependent variable. What makes a random effect different is that each level of a random effect contributes an amount that is viewed as a sample from a population of normally distributed variables, each with mean 0, and an unknown variance, much like the usual random error term that is a part of all linear models. The estimate of the variance associated with the random effect is known as the *variance component* because it is measuring the part of the overall variance contributed by that effect. Thus, PROC VARCOMP estimates the variance of the random variables that are associated with the random effects in your model, and the variance components tell you how much each of the random factors contributes to the overall variability in the dependent variable.

Negative Variance Component Estimates

The variance components estimated by PROC VARCOMP should theoretically be nonnegative because they are assumed to represent the variance of a random variable. Nevertheless, when you are using METHOD=MIVQUE0 (the default) or METHOD=TYPE1, some estimates of variance components may become negative. (Due to the nature of the algorithms used for METHOD=ML and METHOD=REML, negative estimates are constrained to zero.) These negative estimates may arise for a variety of reasons:

- The variability in your data may be large enough to produce a negative estimate, even though the true value of the variance component is positive.
- Your data may contain outliers. Refer to Hocking (1983) for a graphical technique for detecting outliers in variance components models using the SAS System.
- A different model for interpreting your data may be appropriate. Under some statistical models for variance components analysis, negative estimates are an indication that observations in your data are negatively correlated. Refer to Hocking (1984) for further information about these models.

Assuming that you are satisfied that the model PROC VARCOMP is using is appropriate for your data, it is common practice to treat negative variance components as if they are zero.

Computational Methods

Four methods of estimation can be specified in the PROC VARCOMP statement using the METHOD= option. They are described in the following sections.

The Type I Method

This method (METHOD=TYPE1) computes the Type I sum of squares for each effect, equates each mean square involving only random effects to its expected value, and solves the resulting system of equations (Gaylor, Lucas, and Anderson 1970). The $\mathbf{X}'\mathbf{X} \mid \mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix.

The MIVQUE0 Method

Based on the technique suggested by Hartley, Rao, and LaMotte (1978), the MIVQUE0 method (METHOD=MIVQUE0) produces unbiased estimates that are invariant with respect to the fixed effects of the model and that are locally best quadratic unbiased estimates given that the true ratio of each component to the residual error component is zero. The technique is similar to TYPE1 except that the random effects are adjusted only for the fixed effects. This affords a considerable timing advantage over the TYPE1 method; thus, MIVQUE0 is the default method used in PROC VARCOMP. The $\mathbf{X}'\mathbf{X}|\mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix. Each element (i, j) of the form

$$\mathrm{SSQ}(\mathbf{X}_i'\mathbf{M}\mathbf{X}_j)$$

is computed, where

$$\mathbf{M} = \mathbf{I} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{X}_0)^{-}\mathbf{X}_0'$$

and where $\mathbf{X}_0$ is part of the design matrix for the fixed effects, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and SSQ is an operator that takes the sum of squares of the elements. For more information refer to Rao (1971, 1972) and Goodnight (1978).

The Maximum Likelihood Method

The Maximum Likelihood method (METHOD=ML) computes maximum-likelihood estimates of the variance components; refer to Searle, Casella, and McCulloch (1992). The computing algorithm makes use of the W-transformation developed by Hemmerle and Hartley (1973). The procedure uses a Newton-Raphson algorithm, iterating until the log-likelihood objective function converges.

The objective function for METHOD=ML is $\ln(|\mathbf{V}|) + \mathbf{r}'\mathbf{V}^{-1}\mathbf{r}$, where

$$\mathbf{V} = \sigma_0^2\mathbf{I} + \sum_{i=1}^{n_r} \sigma_i^2 \mathbf{X}_i\mathbf{X}_i'$$

and where σ_0^2 is the residual variance, n_r is the number of random effects in the model, σ_i^2 represents the variance components, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and

$$\mathbf{r} = \mathbf{y} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0)^{-}\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{y}$$

is the vector of residuals.

The Restricted Maximum Likelihood Method

The Restricted Maximum Likelihood Method (METHOD=REML) is similar to the maximum likelihood method, but it first separates the likelihood into two parts: one that contains the fixed effects and one that does not (Patterson and Thompson 1971). The procedure uses a Newton-Raphson algorithm, iterating until convergence is reached for the log-likelihood objective function of the portion of the likelihood that does not contain the fixed effects. Using notation from earlier methods, the objective function for METHOD=REML is $\ln(|\mathbf{V}|) + \mathbf{r}'\mathbf{V}^{-1}\mathbf{r} + \ln(|\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0|)$. Refer to Searle, Casella, and McCulloch (1992) for additional details.

Displayed Output

PROC VARCOMP displays the following items:

- Class Level Information for verifying the levels in your data
- Number of observations read from the data set and number of observations used in the analysis
- for METHOD=TYPE1, an analysis-of-variance table with Source, DF, Type I Sum of Squares, Type I Mean Square, and Expected Mean Square, and a table of Type I variance component estimates
- for METHOD=MIVQUE0, the SSQ Matrix containing sums of squares of partitions of the $\mathbf{X}'\mathbf{X}$ crossproducts matrix adjusted for the fixed effects
- for METHOD=ML and METHOD=REML, the iteration history, including the objective function, as well as variance component estimates
- for METHOD=ML and METHOD=REML, the estimated Asymptotic Covariance Matrix of the variance components
- a table of variance component estimates

ODS Table Names

PROC VARCOMP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information on ODS, see Chapter 14, “Using the Output Delivery System.”

Table 79.1. ODS Tables Produced in PROC VARCOMP

ODS Table Name	Description	Statement
ANOVA	Type 1 analysis of variance	METHOD=TYPE1
AsyCov	Asymptotic covariance matrix of estimates	METHOD=ML or REML
ClassLevels	Class level information	default
ConvergenceStatus	Convergence status	METHOD=ML or REML
DepVar	Dependent variable	METHOD=TYPE1, REML, or ML
DependentInfo	Dependent variable info (multiple variables)	
Estimates	Variance component estimates	default
IterHistory	Iteration history	METHOD=ML or REML
NObs	Number of observations	default
SSCP	Sum of squares matrix	METHOD=MIVQUE0

In situations where multiple dependent variables are analyzed that differ in their missing value pattern, separate names for ANOVAn, AsyCov$n$, Estimates$n$, IterHistory$n$, and SSCPn tables are no longer required. The results are combined into a single output data set. For METHOD=TYPE1, ML, or REML, variable Dependent in the output data set identifies the dependent variable. For METHOD=MIVQUE0, a variable is added to the output data set for each dependent variable.

Relationship to PROC MIXED

The MIXED procedure effectively performs the same analyses as PROC VARCOMP and many others, including Type I, Type II, and Type III tests of fixed effects, confidence limits, customized contrasts, and least-squares means. Furthermore, continuous variables are permitted as both fixed and random effects in PROC MIXED, and numerous other covariance structures besides variance components are available.

To translate PROC VARCOMP code into PROC MIXED code, move all random effects to the RANDOM statement in PROC MIXED. For example, the syntax for the example in the "Getting Started" section on page 4832 is as follows:

```
proc mixed;
   class Temp Lab Batch;
   model Cure = Temp;
   random Lab Temp*Lab Batch(Lab Temp);
run;
```

REML is the default estimation method in PROC MIXED, and you can specify other methods using the METHOD= option.

Example

Example 79.1. Using the Four Estimation Methods

In this example, a and b are classification variables and y is the dependent variable. a is declared fixed, and b and a*b are random. Note that this design is unbalanced because the cell sizes are not all the same. PROC VARCOMP is invoked four times, once for each of the estimation methods. The data are from Hemmerle and Hartley (1973). The following statements produce Output 79.1.1.

```
data a;
   input a b y @@;
   datalines;
1 1 237   1 1 254   1 1 246   1 2 178   1 2 179
2 1 208   2 1 178   2 1 187   2 2 146   2 2 145   2 2 141
3 1 186   3 1 183   3 2 142   3 2 125   3 2 136
;

proc varcomp method=type1;
   class a b;
   model y=a|b / fixed=1;
run;

proc varcomp method=mivque0;
   class a b;
   model y=a|b / fixed=1;
run;

proc varcomp method=ml;
   class a b;
   model y=a|b / fixed=1;
run;

proc varcomp method=reml;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 79.1.1. VARCOMP Procedure: Method=TYPE1

```
                 Variance Components Estimation Procedure

                         Class Level Information

                      Class         Levels    Values

                      a                  3    1 2 3

                      b                  2    1 2


                  Number of Observations Read          16
                  Number of Observations Used          16


                         Dependent Variable:    y
```

The “Class Level Information” table displays the levels of each variable specified in the CLASS statement. You can check this table to make sure the data are input correctly.

```
                              Variance Components Estimation Procedure

                                    Type 1 Analysis of Variance

                                  Sum of
Source                 DF        Squares   Mean Square  Expected Mean Square

a                       2          11736   5868.218750  Var(Error) + 2.725 Var(a*b) + 0.1 Var(b) + Q(a)
b                       1          11448         11448  Var(Error) + 2.6308 Var(a*b) + 7.8 Var(b)
a*b                     2     299.041026    149.520513  Var(Error) + 2.5846 Var(a*b)
Error                  10     786.333333     78.633333  Var(Error)
Corrected Total        15          24270             .  .
```

The Type I analysis of variance consists of sequentially partitioning the total sum of squares. The mean square is the sum of squares divided by the degrees of freedom, and the expected mean square is the expected value of the mean square under the mixed model. The “Q” notation in the expected mean squares refers to a quadratic form in parameters of the parenthesized effect.

```
                        Variance Components Estimation Procedure

                                  Type 1 Estimates

                           Variance Component        Estimate

                           Var(b)                      1448.4
                           Var(a*b)                  27.42659
                           Var(Error)                78.63333
```

The Type I estimates of the variance components result from solving the linear system of equations established by equating the observed mean squares to their expected values.

Output 79.1.2. VARCOMP Procedure: Method=MIVQUE0

```
                Variance Components Estimation Procedure

                        Class Level Information

                     Class         Levels    Values

                     a                  3    1 2 3

                     b                  2    1 2


                Number of Observations Read          16
                Number of Observations Used          16
```

The “Class Level Information” is the same as before.

```
                     Variance Components Estimation Procedure

                               MIVQUE(0) SSQ Matrix

Source                 b                 a*b               Error                 y

b               60.84000            20.52000             7.80000           89295.4
a*b             20.52000            20.52000             7.80000           30181.3
Error            7.80000             7.80000            13.00000           12533.5
```

The MIVQUE0 sums-of-squares matrix is displayed in the previous table.

```
                     Variance Components Estimation Procedure

                               MIVQUE(0) Estimates

                        Variance Component             y

                        Var(b)                    1466.1
                        Var(a*b)               -35.49170
                        Var(Error)             105.73660
```

The MIVQUE0 estimates result from solving the equations established by the MIVQUE0 SSQ matrix. Note that the estimate of the variance component for the interaction effect, Var(a*b), is negative for this example.

Output 79.1.3. VARCOMP Procedure: Method=ML

```
                     Variance Components Estimation Procedure

                             Class Level Information

                         Class         Levels    Values

                         a                  3    1 2 3

                         b                  2    1 2


                     Number of Observations Read          16
                     Number of Observations Used          16


                             Dependent Variable:    y
```

The “Class Level Information” is the same as before.

```
                    Variance Components Estimation Procedure

                         Maximum Likelihood Iterations

Iteration        Objective            Var(b)          Var(a*b)       Var(Error)

        0    78.3850371200       1031.49070                 0    74.3909717935
        1    78.2637043807   732.3606453636                 0    77.4011688154
        2    78.2635471161   723.6867470850                 0    77.5301774839
        3    78.2635471152   723.6658365289                 0    77.5304926877


       Convergence criteria met.
```

The Newton-Raphson algorithm used by PROC VARCOMP requires three iterations to converge to the maximum likelihood estimates.

```
                    Variance Components Estimation Procedure

                              Maximum Likelihood
                                  Estimates

                            Variance
                            Component       Estimate

                            Var(b)         723.66584
                            Var(a*b)               0
                            Var(Error)      77.53049
```

The ML estimate of Var(a*b) is zero for this example, and the other two estimates are smaller than their Type I and MIVQUE0 counterparts.

```
                    Variance Components Estimation Procedure

                   Asymptotic Covariance Matrix of Estimates

                               Var(b)        Var(a*b)      Var(Error)

          Var(b)            537826.1               0      -107.33905
          Var(a*b)                 0               0               0
          Var(Error)      -107.33905               0       858.71104
```

One benefit of using likelihood-based methods is that an approximate covariance matrix is available from the matrix of second derivatives evaluated at the ML solution. This covariance matrix is valid asymptotically and can be unreliable in small samples.

Here the variance component estimates for B and the Error are negatively correlated and the elements for Var(a*b) are set to zero because the estimate equals zero. Also, the very large variance for Var(b) indicates a lot of uncertainty about the estimate for Var(b), and one contributing explanation is that B has only two levels in this data set.

Output 79.1.4. VARCOMP Procedure: Method=REML

```
                    Variance Components Estimation Procedure

                            Class Level Information

                         Class          Levels    Values

                         a                   3    1 2 3

                         b                   2    1 2


                    Number of Observations Read          16
                    Number of Observations Used          16


                            Dependent Variable:    y
```

The “Class Level Information” is the same as before.

```
                    Variance Components Estimation Procedure

                                REML Iterations

 Iteration        Objective           Var(b)         Var(a*b)       Var(Error)

         0    63.4134144942       1269.52701                0    91.5581191305
         1    63.0446869787       1601.84199    32.7632417174    76.9355562461
         2    63.0311530508       1468.82932    27.2258186561    78.7548276319
         3    63.0311265148       1464.33646    26.9564053003    78.8431476502
         4    63.0311265127       1464.36727    26.9588525177    78.8423898761


        Convergence criteria met.
```

The REML optimization requires four iterations to converge.

```
                    Variance Components Estimation Procedure

                                 REML Estimates

                            Variance
                            Component       Estimate

                            Var(b)            1464.4
                            Var(a*b)        26.95885
                            Var(Error)      78.84239
```

The REML estimates are all larger than the corresponding ML estimates (adjusting for potential downward bias) and are fairly similar to the Type I estimates.

```
                    Variance Components Estimation Procedure

                    Asymptotic Covariance Matrix of Estimates

                                Var(b)         Var(a*b)       Var(Error)

            Var(b)           4401703.8          1.29359       -273.39651
            Var(a*b)           1.29359           3559.1       -502.85157
            Var(Error)      -273.39651       -502.85157           1249.7
```

The Error variance component estimate is negatively correlated with the other two variance component estimates, and the estimated variances are all larger than their ML counterparts.

References

Gaylor, D.W., Lucas, H.L., and Anderson, R.L. (1970), "Calculation of Expected Mean Squares by the Abbreviated Doolittle and Square Root Methods," *Biometrics*, 26, 641–55.

Goodnight, J.H. (1978), *Computing MIVQUE0 Estimates of Variance Components*, SAS Technical Report R-105. Cary, NC: SAS Institute Inc.

Goodnight, J.H. and Hemmerle, W.J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Hartley, H.O., Rao, J.N.K., and LaMotte, L. (1978), "A Simple Synthesis-Based Method of Variance Component Estimation," *Biometrics*, 34, 233–244.

Hemmerle, W.J. and Hartley, H.O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Hicks, C.R. (1973), *Fundamental Concepts in the Design of Experiments*, New York: Holt, Rinehart and Winston, Inc.

Hocking, R.R. (1983), "A Diagnostic Tool for Mixed Models with Applications to Negative Estimates of Variance Components," *Proceedings of the Eighth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 8, 711–716.

Hocking, R.R. (1984), *The Analysis of Linear Models*, Monterey, CA: Brooks-Cole Publishing Co.

Patterson, H.D. and Thompson, R. (1971), "Recovery of Inter-Block Information When Block Sizes Are Unequal," *Biometrika*, 58, 545–554.

Rao, C.R. (1971), "Minimum Variance Quadratic Unbiased Estimation of Variance Components," *Journal of Multivariate Analysis*, 1, 445–456.

Rao, C.R. (1972), “Estimation of Variance and Covariance Components in Linear Models,” *Journal of the American Statistical Association*, 67, 112–115.

Searle, S.R., Casella, G., and McCulloch, C.E. (1992), *Variance Components*, New York: John Wiley and Sons, Inc.

Chapter 80
The VARIOGRAM Procedure

Chapter Contents

Chapter 80
The VARIOGRAM Procedure

Overview

The VARIOGRAM procedure computes sample or empirical measures of spatial continuity for two-dimensional spatial data. These continuity measures are the regular semivariogram, a robust version of the semivariogram, and the covariance. The continuity measures are written to an output data set, allowing plotting or parameter estimation for theoretical semivariogram or covariance models. Both isotropic and anisotropic measures are available.

The VARIOGRAM procedure produces two additional output data sets that are useful in the analysis of pairwise distances in the original data. The OUTPAIR= data set contains one observation for each pair of points. The coordinates, distance, angle, and values of the analysis variables are written to this data set. The OUTDISTANCE= data set contains histogram information on the count of pairs within distance intervals, which is useful for determining unit lag distances.

Introduction to Spatial Prediction

Spatial prediction, in general, is any prediction method that incorporates spatial dependence. A simple and popular spatial prediction method is ordinary kriging.

Ordinary kriging requires a model of the spatial continuity, or dependence. This is typically in the form of a covariance or semivariogram.

Spatial prediction, then, involves two steps. First, you model the covariance or semivariogram of the spatial process. This involves choosing both a mathematical form and the values of the associated parameters. Second, you use this dependence model in solving the kriging system at a specified set of spatial points, resulting in predicted values and associated standard errors.

SAS/STAT software has two procedures corresponding to these steps for spatial prediction of two-dimensional data. The VARIOGRAM procedure is used in the first step. By computing a sample estimate of the variogram or covariance, you can choose a theoretical model based on graphical or other means.

Getting Started

In activities such as reservoir estimation in mining, petroleum exploration, and environmental modeling of air and water pollution, it often happens that data on one or more quantities are available at given spatial locations, and the goal is to predict the measured quantities at unsampled locations. Often, these unsampled locations are on a regular grid, and the predictions are used to produce surface plots or contour maps.

A popular method of spatial prediction is ordinary kriging, which produces both predicted values and associated standard errors. Ordinary kriging requires the complete specification (the form and parameter values) of the spatial dependence of the spatial process in terms of a covariance or semivariogram model.

Typically the semivariogram model is not known in advance and must be estimated, either visually or by some estimation method.

PROC VARIOGRAM computes the sample semivariogram, from which you can find a suitable theoretical semivariogram by visual methods.

The following example goes through a typical problem to show how you can compute a sample variogram and determine an appropriate theoretical model.

Preliminary Spatial Data Analysis

The simulated data consist of coal seam thickness measurements (in feet) taken over an approximately square area. The coordinates are offsets from a point in the southwest corner of the measurement area, with the north and east distances in units of thousands of feet.

First, the data are input.

```
data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1    2.1  82.7  42.2    4.7  75.1  39.5
    4.8  52.8  34.3    5.9  67.1  37.0    6.0  35.7  35.9
    6.4  33.7  36.4    7.0  46.7  34.6    8.2  40.1  35.4
   13.3   0.6  44.7   13.3  68.2  37.8   13.4  31.3  37.8
   17.8   6.9  43.9   20.1  66.3  37.7   22.7  87.6  42.8
   23.0  93.9  43.6   24.3  73.0  39.3   24.8  15.1  42.3
   24.8  26.3  39.7   26.4  58.0  36.9   26.9  65.0  37.8
   27.7  83.3  41.8   27.9  90.8  43.3   29.1  47.9  36.7
   29.5  89.4  43.0   30.1   6.1  43.6   30.8  12.1  42.8
   32.7  40.2  37.5   34.8   8.1  43.3   35.3  32.0  38.8
   37.0  70.3  39.2   38.2  77.9  40.7   38.9  23.3  40.5
   39.4  82.5  41.4   43.0   4.7  43.3   43.7   7.6  43.1
   46.4  84.1  41.5   46.7  10.6  42.6   49.9  22.1  40.7
   51.0  88.8  42.0   52.8  68.9  39.3   52.9  32.7  39.2
   55.5  92.9  42.2   56.0   1.6  42.7   60.6  75.2  40.1
   62.1  26.6  40.1   63.0  12.7  41.8   69.0  75.6  40.1
   70.5  83.7  40.9   70.9  11.0  41.7   71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;
```

It is instructive to see the locations of the measured points in the area where you want to perform spatial prediction. It is desirable to have these locations scattered evenly around the prediction area. If this is not the case, the prediction error might be unacceptably large where measurements are sparse. The following GPLOT procedure is useful in determining potential problems:

```
proc gplot data=thick;
   title ’Scatter Plot of Measurement Locations’;
   plot north*east / frame cframe=ligr haxis=axis1
                     vaxis=axis2;
   symbol1 v=dot color=blue;
   axis1 minor=none;
   axis2 minor=none label=(angle=90 rotate=0);
   label east   = ’East’
         north  = ’North’
   ;
run;
```

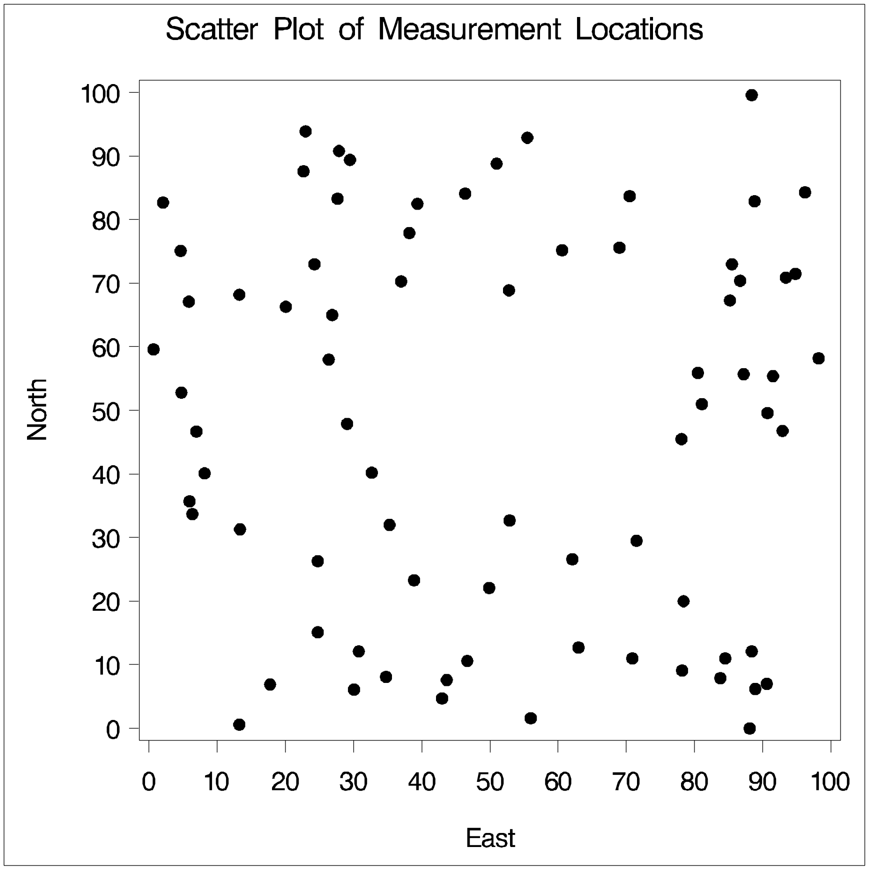


Figure 80.1. Scatter Plot of Measurement Locations

As Figure 80.1 indicates, while the locations are not ideally spread around the prediction area, there are not any large areas lacking measurements. You now can look at a surface plot of the measured variable, the thickness of coal seam, using the G3D procedure. This is a crucial step. Any obvious surface trend has to be removed before you compute and estimate the model of spatial dependence (the semivariogram model).

```
proc g3d data=thick;
   title 'Surface Plot of Coal Seam Thickness';
   scatter east*north=thick / xticknum=5 yticknum=5
      grid zmin=20 zmax=65;
   label east  = 'East'
         north = 'North'
         thick = 'Thickness'
   ;
run;
```

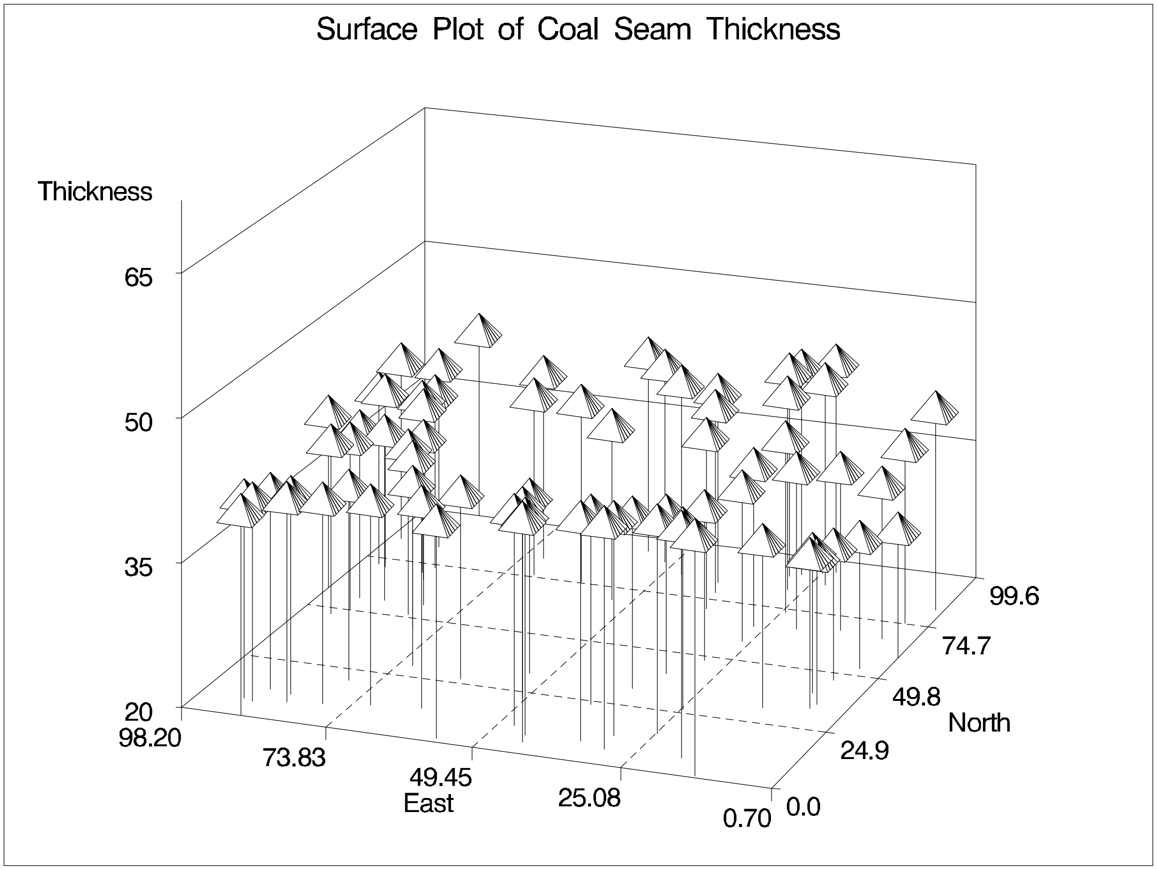


Figure 80.2. Surface Plot of Coal Seam Thickness

Figure 80.2 shows the small-scale variation typical of spatial data, but there does not appear to be any surface trend. Hence, you can work with the original thickness data rather than residuals from a trend surface fit.

Preliminary Variogram Analysis

Recall that the goal of this example is spatial prediction. In particular, you would like to produce a contour map or surface plot on a regular grid of predicted values based on ordinary kriging. Ordinary kriging requires the complete specification of the spatial covariance or semivariogram.

You can use PROC VARIOGRAM, along with a DATA step and PROC GPLOT, to estimate visually a reasonable semivariogram model (both the form and associated parameters) for the thickness data.

Before proceeding with this estimation, consider the formula for the empirical or experimental semivariogram $\gamma_z(h)$. Denote the coal seam thickness process by $\{Z(r), r \in D \subset \mathcal{R}^2\}$. You have measurements $(Z(r_i), i = 1, \ldots, 75)$. The standard formula for $\gamma_z(h)$ (isotropic case) is

$$2\gamma_z(h) = \frac{1}{\mid N(h) \mid} \sum_{N(h)} (Z(r_i) - Z(r_j))^2$$

where $N(h)$ is given by

$$N(h) = \{i, j :\mid r_i - r_j \mid = h\}$$

and $\mid N(h) \mid$ is the number of such pairs (i, j).

For actual data, it is unlikely that any pair (i, j) would exactly satisfy $\mid r_i - r_j \mid = h$, so typically a range of pairwise distances, $\mid r_i - r_j \mid \in [h - \delta h, h + \delta h)$, is used to group pairs (r_i, r_j) for a single term in the expression for $\gamma_z(h)$. Using this range, $N(h)$ is modified by

$$N(h, \delta h) = \{i, j :\mid r_i - r_j \mid \in [h - \delta h, h + \delta h)\}$$

PROC VARIOGRAM performs this grouping with two required options for variogram computation: the LAGDISTANCE= and MAXLAGS= options.

The meaning of the required LAGDISTANCE= option is as follows. Classify all pairs of points into intervals according to their pairwise distance. The width of the distance interval is the LAGDISTANCE= value. The meaning of the required MAXLAGS= option is simply the number of intervals.

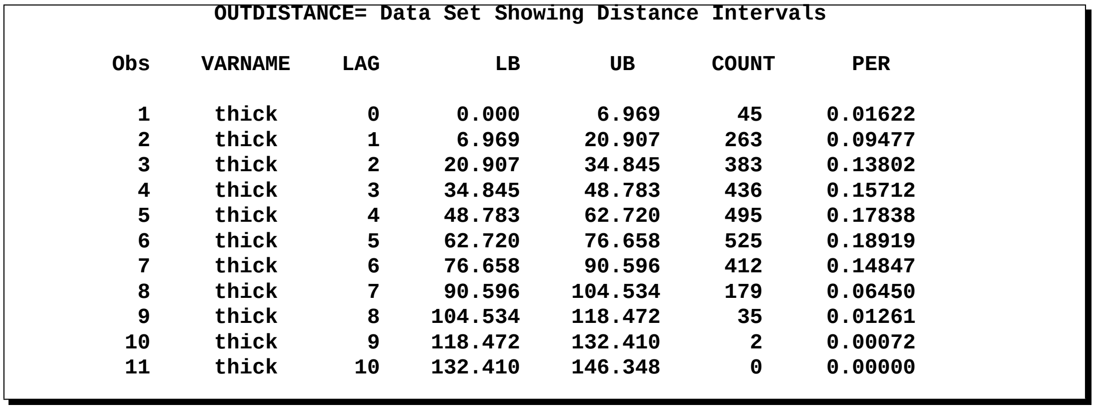
The problem is that a surface plot of the original data, or the scatter plot of the measurement locations, is not very helpful in determining the distribution of these pairwise distances; it is not clear what values to give to the LAGDISTANCE= and MAXLAGS= options.

You use PROC VARIOGRAM with the OUTDISTANCE= option to produce a modified histogram of the pairwise distances in order to find reasonable values for the LAGDISTANCE= and MAXLAGS= options. In the following analysis, you use the

NOVARIOGRAM option in the COMPUTE statement and the OUTDISTANCE= option in the PROC VARIOGRAM statement. You need the NOVARIOGRAM option to keep an error message from being issued due to the absence of the LAGDISTANCE= and MAXLAGS= options.

The DATA step after the PROC VARIOGRAM statement computes the midpoint of each distance interval. This midpoint is then used in the GCHART procedure. Since the number of distance intervals is not specified by using the NHCLASSES= option in the COMPUTE statement, the default of 10 is used.

```
proc variogram data=thick outdistance=outd;
   compute novariogram;
   coordinates xc=east yc=north;
   var thick;
run;

title ’OUTDISTANCE= Data Set Showing Distance Intervals’;
proc print data=outd;
run;

data outd; set outd;
   mdpt=round((lb+ub)/2,.1);
   label mdpt = ’Midpoint of Interval’;
run;

axis1 minor=none;
axis2 minor=none label=(angle=90 rotate=0);
title ’Distribution of Pairwise Distances’;
proc gchart data=outd;
   vbar mdpt / type=sum sumvar=count discrete frame
               cframe=ligr gaxis=axis1 raxis=axis2 nolegend;
run;
```



```
             OUTDISTANCE= Data Set Showing Distance Intervals

      Obs    VARNAME    LAG        LB        UB      COUNT      PER

        1     thick       0     0.000     6.969      45     0.01622
        2     thick       1     6.969    20.907     263     0.09477
        3     thick       2    20.907    34.845     383     0.13802
        4     thick       3    34.845    48.783     436     0.15712
        5     thick       4    48.783    62.720     495     0.17838
        6     thick       5    62.720    76.658     525     0.18919
        7     thick       6    76.658    90.596     412     0.14847
        8     thick       7    90.596   104.534     179     0.06450
        9     thick       8   104.534   118.472      35     0.01261
       10     thick       9   118.472   132.410       2     0.00072
       11     thick      10   132.410   146.348       0     0.00000
```

Figure 80.3. OUTDISTANCE= Data Set Showing Distance Intervals

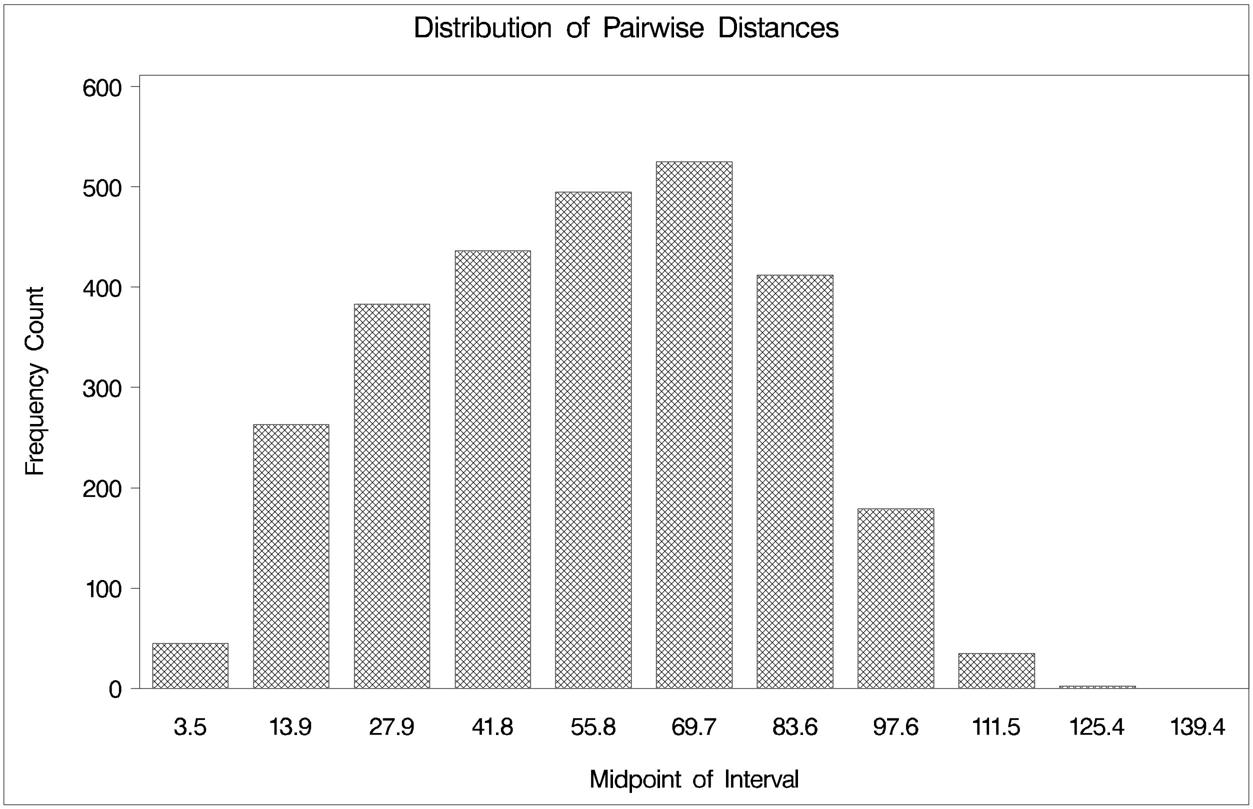


Figure 80.4. Distribution of Pairwise Distances

For plotting and estimations purposes, it is desirable to have as many points as possible for the plot of $\gamma_z(h)$ against h. This corresponds to having as many distance intervals as possible, that is, having a small value for the LAGDISTANCE= option.

However, a rule of thumb used in computing sample semivariograms is to use at least 30 point pairs in computing a single value of the empirical or experimental semivariogram.

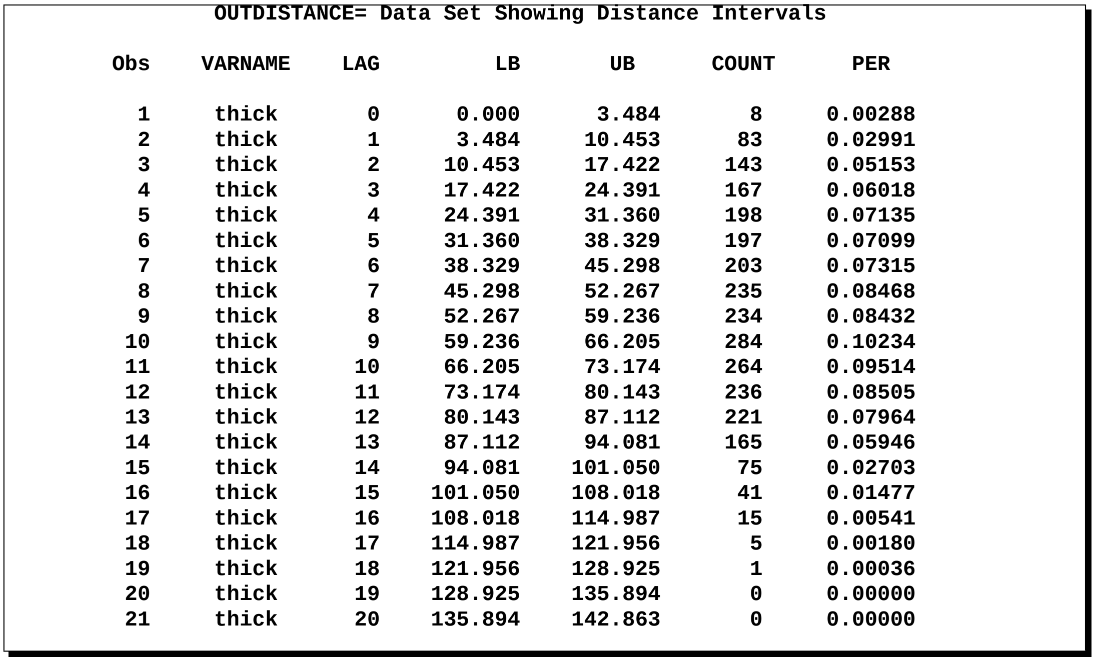
If the LAGDISTANCE= value is set too small, there may be too few points in one or more of the intervals. On the other hand, if the LAGDISTANCE= value is set to a large value, the number of point pairs in the distance intervals may be much greater than that needed for estimation precision, thereby “wasting” point pairs at the expense of variogram points.

Hence, there is a tradeoff between the number of distance intervals and the number of point pairs within each interval.

As discussed in the section “OUTDIST=*SAS-data-set* ” on page 4878 the first few distance intervals, corresponding to lag 0 and lag 1, are typically the limiting intervals. This is particularly true for lag 0 since it is half the width of the remaining intervals. For the default of NHCLASSES=10, the lag 0 class contains 45 points, which is reasonably close to 30, but the lag 1 class contains 263 points.

If you rerun PROC VARIOGRAM with NHCLASSES=20, these numbers become 8 and 83 for lags 0 and 1, respectively. Because of the asymmetrical nature of lag 0, you are willing to violate the rule of thumb for the 0th lag. You will, however, have sufficient numbers in lag 1 and above.

```
proc variogram data=thick outdistance=outd;
   compute nhc=20 novariogram;
   coordinates xc=east yc=north;
   var thick;
run;

title ’OUTDISTANCE= Data Set Showing Distance Intervals’;
proc print data=outd;
run;

data outd; set outd;
   mdpt=round((lb+ub)/2,.1);
   label mdpt = ’Midpoint of Interval’;
run;

axis1 minor=none;
axis2 minor=none label=(angle=90 rotate=0);
title ’Distribution of Pairwise Distances’;
proc gchart data=outd;
   vbar mdpt / type=sum sumvar=count discrete frame
               cframe=ligr gaxis=axis1 raxis=axis2 nolegend;
run;
```



```
             OUTDISTANCE= Data Set Showing Distance Intervals

     Obs    VARNAME    LAG         LB         UB     COUNT       PER

       1     thick       0      0.000      3.484        8     0.00288
       2     thick       1      3.484     10.453       83     0.02991
       3     thick       2     10.453     17.422      143     0.05153
       4     thick       3     17.422     24.391      167     0.06018
       5     thick       4     24.391     31.360      198     0.07135
       6     thick       5     31.360     38.329      197     0.07099
       7     thick       6     38.329     45.298      203     0.07315
       8     thick       7     45.298     52.267      235     0.08468
       9     thick       8     52.267     59.236      234     0.08432
      10     thick       9     59.236     66.205      284     0.10234
      11     thick      10     66.205     73.174      264     0.09514
      12     thick      11     73.174     80.143      236     0.08505
      13     thick      12     80.143     87.112      221     0.07964
      14     thick      13     87.112     94.081      165     0.05946
      15     thick      14     94.081    101.050       75     0.02703
      16     thick      15    101.050    108.018       41     0.01477
      17     thick      16    108.018    114.987       15     0.00541
      18     thick      17    114.987    121.956        5     0.00180
      19     thick      18    121.956    128.925        1     0.00036
      20     thick      19    128.925    135.894        0     0.00000
      21     thick      20    135.894    142.863        0     0.00000
```

Figure 80.5. OUTDISTANCE= Data Set Showing Distance Intervals

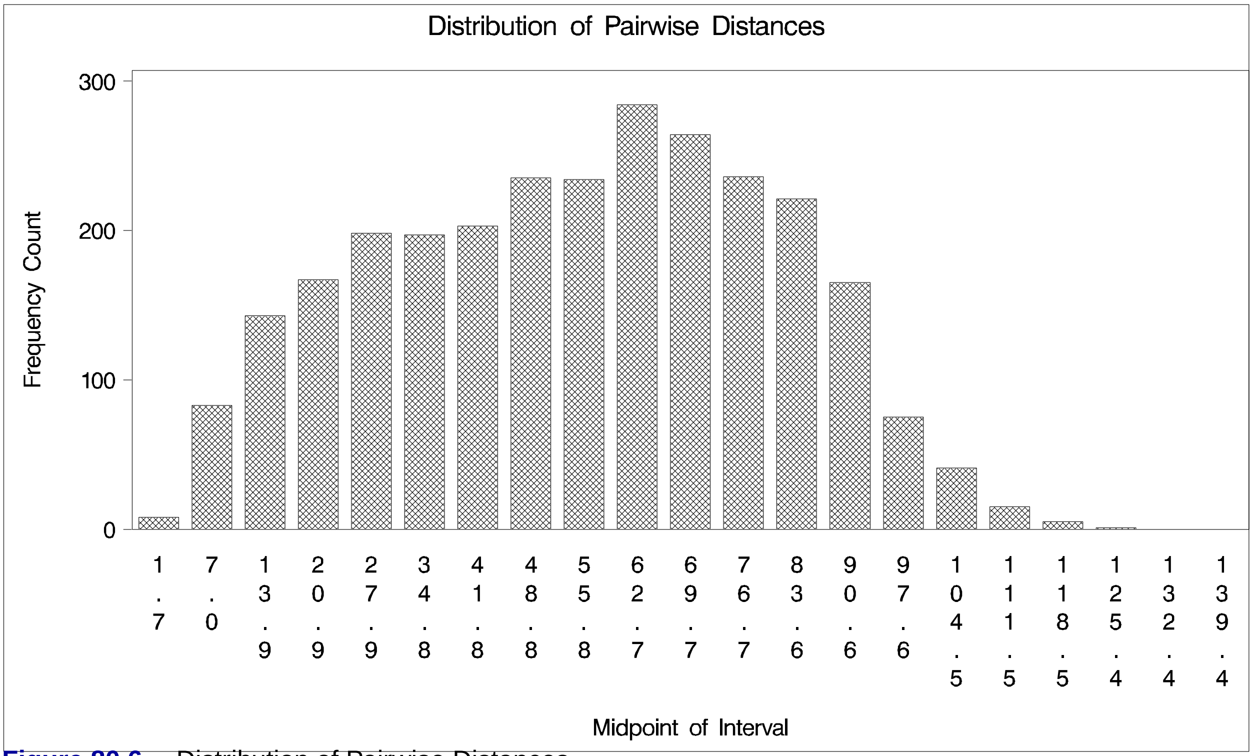


Figure 80.6. Distribution of Pairwise Distances

The length of the lag 1 class $(3.484, 10.453)$ is 6.969. You round off and use LAGDISTANCE=7.0 in the COMPUTE statement.

The use of the MAXLAGS= option is more difficult. From Figure 80.5, note that up to a pairwise distance of 101, you have a sufficient number of pairs. With your choice of LAGDISTANCE=7.0, this yields a maximum number of lags $\frac{101}{7} \approx 14$.

The problem with using the maximum lag value is that it includes pairs of points so far apart that they are likely to be independent. Using pairs of points that are independent adds nothing to the empirical semivariogram plot; they are essentially added noise.

If there is an estimate of correlation length, perhaps from a prior geologic study of a similar site, you can specify the MAXLAGS= value so that the maximum pairwise distance does not exceed two or three correlation lengths. If there is no estimate of correlation length, you can use the following rule of thumb: use $\frac{1}{2}$ to $\frac{3}{4}$ of the "diameter" of the region containing the data. A MAXLAGS= value of 10 is within this range.

You now rerun PROC VARIOGRAM with these values.

Sample Variogram Computation and Plots

Using the values of LAGDISTANCE=7.0 and MAXLAGS=10 computed previously, rerun PROC VARIOGRAM without the NOVARIOGRAM option. Also, request a robust version of the semivariogram; then, plot both results against the pairwise distance of each class.

```
proc variogram data=thick outv=outv;
   compute lagd=7 maxlag=10 robust;
   coordinates xc=east yc=north;
   var thick;
run;

title ’OUTVAR= Data Set Showing Sample Variogram Results’;
proc print data=outv label;
   var lag count distance variog rvario;
run;

data outv2; set outv;
   vari=variog; type = ’regular’; output;
   vari=rvario; type = ’robust’; output;
run;

title ’Standard and Robust Semivariogram for Coal Seam
       Thickness Data’;
proc gplot data=outv2;
   plot vari*distance=type / frame cframe=ligr vaxis=axis2
                             haxis=axis1;
   symbol1 i=join l=1 c=blue   /* v=star   */;
   symbol2 i=join l=1 c=yellow /* v=square */;
   axis1 minor=none
         label=(c=black ’Lag Distance’) /* offset=(3,3) */;
   axis2 order=(0 to 9 by 1) minor=none
         label=(angle=90 rotate=0 c=black ’Variogram’)
         /* offset=(3,3) */;
run;
```

```
                  OUTVAR= Data Set Showing Sample Variogram Results

            Lag Class
            Value (in     Number of    Average Lag
             LAGDIST=      Pairs in     Distance                      Robust
    Obs       units)        Class       for Class     Variogram     Variogram

      1        -1             75            .             .             .
      2         0              8         2.5045       0.02937       0.01694
      3         1             85         7.3625       0.38047       0.19807
      4         2            142        14.1547       1.15158       0.98029
      5         3            169        21.0913       2.79719       3.01412
      6         4            199        27.9691       4.68769       4.86998
      7         5            199        35.1591       6.16018       6.15639
      8         6            205        42.2547       7.58912       8.05072
      9         7            232        48.7775       7.12506       7.07155
     10         8            244        56.1824       7.04832       7.62851
     11         9            285        62.9121       6.66298       8.02993
     12        10            262        69.8925       6.18775       7.92206
```

Figure 80.7. OUTVAR= Data Set Showing Sample Variogram Results

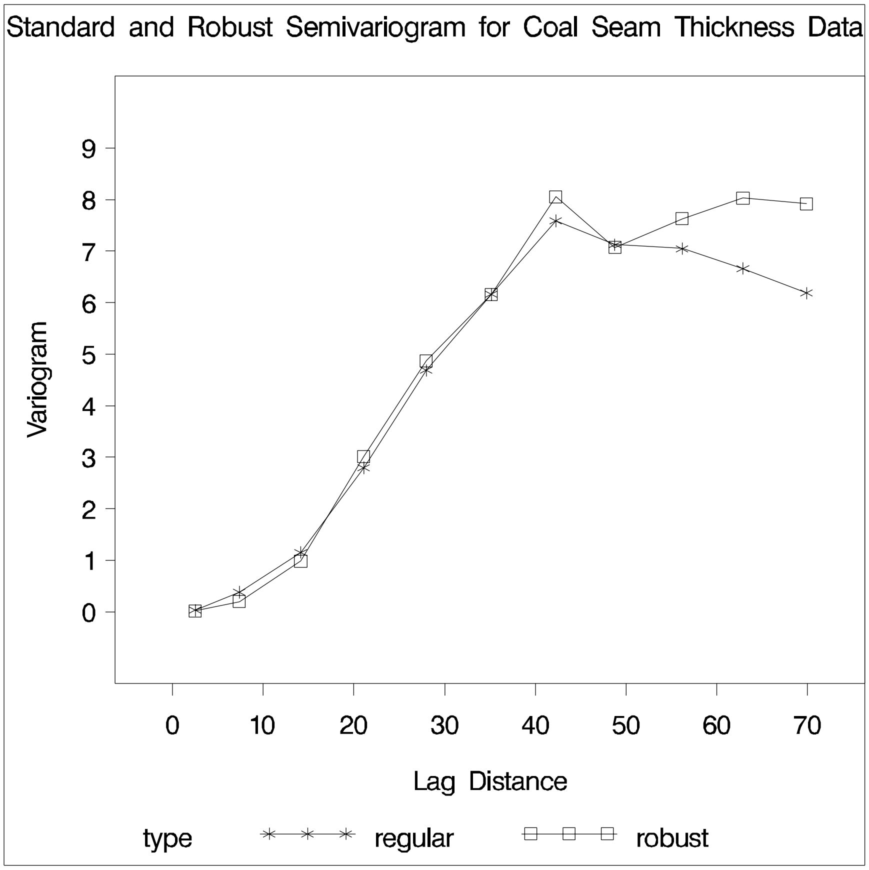


Figure 80.8. Standard and Robust Semivariogram for Coal Seam Thickness Data

Figure 80.8 shows first a slow, then a rapid rise from the origin, suggesting a Gaussian type form:

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h^2}{a_0^2}\right)\right]$$

See the section "Theoretical and Computational Details of the Semivariogram" on page 4872 for graphs of the standard semivariogram forms.

By experimentation, you find that a scale of $c_0 = 7.5$ and a range of $a_0 = 30$ fits reasonably well for both the robust and standard semivariogram

The following statements plot the sample and theoretical variograms:

```
data outv3; set outv;
   c0=7.5; a0=30;
   vari = c0*(1-exp(-distance*distance/(a0*a0)));
   type = 'Gaussian'; output;
   vari = variog; type = 'regular'; output;
   vari = rvario; type = 'robust'; output;
run;

title 'Theoretical and Sample Semivariogram for Coal Seam
       Thickness Data';
proc gplot data=outv3;
   plot vari*distance=type / frame cframe=ligr vaxis=axis2
                             haxis=axis1;
   symbol1 i=join l=1 c=blue    /* v=star    */;
   symbol2 i=join l=1 c=yellow  /* v=square  */;
   symbol3 i=join l=1 c=cyan    /* v=diamond */;
   axis1 minor=none
         label=(c=black 'Lag Distance') /* offset=(3,3) */;
   axis2 order=(0 to 9 by 1) minor=none
         label=(angle=90 rotate=0 c=black 'Variogram')
         /* offset=(3,3) */;
run;
```

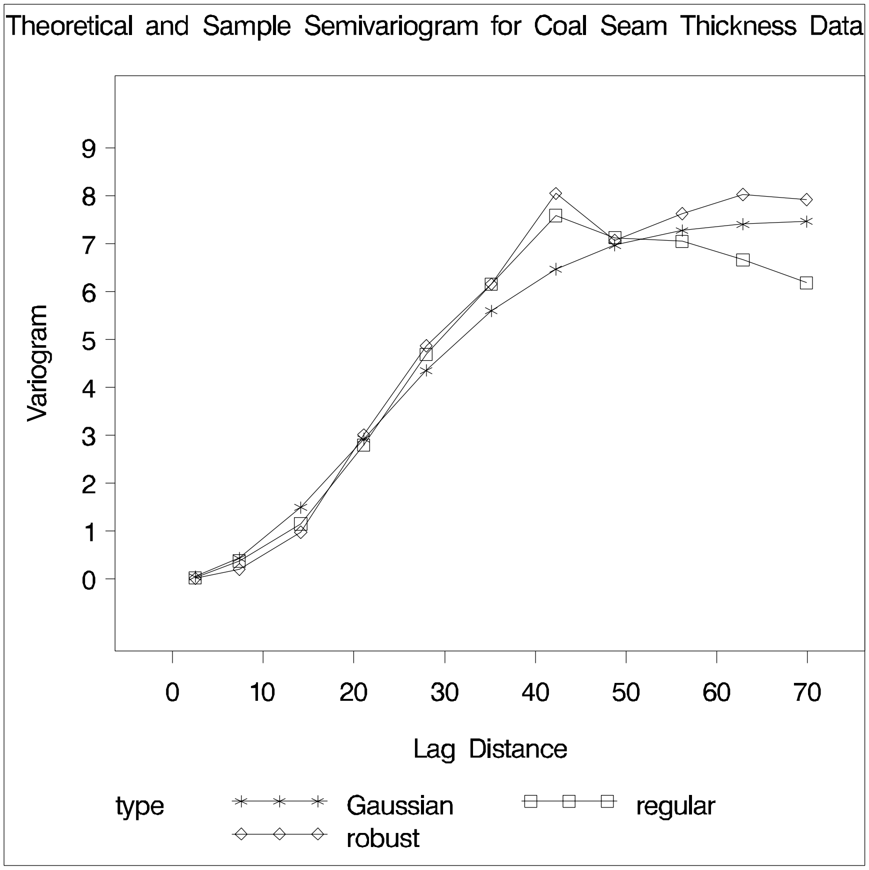


Figure 80.9. Theoretical and Sample Semivariogram for Coal Seam Thickness Data

Figure 80.9 shows that the choice of a semivariogram model is adequate. You can use this Gaussian form and these particular parameters in PROC KRIGE2D to produce a contour plot of the kriging estimates and the associated standard errors.

Syntax

The following statements are available in PROC VARIOGRAM.

PROC VARIOGRAM *options* **;**
    COMPUTE *computation-options* **;**
    COORDINATES *coordinate-variables* **;**
    DIRECTIONS *directions-list* **;**
    VAR *analysis-variables-list* **;**

The COMPUTE and COORDINATES statements are required.

The following table outlines the options available in PROC VARIOGRAM classified by function.

Table 80.1. Options Available in the VARIOGRAM Procedure

Task	Statement	Option
Data Set Options		
specify input data set	PROC VARIOGRAM	DATA=
write spatial continuity measures	PROC VARIOGRAM	OUTVAR=
write distance histogram information	PROC VARIOGRAM	OUTDISTANCE=
write pairwise point information	PROC VARIOGRAM	OUTPAIR=
Declaring the Role of Variables		
specify the analysis variables	VAR	
specify the x, y coordinates in the DATA= data set	COORDINATES	XCOORD= YCOORD=
Controlling Continuity Measure Computations		
specify the basic lag distance	COMPUTE	LAGDISTANCE=
specify the tolerance around the lag distance	COMPUTE	LAGDISTANCE=
specify the maximum number of lags in computations	COMPUTE	MAXLAGS=
specify the number of angle classes	COMPUTE	NDIRECTIONS=
specify the angle tolerances for angle classes	COMPUTE	ANGLETOL=
specify the bandwidths for angle classes	COMPUTE	BANDWIDTH=
compute robust semivariogram	COMPUTE	ROBUST
suppress computation of all continuity measures	COMPUTE	NOVARIOGRAM
Controlling Distance Histogram Data Set		
specify the distance histogram data set	PROC VARIOGRAM	OUTDISTANCE=
specify the number of histogram classes	COMPUTE	NHCLASSES=
Controlling Pairwise Information Data Set		
specify the pairwise data set	PROC VARIOGRAM	OUTPAIR=
specify the maximum distance for the pairwise data set	COMPUTE	OUTPDISTANCE=

PROC VARIOGRAM Statement

PROC VARIOGRAM *options* **;**

You can specify the following options in the PROC VARIOGRAM statement.

DATA=*SAS-data-set*

specifies a SAS data set containing the x and y coordinate variables and the VAR statement variables.

OUTDISTANCE=*SAS-data-set*
OUTDIST=*SAS-data-set*
OUTD=*SAS-data-set*

specifies a SAS data set in which to store summary distance information. This data set contains a count of all pairs of data points within a given distance interval. The number of distance intervals is controlled by the NHCLASSES= option in the COMPUTE statement. The OUTDISTANCE= data set is useful for plotting modified histograms of the count data for determining appropriate lag distances. See the section "OUTDIST=*SAS-data-set* " on page 4878 for details.

OUTPAIR=*SAS-data-set*
OUTP=*SAS-data-set*

specifies a SAS data set in which to store distance and angle information for each pair of points in the DATA= data set. This option should be used with caution when the DATA= data set is large. If n denotes the number of observations in the DATA= data set, the OUTPAIR= data set contains $\frac{n(n-1)}{2}$ observations unless you restrict it with the OUTPDISTANCE= option in the COMPUTE statement. The OUTPDISTANCE= option in the COMPUTE statement excludes pairs of points when the distance between the pairs exceeds the OUTPDISTANCE= value. See the section "OUTPAIR=*SAS-data-set*" on page 4881 for details.

OUTVAR=*SAS-data-set*
OUTVR=*SAS-data-set*

specifies a SAS data set in which to store the continuity measures. See the section "OUTVAR=*SAS-data-set* " on page 4877 for details.

COMPUTE Statement

COMPUTE *computation-options* **;**

The COMPUTE statement provides a number of options that control the computation of the semivariogram, the robust semivariogram, and the covariance.

ANGLETOLERANCE=*angle tolerance*
ANGLETOL=*angle tolerance*
ATOL=*angle tolerance*

specifies the tolerance, in degrees, around the angles determined by the NDIRECTIONS= specification. The default is $\frac{180^o}{2 \times n_d}$, where n_d is the NDIRECTIONS= specification.

See the section "Theoretical and Computational Details of the Semivariogram" on page 4872 for more detailed information.

BANDWIDTH=*bandwidth distance*
BANDW=*bandwidth distance*

specifies the bandwidth, or perpendicular distance cutoff for determining the angle class for a given pair of points. The distance classes define a series of cylindrically shaped areas, while the angle classes radially cut these cylindrically shaped areas. For a given angle class $(\theta_1 - \delta\theta_1, \theta_1 + \delta\theta_1)$, as you proceed out radially, the area encompassed by this angle class becomes larger. The BANDWIDTH= option restricts this area by excluding all points with a perpendicular distance from the line $\theta = \theta_1$ that is greater than the BANDWIDTH= value.

If you do not specify the BANDWIDTH= option, no restriction occurs. See Figure 80.15 on page 4876 for more detailed information.

DEPSILON=*distance value*
DEPS=*distance value*

specifies the distance value for declaring that two distinct points are zero distance apart. Such pairs, if they occur, cause numeric problems. If you specify DEPSILON=ε, then pairs of points P_1 and P_2 for which the distance between them $\mid P_1P_2 \mid < \varepsilon$ are excluded from the continuity measure calculations. The default value of the DEPSILON= option is 100 times machine epsilon; this product is approximately 1E-10 on most computers.

LAGDISTANCE=*distance unit*
LAGDIST=*distance unit*
LAGD=*distance unit*

specifies the basic distance unit defining the lags. For example, a specification of LAGDISTANCE=x results in lag distance classes that are multiples of x. For a given pair of points P_1 and P_2, the distance between them, denoted $\mid P_1P_2 \mid$, is calculated. If $\mid P_1P_2 \mid = x$, then this pair is in the first lag class. If $\mid P_1P_2 \mid = 2x$, then this pair is in the second lag class, and so on.

For irregularly spaced data, the pairwise distances are unlikely to fall exactly on multiples of the LAGDISTANCE= value. A distance tolerance of δx is used to accommodate a spread of distances around multiples of x (the LAGTOLERANCE= option specifies the distance tolerance). For example, if $\mid P_1P_2 \mid$ is within $x \pm \delta x$, you would place this pair in the first lag class; if $\mid P_1P_2 \mid$ is within $2x \pm \delta x$, you would place this pair in the second lag class, and so on.

You can determine the candidate values for the LAGDISTANCE= option by plotting or displaying the OUTDISTANCE= data set.

A LAGDISTANCE= value is required unless you specify the NOVARIOGRAM option.

See the section "Theoretical and Computational Details of the Semivariogram" on page 4872 for more details.

LAGTOLERANCE=*tolerance number*
LAGTOL=*tolerance number*
LAGT=*tolerance number*

specifies the tolerance around the LAGDISTANCE= value for grouping distance pairs into lag classes. See the preceding description of the LAGDISTANCE= option for information on the use of the LAGTOLERANCE= option, and see the section "Theoretical and Computational Details of the Semivariogram" on page 4872 for more details.

If you do not specify the LAGTOLERANCE= option, a default value of $(1/2)$ times the LAGDISTANCE= value is used.

MAXLAGS=*number of lags*
MAXLAG=*number of lags*
MAXL=*number of lags*

specifies the maximum number of lag classes used in constructing the continuity measures. This option excludes any pair of points P_1 and P_2 for which the distance between them, $\mid P_1P_2 \mid$, exceeds the MAXLAGS= value times the LAGDISTANCE= value.

You can determine candidate values for the MAXLAGS= option by plotting or displaying the OUTDISTANCE= data set.

A MAXLAGS= value is required unless you specify the NOVARIOGRAM option.

NDIRECTIONS=*number of directions*
NDIR=*number of directions*
ND=*number of directions*

specifies the number of angle classes to use in computing the continuity measures. This option is useful when there is potential anisotropy in the spatial continuity measures. Anisotropy occurs when the spatial continuity or dependence between a pair of points depends on the orientation or angle between the pair. Isotropy is the absence of this effect: the spatial continuity or dependence between a pair of points depends only on the distance between the points, not the angle.

The angle classes formed from the NDIRECTIONS= option start from N–S and proceed clockwise. For example, NDIRECTIONS=3 produces three angle classes. In terms of compass points, these classes are centered at 0^o (or its reciprocal 180^o), 60^o (or its reciprocal 240^o), and 120^o (or its reciprocal 300^o). For irregularly spaced data, the angles between pairs are unlikely to fall exactly in these directions, so an angle tolerance of $\delta\theta$ is used (the ANGLETOLERANCE= option specifies the angle tolerance). If NDIRECTIONS=n_d, the base angle is $\theta = \frac{180^o}{n_d}$, and the angle classes are

$$(k\theta - \delta\theta, k\theta + \delta\theta) \qquad k = 0, \ldots, n_d - 1$$

If you do not specify the NDIRECTIONS= option, no angles are formed, and the spatial continuity measures are assumed to be isotropic.

The NDIRECTIONS= option is useful for exploring possible anisotropy. The DIRECTIONS statement, described in the "DIRECTIONS Statement" section on page 4870, provides greater control over the angle classes. See the section "Theoretical and Computational Details of the Semivariogram" on page 4872 for more detailed information.

NHCLASSES=*number of histogram classes*
NHCLASS=*number of histogram classes*
NHC=*number of histogram classes*

specifies the number of distance or histogram classes to write to the OUTDISTANCE= data set. The actual number of classes is one more than the NHCLASSES= value since a special lag 0 class is also computed. See the OUTDISTANCE= option on page 4865 and the section "OUTDIST=*SAS-data-set* " on page 4878 for details.

The default value of the NHCLASSES= option is 10. This option is ignored if you do not specify an OUTDISTANCE= data set.

NOVARIOGRAM

prevents the computation of the continuity measures. This option is useful for preliminary analysis when you require only the OUTDISTANCE= or OUTPAIR= data sets.

OUTPDISTANCE=*distance limit*
OUTPDIST=*distance limit*
OUTPD=*distance limit*

specifies the cutoff distance for writing observations to the OUTPAIR= data set. If you specify OUTPDISTANCE=d_{max}, the distance $\mid P_1P_2 \mid$ between each pair of points P_1 and P_2 is checked against d_{max}. If $\mid P_1P_2 \mid > d_{max}$, the observation for this pair is not written to the OUTPAIR= data set. If you do not specify the OUTPDISTANCE= option, all distinct pairs are written. This option is ignored if you do not specify an OUTPAIR= data set.

ROBUST

requests that a robust version of the semivariogram be calculated in addition to the regular semivariogram and covariance.

COORDINATES Statement

COORDINATES *coordinate-variables* **;**

The following two options give the names of the variables in the DATA= data set containing the values of the x and y coordinates of the data.

Only one COORDINATES statement is allowed, and it is applied to all the analysis variables. In other words, it is assumed that all the VAR variables have the same x and y coordinates.

XCOORD= *(variable-name)*
XC= *(variable-name)*

gives the name of the variable containing the x coordinate of the data in the DATA= data set.

YCOORD= *(variable-name)*
YC= *(variable-name)*

gives the name of the variable containing the y coordinate of the data in the DATA= data set.

DIRECTIONS Statement

DIRECTIONS *directions-list* **;**

You use the DIRECTIONS statement to define angle classes. You can specify angle classes as a list of angles, separated by commas, with optional angle tolerances and bandwidths within parentheses following the angle. You must specify at least one angle.

If you do not specify the optional angle tolerance, the default value of 45^o is used. If you do not specify the optional bandwidth, no bandwidth is checked. If you specify a bandwidth, you must also specify an angle tolerance.

For example, suppose you want to compute three separate semivariograms at angles $\theta_1 = 0^o, \theta_2 = 60^o$, and $\theta_3 = 120^o$, with corresponding angle tolerance $\delta\theta_1 = 22.5^o, \delta\theta_2 = 12.5^o$, and $\delta\theta_3 = 22.5^o$, with bandwidths 50 and 40 distance units on the first two angle classes and no bandwidth check on the last angle class.

The appropriate DIRECTIONS statement is

```
directions 0.0(22.5,50), 60.0(12.5,40),120(22.5);
```

VAR Statement

VAR *analysis-variables-list* **;**

Use the VAR statement to specify the analysis variables. You can specify only numeric variables. If you do not specify a VAR statement, all numeric variables in the DATA= data set that are not in the COORDINATES statement are used.

Details

Theoretical Semivariogram Models

The VARIOGRAM procedure computes the sample, or experimental semivariogram. Prediction of the spatial process at unsampled locations by techniques such as ordinary kriging requires a theoretical semivariogram or covariance.

It is necessary, then, to decide on a theoretical variogram based on the sample variogram. While there are methods of fitting semivariogram models, such as least squares, maximum likelihood, and robust methods (Cressie 1993, section 2.6), these techniques are not appropriate for data sets resulting in a small number of variogram points. Instead, a visual fit of the variogram points to a few standard models is often satisfactory. Even when there are sufficient variogram points, a visual check against a fitted theoretical model is appropriate (Hohn 1988, p. 25ff).

In some cases, a plot of the experimental semivariogram suggests that a single theoretical model is inadequate. Nested models, anisotropic models, and the nugget effect increase the scope of theoretical models available. All of these concepts are discussed in this section. The specification of the final theoretical model is provided by the syntax of PROC KRIGE2D.

Note the general flow of investigation. After a suitable choice is made of the LAGDIST= and MAXLAGS= options and, possibly, the NDIR= option (or a DIRECTIONS statement), the experimental semivariogram is computed. Potential theoretical models, possibly incorporating nesting, anisotropy, and the nugget effect, are computed by a DATA step, then they are plotted against the experimental semivariogram and evaluated. A suitable theoretical model is thus found visually, and the specification of the model is used in PROC KRIGE2D. This flow is illustrated in Figure 80.10; also see the "Getting Started" section on page 4852 for an illustration in a simple case.

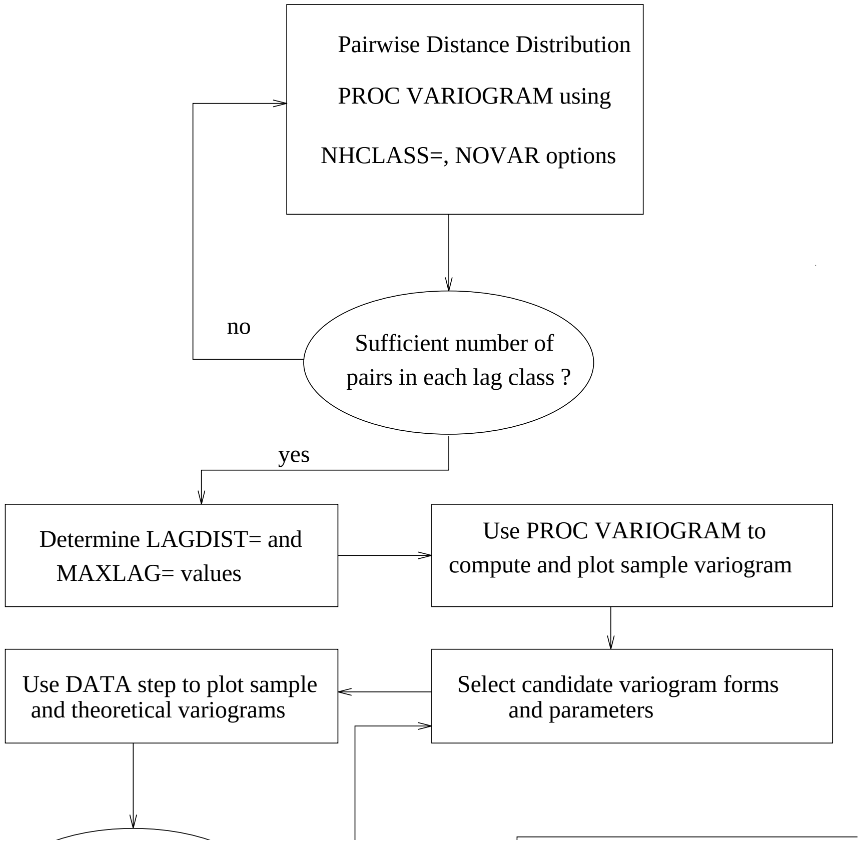

Pairwise Distance Distribution
PROC VARIOGRAM using
NHCLASS=, NOVAR options
no
Sufficient number of
pairs in each lag class ?
yes
Determine LAGDIST= and
MAXLAG= values
Use PROC VARIOGRAM to
compute and plot sample variogram
Use DATA step to plot sample
and theoretical variograms
Select candidate variogram forms
and parameters

Figure 80.10. Flowchart for Variogram Selection

Theoretical and Computational Details of the Semivariogram

The basic starting point in computing the semivariogram is the enumeration of pairs of points for the spatial data. Figure 80.11 shows a spatial domain in which a set of measurements are made at the indicated locations. Two points P_1 and P_2, with coordinates $(x_1, y_1), (x_2, y_2)$, are selected for illustration. A vector, or directed line segment, is drawn between these points. This pair is then categorized first by orientation of this directed line segment and then by its length. That is, the pair P_1P_2 is placed into an angle and distance class.

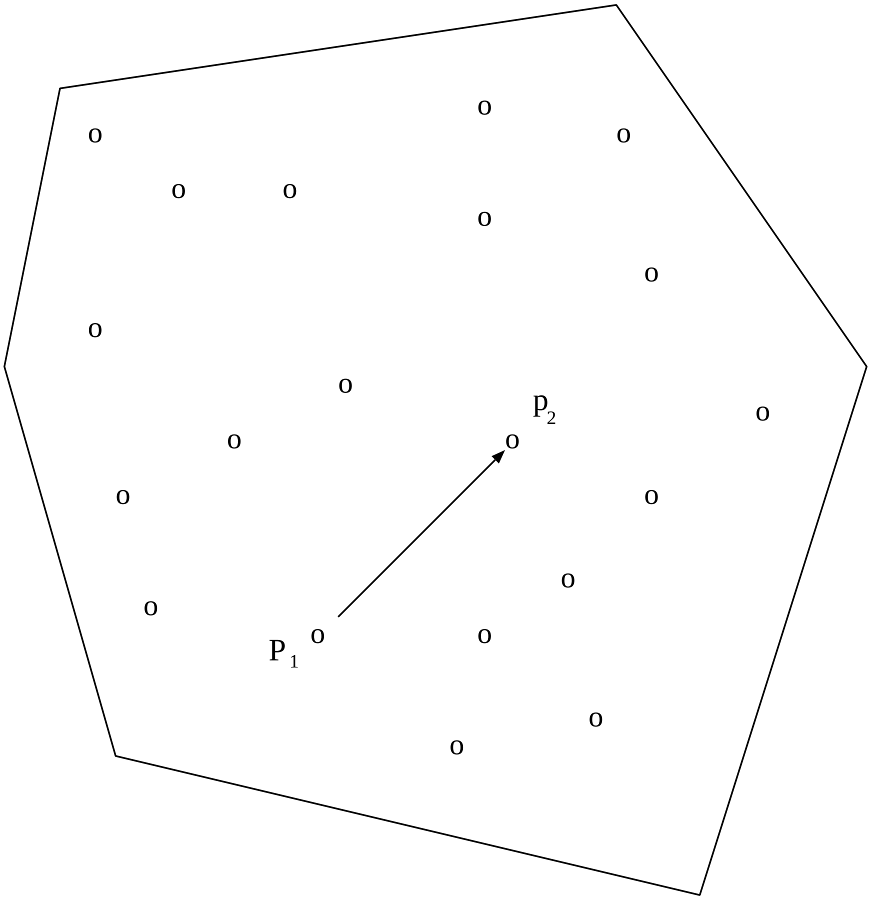


Figure 80.11. Selection of Points P_1 and P_2 in Spatial Domain

Angle Classification

Suppose you specify NDIR=3 in the COMPUTE statement in PROC VARIOGRAM. This results in three angle classes defined by midpoint angles between 0^o and 180^o: $0^o \pm \delta\theta$, $60^o \pm \delta\theta$, and $120^o \pm \delta\theta$, where $\delta\theta$ is the angle tolerance. If you do not specify an angle tolerance using the ATOL= option in the COMPUTE statement, the following default value is used.

$$\delta\theta = \frac{180^o}{2 \times NDIR}$$

For three classes, $\delta\theta = 30^o$. When the example directed line segment P_1P_2 is superimposed on the coordinate system showing the angle classes, its angle, measured clockwise from north, is approximately 45^o. In particular, it falls within $[60^o - \delta\theta, 60^o + \delta\theta) = [30^o, 90^o)$, the second angle class. See Figure 80.12.

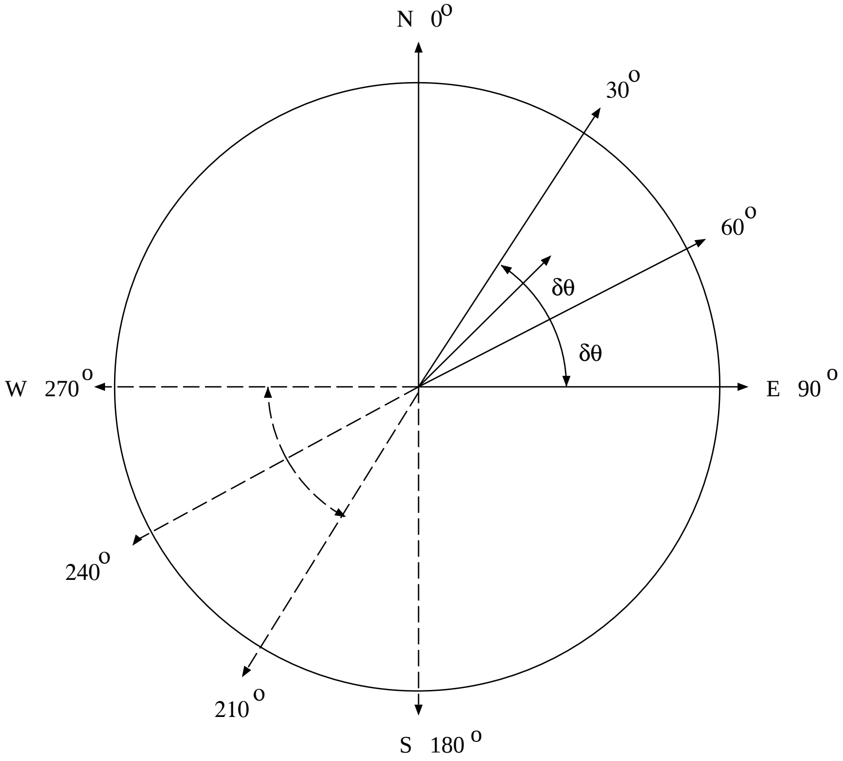


Figure 80.12. Selected Pair P_1P_2 Falls within the Second Angle Class

Note that if the designated points P_1 and P_2 are labeled in the opposite order, the orientation is in a reciprocal direction, that is, approximately 225^o for the point pair instead of approximately 45^o. This does not affect angle class selection; the angle classes $[60^o - \delta\theta, 60^o + \delta\theta)$ and $[240^o - \delta\theta, 240^o + \delta\theta)$ are the same.

If you specify an angle tolerance less than the default, for example, $ATOL = 15^o$, some point pairs might be excluded. For example, the selected point pair P_1P_2 in Figure 80.12, while closest to the 60^o axis, might lie outside $[60 - \delta\theta, 60 + \delta\theta) = [45^o, 75^o)$. In this case, the point pair P_1P_2 would be excluded from the variogram computation.

On the other hand, you can specify an angle tolerance *greater* than the default. This can result in a point pair being counted in more than one angle class. This has a smoothing effect on the variogram and is useful when there is a small amount of data available.

An alternative way to specify angle classes and angle tolerances is with the DIRECTIONS statement. The DIRECTIONS statement is useful when angle classes are not equally spaced. When you specify the DIRECTIONS statement, you should also specify the angle tolerance. The default value of the angle tolerance is 45^o

when a DIRECTIONS statement is used instead of the NDIRECTIONS= option in the COMPUTE statement. This may not be appropriate for a particular set of angle classes. See the "DIRECTIONS Statement" section on page 4870 for more details on the DIRECTIONS statement.

Distance Classification

Next, the distance class for the point pair P_1P_2 is determined. The directed line segment P_1P_2 is superimposed on the coordinate system showing the distance or lag classes. These classes are determined by the LAGD= specification in the COMPUTE statement. Denoting the length of the line segment by $\mid P_1P_2 \mid$ and the LAGD value by Δ, the lag class L is determined by

$$L(P_1P_2) = \left\lfloor \frac{\mid P_1P_2 \mid +.5}{\Delta} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the largest integer $\leq x$.

When the directed line segment P_1P_2 is superimposed on the coordinate system showing the distance classes, it is seen to fall in the first lag class; see Figure 80.13 for an illustration for $\Delta = 1$.

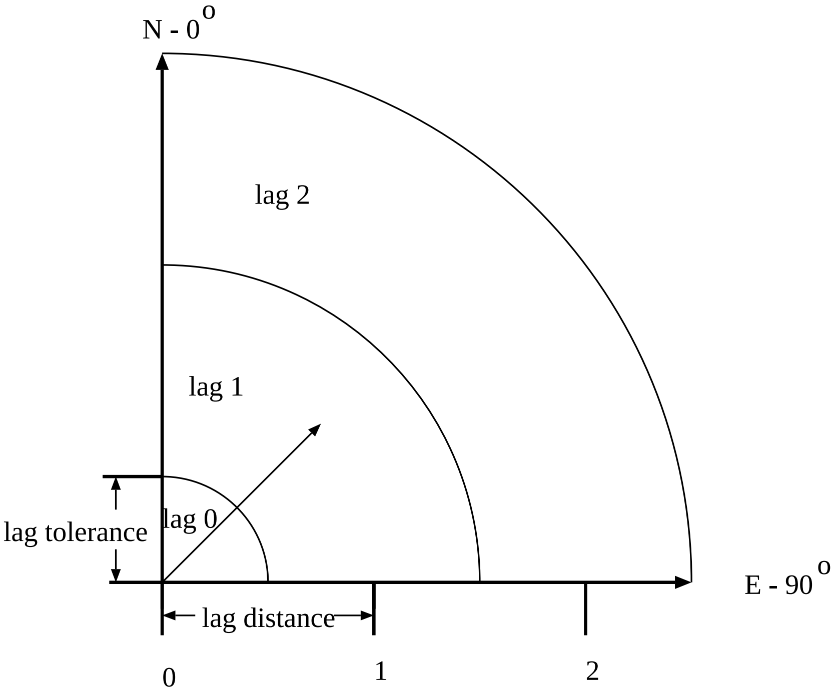


Figure 80.13. Selected Pair P_1P_2 Falls within the First Lag Class

Because pairwise distances are positive, lag class zero is smaller than lag classes $1, \cdots, MAXLAG - 1$. For example, if you specify LAGD=1.0 and MAXLAG=10, and you do not specify a LAGTOL= value in the COMPUTE statement in PROC VARIOGRAM, the ten lag classes generated by the preceding equation are

$$[0,.5),[.5,1.5),[1.5,2.5),\cdots,[8.5,9.5)$$

This is because the default lag tolerance is one-half the LAGD= value, resulting in no gaps between the distance class intervals. This is shown in Figure 80.14.

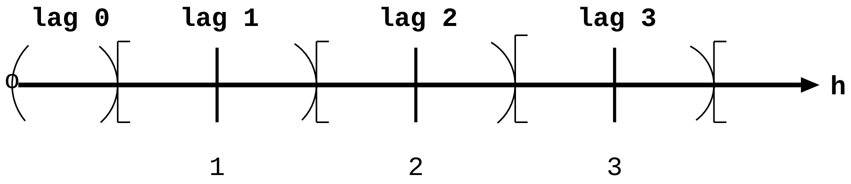


Figure 80.14. Lag Distance Axis Showing Lag Classes

On the other hand, if you do specify a distance tolerance with the DTOL= option in the COMPUTE statement, a further check is performed to see if the point pair falls within this tolerance of the nearest lag. In the preceding example, if you specify LAGD=1.0 and MAXLAG=10 (as before) and also specify LAGTOL=0.25, the intervals become

$$[0,0.25),[0.75,1.25),[1.75,2.25),\cdots,[8.75,9.25)$$

Note that this specification results in gaps in the lag classes; a point pair P_1P_2 might fall, for example, in the interval

$$\mid P_1P_2 \mid \in [1.25,1.75)$$

and hence be excluded from the semivariogram calculation. The maximum LAGTOL= value allowed is half the LAGD= value; no overlap of the distance classes is allowed.

Bandwidth Restriction

Because the areal segments generated from the angle and distance classes increase in area as the lag distance increases, it is sometimes desirable to restrict this area (Duetsch and Journel 1992, p. 45). If you specify the BANDW= option in the COMPUTE statement, the lateral, or perpendicular, distance from the axis defining the angle classes is fixed.

For example, suppose two points P_3, P_4 are picked from the domain in Figure 80.11 and are superimposed on the grid defining distance and angle classes, as shown in Figure 80.15.

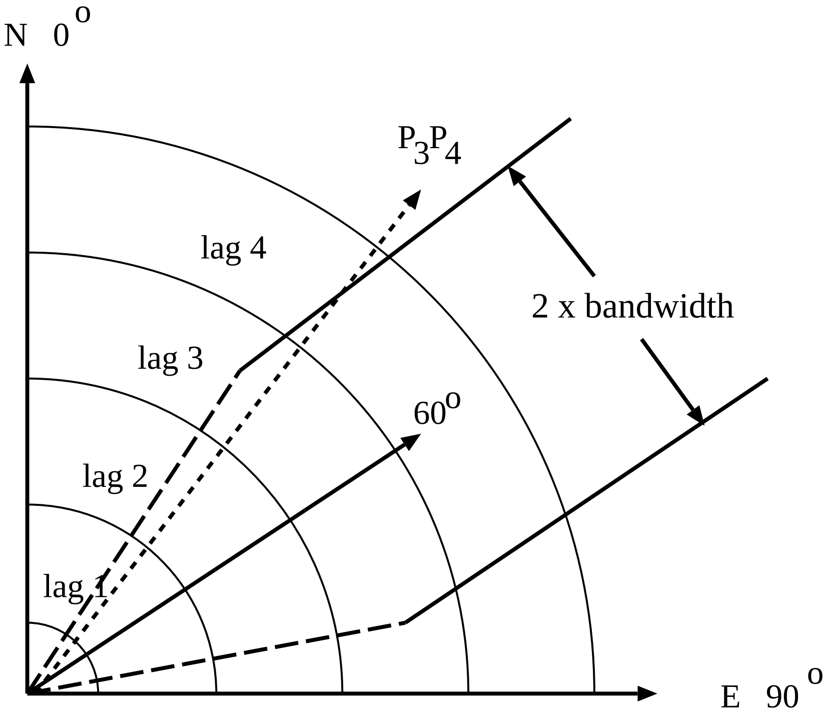


Figure 80.15. Selected Pair P_3P_4 Falls Outside Bandwidth Limit

The endpoint of vector P_3P_4 falls within the angle class around 60^o and the 5th lag class; however, it falls outside the restricted area defined by the bandwidth. Hence, it is excluded from the semivariogram calculation.

Finally, a pair P_iP_j that falls in a lag class larger than the value of the MAXLAG= option is excluded from the semivariogram calculation.

From this description, it is clear that the number of pairs within each angle/distance class is strongly affected by the angle and lag tolerances. Since it is desirable to have the maximum number of point pairs within each class, the angle tolerance and the distance tolerance should usually be the default values.

Semivariogram Computation

With the classification of a point pair P_iP_j into an angle/distance class, as shown in the preceding section, the semivariogram computation proceeds as follows.

Denote all pairs P_iP_j belonging to angle class $[\theta_k - \delta\theta_k, \theta_k + \delta\theta_k)$ and distance class $L = L(P_iP_j)$ by $N(\theta_k, L)$. For example, in the preceding illustration, P_1P_2 belongs to $N(60^o, 1)$.

Let $\mid N(\theta_k, L) \mid$ denote the *number* of such pairs. Let V_i, V_j be the measured values at points P_i, P_j. The component of the standard (or method of moments) semivariogram corresponding to angle/distance class $N(\theta_k, L)$ is given by

$$2\gamma(h_k) = \frac{1}{\mid N(\theta_k, L) \mid} \sum_{P_i P_j \in N(\theta_k, L)} (V_i - V_j)^2$$

where h_k is the average distance in class $N(\theta_k, L)$; that is,

$$h_k = \frac{1}{\mid N(\theta_k, L) \mid} \sum_{P_i P_j \in N(\theta_k, L)} \mid P_i P_j \mid$$

The robust version of the semivariogram, as suggested by Cressie (1993), is given by

$$2\bar{\gamma}(h_k) = \frac{\Psi^4(h_k)}{0.457 + 0.494/N(\theta_k, L)}$$

where

$$\Psi(h_k) = \frac{1}{N(\theta_k, L)} \sum_{P_i P_j \in N(\theta_k, L)} (V_i - V_j)^{\frac{1}{2}}$$

This robust version of the semivariogram is computed when you specify the ROBUST option in the COMPUTE statement in PROC VARIOGRAM.

PROC VARIOGRAM computes and writes to the OUTVAR= data set the quantities $h_k, \theta_k, L, N(\theta_k, L), \gamma(h)$, and $\bar{\gamma}(h)$.

Output Data Sets

The VARIOGRAM procedure produces three data sets: the OUTVAR=*SAS-data-set*, the OUTPAIR=*SAS-data-set*, and the OUTDIST=*SAS-data-set*. These data sets are described in the following sections.

OUTVAR=*SAS-data-set*

The OUTVAR= data set contains the standard and robust versions of the sample semivariogram, the covariance, and other information at each lag class.

The details of the computation of the variogram, the robust variogram, and the covariance is described in the section "Theoretical and Computational Details of the Semivariogram" on page 4872.

The OUTVAR= data set contains the following variables:

- ANGLE, which is the angle class value (clockwise from N–S)
- ATOL, which is the angle tolerance for the lag/angle class
- AVERAGE, which is the average variable value for the lag/angle class

- BANDW, which is the band width for the lag/angle class
- COUNT, which is the number of pairs in the lag/angle class
- COVAR, which is the covariance value for the lag/angle class
- DISTANCE, which is the average lag distance for the lag/angle class
- LAG, which is lag class value (in LAGDISTANCE= units)
- RVARIO, which is the sample robust variogram value for the lag/angle class
- VARIOG, which is the sample variogram value for the lag/angle class
- VARNAME, which is the name of the current VAR= variable

The bandwidth variable, BANDW, is not included in the data set if no bandwidth specification is given in the COMPUTE statement or in a DIRECTIONS statement.

OUTDIST=*SAS-data-set*

The OUTDIST= data set contains counts for a modified histogram showing the distribution of pairwise distances. The purpose of this data set is to enable you to make choices for the value of the LAGDISTANCE= option in the COMPUTE statement in subsequent runs of PROC VARIOGRAM.

For plotting and estimation purposes, it is desirable to have as many points as possible for a variogram plot. However, a rule of thumb used in computing sample semivariograms is to use at least 30 points in each interval whenever possible. Hence, there is a lower limit to the value of the LAGDISTANCE= option.

Since the distribution of pairwise distances is seldom known in advance, the information contained in the OUTDIST= data set enables you to choose, in an iterative fashion, a value for the LAGDISTANCE= parameter. The value you choose is a compromise between the number of pairs making up each variogram point and the number of variogram points.

In some cases, the pattern of measured points may result in some lag or distance classes having a small number of pairs, while the remaining classes have a large number of pairs. By adjusting the value of the LAGDISTANCE= option to honor the rule of thumb (at least 30 pairs), you are “wasting” pairs in the other distance classes.

One strategy for solving this problem is to use less than 30 pairs for these distance classes. Then, either delete the corresponding variogram points or use them and accept the increased uncertainty. Unfortunately, the deficient distance classes are usually those close to the origin ($h = 0$). This is the crucial portion of the experimental variogram curve for determining the form of the theoretical variogram and for detecting the presence of a nugget effect.

Another alternative is to force distance classes to contain approximately the same number of pairs. This results in distance classes of unequal widths.

While PROC VARIOGRAM does not produce such distance classes directly, the OUTPAIR= data set, described in the section “OUTPAIR=*SAS-data-set*” on page 4881, contains information on all distinct pairs of points. You can use this data set, along with the RANK procedure, to produce experimental variogram-based equal numbers of pairs in each distance class.

To request an OUTDIST= data set, you specify the OUTDIST= data set in the PROC VARIOGRAM statement and the NOVARIOGRAM option in the COMPUTE statement. The NOVARIOGRAM option prevents any variogram or covariance computation from being performed.

Computation of the Distribution Distance Classes

The simplest way of determining the distribution of pairwise distances is to determine the maximum distance h_{max} between pairs and divide this distance by some number N of intervals to produce distance classes of length $\delta = \frac{h_{max}}{N}$. The distance between each pair of points P_1, P_2, denoted $\mid P_1P_2 \mid$, is computed, and the pair P_1P_2 is counted in the kth distance class if $\mid P_1P_2 \mid \in [(k-1)\delta, k\delta)$ for $k = 1, \cdots, N$.

The actual computation is a slight variation of this. A bound, rather than the actual maximum distance, is computed. This bound is the length of the diagonal of a bounding rectangle for the data points. This bounding rectangle is found by using the maximum and minimum x and y coordinates, $x_{max}, x_{min}, y_{max}, y_{min}$, and forming the rectangle determined by the points

$(x_{max}, y_{max}), (x_{max}, y_{min}), (x_{min}, y_{min}), (x_{min}, y_{max})$

See Figure 80.16 for an illustration of the bounding rectangle.

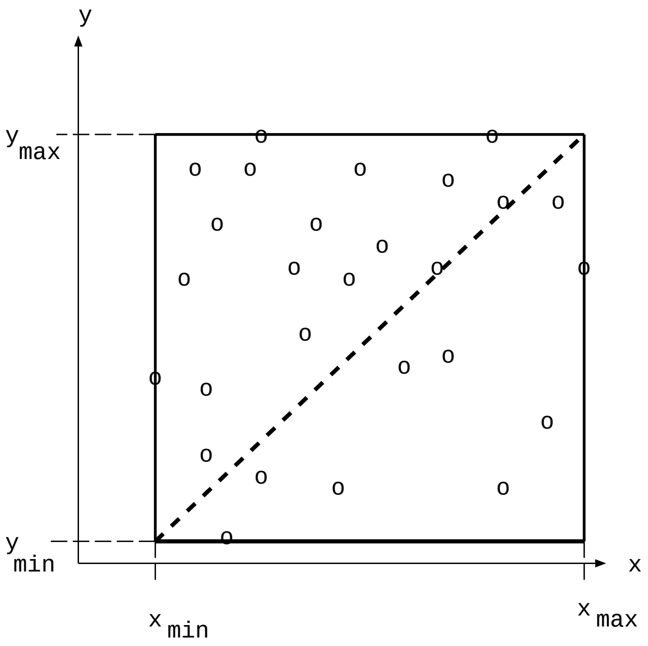


Figure 80.16. Bounding Rectangle to Determine Maximum Pairwise Distance

The pairwise distance bound, denoted by h_b, is given by

$$h_b^2 = (x_{max} - x_{min})^2 + (y_{max} - y_{min})^2$$

Using h_b, the interval $(0, h_b]$ is divided into $N+1$ subintervals, where N is the value of the NHCLASSES= option specified in the COMPUTE statement, or $N = 10$ if the NHCLASSES= option is not specified. The basic distance unit is $h_0 = \frac{h_b}{N}$; the distance intervals are centered on $h_0, 2h_0, \cdots, Nh_0$, with a distance tolerance of $\pm\frac{h_0}{2}$. The extra subinterval is $(0, h_0/2)$, corresponding to the 0th lag. It is half the length of the remaining subintervals, and it often contains the smallest number of pairs.

This method of partitioning the interval $(0, h_b]$ is identical to what is done when you actually compute the sample variogram.

The lag classes corresponding to h_0=1 are shown in Figure 80.17.

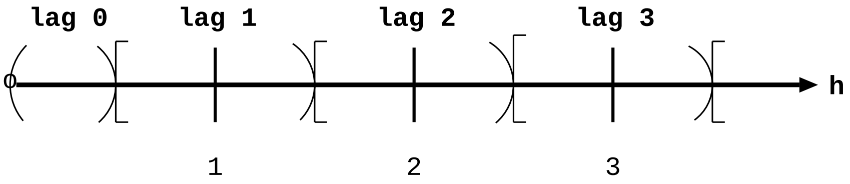


Figure 80.17. Lag Classes Corresponding to $h_0 = 1$

By increasing or decreasing the value of the NHCLASSES= option, you can adjust the lag or distance class with the smallest count so that this count is around 30 or some other value that you judge appropriate.

Once you determine an appropriate value for the NHCLASSES= option, you can use the width of the lag classes as a candidate value for the LAGDIST= option in the COMPUTE statement. The width of the lag classes is determined by the upper bound (UB) and lower bound (LB) variables.

For example, read the observation from the OUTDIST= data set corresponding to lag 1 and compute the quantity UB-LB. Use this value for the LAGDIST= option in the COMPUTE statement.

Note: Do not use the 0th lag class; it is half the length of the other intervals. Use lag 1 instead.

Variables in the OUTDIST= data set

The following variables are written to the OUTDIST= data set:

- COUNT, which is the number of pairs falling into this lag class
- LAG, which is the lag class value
- LB, which is the lower bound of the lag class interval
- UB, which is the upper bound of the lag class interval
- PER, which is the percent of all pairs falling in this lag class
- VARNAME, which is the name of the current VAR= variable

OUTPAIR=*SAS-data-set*

The OUTPAIR= data set contains one observation for each distinct pair of points P_1, P_2 in the original data set, unless you specify the OUTPDISTANCE= option in the COMPUTE statement.

If you specify OUTPDISTANCE=D_{max} in the COMPUTE statement, all pairs P_1, P_2 in the original data set that satisfy the relation $\mid P_1P_2 \mid \leq D_{max}$ are written to the OUTPAIR= data set.

Note that the OUTPAIR= data set can be very large even for a moderately sized DATA= data set. For example, if the DATA= data set has NOBS=500, the OUTPAIR= data set has NOBS$($NOBS $- 1)/2$ =124,750 if no OUTPDISTANCE= restriction is given in the COMPUTE statement.

The OUTPAIR= data set contains information on the distance and orientation for each point pair, and you can use it for specialized continuity measure calculations.

The OUTPAIR= data set contains the following variables:

- AC, which is the angle class value
- COS, which is the cosine of the angle between pairs
- DC, which is the distance (lag) class
- DISTANCE, which is the distance between pairs
- V1, which is the variable value for the first point in the pair
- V2, which is the variable value for the second point in the pair
- VARNAME, which is the variable name for the current VAR variable
- X1, which is the x coordinate of the first point in the pair
- X2, which is the x coordinate of the second point in the pair
- Y1, which is the y coordinate of the first point in the pair
- Y2, which is the y coordinate of the second point in the pair

Computational Resources

The computations of the VARIOGRAM procedure are basically binning: for each pair of observations in the input data set, a distance and angle class is determined and recorded. Let N_d denote the number of distance classes, N_a denote the number of angle classes, and N_v denote the number of VAR variables. The memory requirements for these operations are proportional to $N_d \times N_a \times N_v$. This is typically small.

The CPU time required for the computations is proportional to the number of pairs of observations, or to $N^2 \times N_v$, where N is the number of observations in the input data set.

Example

Example 80.1. A Box Plot of the Square Root Difference Cloud

The Gaussian form chosen for the variogram in the "Getting Started" section on page 4852 is based on the consideration of the plots of the sample variogram. For the coal thickness data, the Gaussian form appears to be a reasonable choice.

It can often happen, however, that a plot of the sample variogram shows so much scatter that no particular form is evident. The cause of this scatter can be one or more outliers in the pairwise differences of the measured quantities.

A method of identifying potential outliers is discussed in Cressie (1993, section 2.2.2). This example illustrates how to use the OUTPAIR= data set from PROC VARIOGRAM to produce a square root difference cloud, which is useful in detecting outliers.

For the spatial process $Z(s), s \in \mathcal{R}^2$, the square root difference cloud for a particular direction $\mathbf{e}$ is given by

$$\mid Z(s_i + he) - Z(s_i) \mid^{\frac{1}{2}}$$

for a given lag distance h. In the actual computation, all pairs of points P_1, P_2 within a distance tolerance around h and an angle tolerance around the direction $\mathbf{e}$ are used. This generates a number of point pairs for each lag class h. The spread of these values gives an indication of outliers.

Following the example in the "Getting Started" section on page 4852, this example uses a basic lag distance of 7 units, with a distance tolerance of 3.5, and a direction of N–S, with a 30^o angle tolerance.

First, input the data, then use PROC VARIOGRAM to produce an OUTPAIR= data set. Then use a DATA step to subset this data by choosing pairs within 30^o of N–S. In addition, compute lag class and square root difference variables. Next, summarize the results using the MEANS procedure and present them in a box plot using the SHEWHART procedure. The box plot facilitates the detection of outliers.

You can conclude from this example that there does not appear to be any outliers in the N–S direction for the coal seam thickness data.

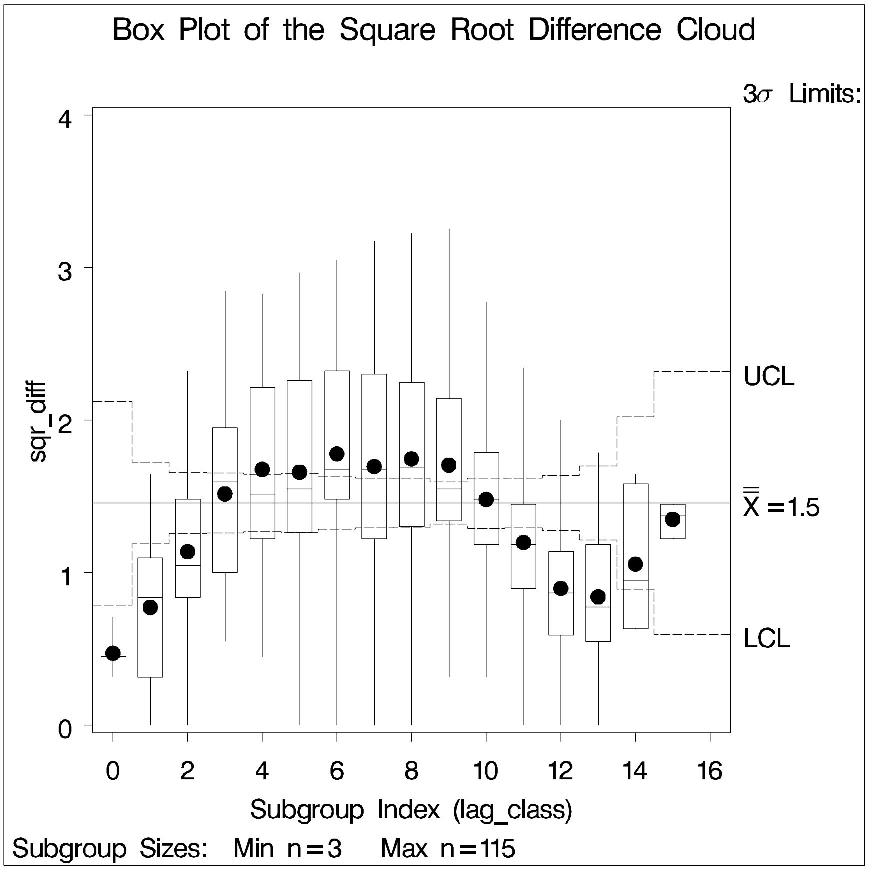
```
title 'Square Root Difference Cloud Example';
data thick;
   input east north thick @@;
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   94.8  71.5  39.7  96.2  84.3  40.3  98.2  58.2  39.5
   ;

proc variogram data=thick outp=outp;
coordinates xc=east yc=north;
var thick;
compute novar;
run;

data sqroot;
   set outp;

/*- Include only points +/- 30 degrees of N-S -------*/
   where abs(cos) < .5;

/*- Unit lag of 7, distance tolerance of 3.5  -------*/
   lag_class=int(distance/7 + .5000001);
   sqr_diff=sqrt(abs(v1-v2));
run;

proc sort data=sqroot;
   by lag_class;
run;

proc means data=sqroot noprint n mean std;
   var sqr_diff;
   by lag_class;
   output out=msqrt n=n mean=mean std=std;
run;

title2 'Summary of Results';
proc print data=msqrt;
   id lag_class;
   var n mean std;
run;
```

```
title ’Box Plot of the Square Root Difference Cloud’;
proc shewhart data=sqroot;
   boxchart sqr_diff*lag_class / cframe=ligr haxis=axis1
                                 vaxis=axis2;
   symbol1 v=dot c=blue height=3.5pct;
   axis1 minor=none;
   axis2 minor=none label=(angle=90 rotate=0);
run;
```

Output 80.1.1. Summary of Results

```
                  Square Root Difference Cloud Example
                           Summary of Results

                   lag_
                  class      n      mean       std

                    0        5    0.47300    0.14263
                    1       31    0.77338    0.41467
                    2       55    1.13908    0.47604
                    3       58    1.51768    0.51989
                    4       63    1.67858    0.60494
                    5       61    1.66014    0.70687
                    6       75    1.77999    0.64590
                    7       85    1.69703    0.75362
                    8       84    1.74687    0.68785
                    9      115    1.70635    0.57173
                   10       82    1.48100    0.48105
                   11       85    1.19877    0.47121
                   12       68    0.89765    0.42510
                   13       38    0.84223    0.44249
                   14        7    1.05653    0.42548
                   15        3    1.35076    0.11472
```

Output 80.1.2. Box Plot of the Square Root Difference Cloud



References

Cressie, N.A.C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons, Inc.

Duetsch, C.V. and Journel, A.G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Hohn, M.E. (1988), *Geostatistics and Petroleum Geology*, New York: Van Nostrand Reinhold.

Appendix A
Special SAS Data Sets

Appendix Contents

Appendix A
Special SAS Data Sets

Introduction to Special SAS Data Sets

All SAS/STAT procedures create SAS data sets. Any table generated by a procedure can be saved to a data set by using the Output Delivery System (ODS), and many procedures also have syntax to enable you to save other statistics to data sets. Some of these data sets are organized according to certain conventions that allow them to be read by a SAS/STAT procedure for further analysis. Such specially organized data sets are recognized by the TYPE= attribute of the data set.

For example, the CORR procedure (refer to the *SAS Procedures Guide*) can create a data set with the attribute TYPE=CORR containing a correlation matrix. This TYPE=CORR data set can be read by the REG or FACTOR procedure, among others. If the original data set is large, using a special SAS data set in this way can save a great deal of computer time by avoiding the recomputation of the correlation matrix in each of several analyses.

As another example, the REG procedure can create a TYPE=EST data set containing estimated regression coefficients. If you need to make predictions for new observations, you can have the SCORE procedure read both the TYPE=EST data set and a data set containing the new observations. PROC SCORE can then compute predicted values or residuals without repeating the entire regression analysis. See Chapter 64, "The SCORE Procedure," for an example.

A special SAS data set may contain different kinds of statistics. A special variable called _TYPE_ is used to distinguish the various statistics. For example, in a TYPE=CORR data set, an observation in which _TYPE_='MEAN' contains the means of the variables in the analysis, and an observation in which _TYPE_='STD' contains the standard deviations. Correlations appear in observations with _TYPE_='CORR'. Another special variable, _NAME_, is needed to identify the row of the correlation matrix. Thus, the correlation between variables X and Y would be given by the value of the variable X in the observation for which _TYPE_='CORR' and _NAME_='Y', or by the value of the variable Y in the observation for which _TYPE_='CORR' and _NAME_='X'.

You can create special SAS data steps directly in a DATA step by specifying the TYPE= option in parentheses after the data set name in the DATA statement. See Example A.2 on page 4896 for an example.

The special data sets created by SAS/STAT procedures can generally be used directly by other procedures without modification. However, if you create an output data set with PROC CORR and use the NOCORR option to omit the correlation matrix from the OUT= data set, you need to set the TYPE= option either in parentheses following the OUT= data set name in the PROC CORR statement or in parentheses following the DATA= option in any other procedure that recognizes the special TYPE= attribute. In either case, the TYPE= option should be set to COV, CSSCP, or SSCP according to what type of matrix is stored in the data set and what data set types are accepted as input by the other procedures you plan to use. If you do not follow these steps and you use the TYPE=CORR data set with no correlation matrix as input to another procedure, the procedure may issue an error message indicating that the correlation matrix is missing from the data set.

If you use a DATA step with a SET statement to modify a special SAS data set, you must specify the TYPE= option in the DATA statement. The TYPE= attribute of the data set in the SET statement is *not* automatically copied to the data set being created.

You can determine the TYPE= attribute of a data set by using the CONTENTS procedure (see Example A.1 on page 4895 and refer to the *SAS Procedures Guide* for details).

Table A.1 summarizes the TYPE= data sets that can be used as input to SAS/STAT procedures and the TYPE= data sets that are created by SAS/STAT procedures. The essential parts of the statements each procedure uses to create its output data set or data sets are shown.

Formulas useful for illustrating differences between corrected and uncorrected matrices in some special SAS data sets are shown in the "Definitional Formulas" section on page 4902.

Table A.1. SAS/STAT Procedures and Types of Data Sets

Procedure	**Input Data Sets TYPE= as shown***	**Output Data Sets (TYPE=null or as shown)**	**Created by Statement and Specification**
ACECLUS	INITIAL= INPUT= data set may be of type: ACE, CORR, COV, SSCP, UCORR, UCOV	ACE	PROC ACECLUS OUTSTAT= PROC ACECLUS OUT=
ANOVA			PROC ANOVA OUTSTAT=
CALIS	CORR, COV, FACTOR, RAM, SSCP, UCORR, UCOV, WEIGHT	CORR COV EST UCORR UCOV RAM WEIGHT	PROC CALIS OUTSTAT= PROC CALIS COV OUTSTAT= PROC CALIS OUTEST= PROC CALIS NOINT OUTSTAT= PROC CALIS NOINT COV OUTSTAT= PROC CALIS OUTRAM= PROC CALIS OUTWGT=
CANCORR	CORR, COV, FACTOR, SSCP, UCORR, UCOV	CORR UCORR	PROC CANCORR OUTSTAT= PROC CANCORR NOINT OUTSTAT= PROC CANCORR OUT=
CANDISC	CORR, COV, SSCP, CSSCP	CORR	PROC CANDISC OUTSTAT= PROC CANDISC OUT=
CATMOD	EST	EST	RESPONSE / OUTEST= RESPONSE /OUT=
CLUSTER	DISTANCE	TREE	PROC CLUSTER OUTTREE=

Table A.1. (continued)

Procedure	**Input Data Sets TYPE= as shown***	**Output Data Sets (TYPE=null or as shown)**	**Created by Statement and Specification**
CORRESP			PROC CORRESP OUTC= PROC CORRESP OUTF=
DISCRIM	CORR, COV, SSCP, CSSCP, LINEAR, QUAD, MIXED	LINEAR QUAD MIXED CORR	PROC DISCRIM POOL=YES OUTSTAT= PROC DISCRIM POOL=NO OUTSTAT= PROC DISCRIM POOL=TEST OUTSTAT= PROC DISCRIM METHOD=NPAR OUTSTAT= PROC DISCRIM OUT= PROC DISCRIM OUTCROSS= PROC DISCRIM OUTD= PROC DISCRIM TESTOUT= PROC DISCRIM TESTOUTD=
DISTANCE		DISTANCE	PROC DISTANCE OUT= PROC DISTANCE OUTSDZ=
FACTOR	ACE, CORR, COV, FACTOR, SSCP, UCORR, UCOV	FACTOR	PROC FACTOR OUTSTAT= PROC FACTOR OUT=
FASTCLUS			PROC FASTCLUS OUT= PROC FASTCLUS OUTSEED= PROC FASTCLUS OUTSTAT= PROC FASTCLUS MEAN=
FREQ			TABLES OUT= OUTPUT OUT=
GENMOD			OUTPUT OUT=
GLM			PROC GLM OUTSTAT= LSMEANS / OUT= OUTPUT OUT=
GLMMOD			PROC GLMMOD OUTDESIGN= PROC GLMMOD OUTPARM=
INBREED			PROC INBREED OUTCOV=
KRIGE2D			PROC KRIGE2D OUTEST= PROC KRIGE2D OUTNBHD=
LATTICE			
LIFEREG		EST	PROC LIFEREG OUTEST= OUTPUT OUT=
LIFETEST			PROC LIFETEST OUTSURV= PROC LIFETEST OUTTEST=
LOGISTIC		EST	PROC LOGISTIC OUTEST= OUTPUT OUT= MODEL / OUTROC=

Table A.1. (continued)

Procedure	**Input Data Sets TYPE= as shown***	**Output Data Sets (TYPE=null or as shown)**	**Created by Statement and Specification**
MDS			PROC MDS OUT= PROC MDS OUTFIT= PROC MDS OUTRES=
MIXED			MODEL OUTPRED= MODEL OUTPREDM= PRIOR OUT= PRIOR OUTG= PRIOR OUTGT=
MODECLUS	DISTANCE		PROC MODECLUS OUT= PROC MODECLUS OUTCLUS= PROC MODECLUS OUTSUM=
MULTTEST			PROC MULTTEST OUT= PROC MULTTEST OUTPERM= PROC MULTTEST OUTSAMP=
NESTED			
NLIN		EST	PROC NLIN OUTEST= OUTPUT OUT=
NPAR1WAY			OUTPUT OUT=
ORTHOREG		EST	PROC ORTHOREG OUTEST=
PHREG		EST	PROC PHREG OUTEST= BASELINE OUT= OUTPUT OUT=
PLAN			OUTPUT OUT=
PLS			OUTPUT OUT=
PRINCOMP	ACE, CORR, COV, EST, FACTOR, SSCP, UCORR, UCOV	CORR COV UCORR UCOV	PROC PRINCOMP OUTSTAT= PROC PRINCOMP COV OUTSTAT= PROC PRINCOMP NOINT OUTSTAT= PROC PRINCOMP NOINT COV OUTSTAT= PROC PRINCOMP OUT=
PRINQUAL			PROC PRINQUAL OUT=
PROBIT		EST	PROC PROBIT OUTEST= OUTPUT OUT=
REG	CORR, COV, SSCP, UCORR, UCOV	EST SSCP	PROC REG OUTEST= PROC REG OUTSSCP= OUTPUT OUT=
RSREG			PROC RSREG OUT= RIDGE OUTR=
SCORE	SCORE= data set can be of any type		PROC SCORE OUT=
SIM2D			PROC SIM2D OUTSIM=

Table A.1. (continued)

Procedure	Input Data Sets TYPE= as shown[*]	Output Data Sets (TYPE=null or as shown)	Created by Statement and Specification
SURVEYSELECT			PROC SURVEYSELECT OUT= PROC SURVEYSELECT OUTSORT=
STDIZE			PROC STDIZE OUT= PROC STDIZE OUTSTAT=
STEPDISC	CORR, COV, SSCP, CSSCP		
TRANSREG			PROC TRANSREG OUTTEST= OUTPUT OUT=
TREE	TREE		PROC TREE OUT=
TTEST			
VARCLUS	CORR, COV, FACTOR, SSCP, UCORR, UCOV	CORR UCORR TREE	PROC VARCLUS OUTSTAT= PROC VARCLUS NOINT OUTSTAT= PROC VARCLUS OUTTREE=
VARCOMP			
VARIOGRAM			PROC VARIOGRAM OUTDISTANCE= PROC VARIOGRAM OUTPAIR= PROC VARIOGRAM OUTVAR=

[*] If no TYPE= is shown, the procedure does not recognize any special data set types except possibly to issue an error message for inappropriate values of TYPE=.

Special SAS Data Sets

TYPE=CORR Data Sets

A TYPE=CORR data set usually contains a correlation matrix and possibly other statistics including means, standard deviations, and the number of observations in the original SAS data set from which the correlation matrix was computed.

Using PROC CORR with an output data set option (OUTP=, OUTS=, OUTK=, OUTH=, or OUT=) produces a TYPE=CORR data set. (For a complete description of the CORR procedure, refer to the *SAS Procedures Guide*). The CALIS, CANCORR, CANDISC, DISCRIM, PRINCOMP, and VARCLUS procedures can also create a TYPE=CORR data set with additional statistics.

A TYPE=CORR data set containing a correlation matrix can be used as input for the ACECLUS, CALIS, CANCORR, CANDISC, DISCRIM, FACTOR, PRINCOMP, REG, SCORE, STEPDISC, and VARCLUS procedures.

The variables in a TYPE=CORR data set are

- the BY variable or variables, if a BY statement is used with the procedure
- _TYPE_, a character variable of length eight with values identifying the type of statistic in each observation, such as ’MEAN’, ’STD’, ’N’, and ’CORR’
- _NAME_, a character variable with values identifying the variable with which a given row of the correlation matrix is associated
- other variables that were analyzed by the CORR procedure or other procedures

The usual values of the _TYPE_ variable are as follows.

TYPE	**Contents**
MEAN	mean of each variable analyzed
STD	standard deviation of each variable
N	number of observations used in the analysis. PROC CORR records the number of nonmissing values for each variable unless the NOMISS option is used. If the NOMISS option is specified, or if the CALIS, CANCORR, CANDISC, PRINCOMP, or VARCLUS procedure is used to create the data set, observations with one or more missing values are omitted from the analysis, so this value is the same for each variable and provides the number of observations with no missing values. If a FREQ statement is used with the procedure that creates the data set, the number of observations is the sum of the relevant values of the variable in the FREQ statement. Procedures that read a TYPE=CORR data set use the smallest value in the observation with _TYPE_=’N’ as the number of observations in the analysis.
SUMWGT	sum of the observation weights if a WEIGHT statement is used with the procedure that creates the data set. The values are determined analogously to those of the _TYPE_=’N’ observation.
CORR	correlations with the variable named by the _NAME_ variable

There may be additional observations in a TYPE=CORR data set depending on the particular procedure and options used.

If you create a TYPE=CORR data set yourself, the data set need not contain the observations with _TYPE_=’MEAN’, ’STD’, ’N’, or ’SUMWGT’, unless you intend to use one of the discriminant procedures. Procedures assume that all of the means are 0.0 and that the standard deviations are 1.0 if this information is not in the TYPE=CORR data set. If _TYPE_=’N’ does not appear, most procedures assume that the number of observations is 10,000; significance tests and other statistics that depend on the number of observations are, of course, meaningless. In the CALIS and CANCORR procedures, you can use the EDF= option instead of including a _TYPE_=’N’ observation.

A correlation matrix is symmetric; that is, the correlation between X and Y is the same as the correlation between Y and X. The CALIS, CANCORR, CANDISC, CORR, DISCRIM, PRINCOMP, and VARCLUS procedures output the entire correlation matrix. If you create the data set yourself, you need to include only one of the two occurrences of the correlation between two variables; the other may be given a missing value.

If you create a TYPE=CORR data set yourself, the _TYPE_ and _NAME_ variables are not necessary except for use with the discriminant procedures and PROC SCORE. If there is no _TYPE_ variable, then all observations are assumed to contain correlations. If there is no _NAME_ variable, the first observation is assumed to correspond to the first variable in the analysis, the second observation to the second variable, and so on. However, if you omit the _NAME_ variable, you will not be able to analyze arbitrary subsets of the variables or list the variables in a VAR or MODEL statement in a different order.

Example A.1: A TYPE=CORR Data Set Produced by PROC CORR

See Output A.1.1 for an example of a TYPE=CORR data set produced by the following SAS statements. Output A.1.2 displays partial output from the CONTENTS procedure, which indicates that the “Data Set Type” is ’CORR’.

```
title 'Five Socioeconomic Variables';
data SocEcon;
   title2 'Harman (1976), Modern Factor Analysis, 3rd ed';
   input pop school employ services house;
   datalines;
5700     12.8      2500      270       25000
1000     10.9      600       10        10000
3400     8.8       1000      10        9000
3800     13.6      1700      140       25000
4000     12.8      1600      140       25000
8200     8.3       2600      60        12000
1200     11.4      400       10        16000
9100     11.5      3300      60        14000
9900     12.5      3400      180       18000
9600     13.7      3600      390       25000
9600     9.6       3300      80        12000
9400     11.4      4000      100       13000
;
proc corr noprint out=corrcorr;
run;

proc print data=corrcorr;
run;

proc contents data=corrcorr;
run;
```

Output A.1.1. A TYPE=CORR Data Set Produced by PROC CORR

```
                          Five Socioeconomic Variables
                 Harman (1976), Modern Factor Analysis, 3rd ed

Obs    _TYPE_    _NAME_          pop     school     employ    services       house

 1      MEAN                 6241.67    11.4417    2333.33     120.833    17000.00
 2      STD                  3439.99     1.7865    1241.21     114.928     6367.53
 3      N                      12.00    12.0000      12.00      12.000       12.00
 4      CORR     pop            1.00     0.0098       0.97       0.439        0.02
 5      CORR     school         0.01     1.0000       0.15       0.691        0.86
 6      CORR     employ         0.97     0.1543       1.00       0.515        0.12
 7      CORR     services       0.44     0.6914       0.51       1.000        0.78
 8      CORR     house          0.02     0.8631       0.12       0.778        1.00
```

Output A.1.2. Contents of a TYPE=CORR Data Set

```
                             The CONTENTS Procedure

 Data Set Name  WORK.CORRCORR                      Observations          8
 Member Type    DATA                               Variables             7
 Engine         V8                                 Indexes               0
 Created        13:56 Wednesday, July 25, 2001     Observation Length    56
 Last Modified  13:56 Wednesday, July 25, 2001     Deleted Observations  0
 Protection                                        Compressed            NO
 Data Set Type  CORR                               Sorted                NO
 Label          Pearson Correlation Matrix
```

Example A.2: Creating a TYPE=CORR Data Set in a DATA Step

This example creates a TYPE=CORR data set by reading a correlation matrix in a DATA step. Output A.2.2 shows the resulting data set.

```
title 'Five Socioeconomic Variables';
data datacorr(type=corr);
   infile cards missover;
   type_='corr';
   input _name_ $ pop school employ services house;
   datalines;
POP         1.00000
SCHOOL      0.00975   1.00000
EMPLOY      0.97245   0.15428   1.00000
SERVICES    0.43887   0.69141   0.51472   1.00000
HOUSE       0.02241   0.86307   0.12193   0.77765   1.00000
;
run;

proc print data=datacorr;
run;
```

Output A.2.2. A TYPE=CORR Data Set Created by a DATA Step

```
                              Five Socioeconomic Variables

Obs    type_    _name_        pop       school     employ     services     house

 1     corr     POP         1.00000      .          .            .           .
 2     corr     SCHOOL      0.00975    1.00000      .            .           .
 3     corr     EMPLOY      0.97245    0.15428    1.00000        .           .
 4     corr     SERVICES    0.43887    0.69141    0.51472      1.00000       .
 5     corr     HOUSE       0.02241    0.86307    0.12193      0.77765       1
```

TYPE=UCORR Data Sets

A TYPE=UCORR data set is almost identical to a TYPE=CORR data set, except that the correlations are uncorrected for the mean. The corresponding value of the _TYPE_ variable is 'UCORR' instead of 'CORR'. Uncorrected standard deviations are in observations with _TYPE_='USTD'.

A TYPE=UCORR data set can be used as input for every SAS/STAT procedure that uses a TYPE=CORR data set, except for the CANDISC, DISCRIM, and STEPDISC procedures. TYPE=UCORR data sets can be created by the CALIS, CANCORR, PRINCOMP, and VARCLUS procedures.

TYPE=COV Data Sets

A TYPE=COV data set is similar to a TYPE=CORR data set except that it has _TYPE_='COV' observations containing covariances instead of or in addition to _TYPE_='CORR' observations containing correlations. The CALIS and PRINCOMP procedures create a TYPE=COV data set if the COV option is used. You can also create a TYPE=COV data set by using PROC CORR with the COV and NOCORR options and specifying the data set option TYPE=COV in parentheses following the name of the output data set. You can use only the OUTP= or OUT= options to create a TYPE=COV data set with PROC CORR.

Another way to create a TYPE=COV data set is to read a covariance matrix in a data set, in the same manner as shown in Example A.2 on page 4896 for a TYPE=CORR data set.

TYPE=COV data sets are used by the same procedures that use TYPE=CORR data sets.

TYPE=UCOV Data Sets

A TYPE=UCOV data set is similar to a TYPE=COV data set, except that the covariances are uncorrected for the mean. Also, the corresponding value of the _TYPE_ variable is 'UCOV' instead of 'COV'.

A TYPE=UCOV data set can be used as input for every SAS/STAT procedure that uses a TYPE=COV data set, except for the CANDISC, DISCRIM, and STEPDISC procedures. TYPE=UCOV data sets can be created by the CALIS and PRINCOMP procedures.

TYPE=SSCP Data Sets

A TYPE=SSCP data set contains an uncorrected sum of squares and crossproducts (SSCP) matrix. TYPE=SSCP data sets are produced by PROC REG when the OUTSSCP= option is specified in the PROC REG statement. You can also create a TYPE=SSCP data set by using PROC CORR with the SSCP option and specifying the data set option TYPE=SSCP in parentheses following the name of the OUTP= or OUT= data set. You can also create TYPE=SSCP data sets in a DATA step; in this case, TYPE=SSCP must be specified as a data set option.

The variables in a TYPE=SSCP data set include those found in a TYPE=CORR data set. In addition, there is a variable called Intercept that contains crossproducts for the intercept (sums of the variables). The SSCP matrix is stored in observations with _TYPE_='SSCP', including a row with _NAME_='Intercept'. PROC REG also outputs an observation with _TYPE_='N'. PROC CORR includes observations with _TYPE_='MEAN' and _TYPE_='STD' as well.

TYPE=SSCP data sets are used by the same procedures that use TYPE=CORR data sets.

Example A.3: A TYPE=SSCP Data Set Produced by PROC REG

Output A.3.1 shows a TYPE=SSCP data set produced by PROC REG from the SocEcon data set created in Example A.1 on page 4895.

```
proc reg data=SocEcon outsscp=regsscp;
   model house=pop school employ services / noprint;
run;

proc print data=regsscp;
run;
```

Output A.3.1. A TYPE=SSCP Data Set Produced by PROC REG

```
Obs   _TYPE_   _NAME_      Intercept          pop       school      employ  services        house

  1    SSCP    Intercept        12.0        74900       137.30       28000      1450       204000
  2    SSCP    pop           74900.0    597670000    857640.00   220440000  10959000   1278700000
  3    SSCP    school          137.3       857640      1606.05      324130     18152      2442100
  4    SSCP    employ        28000.0    220440000    324130.00    82280000   4191000    486600000
  5    SSCP    services       1450.0     10959000     18152.00     4191000    320500     30910000
  6    SSCP    house        204000.0   1278700000   2442100.00   486600000  30910000   3914000000
  7    N                        12.0           12        12.00          12        12           12
```

TYPE=CSSCP Data Sets

A TYPE=CSSCP data set contains a corrected sum of squares and crossproducts (CSSCP) matrix. TYPE=CSSCP data sets are created by using the CORR procedure with the CSSCP option and specifying the data set option TYPE=CSSCP in parentheses following the name of the OUTP= or OUT= data set. You can also create TYPE=CSSCP data sets in a DATA step; in this case, TYPE=CSSCP must be specified as a data set option.

The variables in a TYPE=CSSCP data set are the same as those found in a TYPE=SSCP data set, except that there is not a variable called Intercept or a row with _NAME_='Intercept'.

TYPE=CSSCP data sets are read by only the CANDISC, DISCRIM, and STEPDISC procedures.

TYPE=EST Data Sets

A TYPE=EST data set contains parameter estimates. The CALIS, CATMOD, LIFEREG, LOGISTIC, NLIN, ORTHOREG, PHREG, PROBIT, and REG procedures create TYPE=EST data sets when the OUTEST= option is specified. A TYPE=EST data set produced by PROC LIFEREG, PROC ORTHOREG, or PROC REG can be used with PROC SCORE to compute residuals or predicted values.

The variables in a TYPE=EST data set include

- the BY variables, if a BY statement is used
- _TYPE_, a character variable of length eight, that indicates the type of estimate. The values depend on which procedure created the data set. Usually a value of 'PARM' or 'PARMS' indicates estimated regression coefficients, and a value of 'COV' or 'COVB' indicates estimated covariances of the parameter estimates. Some procedures, such as PROC NLIN, have other values of _TYPE_ for special purposes.
- _NAME_, a character variable that contains the values of the names of the rows of the covariance matrix when the procedure outputs the covariance matrix of the parameter estimates.
- variables that contain the parameter estimates, usually the same variables that appear in the VAR statement or in any MODEL statement. See Chapter

19, “The CALIS Procedure,” Chapter 22, “The CATMOD Procedure,” and Chapter 50, “The NLIN Procedure,” for details on the variable names used in output data sets created by those procedures.

Other variables can be included depending on the particular procedure and options used.

Example A.4: A TYPE=EST Data Set Produced by PROC REG

Output A.4.1 shows the TYPE=EST data set produced by the following statements:

```
proc reg data=SocEcon outest=regest covout;
   full:   model house=pop school employ services / noprint;
   empser: model house=employ services / noprint;
run;
proc print data=regest;
run;
```

Output A.4.1. A TYPE=EST Data Set Produced by PROC REG

```
Obs    _MODEL_    _TYPE_    _NAME_       _DEPVAR_    _RMSE_        Intercept

  1    full       PARMS                   house      3122.03         -8074.21
  2    full       COV       Intercept     house      3122.03     109408014.44
  3    full       COV       pop           house      3122.03         -9157.04
  4    full       COV       school        house      3122.03      -9784744.54
  5    full       COV       employ        house      3122.03         20612.49
  6    full       COV       services      house      3122.03        102764.89
  7    empser     PARMS                   house      3789.96         15021.71
  8    empser     COV       Intercept     house      3789.96       5824096.19
  9    empser     COV       employ        house      3789.96         -1915.99
 10    empser     COV       services      house      3789.96         -1294.94

Obs         pop         school       employ      services    house

  1        0.65        2140.10        -2.92         27.81      -1
  2    -9157.04    -9784744.54     20612.49     102764.89       .
  3        2.32         852.86        -6.20         -5.20       .
  4      852.86      907886.36     -2042.24      -9608.59       .
  5       -6.20       -2042.24        17.44          6.50       .
  6       -5.20       -9608.59         6.50        202.56       .
  7         .              .          -1.94         53.88      -1
  8         .              .        -1915.99      -1294.94       .
  9         .              .           1.15         -6.41       .
 10         .              .          -6.41        134.49       .
```

TYPE=ACE Data Sets

A TYPE=ACE data set is created by the ACECLUS procedure, and it contains the approximate within-cluster covariance estimate, as well as eigenvalues and eigenvectors from a canonical analysis, among other statistics. It can be used as input to the ACECLUS procedure to initialize another execution of PROC ACECLUS. It can also be used to compute canonical variable scores with the SCORE procedure and as input to the FACTOR procedure, specifying METHOD=SCORE, to rotate the canonical variables. See Chapter 16, “The ACECLUS Procedure,” for details.

TYPE=DISTANCE Data Sets

You can create a TYPE=DISTANCE data set containing distance or dissimilarity measures using the DISTANCE procedure. The proximity measures are stored as a lower triangular matrix or a square matrix in the OUT= data set (depending on the SHAPE= option). See Chapter 26, "The DISTANCE Procedure," for details. You can also create a TYPE=DISTANCE data set in a DATA step by reading or computing a lower triangular or symmetric matrix of dissimilarity values, such as a chart of mileage between cities. The number of observations must be equal to the number of variables used in the analysis. This type of data set is used as input by the CLUSTER and MODECLUS procedures. PROC CLUSTER ignores the upper triangular portion of a TYPE=DISTANCE data set and assumes that all main diagonal values are zero, even if they are missing. PROC MODECLUS uses the entire distance matrix and does not require the matrix to be symmetric. See Chapter 23, "The CLUSTER Procedure," and Chapter 47, "The MODECLUS Procedure," for examples and details.

TYPE=FACTOR Data Sets

A TYPE=FACTOR data set is created by PROC FACTOR when the OUTSTAT= option is specified. The CALIS, CANCORR, FACTOR, PRINCOMP, SCORE, and VARCLUS procedures can use TYPE=FACTOR data sets as input. The variables are the same as in a TYPE=CORR data set. The statistics include means, standard deviations, sample size, correlations, eigenvalues, eigenvectors, factor patterns, residual correlations, scoring coefficients, and others depending on the options specified. See Chapter 27, "The FACTOR Procedure," for details.

When the NOINT option is used with the OUTSTAT= option in PROC FACTOR, the value of the _TYPE_ variable is set to 'USCORE' instead of 'SCORE' to indicate that the scoring coefficients have not been corrected for the mean. If this data set is used with the SCORE procedure, the value of the _TYPE_ variable tells PROC SCORE whether or not to subtract the mean from the scoring coefficients.

TYPE=RAM Data Sets

The CALIS procedure creates and accepts as input a TYPE=RAM data set. This data set contains the model specification and the computed parameter estimates. A TYPE=RAM data set is intended to be reused as an input data set to specify good initial values in subsequent analyses by PROC CALIS. See Chapter 19, "The CALIS Procedure," for details.

TYPE=WEIGHT Data Sets

The CALIS procedure creates and accepts as input a TYPE=WEIGHT data set. This data set contains the weight matrix used in generalized, weighted, or diagonally weighted least-squares estimation. See Chapter 19, "The CALIS Procedure," for details.

TYPE=LINEAR Data Sets

A TYPE=LINEAR data set contains the coefficients of a linear function of the variables in observations with _TYPE_='LINEAR'.

The DISCRIM procedure stores linear discriminant function coefficients in a TYPE=LINEAR data set when you specify METHOD=NORMAL (the default method), POOL=YES, and an OUTSTAT= data set; the data set can be used in a subsequent invocation of PROC DISCRIM to classify additional observations. Many other statistics can be included depending on the options used. See Chapter 25, "The DISCRIM Procedure," for details.

TYPE=QUAD Data Sets

A TYPE=QUAD data set contains the coefficients of a quadratic function of the variables in observations with _TYPE_='QUAD'.

The DISCRIM procedure stores quadratic discriminant function coefficients in a TYPE=QUAD data set when you specify METHOD=NORMAL (the default method), POOL=NO, and an OUTSTAT= data set; the data set can be used in a subsequent invocation of PROC DISCRIM to classify additional observations. Many other statistics can be included depending on the options used. See Chapter 25, "The DISCRIM Procedure," for details.

TYPE=MIXED Data Sets

A TYPE=MIXED data set contains coefficients of either a linear or a quadratic function, or both if there are BY groups.

The DISCRIM procedure produces a TYPE=MIXED data set when you specify METHOD=NORMAL (the default method), POOL=TEST, and an OUTSTAT= data set. See Chapter 25, "The DISCRIM Procedure," for details.

Definitional Formulas

This section contrasts corrected and uncorrected SSCP, COV, and CORR matrices by showing how these matrices can be computed.

In the following formulas, assume that the data consist of two variables, X and Y, with n observations.

$$\text{SSCP} = \begin{bmatrix} n & \sum X & \sum Y \\ \sum X & \sum X^2 & \sum XY \\ \sum Y & \sum XY & \sum Y^2 \end{bmatrix}$$

$$\text{CSSCP} = \begin{bmatrix} \sum(X-\bar{X})^2 & \sum(X-\bar{X})(Y-\bar{Y}) \\ \sum(X-\bar{X})(Y-\bar{Y}) & \sum(Y-\bar{Y})^2 \end{bmatrix}$$

$$\text{COV} = \frac{\text{CSSCP}}{n-1} = \frac{1}{n-1}\begin{bmatrix} \sum(X-\bar{X})^2 & \sum(X-\bar{X})(Y-\bar{Y}) \\ \sum(X-\bar{X})(Y-\bar{Y}) & \sum(Y-\bar{Y})^2 \end{bmatrix}$$

$$\text{UCOV} = \frac{1}{n}\begin{bmatrix} \sum X^2 & \sum XY \\ \sum XY & \sum Y^2 \end{bmatrix}$$

$$\text{CORR} = \begin{bmatrix} 1 & \dfrac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}} \\ \dfrac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}} & 1 \end{bmatrix}$$

$$\text{UCORR} = \begin{bmatrix} 1 & \dfrac{\sum XY}{\sqrt{\sum X^2\sum Y^2}} \\ \dfrac{\sum XY}{\sqrt{\sum X^2\sum Y^2}} & 1 \end{bmatrix}$$

Appendix B
Using the %PLOTIT Macro

Appendix Contents

Appendix B
Using the %PLOTIT Macro

Overview

You can use the PLOT procedure to create low-resolution printer plots of labeled points. Alternatively, you can use the %PLOTIT macro to create high-resolution graphical scatter plots of labeled points.

The %PLOTIT macro is designed to make it easy to display raw data, regression results, and results from the CORRESP, MDS, PRINCOMP, PRINQUAL, and TRANSREG procedures. You can use this macro to position labels, draw curves, vectors, and circles, and shade to show density or a third variable. You can also use the %PLOTIT macro to control the colors, sizes, fonts, and general appearance of the plots and to create contour plots for discriminant analysis.

The %PLOTIT macro is a part of the SAS autocall library. If your site has installed the autocall libraries supplied by SAS Institute and uses the standard configuration of SAS software supplied by the Institute, you need only to ensure that the SAS system option MAUTOSOURCE is in effect to begin using the autocall macros.

For more information about autocall libraries, refer to *SAS Macro Language: Reference, First Edition*, 1997. The %PLOTIT macro can also be found at SAS Institute's Web site [http://www.sas.com]. Refer to "Experimental Design and Choice Modeling Macros" and "Graphical Scatter Plots of Labeled Points" at [http://support.sas.com/techsup/tnote/tnote_stat.html#market].

%PLOTIT Macro Options Used in This Book

Most of the examples in this book that invoke the %PLOTIT macro are created with a specific set of options. The graphics are generated by using a special macro variable called plotitop.

The code you see in the examples creates the color graphics that appear in the online (CD) version of the manual. A slightly different set of options and statements are used to create the black-and-white graphics that appear in the printed version of the book.

To create the online (color) version of the graphic output, the plotitop variable is defined as follows.

```
%let plotitop =  gopts   = gsfmode = replace
                 gaccess = gsasfile device = gif
                 hsize   = 5.63     vsize  = 3.5
                 cback   = white,
                 cframe  = ligr,
                 color   = black,
                 colors  = red blue white,
                 options = noclip expand, post=myplot.gif;
```

To create the black-and-white version of the graphic output, which appears in the printed version of the manual, the plotitop variable is defined as follows:

```
%let plotitop =  gopts   = gsfmode = replace
                 gaccess = gsasfile device = pslepsf
                 hsize   = 5.63     vsize  = 3.5
                 cback   = white,
                 color   = black,
                 colors  = black,
                 options = noclip border expand, post=myplot.ps;
```

For information on graphics options used in other examples, see Chapter 1, “Introduction.”

Subject Index

Syntax Index

Your Turn

If you have comments or suggestions about *SAS/STAT® 9.1 User's Guide,* please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: `yourturn@sas.com`

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: `suggest@sas.com`